

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»**

Кафедра математической теории упругости и биомеханики

**Создание сервиса для обработки музыкального контента**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 442 группы  
направления 09.03.03 — Прикладная информатика

механико-математического факультета

Горбачева Никиты Игоревича

Научный руководитель  
к.ю.н., доцент

\_\_\_\_\_

Р.В. Амелин

подпись, дата

Зав. кафедрой  
д.ф.-м.н., профессор

\_\_\_\_\_

Л.Ю. Коссович

подпись, дата

Саратов 2024

**Введение.** В последнее время наблюдается значительный рост интереса к сервисам по обработке музыкального контента, в том числе с использованием технологий искусственного интеллекта. При этом направлений для развития технологий по обработке музыкального контента становится всё больше.

Актуальность данной темы обусловлена тем, что технологии искусственного интеллекта уже активно используются в различных сферах музыкальной индустрии, например, в создании музыки для фильмов или в процессе обработки звука. Так, некоторые проекты используют нейронные сети и собственные технологии для удаления шумов и помех, улучшения баланса звука и оптимизации уровня громкости. И наконец, использование искусственного интеллекта в музыкальной индустрии может привести к созданию новых и уникальных музыкальных произведений, которые могут стать важным шагом в развитии индустрии.

Цель данной дипломной работы состоит в создании сервиса для обработки музыкального контента с использованием обычных алгоритмов и технологий искусственного интеллекта.

Для достижения этой цели были поставлены следующие задачи:

- проанализировать основные технологии искусственного интеллекта, которые могут быть использованы в музыкальной индустрии;
- рассмотреть существующие примеры сервисов в обработке музыкального контента и выбрать наиболее подходящее направление;
- выявить возможные преимущества и недостатки существующих сервисов для обработки музыкального контента;
- создать своё уникальное приложение, которое было бы актуальным на текущий момент времени.

В данной работе будут рассмотрены основные принципы работы технологий по обработке аудиофайлов, которые могут использоваться в музыкальной индустрии, а также их преимущества и недостатки. Также будут проанализированы примеры использования искусственного интеллекта в создании и обработке музыки, влияние искусственного интеллекта на

творческий процесс создания музыки и на роль музыкантов в индустрии.

И, наконец, будет спроектировано и создано собственное оригинальное приложение по обработке музыкального контента.

**Структура работы.** Основная часть состоит из 3 разделов:

- Обзора и анализа предметной области, а также обоснованием актуальности создания такого сервиса;
- Проектирования приложения;
- Этапы разработки приложения и его тестирования.

**В первом разделе** собрана вся информация о существующих сервисах по обработке музыкального контента и их алгоритмах, актуальность разрабатываемого приложения.

Существуют различные алгоритмы обработки музыки, включая те, что используют искусственный интеллект. Одни удаляют шумы и помехи, применяя частотную коррекцию, редактирование цифрового аудио и другие методы. Алгоритмы искусственного интеллекта используют эти методы, включая фильтрацию на основе нейронных сетей, спектральную маскировку и другие. Популярные сервисы для удаления шумов и помех включают Vocalremover, Phonicmind и Filmora.

Другие производят апскейлинг звука — это процесс улучшения качества звуковых файлов, особенно с низким качеством записи или шумами. Алгоритмы искусственного интеллекта могут использоваться для обработки сигналов, удаления шумов и других помех, изменения тональности, темпа и скоростивоспроизведения.

Третьи алгоритмы искусственного интеллекта могут быть использованы для создания новых музыкальных композиций. Например, генеративно-состязательные сети могут создавать музыкальные композиции на основе обучения на большом количестве существующих произведений.

Другие методы создания музыки с помощью ИИ включают использование алгоритмов обработки естественного языка для анализа текстов и создания композиций, а также использование алгоритмов машинного

обучения для анализа стиля и звуковых характеристик различных музыкальных жанров.

И четвертые используются для создания ремиксов и аранжировок на основе уже существующих композиций. ИИ может генерировать новые звуковые эффекты, создавать ритмические и мелодические паттерны, а также анализировать тексты песен и определять их стиль и настроение. ИИ также может автоматически выравнивать биты и создавать петли.

Анализ предметной области же показал, что наиболее перспективной сферой для создания приложения является очищение аудиофайла от шума и помех. Актуальность создания приложения обусловлена тем, что при работе с аудио контентом всегда возникают проблемы шума из-за несовершенства записывающего оборудования или технических ошибок. Разрабатываемое приложение отличается универсальностью благодаря дополнительным функциями гибкости при использовании.

**Второй раздел** относится к выбору необходимого программного обеспечения для разработки и описанием функциональности приложения.

Некоторые из наиболее популярных языков:

- Python – язык программирования, широко используемый в области машинного обучения и искусственного интеллекта с множеством библиотек и фреймворков.

- C++ – язык программирования для создания высокопроизводительных приложений, включая те, которые используют нейросети.

- Java – язык программирования для создания приложений для обработки музыки, таких как плееры, эффекты и инструменты для записи звука.

- JavaScript – язык программирования для создания веб-приложений для обработки музыки, таких как веб-синтезаторы и эффекты.

Для создания ИИ и алгоритмов для обработки звука был выбран язык программирования Python. Преимущества этого выбора:

- Расширенная экосистема библиотек и фреймворков для задач

обработки аудио и машинного обучения, таких как Librosa, NumPy, SciPy, TensorFlow, PyTorch, MuseScore, Madlib и другие.

- Универсальность: Python может использоваться не только для обработки аудио, но и для решения других задач, таких как веб-разработка, анализ данных, автоматизация и так далее.

- Популярность в сфере машинного обучения: Python является доминирующим языком программирования в этой области, что позволяет использовать готовые решения и алгоритмы для создания сервисов обработки музыкального контента.

- Производительность: Python можно оптимизировать и использовать библиотеки, позволяющие ему работать достаточно быстро для задач обработки аудио.

- Поддержка GPU: Python имеет библиотеки, такие как TensorFlow и PyTorch, которые позволяют использовать вычислительные возможности GPU для ускорения обучения нейронных сетей и обработки аудиоданных.

Было решено реализовать два алгоритма для очистки аудиофайлов от шума: на основе библиотек Python и рекуррентных нейронных сетей (RNN).

RNN хорошо подходят для обработки последовательных данных, таких как аудиосигналы, и могут моделировать долгосрочные зависимости в данных. В качестве основы нейронной сети будет использована библиотека RNNNoise с открытым исходным кодом, предназначенная для шумоподавления в аудиофайлах. Она будет доработана и переобучена на массивах данных с зашумленными аудиофайлами с русскоязычным текстом.

Для реализации алгоритма очистки аудиофайлов от шума на основе библиотек Python были выбраны:

1. Noisereduce — библиотека для шумоподавления аудиозаписей, использующая различные алгоритмы, такие как спектральное вычитание, спектральное и временное усреднение.

2. Functools (reduce) — стандартная библиотека Python для создания функциональных инструментов, включая функции, связанные с

шумоподавлением.

Необходимо было уточнить работу каждой функции и её взаимодействие с пользователем:

1. Изменение громкости в LUFs для согласования уровней громкости различных аудиофайлов.
2. Изменение темпа в BPM для работы с музыкальной сферой.
3. Изменение тональности в полутонах через интерфейс приложения.
4. Распознавание и удаление шумов с помощью двух алгоритмов: процедурного и нейросетевого.
5. Вывод графика аудиофайла для визуального анализа его характеристик и качества звучания.

Все функции будут взаимосвязаны для поэтапной обработки аудиофайла. Кнопка «Сохранить» будет компилировать результаты обработки каждой функции в один файл.

И, наконец, были определены основные требования к интерфейсу: простота использования, функциональность, визуализация, актуальность дизайна.

Черты будущего интерфейса: простота, цветовая гамма, иконки, шрифты, типографика, макет, информативность.

**В третьем разделе** описаны этапы разработки сервиса для обработки музыкального контента.

Вначале были описаны алгоритмы очистки аудиофайлов от шума.

Первый алгоритм основан на библиотеках NumPy, NoiseReduce и PyDub. Реализация: если выбран аудиофайл, функция загружает его и преобразует в массив. Задаются начальные значения параметров: уменьшение шума, количество проходов обработки, шаг уменьшения параметра. Затем происходят множественные проходы с уменьшением шума. После завершения обработки, полученный очищенный сигнал конвертируется обратно в аудиоформат, и очищенный аудиофайл сохраняется в новый файл.

Второй алгоритм основан на нейронной сети, а именно на библиотеке с

открытым исходным кодом RNNoise, предназначенную для шумоподавления в аудиофайлах.

Опираясь на специфику своего приложения, сначала был доработан алгоритм шумоподавления аудиофайла, и дополнительно была обучена эта модель на собственной выборке данных.

Было автоматизировано извлечение фреймов из аудиозаписи без необходимости извлекать их вручную, как это реализовано в исходной библиотеке.

Вторым этапом было создание дополнительных функций для работы с аудиофайлами:

1. Изменение громкости.

Осуществляется с помощью алгоритма нормализации аудиосигнала. Процесс состоит из нескольких шагов: определение текущего уровня громкости, расчет необходимого изменения громкости, нормализация аудиосигнала.

2. Изменение тональности.

Включает в себя изменение его высоты звука без изменения продолжительности. Процесс состоит из следующих шагов: получение значения сдвига тональности, расчет новой частоты дискретизации, применение изменений к аудиосигналу.

3. Изменение темпа.

Позволяет ускорять или замедлять его воспроизведение без изменения высоты звука. Состоит из следующих шагов: получение желаемого BPM, расчет новой продолжительности, применение изменений к аудиосигналу.

Также для удобства пользования в приложении были созданы дополнительные опции: выбор аудиофайла, плашка «Ваш выбранный аудиофайл», возможность сохранения, возможность сбросить изменения и возможность удалить полученный на выходе аудиофайл.

Третьим этапом было создание интерфейса, состоящее из следующих шагов:

- создание макета интерфейса: на этом этапе было придумано расположение элементов на экране;
- выбор цветовой палитры и фонов: были определены основные цвета для интерфейса и подходящий фон;
- создание элементов интерфейса: были созданы элементы интерфейса, такие как кнопки, поля ввода и изображения;
- сборка и корректировка: после завершения создания элементов интерфейса они были соединены в готовый дизайн. При этом было проверено, что все элементы выглядят правильно, а неудачные решения были скорректированы и исправлены;
- экспорт и сохранение: после завершения дизайна он был экспортирован в формате JPEG, как и все его элементы, для дальнейшего использования в разработке программы (рисунок 1).

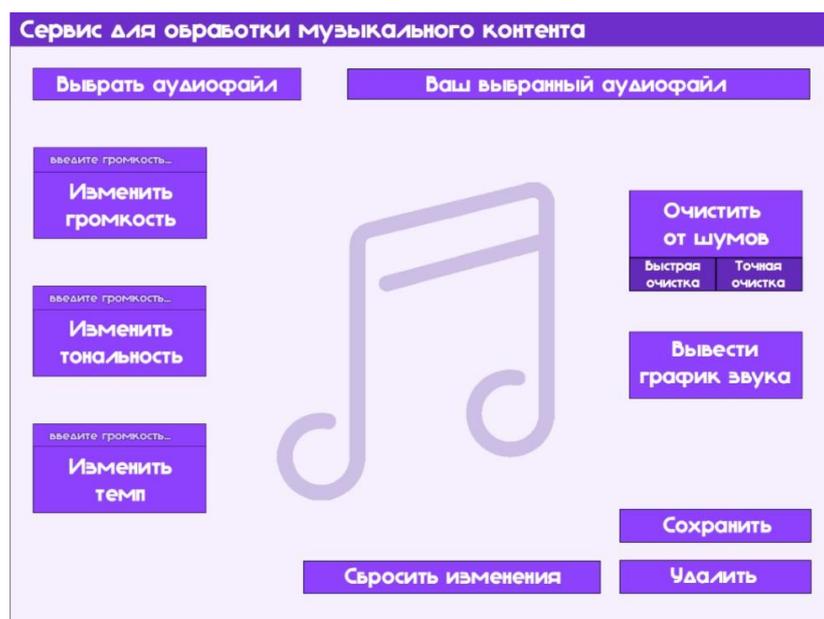


Рисунок 1 — Результат создания дизайна приложения

После создания приложения заключительным этапом стало его тестирование на примере зашумленной аудиозаписи с голосом и фрагментом звучания инструментальной партии (барабанов), который также был деформирован шумом.

Изначальные графики звуковой волны аудиозаписей можно увидеть на рисунках 2 и 3.

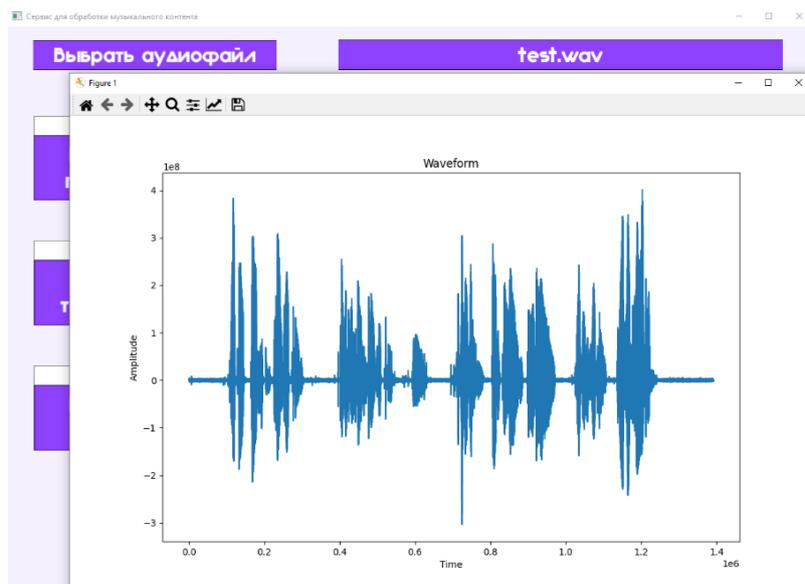


Рисунок 2 – Волновой график зашумленной аудиозаписи голоса

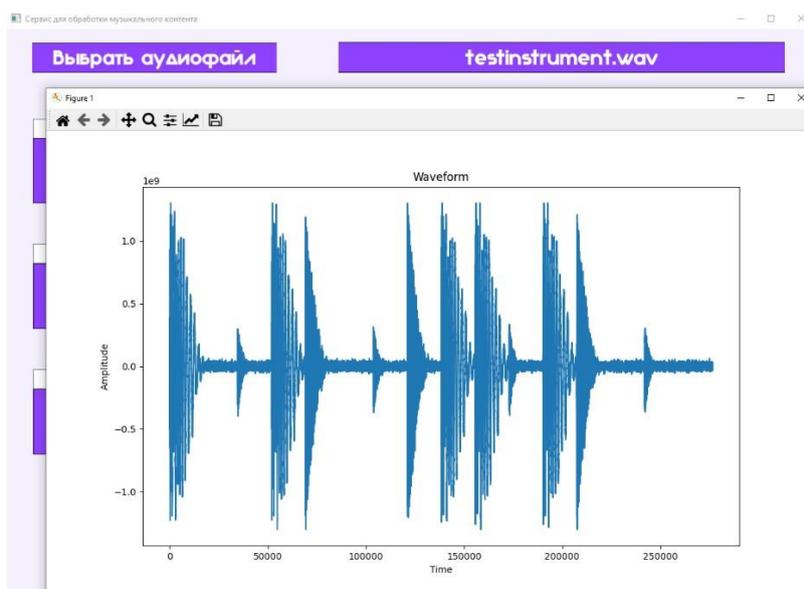


Рисунок 3 – Волновой график зашумленной инструментальной аудиозаписи

В результате обработки аудиозаписи голоса нейросетевым алгоритмом получили следующий результат:

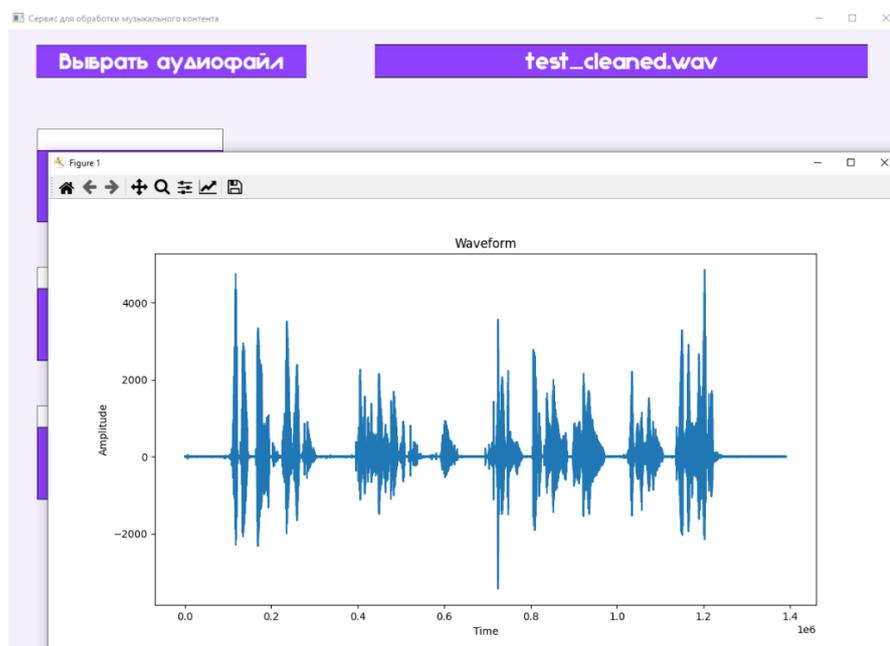


Рисунок 3 – Волновой график очищенной аудиозаписи голоса

И также в результате обработки обычным алгоритмом аудиозаписи с барабанной партией получим следующий график:

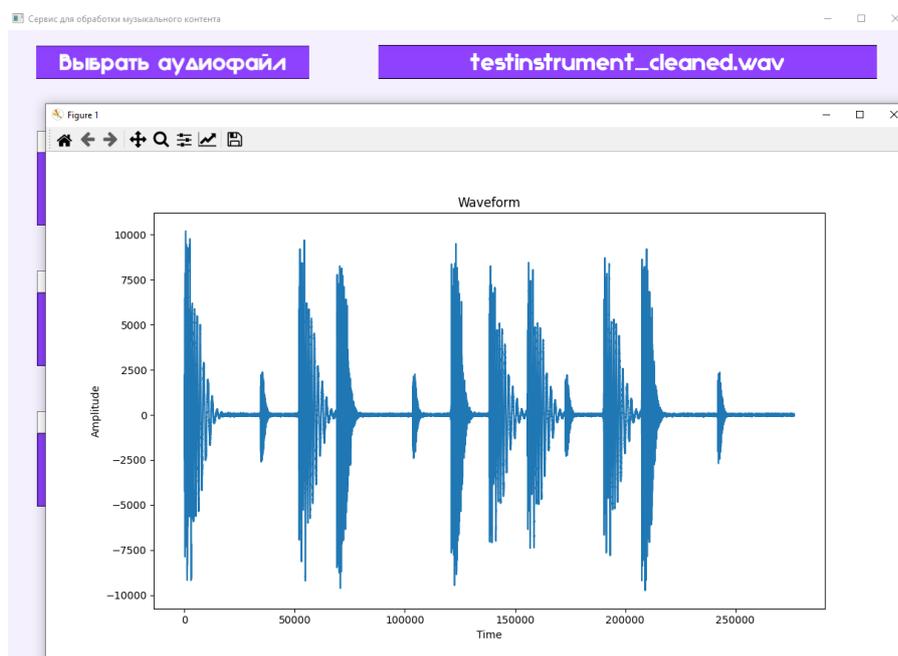


Рисунок 4 – Волновой график очищенной аудиозаписи барабанной партии

**В приложениях** представлены исходные программные коды реализации.

**Заключение.** В работе были рассмотрены три важных этапа становления приложения: анализ предметной области, проектирование и

реализация.

Было установлено, что использование ИИ в музыкальной индустрии обладает значительным потенциалом, открывая новые возможности для создания, обработки и распространения музыки.

На основе проведенного анализа был создан сервис для обработки музыкального контента, который позволяет пользователям выполнять различные операции с аудиофайлами, такие как очистка от шумов, улучшение качества звука, изменение темпа и тональности. При этом разработанное приложение является оригинальным и актуальным решением, которое может быть востребовано как профессиональными музыкантами, так и обычными пользователями.

В целом, данная работа является актуальным и перспективным исследованием и практическим проектом, который вносит вклад в развитие технологий обработки музыкального контента с использованием искусственного интеллекта.