

Инференс импульсной нейронной сети, полученной путем перевода её из формальной, на мемристорах

В настоящее время технологии искусственного интеллекта находят широкое применение в сфере обработки и анализа данных. Хотя формальные искусственные нейронные сети (ИНС) получили широкое распространение, всё большее внимание привлекают спайковые, или импульсные нейронные сети (СИНС), которые считаются более перспективными. Эти нейронные сети работают по временному принципу передачи информации между нейронами и нейроны в таких сетях обмениваются между собой спайками одинакового значения (одинаковой амплитуды) за определенный промежуток времени.

Из-за своего принципа работы СИНС имеет ряд преимуществ – это разреженность вычислений, масштабируемость, стабильная латентность и возможность работы на энергоэффективных электронных компонентах - мемристорах.

Нейрон в СИНС схож с по принципам своей работы с биологическим: его входы – дендриты у биологического, связи – синапсы, тело – это сома и выход – аксон. Спайки, выходящие из первого нейрона, идут на синапс следующего, где они суммируются с другими поступающими сигналами, определяя дальнейшее поведение нейрона. [1].

Спайк, поступивший на вход нейрона, после скалярного умножения на соответствующий вес, накапливается на мембране с другими импульсами, поступившими за некоторое время, определяя дальнейшее поведение этого нейрона – если мембранный порог превышен или равен сумме сигналов, то нейрон испускает спайк, сбрасывая мембранный потенциал до определенного значения. Таким образом получается, что спайковый нейрон реагирует только на определенную комбинацию входных спайков, детектируя определенные признаки во входных данных [2].

Наиболее быстрым вариантом решения практических задач с помощью СИНС является такой подход, при котором обученная формальная нейронная сеть конвертируется в модель спайковой ИНС [3].

В докладе продемонстрирована программная реализация алгоритма преобразования формальной ИНС в СИНС, реализованная в библиотеке. Созданная библиотека позволяет импортировать модель формальной ИНС из фреймворка Keras, конвертировать входные данные в спайковые последовательности, конвертировать формальные нейроны в спайковые и провести модельный инференс СИНС. Также в работе показан инференс конвертированной модели на реальных мемристивных устройствах в топологии кроссбар-массива 32x8 1T1R с применением программно-аппаратного комплекса.

Литература

1. Киселев М.В., Моделирование импульсных нейронных сетей - цели, методы, аппаратные платформы / I Национальный конгресс по когнитивным исследованиям, искусственному интеллекту и нейроинформатике. Сборник пленарных и индустриальных докладов. Доклады симпозиумов. Москва. 2020. С. 81-84.
2. В.А. Евграфов, Е.А. Ильюшин. Спайковые нейронные сети / International Journal of Open Information Technologies. 2021. Vol. 9. №7.
3. А. Г. Сбоев, Р. Б. Рыбка, А. В. Серенко, А. Г. Селиванов. Классификация звука импульсной нейронной сетью, удовлетворяющей ограничениям нейропроцессора Алтай и при этом использующей полный динамический диапазон весов / Вестник военного инновационного технополиса «Эра». 2022. Vol 3. № 3. С. 314–321.

Балдин Н.Д.

Руководитель: к.т.н. доцент Канунова Е.Е.

*Муромский институт (филиал) федерального государственного бюджетного
образовательного учреждения высшего образования «Владимирский
государственный университет имени Александра Григорьевича и Николая
Григорьевича Столетовых»*

602264, г. Муром, Владимирская обл., ул. Орловская, 23

E-mail: baldinnikita691@gmail.com

Программное обеспечение информационной системы стоматологической клиники

Современный уровень развития медицинских информационных систем требует создания сложных программных комплексов, отвечающих требованиям гибкости, надежности и удобства использования. Наиболее эффективным для таких задач является объектно-ориентированный подход (ООП), позволяющий представить систему как совокупность взаимодействующих объектов, каждый из которых инкапсулирует данные и методы работы с ними. Это обеспечивает возможность поэтапной разработки и легкой модификации системы.

В докладе рассматривается разработка специализированного программного обеспечения для автоматизации работы стоматологической клиники. В условиях цифровизации здравоохранения подобные системы становятся необходимым инструментом для повышения качества обслуживания пациентов и эффективности работы медицинского персонала.

Существует несколько коммерческих аналогов разрабатываемой системы, таких как "Битрикс24 Медицина" и "IDENT". Однако эти решения имеют существенные недостатки: "Битрикс24" является слишком универсальным и не учитывает специфику стоматологии, а "IDENT" требует значительных ресурсов для внедрения и сложен в освоении. Разрабатываемая программа устраняет эти недостатки, предлагая специализированное решение с оптимальным набором функций.

Приложение реализовано на языке Python с использованием фреймворка PyQt6 для графического интерфейса и SQLite в качестве базы данных. Система обладает интуитивно понятным интерфейсом и предоставляет полный набор инструментов для управления клиникой, включая возможность ведения электронных медицинских карт и интеграции с диагностическим оборудованием.

Основные компоненты системы:

1. Модуль авторизации и управления доступом;
2. Модуль работы с пациентами;
3. Модуль расписания.

Последовательность работы с системой:

1. Авторизация пользователя в системе;
2. Выбор рабочего режима (прием пациентов, работа с расписанием и т.д.);
3. Ввод и обработка информации;
4. Формирование отчетности.

Литература

1. docs.python.org/3/ - Официальная документация Python
2. doc.qt.io/qtforpython/ - Документация PyQt6
3. www.sqlite.org/docs.html - Руководство по SQLite

Бобров И.С.

Руководитель: к.т.н. доцент Канунова Е.Е.

*Муромский институт (филиал) федерального государственного бюджетного
образовательного учреждения высшего образования «Владимирский
государственный университет имени Александра Григорьевича и Николая
Григорьевича Столетовых»*

602264, г. Муром, Владимирская обл., ул. Орловская, 23

E-mail: kanunovae@list.ru

Обучение нейронной модели для определения опухолей на маммографических снимках

В современном мире машинное обучение – это мощный инструмент, который помогает значительно повысить точность диагностики заболеваний и обнаружить патологии, которые могут быть пропущены при использовании традиционных методов. Оно автоматизирует рутинные процессы, освобождая время врачей для более сложных задач, и улучшает качество ухода за пациентами.

Благодаря способности обрабатывать большие объемы информации, алгоритмы машинного обучения улучшают исследовательскую работу и клиническую практику, а так же позволяют разработать индивидуальные планы лечения на основе индивидуальных данных о каждом пациенте. С помощью анализа данных и прогнозирования, машинное обучение способствует раннему выявлению заболеваний, что повышает шансы на успешное лечение. Оно так же может снизить затраты на здравоохранение за счет оптимизации диагностических процессов, делая медицинские услуги более доступными. Кроме того, непрерывное обучение и адаптация алгоритмов машинного обучения к новым данным обеспечивает постоянное улучшение их эффективности и соответствие актуальным медицинским стандартам.

В докладе представлены несколько основных этапов обучение модели на маммографических снимках для выявления опухолей:

1) Сначала необходимо собрать и подготовить данные, состоящие из большого количества маммографий с соответствующими метками, которые указывают наличие или отсутствие опухолей;

2) Следующим шагом является предобработка изображений, включающая в себя изменение размера, нормализацию и аугментацию (Рис.1);

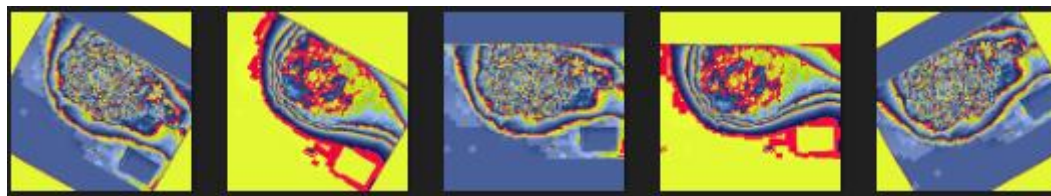


Рисунок 1 – Предобработанные снимки.

3) Далее – разработка модели. Для задач классификации изображений, включая обнаружение опухолей, чаще всего используются сверточные нейронные сети. Эти сети хороши для работы с 2D-данными, такими как маммографические изображения. Описание архитектуры нейронной сети:

1. Входной слой:

- Нормализует изображения, преобразуя значения пикселей из диапазона $[0, 255]$ в $[0, 1]$ для улучшения сходимости модели.

2. Сверточные слои:

- Первый сверточный слой извлекает базовые признаки (например, края) с помощью 32 фильтров.

- Слой подвыборки уменьшает размер карты признаков.

- Второй сверточный слой с 64 фильтрами для выявления более сложных признаков.

- Еще один слой подвыборки.

- Третий сверточный слой с 128 фильтрами для извлечения высокоуровневых признаков.
- Последний слой подвыборки перед полносвязными слоями.

3. Полносвязные слои:

- Преобразует двумерные данные в одномерный вектор для дальнейшей обработки.
- Полносвязный слой с 128 нейронами и функцией активации, добавляющей нелинейность.

4. Выходной слой:

- Один нейрон с функцией активации sigmoid, который выдает вероятность принадлежности к положительному классу, что подходит для бинарной классификации.

5. Компиляция модели:

- Использует оптимизатор Adam и функцию потерь бинарной кросс-энтропии, что делает модель подходящей для бинарной классификации.

4) На следующем этапе происходит обучение модели. Важно разделить данные на обучающую, валидационную и тестовую выборки.

5) Заключительным шагом является оценка модели. После обучения важно провести тестирование и оценку производительности метриками оценки: accuracy, precision, recall и F1-score.

На рисунке 2 представлены результаты работы нейронной сети.

```
Accuracy: 0.5451
Precision: 0.5407
Recall: 0.9858
F1-score: 0.6983
```

Рисунок 2 – результат работы нейронной сети.

Большой Recall в контексте оценки моделей машинного обучения означает, что модель хорошо справляется с задачей выявления всех положительных примеров из набора данных. Recall измеряет, какую долю из всех реальных положительных случаев модель правильно классифицировала как положительные.

Recall – это количество истинно положительных предсказаний (верно классифицированные положительные примеры), а FN (False Negatives) — количество ложно отрицательных предсказаний (положительные примеры, ошибочно классифицированные как отрицательные). Высокий Recall может указывать на то, что модель эффективно выявляет положительные примеры, что критично в задачах, где пропуск положительного случая может иметь серьезные последствия, например, в медицинской диагностике или обнаружении мошенничества. Однако, при этом стоит отметить, что модель может иметь низкий уровень точности, что означает, что она также может делать много ложных срабатываний, то есть неправильно классифицировать отрицательные примеры как положительные. Это может быть допустимо в ситуациях, когда более важна полнота.

Машинное обучение представляет собой современный инструмент в области медицины, который способен не только повысить точность диагностики заболеваний, но и оптимизировать рабочие процессы в сфере здравоохранения. Особенное внимание стоит уделить процессу предварительной обработки данных, поскольку грамотно подготовленные маммографические снимки являются ключом к успешному обучению модели. Этапы, начиная от сбора и подготовки данных и заканчивая обучением и оценкой модели, формируют основу для надежной и эффективной диагностики опухолей. В итоге, внедрение технологий машинного обучения обещает значительные изменения в подходах к выявлению и лечению заболеваний, улучшая качество медицинского обслуживания и снижая на него затраты.

Литература

1. Автоматизированная обработка и анализ маммографических снимков: монография/ С.С. Садыков, Ю.А. Буланова, Е.А. Захарова; Владим. Гос. Ун-т имени Александра Григорьевича и Николая Григорьевича Столетовых. – Владимир: Изд-во ВлГУ, 2014.
2. Извлечение знаний методами машинного обучения : учебное пособие по курсам «Модели и методы инженерии знаний», «Методы машинного обучения» / А. Н. Целых, Э. М.

Котов. — Ростов-на-Дону, Таганрог : Издательство Южного федерального университета, 2022. — 105 с. — ISBN 978-5-9275-4215-4. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/131448.html>

3. Цифровая обработка изображений: учебное пособие / Е. А. Шефер. — Санкт-Петербург: Санкт-Петербургский государственный университет промышленных технологий и дизайна, 2019. — 100 с. - <https://www.iprbookshop.ru/102493.html>

4. Цифровая обработка изображений / Гонсалес Рафаэл, Вудс Ричард ; перевод Л. И. Рубанов, П. А. Чочиа ; под редакцией П. А. Чочиа. — Москва : Техносфера, 2012. — 1104 с. — ISBN 978-5-94836-331-8. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/26905.html>

Глазунов Д. Д.

Научный руководитель: к.т.н. Симаков Р.А.

*Муромский институт (филиал) федерального государственного бюджетного
образовательного учреждения высшего образования «Владимирский
государственный университет имени Александра Григорьевича и Николая
Григорьевича Столетовых»*

602264, г. Муром, Владимирская обл., ул. Орловская, 23

Сравнительный анализ библиотек для парсинга JSON в Python: выбор оптимального решения на основе производительности

Обработка данных в формате JSON является одной из ключевых задач в современных Python-приложениях: от веб-разработки и API-интеграций до анализа данных и конфигурационных файлов. При выборе библиотеки для парсинга JSON ключевыми факторами становятся скорость обработки и эффективность использования памяти, особенно при работе с большими объемами данных.

В данной статье приводится сравнительный анализ производительности следующих библиотек:

- Стандартная библиотека Python json[1];
- orjson - парсер на Rust[4]
- msgspec - парсер с поддержкой схем[3];
- json_stream - потоковый парсер[2].

Исследование фокусируется на двух ключевых метриках: скорость парсинга и использование памяти.

Для оценки применяется специально разработанный тест.

Результаты помогут выбрать подходящую библиотеку для парсинга метрик системы мониторинга.

1. Методология исследования

Для объективного сравнения библиотек использовались тестовые данные, приближенные к реальным. Тестовые данные состоят из ста тысяч объектов структуры, описывающей метрики запросов в базе данных.

Для сравнения времени парсинга используется профилирование выполнения кода с помощью встроенной библиотеки cPython, а для сравнения использования памяти - встроенная библиотека tracemalloc.

В качестве окружения был выбран интерпретатор Python версии 3.11, операционная система Ред ОС 8 без графической оболочки и центральный процессор на 4 ядра. Окружение было запущено на виртуальной машине.

2. Анализ производительности

Библиотека	Время парсинга	Потребление памяти
json	947 мс	178 Мб
orjson	735 мс	239 Мб
msgspec	757 мс	225 Мб
json_stream	32 часа	30 Мб

Проведённый сравнительный анализ библиотек парсинга JSON в Python выявил значительные различия в производительности и потреблении памяти.

Стандартная библиотека json оказалась не самой медленной, но и не самой быстрой. Её преимущество — доступность без установки дополнительных пакетов.

Библиотеки `orjson` и `msgspec` продемонстрировали близкие результаты по скорости, но при этом потребляли больше памяти, чем стандартный `json`. Это может быть критично в средах с ограниченными ресурсами.

Библиотека `json_stream` показала крайне низкую скорость (32 часа против сотен миллисекунд у других библиотек), однако его главное преимущество — минимальное потребление памяти (всего 30 МБ). Это делает его единственным вариантом для обработки огромных JSON-файлов, так как потребление памяти постоянно.

Таким образом, стандартная библиотека `json` показала неожиданно конкурентные результаты (947 мс), что ставит под сомнение необходимость использования альтернативных библиотек. Библиотеки `orjson` (735 мс) и `msgspec` (757 мс) действительно быстрее стандартного решения, но выигрыш составляет всего 20-25%, а не кратные величины. `json_stream` продемонстрировал катастрофически низкую производительность (32 часа), что полностью исключает его использование в задачах, где важна скорость.

Литература

1. JSON encoder and decoder // Python URL: <https://docs.python.org/3.11/library/json.html>.
2. json-stream // PyPI URL: <https://pypi.org/project/json-stream/>
3. msgspec // jcristharif URL: <https://jcristharif.com/msgspec/>
4. orjson // PyPI URL: <https://pypi.org/project/orjson/2.0.1/>

Голубев Н.В., Костюхина А.М., Гудзь К.С., Земсков С.А., Мурындин Д.С.
Научный руководитель: к.т.н. доцент Канунова Е.Е.
*Муромский институт (филиал) федерального государственного бюджетного
образовательного учреждения высшего образования «Владимирский
государственный университет имени Александра Григорьевича и Николая
Григорьевича Столетовых»
602264, г. Муром, Владимирская обл., ул. Орловская, 23
E-mail: kostyuhinastasia@yandex.ru*

Разработка телеграм-бота для приемной комиссии Муромского института Владимирского государственного университета

В современном мире технологии играют ключевую роль в оптимизации и автоматизации различных процессов, в том числе в сфере образования. Одной из актуальных задач для учебных заведений является упрощение взаимодействия с потенциальными абитуриентами. Это становится особенно важным в условиях растущей конкуренции и увеличения потока заявлений, когда каждая подробность может оказать значительное влияние на выбор учебного заведения.

Создание телеграм-бота для приемной комиссии института открывает новые возможности, позволяя не только автоматизировать процессы, но и облегчить доступ абитуриентов к необходимой информации. Телеграм, как один из самых популярных мессенджеров мира, предоставляет уникальную платформу для реализации сервисов взаимодействия с пользователями, что способствует повышению вовлеченности и удовлетворенности абитуриентов.

Разработанный телеграм-бот предлагает широкий набор функций для абитуриентов. Он предоставляет актуальную информацию о правилах поступления, необходимых документах, сроках подачи заявлений и стоимости обучения. Пользователи могут узнать, какие специальности доступны для поступления в зависимости от предметов, сданных на ЕГЭ, а также получить сведения о направлениях обучения, которые востребованы на рынке труда. Кроме того, бот позволяет абитуриентам записываться на вступительные испытания, а также информирует о возможностях заселения в общежитие.

Данная система была разработана с использованием языка программирования Python и библиотеки Telebot, что обеспечивает гибкость для создания интерактивных сценариев взаимодействия. [1] Бот способен обрабатывать входящие сообщения, предоставлять ответы на разнообразные запросы и даже реализовывать сложные алгоритмы взаимодействия, делая процесс общения максимально удобным и понятным для пользователей.

Важно отметить, что на данный момент бот уже разработан и готов к внедрению в учебное заведение. В процессе эксплуатации будут проводиться регулярные доработки, основанные на обратной связи от абитуриентов, их родителей и членов приемной комиссии. Это позволит не только повышать качество сервиса, но и настраивать его под актуальные потребности пользователей, создавая таким образом систему, которая действительно отвечает вызовам времени.

Литература

1. Документация PyTelegramBotAPI [Электронный ресурс] // URL: <https://pytba.readthedocs.io/ru/latest/index.html> (дата обращения 05.04.2025)

Елизарова М.А.

Научный руководитель: к.т.н. Симаков Р.А.

*Муромский институт (филиал) федерального государственного бюджетного
образовательного учреждения высшего образования «Владимирский
государственный университет имени Александра Григорьевича и Николая
Григорьевича Столетовых»*

602264, г. Муром, Владимирская обл., ул. Орловская, 23
seagirl1991@yandex.ru

Разработка универсального сервера на Node.js для СУБД Ред База Данных

Традиционные подходы к работе с базами данных предполагают создание отдельных серверных приложений для каждой базы. В таких случаях клиентские приложения подключаются к конкретной базе данных через специализированный сервер, жестко привязанный к ее структуре и настройкам. Это усложняет архитектуру, увеличивает затраты на разработку и поддержку, а также снижает адаптивность системы при изменении данных.

Разработка универсального сервера на Node.js решает данную проблему, так как универсальный сервер является эффективным инструментом для работы с различными базами данных внутри СУБД Ред База Данных.

Динамическое подключение делает систему гибкой, поскольку клиент сам определяет, с какой базой данных работать, передавая необходимую информацию в запросе. Использование REST API позволяет клиентам выполнять SQL-запросы, не производя дополнительных настроек серверной части. Благодаря такому подходу разработчики могут сосредоточиться на разработке клиентской части и не беспокоиться о реализации серверных решений для каждой базы данных, упрощая развертывание и администрирование серверной инфраструктуры.

Сервер выполняет операции с базой данных и возвращает результаты в формате JSON, что упрощает взаимодействие с различными web-приложениями.

Для реализации сервера используются следующие технологии:

- node.js – серверная среда выполнения JS,
- express.js – фреймворк для создания REST API,
- node-firebird – драйвер, обеспечивающий подключение к Ред Базе Данных.

Литература

1. <https://rdb.red-soft.ru/> — Официальный сайт СУБД Ред База Данных
2. <https://nodejs.org/en> — Официальный сайт node.js
3. <https://expressjs.com/ru/> — Официальный сайт express.js
4. <https://www.npmjs.com/package/node-firebird> — Руководство пользователя для node-firebird

Есин Д.И.

Научный руководитель: к.т.н. Комкова С. В.

*Муромский институт (филиал) федерального государственного бюджетного
образовательного учреждения высшего образования «Владимирский
государственный университет имени Александра Григорьевича и Николая
Григорьевича Столетовых»*

602264, г. Муром, Владимирская обл., ул. Орловская, 23

Разработка симулятора экономических моделей

Эффективное управление современными предприятиями в значительной степени зависит от точности анализа и прогнозирования экономических процессов. В условиях цифровизации экономики особенно актуальной становится задача автоматизации процессов принятия решений на основе достоверного математического и имитационного моделирования.

Целью данной работы является создание полноценного специализированного программного комплекса — симулятора экономических моделей, предназначенного для автоматизации построения, анализа и оценки математических и имитационных экономических моделей с применением современных технологий искусственного интеллекта. Разрабатываемый программный продукт ориентирован на экономистов, аналитиков и специалистов в области стратегического управления бизнесом.

Планируемый функционал итогового продукта включает:

- Возможность создания, настройки и управления экономическими и имитационными моделями;
- Проведение математических расчётов и симуляций на основе созданных моделей;
- Визуализацию результатов расчётов в виде графиков и диаграмм;
- Экспорт отчётов и результатов моделирования в популярные форматы (Excel, PDF);
- Автоматический анализ финансовой документации и отчётности с использованием обученных AI-моделей;
- Генерацию рекомендаций и предложений по улучшению экономических моделей на основе AI-анализа;
- Механизмы защиты данных, авторизации пользователей и управления сессиями.

Создание такого симулятора позволит существенно автоматизировать трудоёмкие процессы обработки данных, повысить точность и оперативность принятия управленческих решений, а также снизить влияние человеческого фактора на результаты анализа экономических данных. Реализация данного программного комплекса будет способствовать повышению эффективности стратегического управления предприятиями.

В качестве интегрированной среды разработки был выбран IntelliJ IDEA. Среда обладает возможностями и расширенной поддержкой при работе с Java, Spring framework, обладает встроенным статическим анализатором кода, позволяет управлять проектами и работать с базами данных и инструментарий для профилирования и отладки. Такой набор функций позволяет вести удобную разработку и избегать сложностей с развертыванием и интеграцией продукта.

Земсков С.А.

Научный руководитель: к.т.н., доцент Щаников С.А.

*Муромский институт (филиал) федерального государственного бюджетного
образовательного учреждения высшего образования «Владимирский
государственный университет имени Александра Григорьевича и Николая
Григорьевича Столетовых»*

602264, г. Муром, Владимирская обл., ул. Орловская, 23

E-mail: zemskovserj2006@gmail.com

Перспективы создания систем нейрорегуляции колесными робототехническими комплексами с применением вычислителей на базе мемристивных устройств

Управление робототехническими системами с помощью нейроинтерфейсов — это перспективное направление [1], открывающее новые возможности для людей с ограниченными возможностями здоровья [2], а также для автоматизации сложных задач. Однако существующие системы сталкиваются с рядом проблем: высокая задержка обработки сигналов, вследствие не достаточного быстродействия мобильных вычислителей, необходимость периодической сложной калибровки и т.д. Создание энергоэффективных и быстродействующих систем нейрорегуляции для колесных роботов требует интеграции передовых технологий, таких как нейросетевые алгоритмы, мемристивные вычислители и методы обработки сигналов в реальном времени.

Современные нейроинтерфейсы обрабатывают различные биологические сигналы — электроэнцефалограммы (ЭЭГ), электромиограммы (ЭМГ), кортикограммы и другие [3]. Для их обработки применяются различные алгоритмы машинного обучения, такие как сверточные нейросети (CNN) или рекуррентные нейросети (RNN), которые классифицируют намерения пользователя (например, «двигаться вперед», «повернуть направо» или «ехать прямо»).

Перспективным подходом к аппаратной реализации нейросетевых моделей является использование мемристивных устройств. Анализ научно-технических публикаций показывает, что вычислители на базе мемристоров обладают низким энергопотреблением, при этом сохраняя высокие показатели быстродействия и точности работы нейросетевых моделей [4].

Таким образом, в системах нейрорегуляции колесными роботами предлагается использовать мемристивные устройства для запуска моделей нейросетей непосредственно на борту. Управляющие команды передаются на колесный робот, обрабатываются мемристивным вычислителем и управляют актуаторами. Такие мемристивные системы нейрорегуляции колесными робототехническими комплексами могут применяться в медицине: нейрорегулируемые инвалидные коляски для пациентов с параличом, роботы-реабилитаторы для восстановления двигательных функций.

Литература

1. Нейроинтерфейс: сайт. — URL: <https://cmi.to/нейроинтерфейс/> (дата обращения: 02.04.2025)
2. Моторная нейрореабилитация: сайт. — URL: <https://neurobotics.ru/neurophysiology/nejroreabilitacziya/> (дата обращения: 05.04.2025)
3. Ритм ЭЭГ: сайт. — URL: <https://cmi.to/ритмы-ээг/> (дата обращения: 02.04.2025)
4. На пути к реализации высокопроизводительных вычислений в памяти на основе мемристивной электронной компонентной базы / А. Н. Михайлов, Е. Г. Грязнов, В. И. Лукоянов [и др.] // Физмат. — 2023. — Т. 1, № 1. — С. 42-64. — DOI 10.56304/S2949609823010021. — EDN HTSRZQ.

Зуев Е.И.

Научный руководитель: к.т.н., доцент Еремеев С.В.

Муромский институт (филиал) федерального государственного бюджетного образовательного учреждения высшего образования «Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»

602264, г. Муром, Владимирская обл., ул. Орловская, 23

E-mail: zuevgenyua85@gmail.com

Разработка библиотеки алгоритмов для работы с изображениями

Современные технологии обработки изображений стали неотъемлемой частью цифровой эпохи, проникнув во множество сфер человеческой деятельности. Одним из важных аспектов работы с изображениями является возможность их многослойной обработки, что позволяет создавать сложные композиции и применять различные эффекты независимо для каждого слоя.

Многослойная организация изображений представляет собой мощный инструмент для создания сложных графических композиций. Этот подход, ставший стандартом в профессиональных графических редакторах, позволяет применять различные эффекты и трансформации к отдельным элементам изображения независимо друг от друга. Такая архитектура обеспечивает не только гибкость в работе, но и сохраняет возможность возврата к предыдущим стадиям обработки, что особенно ценно в творческом процессе. В контексте обработки изображений особое значение имеют базовые, но фундаментальные операции: инвертирование цветовых каналов, масштабирование с сохранением детализации, пространственные преобразования, включая трехмерные вращения. Эти алгоритмы, будучи относительно простыми в реализации, тем не менее требуют тщательной оптимизации для обеспечения высокой производительности, особенно при работе с изображениями большого разрешения или в режиме реального времени.

Актуальность темы обусловлена растущими требованиями к функциональности графических редакторов и необходимости оптимизации алгоритмов обработки изображений. Разработка собственной библиотеки даёт возможность глубже понять принципы работы таких систем и создать гибкое решение для конкретных задач.

Цель работы: разработка статической библиотеки, предоставляющей набор алгоритмов для обработки изображений.

В данной работе была реализована статическая библиотека в среде Visual Studio 2022, что обеспечило удобство интеграции и тестирования алгоритмов. Она включает в себя алгоритмы для обработки изображений, а также графическое приложение на Windows Forms, демонстрирующее её возможности. Основное внимание уделено таким функциям, как инвертирование цветов, 3D-вращение, масштабирование и работа со слоями.

Реализация функций для обработки изображений была осуществлена по следующим алгоритмам [1-3]:

1. Инвертирование цветов — замена каждого цветового канала (R, G, B) на его дополнение до 255, что создаёт эффект негатива.

2. Масштабирование — изменение размера изображения с использованием билинейной интерполяции для сохранения плавности границ.

3. 3D-вращение — применение матричных преобразований для поворота изображения вокруг осей X, Y и Z с учётом перспективы.

4. Работа со слоями — реализация системы управления слоями, позволяющая накладывать несколько изображений друг на друга и применять к ним отдельные преобразования. В качестве примера управления слоями можно привести функционал, аналогичный QGIS или Photoshop, где каждый слой обрабатывается независимо.

Для визуализации работы алгоритмов создано Windows Forms-приложение с интуитивно понятным интерфейсом, включающим панель инструментов для основных операций (инвертирование, масштабирование, вращение), трекбары для точной настройки параметров преобразований, а также панель управления слоями с возможностью их переупорядочивания,

включения/выключения и применения индивидуальных трансформаций. Интерфейс, представленный на рис. 1 спроектирован с учетом эргономики: часто используемые функции вынесены на главную панель, а дополнительные параметры доступны через контекстные меню, что делает работу с приложением удобной как для начинающих, так и для опытных пользователей.

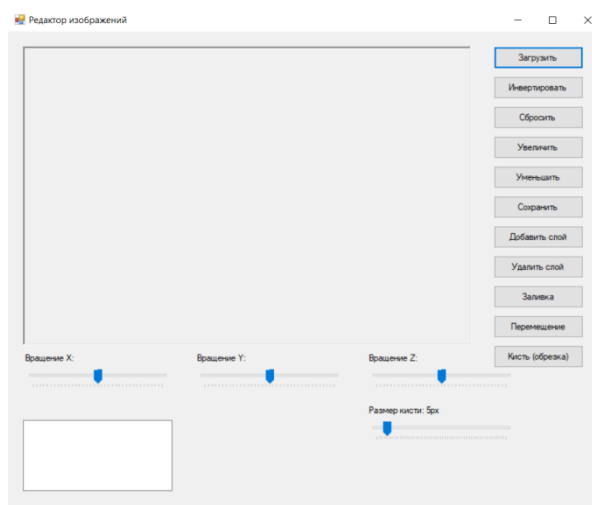


Рис. 1 – Интерфейс Windows Forms-приложения

Интерфейс приложения включает в себя: слайдеры для вращения, изменения размера кисти, кнопки масштабирования, загрузки, инвертирования, сброса, сохранения, добавления/удаления слоя, заливки, перемещения, кисти (обрезки), список слоёв.

Примеры изображений, которые могут быть реализованы с помощью данной библиотеки приведены на рис. 2 а) пример изображения, без применения дополнительных функций, состоящее из нескольких слоев; б) пример изображения, с применением дополнительных функций (инвертирование, 3D-вращение, добавление нового слоя, кисть(обрезка), заливка, масштабирование);

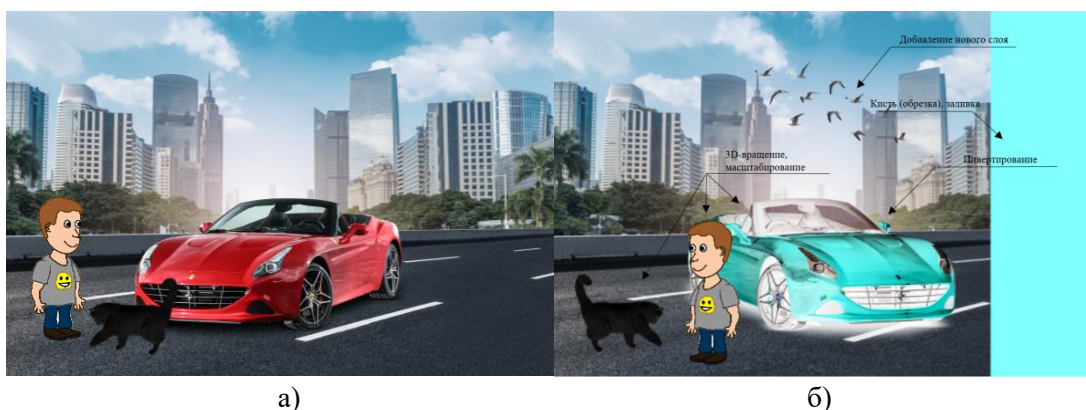


Рис. 2 – Примеры изображений, реализованные с помощью библиотеки

Данный проект демонстрирует основы работы с изображениями и может служить базой для более сложных решений в области компьютерной графики. Реализованные функции могут быть расширены для поддержки дополнительных эффектов и оптимизации производительности.

Литература

1. Программирование графики на C++. Теория и примеры : учебное пособие / В.И. Корнеев, Л.Г. Гагарина, М.В. Корнеева. — Москва : ИНФРА-М, 2023. — 517 с. + Доп. материалы [Электронный ресурс]. — (Высшее образование: Бакалавриат). — РОГ 10.12737/23113.

Секция 3. Разработка информационных систем

2. Лафоре Р. Объектно-ориентированное программирование в C++. Классика Computer Science. 4-е изд. — СПб.: Питер, 2022. — 928 с.: ил. — (Серия «Классика computer science»). ISBN 978-5-4461-0927-2.

3. Поляков А. Ю. Методы и алгоритмы компьютерной графики в примерах на Visual C++. — СПб.: БХВ-Петербург, 2002. — 416 с.: ил. ISBN 5-94157-136-4

Научный руководитель: к.т.н., доцент Щаников С.А.
Муромский институт (филиал) федерального государственного бюджетного
образовательного учреждения высшего образования «Владимирский
государственный университет имени Александра Григорьевича и Николая

Григорьевича Столетовых»

602264, г. Муром, Владимирская обл., ул. Орловская, 23

E-mail: sashakirilin000@gmail.com

Обзор архитектур нейроморфных процессоров и особенностей их реализации на ПЛИС

Целью данной работы является исследование архитектур нейроморфных процессоров и особенностей их реализации на программируемых логических интегральных схемах (ПЛИС) [1]. В работе рассматриваются существующие нейроморфные процессоры и их преимущества перед традиционными архитектурами, а также особенности их реализации на ПЛИС для достижения высокой производительности и энергоэффективности.

В рамках исследования проанализированы несколько современных нейроморфных процессоров, таких как Intel Loihi [2], IBM NorthPole [3], IBM TrueNorth [4], Tianjic [5] и российский процессор «Алтай» [6]. Loihi 2 и NorthPole характеризуются высокой степенью гибкости и производительности, однако их разработка и внедрение сопряжены с существенными техническими и ресурсными затратами. Tianjic, обладая гибридной архитектурой, которая поддерживает как искусственные нейронные сети (ANN), так и спайковые нейронные сети (SNN), представляет собой многофункциональное решение, применимое для широкого диапазона задач в области искусственного интеллекта. Алтай демонстрирует высокие показатели энергоэффективности и масштабируемости, однако его производительность ограничена использованием устаревшего технологического процесса. TrueNorth и Loihi 1, являясь значимым этапом в эволюции нейроморфных технологий, в настоящее время уступают современным аналогам по производительности.

Каждый из рассмотренных нейроморфных процессоров обладает уникальными преимуществами и специализированными функциями, что определяет их применимость для различных задач. Тем не менее, Tianjic выделяется благодаря своей универсальности, гибридной архитектуре и высокой производительности. Способность одновременно обрабатывать как традиционные нейронные сети (ANN), так и импульсные нейронные сети (SNN) делает его оптимальным решением для широкого спектра приложений, включая компьютерное зрение, робототехнику и обработку данных в режиме реального времени.

Таким образом, нейроморфные процессоры демонстрируют высокий потенциал для энергоэффективных вычислений. В докладе будут подробно рассмотрены все характеристики данных нейропроцессоров, а также особенности архитектур и их реализации на ПЛИС.

Литература

1. FPGA Computing [Электронный ресурс]//URL: <https://ieeexplore.ieee.org/document/9473061> (Дата обращения 10.10.24).

2. Intel-Loihi [Электронный ресурс]//URL: <https://www.intel.com/content/www/us/en/research/neuromorphic-computing-loihi-2-technology-brief.html> (Дата обращения 10.10.24).

3. IBM NorthPole [Электронный ресурс] // URL: <https://research.ibm.com/blog/northpole-ibm-ai-chip> (Дата обращения 10.10.24).

4. IBM TrueNorth [Электронный ресурс] // URL: <https://research.ibm.com/publications/truenorth-design-and-tool-flow-of-a-65-mw-1-million-neuron-programmable-neurosynaptic-chip> (Дата обращения 10.10.24).

5. Tianjic: A unified and scalable chip bridging spike-based and continuous neural computation [Электронный ресурс]//URL: https://www.researchgate.net/publication/339248581_Tianjic_A_Unified_and_Scalable_Chip_Bridging_Spike-Based_and_Continuous_Neural_Computation (Дата обращения 10.10.24).

Колоколов К.А.

Научный руководитель: старший преподаватель Булаев А.В.
Муромский институт (филиал) федерального государственного бюджетного образовательного учреждения высшего образования «Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»
602264, г. Муром, Владимирская обл., ул. Орловская, 23
Email: kirillkolokolov@bk.ru

Система мультисинхронизации данных

Работа с электронной информацией уже давно стала неотъемлемой частью профессиональной деятельности в различных сферах. При этом сотрудники могут выполнять свою деятельность не только находясь на рабочем месте в офисе, но и дома. Поэтому является актуальной задача синхронизации файлов между несколькими рабочими местами.

Организация процесса обмена может быть организована разными подходами:

- напрямую между двумя персональными компьютерами (рис. 1);
- посредством некоторого промежуточного хранилища (рис. 2).



Рис. 1 — Прямая схема синхронизации между персональными компьютерами



Рис. 2 — Схема синхронизации с использованием промежуточного хранилища

Вне зависимости от выбранного метода синхронизации обмен данными будет производиться между двумя объектами хранения.

Традиционные методы синхронизации зачастую передают файлы целиком, что приводит к избыточному использованию сетевого трафика и увеличению времени передачи. Разработка оптимизированной системы, использующей инкрементальную синхронизацию и кэширование метаданных, позволяет минимизировать нагрузку на сеть и ускорить процесс передачи данных.

Цель работы - создание открытой системы облачной синхронизации файлов, обеспечивающей надежное хранение и передачу данных с возможностью инкрементального обновления файлов и кэширования метаданных.

Основные задачи:

- разработка алгоритмов инкрементальной синхронизации, позволяющих передавать только измененные файлы;
- внедрение механизма кэширования метаданных для быстрого обнаружения изменений и сокращения числа операций чтения/записи;
- реализация поддержки различных типов хранилищ, включая локальные диски, сетевые ресурсы и облачные сервисы (FTP, SFTP, FTPS);
- обеспечение надежной проверки целостности переданных данных;
- разработка пользовательского интерфейса для настройки синхронизации и мониторинга процессов.

Синхронизация каталогов может выполняться с участием пользователя (когда решение о приоритетности направления синхронизации выбирается вручную) или в автоматическом режиме (на основании сформированных правил в системе).

Алгоритм синхронизации сводится к следующей последовательности шагов:

- 1) сформировать список файлов на стороне А;
- 2) сформировать список файлов на стороне Б;
- 3) выполнить сличение списков файлов А и Б;
- 4) в соответствии с приоритетностью сформировать сценарий синхронизации с определением действий, применяемых к файлам составляющих расхождение списков А и Б;
- 5) выполнить файловые операции.

В данном алгоритме под сторонами А и Б подразумеваются объекты синхронизации — как правило это каталоги на локальных компьютерах или удаленных ресурсах.

При выполнении этапов 1 и 2 возникает проблема — выявление локальных изменений на каждой из сторон. Выполнение сканирования каждой из синхронизируемых сторон, позволяет получить текущую структуру файлов и каталогов, но не позволяет выявить изменения, которые претерпела файлово-каталожная структура (ФКС) объекта синхронизации.

Для решения этой задачи, предлагается для каждого из объектов синхронизации хранить срез его структуры на определенный момент времени.

Для выявления локальных изменений в объекте синхронизации требуется сравнить его текущую ФКС и ФКС его последнего среза. Алгоритм сравнения будет включать следующие шаги:

- 1) сформировать список текущей файлово-каталожной структуры объекта А - ФКС_{ТА};
- 2) загрузить список последнего среза файлово-каталожной структуры объекта А — ФКС_{СА};
- 3) для файла 1 в ФКС_{ТА} находим соответствующий ему в ФКС_{СА};
- 4) если абсолютные пути ФКС_{ТА}1 и ФКС_{СА}1 совпадают, то выполняется проверка их атрибутов, при совпадении атрибутов можно утверждать, что ФКС_{ТА}1 = ФКС_{СА}1 (действия не требуется);
- 5) при расхождении атрибутов — ФКС_{ТА}1 <> ФКС_{СА}1 — файл был изменен позднее последней синхронизации (включить в список синхронизации);
- 6) если существует путь ФКС_{ТА}1, но путь отсутствует для ФКС_{СА}1, то файл 1 был создан позднее последней синхронизации (включить в список синхронизации);
- 7) если отсутствует путь ФКС_{ТА}1, но существует путь ФКС_{СА}1, то файл 1 был удален позднее последней синхронизации (включить в список синхронизации).

Для выявления различий файлов требуется выполнить сравнение их атрибутов, поскольку атрибутивная информация должна будет сохраняться после выполнения каждой синхронизации в файле ФКС_С, то ее объем должен быть минимизирован. Достаточный набор атрибутов составляет: полный путь до файла относительно корня объекта синхронизации и дата последнего изменения файла.

Помимо операций изменения, создания и удаления файлов и каталогов пользователи используют операции переименования, как файлов так и каталогов. При переименовании каталога изменяется значение атрибутов не одного, а группы файлов и каталогов образующих дерево. В других системах синхронизации это выполняется путем удаления файлов и каталогов в поддереве до переименования и выполнение копирования файлов и каталогов поддерева после переименования.

При небольшом объеме файлов — это не представляет проблемы, но при большом количестве различающихся файлов или при их большом объеме это может привести к увеличению времени синхронизации, объема сетевого трафика и приводить к большему износу носителей устройств участвующих в синхронизации. Поэтому является актуальным разработка алгоритма выявления переименования каталогов и применение операции переименования в удаленном хранилище.

Однозначно установить факт переименования файла возможно только при хранении расширенной атрибутивной информации, которая позволит характеризовать содержимое файла, например, хеш-функцию. Если файл был только переименован, без изменения содержимого, то выявить его можно по проверке хеш-функции. Если помимо переименования изменялось и содержимое файла, то выявить его не представляется возможным.

Секция 3. Разработка информационных систем

Установление факта переименования каталога требует анализа субдерева ФКС и может быть подтверждено 100% соответствием файлов субдерева или наличием среди них новых, подлежащих синхронизации, или удаляемых файлов.

При реализации проекта используются следующие методы и технологии:

- использование собственного формата файлов для хранения информации о файлово-каталожной структуре объекта синхронизации;
- оптимизация передачи данных за счет многопоточной обработки и параллельного выполнения файловых операций;
- оптимизация обмена данными за счет применения операций переименования файлов и каталогов;
- применение языков программирования Python и C++.

Разрабатываемая система обеспечивает высокую эффективность синхронизации данных за счет инкрементального обновления и использования кэширования. Решение подходит для использования как в локальных, так и в распределенных средах, снижая нагрузку на сеть и ускоряя процесс передачи файлов. В перспективе возможна интеграция с дополнительными облачными сервисами и реализация мобильного клиента.

Разработка информационной системы для оценки знаний студентов при выполнении практических работ по информатике

Современные требования к образовательным системам предъявляют высокие ожидания в области автоматизации процессов контроля и оценки знаний. Одним из наиболее эффективных решений является разработка специализированных информационных систем с индивидуальной проверкой практических заданий. В рамках данной работы реализована система, предназначенная для оценки знаний студентов при выполнении практических заданий по информатике.

Система разработана с использованием объектно-ориентированного подхода, что обеспечивает высокую модульность и масштабируемость проекта. Серверная часть системы реализована с использованием фреймворка Spring Boot на языке Java. Клиентская часть создана с применением JavaFX. В качестве среды разработки использовалась IntelliJ IDEA. Хранение данных осуществляется в СУБД RedDataBase.

Программа включает окно авторизации, после которого пользователю предоставляется доступ к основному окну с навигационным меню и основной рабочей областью.

Для преподавателя предусмотрены функции просмотра и редактирования списков учебных групп, управления списками студентов, создания и назначения практических работ, а также просмотра результатов выполнения заданий.

Со стороны студента реализована возможность просмотра списка назначенных практических работ, получения индивидуально сгенерированных заданий и автоматической проверки выполнения с отображением результатов.

Основной особенностью является уникальность заданий, генерируемых для каждого студента, а также автоматическая проверка корректности выполнения, что минимизирует влияние человеческого фактора и повышает объективность оценки.

Интерфейс клиента разработан с акцентом на удобство и интуитивную понятность, что позволяет использовать систему без предварительного обучения.

Литература

1. Васильев, А. Н. Java: основы программирования / А. Н. Васильев. — СПб.: Наука и Техника, 2021. — 480 с.
2. Шилдт Г. Java. Полное руководство. 11-е изд. — М.: Вильямс, 2022.
3. Oracle. JavaFX Documentation. [Электронный ресурс]. URL: <https://openjfx.io> (дата обращения: 01.04.2025).
4. Официальная документация RedDataBase. [Электронный ресурс]. URL: <https://reddatabase.ru/ru/documentation> (дата обращения: 01.04.2025).

Королев Л. Я.

Научный руководитель: к.т.н., доцент Щаников С. А.
*Муромский институт (филиал) федерального государственного бюджетного
образовательного учреждения высшего образования «Владимирский
государственный университет имени Александра Григорьевича и Николая
Григорьевича Столетовых»*
602264, г. Муром, Владимирская обл., ул. Орловская, 23
Email: madimtor@ya.ru

Разработка и исследование прототипа-демонстратора работы нейронных сетей на базе мемристивных устройств в архитектуре кроссбара

В настоящее время основной аппаратной платформой для запуска моделей искусственных нейронных сетей являются графические процессоры GPU и нейропроцессорные ядра NPU. Они имеют большое энергопотребление (десятки и сотни ватт) и работают на высоких частотах (гигагерцы), потому что построены по архитектуре Джона фон Неймана с разделенной памятью и вычислителем.

Нейронные сети живых организмов потребляют милливатты и работают на низких герцовых частотах. Результаты современных исследований показывают, что создание нейронных сетей на базе мемристоров позволяет реализовывать вычисления в памяти, которые являются более биоподобными, так как вычислитель и память являются одним устройством.

Мемристор – это электронный компонент, который может менять свое сопротивление при воздействии электрического тока и сохранять это записанное значение длительное время [1].

В искусственных нейронных сетях мемристор может быть использован как аппаратный искусственный синапс. Для применения в нейронных сетях мемристоры объединяются в массивы. В таком случае матричное умножение, как одна из наиболее частых и затратных операций в нейронных сетях, выполняется естественным образом на основе законов Ома и Кирхгофа, что позволяет повысить их скорость работы и энергоэффективность вычислений [2].

В докладе рассмотрены: прототип-демонстратор, принцип его работы, и результаты исследования устройства.

Разработанный прототип-демонстратор имеет следующие характеристики:

1. подключение кроссбар-массива 32x8 или работа с отдельными ячейками;
2. кол-во мемристивных устройств до 256;
3. подключение по USB;
4. съемный кроссбар, можно работать с несколькими;
5. питание от розетки 220В 50Гц;
6. базовое ПО содержит необходимый функционал для проведения исследований мемристивных устройств и для запуска моделей ИНС в кроссбар-массиве;
7. открытый исходный код на Python, любой может доработать под свои нужды.

Литература

1. Shchanikov S. et al. Modeling and hardware implementation of vector-matrix multiplier based on 32x8 1T1R memristive crossbar array //2023 7th Scientific School Dynamics of Complex Networks and their Applications (DCNA). – IEEE, 2023. – С. 249-251.
2. Amirsoleimani A. et al. In-Memory Vector-Matrix Multiplication in Monolithic Complementary Metal–Oxide–Semiconductor-Memristor Integrated Circuits: Design Choices, Challenges, and Perspectives //Advanced Intelligent Systems. – 2020. – Т. 2. – №. 11. – С. 2000115.

Крайнов С. А.

Научный руководитель: к.т.н., доцент Еремеев С.В.

*Муромский институт (филиал) федерального государственного бюджетного
образовательного учреждения высшего образования «Владимирский
государственный университет имени Александра Григорьевича и Николая*

Григорьевича Столетовых»

602264, г. Муром, Владимирская обл., ул. Орловская, 23

E-mail: an0nim2020@ya.ru

Исследование искусственного интеллекта с функцией мышления

В современном мире искусственный интеллект (ИИ) активно используется для решения различных задач, в том числе в системах машинного обучения и робототехнике. Одним из направлений развития ИИ является создание агентов с функцией мышления, которые способны адаптироваться к изменениям окружающей среды и самостоятельно принимать решения [1]. Такая возможность позволяет моделировать более сложные сценарии поведения, приближенные к человеческому интеллекту.

Данная тема исследования посвящена анализу работы искусственного интеллекта на примере проекта “PushBlockCollab”, созданного с использованием ML-Unity. В этом проекте агенты должны совместными усилиями перемещать блоки в определенное место. Однако, при возникновении нетипичной ситуации, например, если один агент оказывается в ловушке, остальные начинают демонстрировать хаотичное поведение и не могут продолжить выполнение задачи [2]. Это свидетельствует о недостаточности их текущей системы принятия решений и отсутствии адаптивных механизмов.

Для решения данной проблемы можно предложить несколько методов:

1. Гибкая координация и передача информации – введение механизма обмена данными между агентами, позволяющего передавать сигналы о статусе членов группы и корректировать стратегию действий [3].

2. Поведенческое обучение через систему вознаграждений – добавление штрафов за состояние “застрявшего” агента и бонусов за успешное решение проблемы, что мотивирует агентов избегать подобных ситуаций [3].

3. Анализ окружающей среды и планирование – внедрение алгоритмов предсказания последствий действий, позволяющих оценивать возможные препятствия заранее [1].

4. Разделение ролей среди агентов – создание специализированных агентов, где одни управляют объектами, а другие анализируют ситуацию и координируют работу команды [2].

5. Имитация человеческого мышления – применение когнитивных моделей, позволяющих агентам предугадывать поведение окружающих и корректировать свою стратегию [4].

Для проверки эффективности предлагаемых решений потребуется провести обучение модели с использованием новых параметров и оценить изменения в поведении агентов. Возможные метрики для оценки включают скорость выполнения задачи, частоту возникновения нештатных ситуаций и уровень координации между агентами.

В заключении, развитие ИИ с функцией мышления является перспективным направлением, способствующим созданию более гибких и адаптивных систем. Решение проблемы потери координации в “PushBlockCollab” позволит продвинуться в изучении автономных агентов, способных эффективно работать в динамически изменяющейся среде.

Литература

1. Ha, D., Dai, A. & Le, Q. V. Hypernetworks. In International Conference on Learning Representations (OpenReview.net, 2017) - <https://openreview.net/forum?id=rkpACe1lx>

2. Ilharco, G. et al. Editing models with task arithmetic. In The Eleventh International Conference on Learning Representations (OpenReview.net, 2023) - <https://openreview.net/forum?id=6t0Kwf8-jrj>

3. Yu, L., Yu, B., Yu, H., Huang, F. & Li, Y. Language models are Super Mario: absorbing abilities from homologous models as a free lunch. In International Conference on Machine Learning

- Vol. 235 (eds Salakhutdinov, R. et al.) 57755–57775 (PMLR, 2024) -
<https://proceedings.mlr.press/v235/yu24p.html>
4. nostalgebraist. Interpreting gpt: the logit lens. LessWrong -
<https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens> (2021)

Предварительная обработка данных для сжатия без потерь

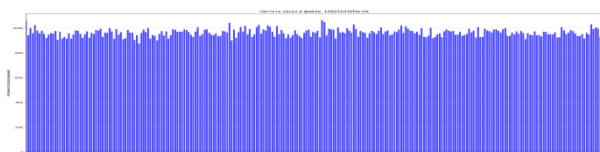
В современном мире управление данными стало неотъемлемой частью нашего общества. Одной из основных технологий для эффективного хранения и передачи информации является сжатие данных. С помощью алгоритмов сжатия можно уменьшить размер файлов, что позволяет экономить место на устройствах и ускорять передачу данных по сетям. Сжатие данных работает на основе выявления и удаления избыточной информации, что дает возможность увеличить скорость обработки и передачи данных.

Данные в процессе сжатия представлены в виде различных форматов, что позволяет оптимизировать их хранение и уменьшить объем передаваемой информации. Однако увеличение степени сжатия может также привести к потере качества, особенно в случае сжатия медиафайлов, а также изображения и видео.

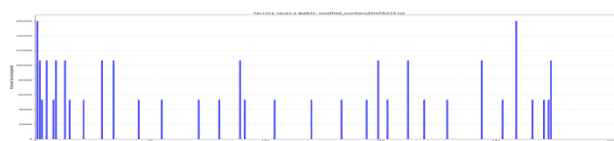
Качество сжатия информации определяется коэффициентом сжатия и уровнем потерь данных. Коэффициент сжатия — это отношение размера исходного файла к размеру сжатого файла. Он является ключевым показателем эффективности алгоритмов сжатия. Обычно коэффициент сжатия увеличивается с применением более сложных алгоритмов, но это также может привести к увеличению времени обработки и потенциальным потерям качества.

Цель работы - разработка эффективного инструмента, который позволит пользователям сократить объем файлов для оптимизации хранения, ускорить передачу данных по сетям, сохранить качество информации, особенно для медиафайлов, а также обеспечить интуитивно понятный интерфейс и совместимость с различными форматами файлов, что в итоге улучшит управление данными в современном цифровом мире.

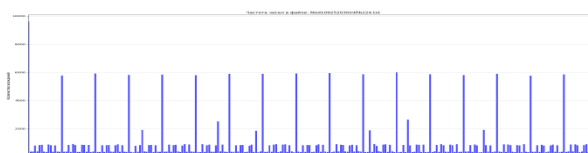
В качестве дополнительного инструмента будет использоваться архиватор 7Z. Архиватор 7Z использует алгоритм Хаффмана и строит приблизительно равномерное распределение (рис. 1а). Идея программы заключается в том чтобы числа, строящие равномерное распределение, умножить на некоторую последовательность (рис. 2б), тем самым «расшевелить» данные, превратив равномерное распределение в другой вид (рис. 3в).



(а)



(б)



(в)

Литература

1. Страуструп Б. "Язык программирования С++". Издательство "Бином", 2022, 1216 с.

Миханов А.В.

Научный руководитель: старший преподаватель Булаев А.В.
*Муромский институт (филиал) федерального государственного бюджетного
образовательного учреждения высшего образования «Владимирский
государственный университет имени Александра Григорьевича и Николая
Григорьевича Столетовых»*
602264, г. Муром, Владимирская обл., ул. Орловская, 23
artem.mikhanov@mail.ru

Построения баркодов для топологического анализа изображений: сравнение C++ и Python

Топологический анализ данных — это современный математический подход, позволяющий изучать глобальную структуру данных через их топологические признаки. В компьютерном зрении и анализе изображений персистентные баркоды применяются, для выделения значимых компонент связности, для классификации изображений по структурным особенностям, для обнаружения аномалий в медицинских, спутниковых и микроскопических изображениях. Основные шаги в построении баркодов:

- предварительная обработка изображений;
- извлечение признаков;
- вычисление топологических особенностей;
- соединение точек;
- построение баркода;
- сравнение баркодов;
- выдача результата.

Имеется уже реализованный алгоритм создания баркодов изображений, написанный на языке программирования C++, но наравне с ним популярным является язык программирования Python — очень удобный для разработки алгоритмов машинного обучения и искусственного интеллекта именно поэтому требуется перевести код из C++ кода в Python-библиотеку, для удобства использования в подключении данного алгоритма к произвольным Python проектам.

Перевод кода может осуществляться двумя способами, оба из которых основаны на анализе древовидной иерархии.

Программный код может рассматриваться как дерево — точка входа в программу может считаться корнем программного кода, а вызываемые методы и подпрограммы образуют ветви, подпрограммы, которые являются конечными в цепочке вызовов — можно назвать листьями.

Первый способ — анализ дерева с его корня - от точки входа main метода, заменяя заглушками непереведенные отсутствующие вызываемые элементы и постепенно реализуя все недостающие объекты, но этот способ требует понимания типов и структуры входных и выходных данных, чтобы симитировать работы программы на C++.

Второй способ — анализ дерева с его ветвей - он будет использован в данной статье для перевода кода из C++ на Python — это перевод начиная с вызываемых методов самого низкого уровня стека вызова, благодаря тому, что не нужно реализовывать заглушки, так как эти методы ничего не вызывают, то можно оптимизировать процесс переноса алгоритма и сэкономить много времени.

Алгоритм распознавания объектов с использованием баркодов, реализованный на C++ имеет следующую структуру:

- считывание изображения из памяти компьютера и создание представление изображения в виде цветовой матрицы;
- предобработка изображения — бинаризация;
- генерация баркода по цветовым признакам;
- распознавание изображения на основе баркода;
- выделение распознанных областей.

Первое, на что стоит обратить внимание при переводе модулей программы – это специфические математические библиотеки, при использовании которых результаты работы алгоритмов могут сильно отличаться.

В C++ были использованы:

- `<vector>` для хранения данных;
- `<algorithm>` для сортировки и преобразований;
- `<numeric>` для вычисления матриц расстояний.

В Python для этого можно использовать:

- NumPy для векторных операций;
- SciPy для построения матриц расстояний;
- библиотеки `gudhi` или `gipser` для вычисления персистентных гомологий.

После нахождения аналогов библиотек, можно начинать выполнять перевод программы из одного языка в другой. Считывание изображения происходит практически одинаково на выбранных языках, и проблем с приводом не возникает.

В качестве предобработки изображения были использованы методы преобразование цветовой матрицы в градацию серого. Для этого можно использовать стандартный метод `python rgb2gray`.

Для генерации баркода, автор программы на C++ использовал рекурсивный способ, где для каждой матрицы со смежными цветами.

Во время тестирования реализации алгоритма построения баркодов, переложённого на Python, было обнаружено, что рекурсивный способ построения баркодов потребляет очень много ресурсов, что может привести к исчерпанию ресурсов ПК, преждевременной остановке работы алгоритма и не получению конечного результата.

В обработке изображений на языке Python лучше избегать рекурсивных методов для оптимизации использования ресурсов компьютера. После изменения программного кода и проведения тестов, было выявлено, что отказ от рекурсивного алгоритма позволил снизить потребляемые алгоритмом ресурсы в 3 раза. При работе с распознаванием самый простой способ перевода более низкоуровневого кода в более высокоуровневый самым быстрым и эффективным решением оказалось, написание таких-же методов с удалением всех C++ конструкций контроля памяти, так как на Python этого не требуется.

В результате тестирования переведённой программы стало ясно, что программа на C++ работает быстрее, но код становится очень большой, и трудночитаемым. Код на Python, получился коротким и более читаемым.

В результате проделанной работы была получена Python библиотека, выдающая такой же результат, как и аналогичная библиотека, написанная на C++, с возможностью использовать её в алгоритмах машинного обучения и процессах распознавания объектов.

Литература

При переложении алгоритма с языка C++ на Python были использованы его описания и математическая модель излагавшаяся в следующих источниках:

1) Еремеев С.В. Метод декомпозиции изображений по топологическим особенностям и его приложения / С. В. Еремеев, Д. Е. Андрианов // Оптико-электронные приборы и устройства в системах распознавания образов и обработки изображений : сборник материалов XVII Международной научно-технической конференции, Курск, 12–15 сентября 2023 года. – Курск: Юго-Западный государственный университет, 2023. – С. 92-94. – EDN JNEJVU.

2) Метод векторизации спутниковых снимков на основе их разложения по топологическим особенностям / С. В. Еремеев, А. В. Абакумов, Д. Е. Андрианов, Т. А. Ширабакина // Информатика и автоматизация. – 2023. – Т. 22, № 1. – С. 110-145. – DOI 10.15622/ia.22.1.5. – EDN ROYLBD.

3) Метод разложения изображения по топологическим признакам / С. В. Еремеев, А. В. Абакумов, Д. Е. Андрианов, Д. В. Титов // Компьютерная оптика. – 2022. – Т. 46, № 6. – С. 939-947. – DOI 10.18287/2412-6179-CO-1080. – EDN FPBEUC.

4) Андрианов Д.Е. Модели и алгоритмы обработки и анализа данных в геоинформационных системах / Д. Е. Андрианов, С. В. Еремеев, К. В. Купцов. – Москва : ВНИИГеосистем, 2020. – 98 с. – ISBN 978-5-8481-0246-8. – EDN JZSJQ.

Набиев Х.П.

Научный руководитель: Вершинин В.В.

Федеральное государственное бюджетное образовательное учреждение высшего образования «Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»

600005, г. Владимир, Владимирская обл., ул. Горького, д. 87-а

Email: hamza-nabiyev88@mail.ru

Разработка инфраструктуры и серверного приложения для проекта «Программа лояльности»

Современные программы лояльности выступают важным инструментом удержания и привлечения клиентов, что требует надежной серверной инфраструктуры. Данная инфраструктура должна обеспечивать масштабируемость, отказоустойчивость и безопасность данных, а также автоматизацию процессов развертывания и интеграции приложений.

Целью исследования является проектирование автоматизированной инфраструктуры для развертывания и обновления серверного приложения программы лояльности, обеспечивающей надежность, гибкость и отказоустойчивость.

В рамках работы был проведен сравнительный анализ современных решений в области оркестрации контейнеров, управления конфигурацией, баз данных и мониторинга. В результате был выбран комплекс технологий, включающий Kubernetes, PostgreSQL, Django REST Framework, Prometheus и Grafana, а также инфраструктурные инструменты Terraform и Ansible.

Архитектура системы представлена распределённой микросервисной моделью, центральным компонентом которой является Kubernetes, осуществляющий автоматическое масштабирование и балансировку нагрузки между контейнерами приложения. Для обеспечения безопасности и высокой доступности, запросы от пользователей поступают через Nginx Ingress, выполняющий функции маршрутизации.

Важным элементом инфраструктуры является реализация автоматизированного CI/CD-конвейера на базе GitLab и GitLab Runner, обеспечивающего автоматическую сборку, тестирование и развертывание приложения в Kubernetes. Использование IaC-инструментов Terraform и Ansible позволило значительно сократить время развертывания инфраструктуры, минимизировать вероятность ошибок и ускорить адаптацию системы к изменениям.

Для долговременного хранения информации была выбрана система управления базами данных PostgreSQL, обеспечивающая транзакционную целостность и возможность масштабирования при увеличении нагрузки.

Особое внимание уделено мониторингу и логированию инфраструктуры. Применение Prometheus и Grafana обеспечивает оперативный мониторинг состояния компонентов системы, позволяя своевременно выявлять и устранять возможные сбои. Для централизованного анализа логов используется ELK-стек.

Разработанная инфраструктура и серверное приложение способствуют повышению общей операционной эффективности, значительно уменьшают временные затраты на обновление компонентов системы и снижают риск возникновения ошибок на этапе внедрения, что приводит к росту общей надежности и эффективности управления серверным приложением программы лояльности.

Литература

1. Kubernetes Documentation. – [Электронный ресурс]. – URL: <https://kubernetes.io/docs/>
2. PostgreSQL: документация [Электронный ресурс]. – URL: <https://www.postgresql.org/docs/>
3. GitLab CI/CD Documentation. – [Электронный ресурс]. – URL: <https://docs.gitlab.com/ee/ci/>
4. Ansible Documentation. – [Электронный ресурс]. – URL: <https://docs.ansible.com/>

Осинцев Д.Н.

Научный руководитель: к.т.н. Комкова С.В.

*Муромский институт (филиал) федерального государственного бюджетного образовательного учреждения высшего образования «Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»
602264, г. Муром, Владимирская обл., ул. Орловская, 23*

Разработка системы для проксирования трафика с поддержкой Xray VLESS+Reality и Tun Mode

В современных условиях цифровизации и растущей потребности в обеспечении безопасного и устойчивого доступа к интернет-ресурсам особое значение приобретает разработка инструментов для конфиденциальной маршрутизации трафика. Безопасность, гибкость настройки и автоматизация — ключевые требования к подобным системам.

Данная посвящена созданию программного продукта, который позволяет проксировать трафик через протокол Xray VLESS с поддержкой Reality и дополнительно функционировать в режиме Tun Mode, обеспечивая прозрачную маршрутизацию трафика приложений. Программа способна работать как классический прокси-сервер, а также предоставляет возможности безопасного туннелирования данных.

Ключевые особенности и функциональные возможности разрабатываемого решения включают:

- Поддержку VLESS+Reality — современного, легковесного и защищённого протокола проксирования с обходом DPI;
- Tun Mode — режим, позволяющий направлять трафик всего устройства через туннель, не требуя дополнительной настройки отдельных приложений;
- TLS-шифрование, обеспечивающее защиту передаваемых данных от перехвата и анализа;
- Возможность автоматической настройки удалённого сервера (в том числе установки Xray и конфигурации) через SSH-подключение, упрощая развёртывание и экономя время пользователя;
- Гибкие параметры конфигурации и управления, адаптированные под различные сценарии использования.

Программа ориентирована на пользователей, которым необходим стабильный и защищённый доступ к интернет-ресурсам, а также специалистов, занимающихся настройкой и администрированием прокси-серверов. Автоматизация настройки и встроенные механизмы безопасности делают продукт удобным инструментом как для профессионального, так и для повседневного использования.

Разработка ведётся с использованием языка программирования Go и графической библиотеки Fyne, которая позволяет создавать кроссплатформенные приложения с современным и удобным пользовательским интерфейсом. Выбор Go обусловлен не только его высокой производительностью и эффективной работой с сетевыми операциями, но и тем, что сам Xray написан на Go — это обеспечивает лучшую совместимость, возможность прямой интеграции и более глубокий контроль над настройками и взаимодействием с ядром Xray. Использование Fyne позволяет создавать лёгкое и стабильное GUI-приложение, пригодное для работы на различных операционных системах без зависимости от сторонних фреймворков.

Панов Е.А.

Научный руководитель: к.т.н. доцент Канунова Е.Е.

*Муромский институт (филиал) федерального государственного бюджетного
образовательного учреждения высшего образования «Владимирский
государственный университет имени Александра Григорьевича и Николая
Григорьевича Столетовых»*

602264, г. Муром, Владимирская обл., ул. Орловская, 23

E-mail: janypanov@gmail.com

Информационная система формирования отчетной документации руководителя практики студентов

Автоматизация документооборота в образовательных учреждениях – важный шаг к оптимизации рабочего процесса. Руководители практики студентов ежедневно сталкиваются с необходимостью формирования большого количества отчетных документов: приказов, ведомостей, индивидуальных планов и итоговых отчетов. В настоящее время этот процесс выполняется вручную, что приводит к значительным временным затратам, возможным ошибкам и дублированию данных.

В МИ ВлГУ ежегодно проходят практику сотни студентов, распределенных по разным предприятиям и организациям. Для учета их деятельности, контроля сроков практики и формирования отчетности требуется единая информационная система, которая исключит ручной ввод данных и автоматизирует создание документов.

Разработанная программа позволяет:

- автоматически заполнять отчетные документы на основе данных из базы (например, приказы о назначении практики, ведомости посещаемости);
- управлять сроками и видами практики (учебная, производственная, преддипломная) с возможностью гибкой настройки;
- учитывать распределение студентов по местам прохождения практики и их переводы между группами;
- генерировать отчеты в стандартизированном формате (PDF, Excel) с использованием шаблонов JasperReports;
- импортировать и экспортировать данные.

Система разрабатывается на языке Java с применением JasperReports для формирования документов и клиент-серверной архитектуры для обеспечения стабильной работы. В качестве базы данных используется FireBird/RedDataBase. Пользовательский интерфейс создается с помощью JavaFX, обеспечивая удобное взаимодействие с системой.

Разработанная система может быть внедрена в учебных заведениях для упрощения работы руководителей практики, сокращения времени на подготовку документов и повышения точности отчетности.

Литература

1. Герберт Шилдт. «Java: Полное руководство» (11-е издание). — М.: Диалектика, 2022. — 1488 с. ISBN: 978-5-907203-36-1.

2. ДжасперReports: Официальная документация. // URL: <https://community.jaspersoft.com/documentation> (Дата обращения: 10.05.2024).

Привезенцев Е. Д.
Научный руководитель: к.т.н., доцент Еремеев С.В.
*Муромский институт (филиал) федерального государственного бюджетного
образовательного учреждения высшего образования «Владимирский
государственный университет имени Александра Григорьевича и Николая
Григорьевича Столетовых»
602264, г. Муром, Владимирская обл., ул. Орловская, 23
E-mail: mc.enzy@mail.ru*

Сравнительный анализ методов поиска глобального минимума на геометрических данных

Современные задачи геоинформатики и анализа пространственных данных требуют применения эффективных методов оптимизации для решения задач пространственного анализа природных объектов. В данной работе проводится сравнительный анализ алгоритмов оптимизации — градиентного спуска, стохастического градиентного спуска, RMSProp и Adam — для задачи нахождения самой глубокой вдающейся точки берега в озеро, то есть глобального минимума расстояний от береговой линии до центра озера.

Целью исследования стало определение наиболее эффективного метода для точного и быстрого нахождения береговой точки, наиболее удалённой внутрь озера. Такая задача может быть актуальна, например, для определения потенциальных мест пристаней, углублений или анализа формы береговой линии.

В рамках проекта были реализованы два программных инструмента. Первый — интерфейс на Python с возможностью выбора метода оптимизации и отображением траектории поиска. Второй — визуализатор, отображающий процесс поиска на фоне карты озера, с построением графиков расстояний и координат найденной точки.

В качестве исходных данных использовались координаты береговой линии, заданные в двумерном пространстве. Для каждого метода производился запуск с одинаковыми начальными условиями, после чего фиксировались координаты найденной точки, длина траектории, число итераций и визуализировался сам путь оптимизации.

Результаты показали, что градиентный спуск обеспечивает быструю сходимость, но может быть чувствителен к форме береговой линии. Стохастический градиентный спуск показал более гибкое поведение, RMSProp — устойчивость к шуму, а Adam оказался наиболее универсальным и сбалансированным по скорости и стабильности.

Данный подход может быть применён в задачах анализа берегов, планирования инфраструктуры, а также в образовательных целях — для демонстрации работы оптимизационных алгоритмов на геометрических данных.

Литература

1. Mikhlyayev SV. Relaxation technique for stereo images matching. Proc ACIT 2002: 376-381.
2. Chetverikov D, Szabo Z. Detection of high curvature points in planar curves. Source: <http://visual.ipan.sztaki.hu/corner/>.
3. Tou JT, Gonzalez RC. Pattern recognition principles. 2nd ed. Addison-Wesley; 1977.
4. Furman YaA, ed. Introduction to contour analysis and its applications to image and signal processing [In Russian]. Moscow: "Fizmatlit" Publisher; 2002.
5. Khmelev RV. Searching for cavities in a closed non-convex contour. Computer Optics 2002; 24: 164-172.

Сакеев А. М.

Научный руководитель: к.т.н., доцент Еремеев С.В.

Муромский институт (филиал) федерального государственного бюджетного образовательного учреждения высшего образования «Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»
602264, г. Муром, Владимирская обл., ул. Орловская, 23
E-mail: sakeev123@gmail.com

Программа для построения карты глубин по изображению

Алгоритмы построения карт глубин по изображению нужны для получения информации о расстоянии до объектов. Это важный элемент для анализа и визуализации трёхмерных данных. Этот алгоритм используется в таких сферах, как компьютерное зрение, виртуальная реальность, робототехника и обработка фотографий.

Существуют 2 вида систем компьютерного зрения (по активности):

- Активные (излучающие)
- Пассивные

Вторые имеют преимущество, так как лидары, являющиеся активными устройствами, излучают сигналы в окружающую среду, при перехвате которых не трудно вычислить их положение в пространстве, в то время как системы, использующие видеокамеры гораздо сложнее обнаружить. Для формирования 3D сцен подобные системы, используя видеоданные, полученные от двух и более видеокамер, формируют карты глубин, представляющие собой изображение с градациями серого цвета. Яркость элементов указывает на расстояние от видеокамеры до объектов [1].

Актуальность темы обусловлена распространением систем компьютерного зрения в разных сферах, таких как промышленность, транспорт, научные исследования и другие. Оно позволяет роботам анализировать пространство в реальном времени и строить оптимальные маршруты движения. Однако, эта технология не совершенна, так как работает достаточно медленно. С этой целью ведутся исследования на тему улучшения алгоритмов построения карт глубин и преобразования их в 3D-сцены.

Цель работы: создание программы для построения карты глубин по стереопаре.

В данной работе были реализованы алгоритмы построения карты глубин на основе стереопары и преобразования карты глубин в 3D-сцену на языке C++ в среде разработки Microsoft Visual Studio 2022. Также было создано оконное приложение Windows Forms для работы в приложении через пользовательский интерфейс.

Реализация функций для обработки изображений была осуществлена по следующим алгоритмам:

1. Загрузка стереопары
2. Построение карты глубин, включающее обработку, улучшающую качество полученной карты.
3. Преобразование карты глубин в облако точек, составляющих 3D-сцену.

Для визуализации работы алгоритмов создано Windows Forms-приложение с интуитивно понятным интерфейсом, дающим возможность загрузить изображения, обработать их, построить 3D-сцену и сохранить полученный результат или вывести его непосредственно в самом приложении.

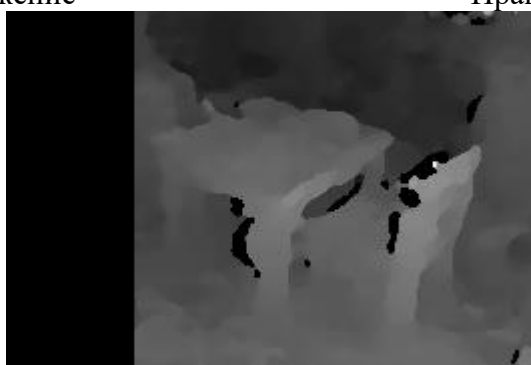
Для примера была взята стереопара, представляющая собой 2 фотографии комнаты со смещением, на основе которой была построена карта глубин (рис. 1) [2].



Левое изображение



Правое изображение



Карта глубин

Рис. 1 – Примеры изображений

Данный проект демонстрирует основы обработки изображений алгоритмом построения карт глубин, и преобразования её в облако точек. Функции приложения могут быть расширены при помощи совершенствования методов обработки и постобработки.

Литература

1 Бобырь М.В., Емельянов С.Г., Милостная Н.А. Алгоритм построения 3d сцен распознанных объектов по картам глубин. Известия Юго-Западного государственного университета. 2023;27(2):90-104.

2 Рощупкина, С. Н. Построение карты диспаратности по неоткалиброванной паре изображений / С. Н. Рощупкина, Л. К. Асташенкова, В. С. Булгакова, К. В. Кудринская. — Текст : непосредственный // Молодой ученый. — 2019. — № 26 (264). — С. 43-46.

Разработка подсистемы приёма и отправки электронных документов в подразделения контрагентов с использованием формата межведомственного электронного документооборота версии 2.7.1

В современном обществе цифровизация затрагивает все сферы деятельности, и одной из важнейших областей является автоматизация процессов документооборота. Электронный документооборот играет ключевую роль в обеспечении оперативного и безопасного взаимодействия между организациями, государственными структурами и коммерческими предприятиями. В условиях растущего объема информации и необходимости ее быстрой и безопасной передачи особое значение приобретает соблюдение требований по защите данных, особенно когда речь идет о документах, содержащих служебную информацию ограниченного распространения (ДСП).

В соответствии с Приказом Министерства цифрового развития, связи и массовых коммуникаций Российской Федерации и Федеральной службы охраны Российской Федерации от 4 декабря 2020 года № 667/233 "Об утверждении Требований к организационно-техническому взаимодействию государственных органов и государственных организаций" было установлено, что для обеспечения эффективного и безопасного обмена электронными документами между ведомствами и государственными организациями требуется строгое соблюдение организационно-технических норм и стандартов. В связи с этим возникает необходимость доработки существующих информационных систем, задействованных в межведомственном взаимодействии, включая Автоматизированную информационную систему Федеральной службы судебных приставов (АИС ФССП) России.

Основной целью данной работы является проектирование и доработка системы АИС ФССП, которая должна обеспечивать корректный прием и отставку электронных документов в подразделения контрагентов с использованием формата межведомственного электронного документооборота (МЭДО) версии 2.7.1. Доработка системы предполагает создание централизованного механизма ведения справочников «Абоненты системы МЭДО» и «Подразделения абонентов МЭДО». Важной задачей является также реализация механизма блокировки отправки документов с грифом «ДСП» тем абонентам, у которых отсутствуют права и техническая возможность на их обработку.

Для реализации поставленной цели необходимо выполнить ряд задач, включающих детальный анализ существующей архитектуры системы, выявление ключевых проблем в процессе передачи данных и определение требований к новой функциональности. В рамках проекта предполагается проектирование алгоритма обработки данных, а также разработка схем взаимодействия между подсистемами, участвующими в процессе документооборота.

Таким образом, в результате выполнения данной работы будет разработана и представлена новая функциональность системы, обеспечивающая надежную и безопасную отставку электронных документов в соответствии с требованиями нормативных актов. Внедрение предложенной подсистемы позволит повысить эффективность межведомственного взаимодействия, минимизировать риски ошибочной отправки документов неподготовленным контрагентам и обеспечить соответствие приказу министерства цифрового развития, установленному на государственном уровне.

Миграция с PostgreSQL на Ред Базу Данных

Системы управления базами данных уже давно стали неотъемлемой частью любой информационной системы.

Хранение производственной или рабочей информации предприятия является ключевым аспектом в организации его работы. Потеря данных может привести к нарушению контрактных обязательств или потере существующих производственных наработок.

Поэтому СУБД принимают на себя всю ответственность по обеспечению хранения информации. При этом СУБД со временем могут терять свою актуальность (когда на рынке появляются решения с расширенными функциональными возможностями).

Так же СУБД, в немалой степени, несет ответственность за быстроедействие корпоративной информационной системы и расхождение роста производительности СУБД с ростом производительности других компонентов вычислительной системы может стать причиной снижения продуктивности работу сотрудников компании, а это, в свою очередь, может привести к снижению темпов роста компании и потери ее позиций в сегменте рынке.

Поэтому использование наиболее эффективной СУБД является важнейшим аспектом деятельности компании.

В связи с этим может возникнуть потребность по переводу информационной системы с одной СУБД на другую.

При переносе БД с одной СУБД на другую инженеры сталкиваются со следующими проблемами:

- разнотипность типов данных при переносе структуры таблиц;
- разнотипность нотаций языка SQL, которые поддерживаются СУБД источником и целевой СУБД;
- разнотипность метаданных БД и структур процедур, функций, триггеров и других компонентов, обеспечивающих логику работы БД;
- различия организации разграничения прав доступа к данным.

На рынке СУБД, среди свободно распространяемых систем исторически выделялись два проекта PostgreSQL и Firebird. Российскими разработчиками на базе указанных СУБД реализованы проекты PostgreSQL Pro и Ред База Данных. При этом необходимо отметить, что программные продукты семейства PostgreSQL Pro распространяются на коммерческой основе, в то время как Ред База Данных имеет свободно-распространяемую редакцию. Поэтому в дальнейшем в рассмотрении будут участвовать PostgreSQL и Ред База Данных.

Одним из важнейших аспектов при миграции из СУБД PostgreSQL на СУБД Ред База Данных — это воссоздание структуры БД. Процесс миграции структуры БД из исходной системы в целевую возможен двумя способами – в ручном или автоматизированном режимах. Для крупных проектов, ручная миграция — это процесс весьма трудоемкий и порождающий потенциальные ошибки в воссоздании структуры. Поэтому разработка автоматизированной системы миграции является актуальной задачей.

Каждая из СУБД внесли свои индивидуальные изменения в синтаксис SQL и внутреннюю структуру, что необходимо учитывать при создании автоматизированной системы миграции.

Для разработки подобной системы рассматривались и сопоставлялись такие элементы создания структуры БД и управления, как: типы данных, элементы языка SQL, создание управляющих функций.

Таблица 1 – Сопоставление типов данных, поддерживаемых PostgreSQL и Ред База Данных

Секция 3. Разработка информационных систем

Тип данных	PSQL	РедБД	Тип данных	PSQL	РедБД	Тип данных	PSQL	РедБД
Integer	+	+	Char	+	+	Circle	+	-
Bigint	+	+	VarChar	+	+	Iseq	+	-
Decimal	+	+	NChar	-	+	Poligon	+	-
Numeric	+	+	NVarChar	-	+	Cidr	+	-
Real	+	-	Text	+	-	Inet	+	-
Float	-	+	Bytea	+	-	Macaddr	+	-
SmallInt	+	+	Blob	-	+	Maccaddr8	+	-
Double Precision	+	+	Bit	+	-	Uuid	+	-
SmallSerial	+	-	BitVarying	+	-	Xml	+	-
Serial	+	-	Boolean	+	+	Json	+	-
BigSerial	+	-	Point	+	-	JsonB	+	-
Money	+	-	Line	+	-	Перечесления	+	-

Таблица 2 – Сопоставление синтаксиса языка SQL

Название запроса	PostgreSQL	Ред База Данных
1. Создание таблицы	CREATE TABLE	CREATE TABLE
2. Изменение таблицы	ALTER TABLE	ALTER TABLE
3. Удаление таблицы	DROP TABLE table_name;	DROP TABLE table_name;
4. Добавление данных	INSERT INTO	INSERT INTO
5. Получение данных	SELECT FROM	SELECT FROM
6. Фильтрация данных	WHERE условие	WHERE условие
7. Обновление данных	UPDATE	UPDATE
8. Удаление данных	DELETE FROM	DELETE FROM
9. Выборка уникальных значений	SELECT DISTINCT	SELECT DISTINCT
10. Сортировка	ORDER BY	ORDER BY
11. Получение диапазона строк	LIMIT и OFFSET	FIRST и SKIP
12. Агрегатные функции	AVG, BIT_AND, BIT_OR, BOOL_AND, BOOL_OR, COUNT (*), COUNT (expression), SUM, MIN, MAX, STRING_AGG (expression, delimiter)	AVG() COUNT() LIST() MAX() MIN() SUM()
13. Группировка	GROUP BY	GROUP BY
14. Фильтрация групп	HAVING	HAVING
15. Соединение таблиц	INNER JOIN, OUTER JOIN, CROSS JOIN	INNER JOIN, OUTER JOIN, CROSS JOIN
16. Объединение множеств	UNION	UNION
17. Создание триггеров	CREATE TRIGGER	CREATE TRIGGER

18. Создание хранимой процедуры	CREATE PROCEDURE	CREATE PROCEDURE
19. Создание хранимой функции	CREATE FUNCTION	CREATE FUNCTION
20. Создание пакетов	CREATE PACKAGE	CREATE PACKAGE
21. Внешние хранимые процедуры, функции и триггеры	-	+

Таблица 3 – Пример расхождения в синтаксисе оформления триггеров

PostgreSQL	РедБазаДанных
CREATE TABLE catalog (id SERIAL PRIMARY KEY, name CHAR);	CREATE TABLE catalog (id INT PRIMARY KEY, name CHAR); CREATE SEQUENCE id_catalog;

На основании сравнительного анализа можно сделать следующие выводы: в структуре типов СУБД наблюдаются расхождения, которые частично могут компенсироваться, типами с другими именами; возможен перенос в автоматическом режиме триггеров, функций, процедур, представлений и других внутренних структур; несмотря на поддержку стандарта SQL-2016, перенос БД с PostgreSQL на Ред Базу Данных может быть невозможен из-за внесения вендорских изменений и дополнения в SQL.

Литература

1. Токмаков Г.П. Базы данных: Модели и структуры данных, язык SQL, программирование баз данных: учебное пособие / Г. П. Токмаков. – Ульяновск: УлГТУ, 2021. – 362 с
2. Маркин А. В. СУБД «РЕД БАЗА ДАННЫХ». Основы SQL. Учебное пособие - Москва: Ай Пи Ар Медиа, 2022. — 460 с. — ISBN 978-5-4497-1605-7.
3. Радыгин В.Ю., Куприянов Д.Ю. Базы данных: основы, проектирование, разработка информационных систем, проекты: курс лекций. Учебное пособие М.: НИЯУ МИФИ, 2020. – 244 с.

Солдатов А.Г.

Научный руководитель: Вершинин В.В.

Федеральное государственное бюджетное образовательное учреждение высшего образования "Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых"

600000, Россия, Центральный Федеральный округ, Владимирская область, г.

Владимир, ул. Горького, дом № 87

drewskirus@yandex.ru

Программная система библиотеки алгоритмов hmi-модулей для комбинации приборов автомобиля

Повышение эффективности разработки программного обеспечения в рамках разработки и реализации HMI внутри автомобиля заключается в том, чтобы уменьшить время на разработку стандартных модулей, на подобие спидометров шкал уровня топлива, температуры, модулей часов и подобного.

В рамках технологического перехода на новую ступень комбинаций приборов, а именно полностью цифровых, либо цифровых с небольшим количеством аналоговых элементов, будет разработана данная система.

Раньше была проблема адаптации стрелочного прибора, поскольку для разных автомобилей могло быть разное аппаратно-компонентное обеспечение, поэтому каждый раз разработчики тратили много времени на доработку и корректировку стрелок и их показаний. В современных экранах данной проблемы нет, но существует проблема повторной разработки существующих модулей для комбинаций приборов. Если брать пример, то комбинации от Lada Vesta с 10-ти дюймовой комбинацией приборов и Lada E-Largus, достаточно схожи, показано на рисунках 1 и 2, как внешне, так и программно, поэтому некоторые модули могут быть едины в основном своём костяке.



Рис. 1. – HMI Lada E-Largus



Рис. 2. – HMI Lada Vesta Sport

Секция 3. Разработка информационных систем

Для данной программной системы необходимо собрать существующие решения из максимального возможного кол-ва модулей, и переписать их параллельно подвергая тестам, на корректные показания в рамках чек листов.

Предлагается переписать модули, снабдив их комментариями и придерживаться принципов разработки программного кода, изложенных в различных источниках[1,2]. Помимо этого, стоит снабдить документацией все существующие модули, поскольку программисты могут возвращаться для правок или внедрения дополнительных функций в модули, и на вникание в существующих код можно потратить много времени.

Для именно программной системы, которая предлагает быстрый и удобный доступ к данной библиотеки могут быть предъявлены следующие требования:

1. Простой интерфейс
2. Увеличенный размер шрифта
3. Смена темы (светлая/тёмная)
4. Большие кнопки «назад» и «копировать» для старшего поколения
5. Быстрый доступ к документации
6. Простота добавления новых модулей в библиотеку
7. Возможность динамического изменения кода в рамках выделенных параметров
8. Понятный конструктор и подбор модуля для необходимого аппаратного обеспечения

Следующие требования необходимы для освоения людьми старшего поколения, поскольку избыточно сложные системы могут вызвать у них трудности, помимо этого, чем проще и удобнее будет система, тем более модульной она сама будет, что поможет её поддерживать в будущем.

Для вариативности модулей НМІ предполагается делать поля, к примеру цвет стрелки, и оптимизировать в коде данный параметр через define, поскольку мы отходим от строгих переменных внутри кода, а стараемся выносить их в шапку, что позволит быстро и удобно редактировать их при необходимости.

Литература

1. Маккóннелл С. Совершенный код / С. Маккóннелл. – США: Русская Редакция, 1993. – 896 с. – ISBN 978-5-7502-0064-1
2. Мартин Р. Чистая архитектура. Искусство разработки программного обеспечения. / Р. Мартин. – СПб: Питер, 2021. – 352 с.

Шмелькова Е. В.

Научный руководитель: к.т.н. доцент Канунова Е.Е.

Муромский институт (филиал) федерального государственного бюджетного образовательного учреждения высшего образования «Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»

602264, г. Муром, Владимирская обл., ул. Орловская, 23

E-mail: smelkovaevgenia2003@gmail.com

Разработка информационной системы для анализа поисковых запросов в социальных сетях

В современном обществе социальные сети стали неотъемлемой частью повседневной жизни и важным инструментом для обмена информацией, получения новостей и формирования общественного мнения. В условиях постоянного роста объемов данных и разнообразия поисковых запросов, возникающих в этих платформах, становится очевидной необходимость разработки эффективных информационных систем, способных анализировать и интерпретировать данные.

Существует множество аналогов разработанной программы, таких как:

1. Hootsuite - платформа для управления социальными медиа, которая позволяет отслеживать упоминания, анализировать взаимодействие и планировать контент.
2. Brandwatch - инструмент для анализа социальных медиа, который предлагает глубокую аналитику, включая анализ настроений и выявление трендов.
3. Sprout Social - платформа, которая сочетает в себе управление социальными сетями и аналитику, позволяя отслеживать упоминания и взаимодействия с пользователями.
4. Talkwalker - инструмент для анализа социальных медиа и мониторинга брендов, который предлагает мощные функции визуализации и анализа данных.

В докладе рассматриваются вопросы разработки информационной системы для анализа поисковых запросов в социальных сетях. В условиях стремительного роста популярности социальных платформ возникает необходимость в эффективном анализе данных, генерируемых пользователями. Данная система нацелена на предоставление пользователям инструмента для сбора, обработки и анализа информации о поисковых запросах, что позволяет выявлять их актуальные предпочтения и в дальнейшем предоставлять необходимый контент. В нашем случае данные будут собираться из таких социальных сетей как: ВКонтакте, Одноклассники и YouTube.

Разработанная программа позволяет собирать поисковые запросы пользователя из различных социальных сетей и выделять из них наиболее часто используемые. Так же для визуального представления данных программа будет строить облако слов и график с указанными чаще всего словами и количеством их использования.

Для разработки такого приложения в качестве языка программирования был выбран Python [1], интегрированная среда разработки PyCharm. Для анализа полученных запросов реализован метод Text Mining [3].

В данной системе реализованы следующие задачи:

- сбор данных поисковых запросов пользователя;
- их обработка [2];
- анализ обработанных данных;
- визуальное представление.

Программа может быть полезна маркетологам в понимании потребностей и предпочтений целевой аудитории, что позволит создавать более эффективные рекламные кампании и контент для различных категорий пользователей.

Литература

1. Григорьев Е. А. Разведочный анализ данных с помощью Python / Григорьев Е. А., Климов Н.С. // E-Scio. 2021. №3 (42). URL: <https://cyberleninka.ru/article/n/razvedochnyy-analiz-dannyh-s-pomoschyu-python/viewer> (дата обращения: 24.02.2025).

Секция 3. Разработка информационных систем

2. Корелов С.В. Предобработка текстов электронных писем в задаче обнаружения спама / С.В. Корелов, А. М. Петров, Л.Ю. Ротков, А. А. Горбунов // Труды учебных заведений связи, 2021. – №5. – с. 81–92

3. Сафаров Л. С. Использование технологии text mining при автоматической обработке текста. «Экономика и социум». 2023.//URL: <https://cyberleninka.ru/article/n/ispolzovanie-tehnologii-text-mining-pri-avtomaticheskoy-obrabotke-teksta/viewer> (Дата обращения: 16.03.2025)

Шпидонов Н.А.

Научный руководитель: к.т.н., доцент Еремеев С.В.

Муромский институт (филиал) федерального государственного бюджетного образовательного учреждения высшего образования «Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»

602264, г. Муром, Владимирская обл., ул. Орловская, 23

E-mail: 1921688101@mail.ru

Разработка диффузионной модели для генерации изображений объектов реального мира

На текущий момент актуальным является развитие методов генерации изображений с использованием глубоких нейросетей. Традиционные генеративные модели, такие как GAN (Generative Adversarial Networks) и VAE (Variational Autoencoder), показали свою эффективность, но имеют ограничения, связанные со сложностью обучения и контролем качества. В последнее время диффузионные модели привлекают внимание исследователей благодаря их способности генерировать высококачественные изображения, устойчивые к артефактам.

В основе алгоритма лежит постепенное добавление шума к изображению на этапе обучения и его восстановление в обратном порядке во время генерации. Это достигается путем использования стохастического дифференциального уравнения (SDE), где модель учится предсказывать шум, а затем корректировать изображение на каждом шаге итерации. Процесс генерации основан на последовательном уменьшении шума и восстановлении деталей изображения с учетом обученных параметров.

Примеры исходного и зашумленного изображений представлены на рисунке 1:



Рисунок 1 –Исходное и зашумленное изображение

Основная суть модели заключается в обучении нейросетевой структуры на выборке изображений и последующем восстановлении исходных данных из зашумленного состояния. В работе реализация основана на обработке изображений размером 128x128 пикселей, использовании оптимизатора Adam и обучении модели на загруженном наборе данных. Датасет содержит в себе более 4300 изображений природы, обработанных необходимым образом, чтобы подходить под условия для обучения модели.

Данная реализация диффузионной модели написана на языке C++ с использованием следующих библиотек:

stb_image.h и stb_image_write.h – для загрузки и сохранения изображений в формате JPEG;

vector, random, thread, mutex – для обработки данных, многопоточного выполнения и генерации шума;

filesystem, fstream – для работы с файлами и загрузки датасета.

Реализация может использоваться на различных платформах (Windows, Linux) с соответствующими адаптациями для сетевого взаимодействия. В перспективе возможна интеграция с Python через интерфейсы C++ API (например, PyBind11) для гибридного использования возможностей данной модели.

Литература

1. Бекман, И. Н. Высшая математика: математический аппарат диффузии : учебник для вузов / И. Н. Бекман. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2025. — 459 с.
2. Generative Diffusion Modeling: A Practical Handbook/Zihan Ding, Chi Jin [Электронная статья] <https://arxiv.org/pdf/2412.17162>