Разработка и исследование программного обеспечения для моделирования певческого голоса на основе применения технологии SoftVC VITS

М.А. Киреенко, Е.Н. Антонянц, Е.Е. Истратова

***Аннотация* — В статье приведены результаты разработки и исследования программного обеспечения для моделирования певческого голоса на основе применения комплексного подхода, основанного на использовании нейронных сетей и технологий моделирования голоса на основе поиска и дифференцируемой цифровой обработки сигналов. В результате проведенного сравнительного анализа современных решений для моделирования голоса был выбран стек технологий для проектирования программного обеспечения. Готовая комплексная система включает четыре взаимосвязанных модуля для разделения аудиоконтента на основе технологии Spleeter, для генерации модели певческого голоса с применением архитектуры SoftVC VITS, для обучения сгенерированной модели, а также для совмещения аудиофайлов и получения финального результата. В рамках исследования программного обеспечения было проведено тестирование процесса обучения модели с использованием специально подготовленного набора данных, включающего 80 аудиозаписей продолжительностью по 9 секунд каждая. В качестве метрик для проведения исследования были использованы значения потерь дискриминатора, расстояния Кульбака-Лейблера, а также значения частотно-модуляционных и мел-кепстральных потерь. Полученные на протяжении 60000 итераций обучения количественные показатели подтвердили устойчивость конвергенции модели. В ходе исследования была отмечена способность модели сохранять характерные тембральные особенности голоса при одновременном обеспечении высокого качества синтезируемого певческого голоса, что имеет существенное практическое значение для различных приложений в области обработки аудиоконтента. Таким образом, результаты исследования продемонстрировали высокую эффективность предложенного комплексного подхода в решении задач моделирования певческого голоса.**

***Ключевые слова* — SoftVC VITS, генеративно-состязательные сети, нейронные сети, обработка аудиосигналов, моделирование певческого голоса.**

I. Введение

В области моделирования певческого голоса и обработки аудиосигналов в настоящее время применяется несколько ключевых технологических подходов, каждый из которых имеет свои особенности и ограничения. Так, наиболее распространенными являются следующие: моделирование певческого голоса на основе поиска (Retrieval based Voice Conversion), применение нейронной сети SoftVC VITS (Soft Voice Conversion VITS) и использование дифференцируемой цифровой обработки сигналов (Differentiable Digital Signal Processing) [1,2].

Разработанная в июне 2023 года технология обработки и моделирования голоса на основе поиска является ключевым трендом в области создания AI-каверов. Это объясняется тем, что в отличие от традиционных подходов, она позволяет минимизировать проблему утечки тембра за счет замены признаков входного источника признаками обучающего набора. При этом существенным преимуществом выступает возможность эффективного обучения даже при использовании видеокарт с относительно невысокой производительностью, что, в свою очередь, позволяет существенно расширить возможности и сферы ее применения. Кроме того, технология моделирования певческого голоса достаточно эффективна при работе с небольшими наборами данных. Так, для получения необходимого результата требуется от 10 минут исходных данных с низким уровнем шума. Данная технология также позволяет модифицировать тембр за счет слияния данных модели при помощи инструмента CKpt-merge. Интеграция с Ultimate Vocal Remover 5 (UVR 5) предоставляет дополнительную возможность для разделения вокала и аккомпанемента. Однако, несмотря на перечисленные преимущества, данная технология имеет определенные ограничения, связанные как с отсутствием автоматического определения оптимального количества эпох обучения, так и с необходимостью настройки параметров обучения и отбора результатов модели в ручном режиме [3].

Применение технологии моделирования певческого голоса на основе нейронной сети (SVC) открывает новые возможности в музыкальной индустрии, позволяя преобразовывать голос одного исполнителя в голос другого при сохранении содержания и мелодии оригинального произведения [4,5].

Параллельно с развитием технологии моделирования голоса на основе поиска значительный прогресс был

Статья получена 10 января 2025 г.

Киреенко Михаил Андреевич, Новосибирский государственный технический университет, Россия (e-mail: mikhail.lol.03@gmail.com).

Антонянц Егор Николаевич, Новосибирский государственный технический университет, Россия (e-mail: bax201438@gmail.com).

Истратова Евгения Евгеньевна, Новосибирский государственный технический университет, Россия (e-mail: istratova@mail.ru).

достигнут в области дифференцируемой цифровой обработки сигналов. Так, методика, приведенная в статье [6], представляет собой интеграцию классических методов обработки сигналов с использованием нейронных сетей. В данном подходе применяется дифференцируемая модель сигнала для обратного распространения ошибок и вычисления градиентов. Этот метод подтвердил свою эффективность в таких областях, как: синтез музыкального исполнения, синтез речи, моделирование звучания музыкальных инструментов, генерация звуковых эффектов, подбор звука синтезатора и моделирование певческого голоса.

II. Анализ существующих решений

Сравнительный анализ современных технологий для моделирования певческого голоса показал, что технология моделирования голоса на основе поиска обладает рядом преимуществ по сравнению с таким традиционным решением, как нейронная сеть So-Vits-SVC. В частности, данная технология обеспечивает более высокое качество выходного аудио при значительно меньшем времени обучения, которое составляет всего несколько часов для достижения максимального качества. При этом она предъявляет минимальные требования к обучающим данным, ограничиваясь необходимостью в качественных записях продолжительностью от 10 до 60 минут.

В то же время, технология дифференцируемой цифровой обработки сигналов базируется на фундаментальном подходе к решению проблем моделирования аудиосигналов, особенно в контексте устранения артефактов сигнала и обеспечения фазовой согласованности между последовательными кадрами. Несмотря на это, развитие обеих технологий продолжается в направлении повышения качества синтеза, уменьшения требований к вычислительным ресурсам и расширения возможностей применения в различных областях аудиообработки. Данные технологические решения вносят существенный вклад в сферу обработки аудиосигналов и открывают новые перспективы для дальнейших исследований и практического применения.

Независимо от выбранной технологии, практически все системы SVC включают два этапа: распознавание и синтез. На первом этапе используются независимые от голоса певца признаки, такие как: фонетические апостериограммы (PPG) [7-11] и результаты самообучения (SSL) [12,13], полученные на больших объемах немаркированных исходных данных. Эти результаты служат посредником для технологии SVC, которая может эффективно извлекать содержание и семантическую информацию из форм волн. На втором этапе акустические модели участвуют в генерации целевого аудио или акустических признаков из этих непосредственных данных.

Несмотря на существенные достижения в этой области, современные технологии SVC сталкиваются с рядом серьезных проблем. Наиболее значимой из которых является создание универсальной системы моделирования певческого голоса по принципу «любой-в-любой», способной работать с неизвестными целевыми исполнителями на основе лишь короткого эталонного образца их голоса. Особую сложность при этом представляет задача точного разделения и последующего воссоздания уникального вокального тембра исполнителя при сохранении содержания и мелодической структуры исходной песни. Сложность данной задачи обусловлена тем, что существующие подходы к ее решению часто сталкиваются с проблемой утечки тембра, особенно заметной при преобразовании голосов между исполнителями разного пола. Это происходит потому, что промежуточные данные модели, такие как PPG и признаки, полученные с помощью SSL, включают не только информацию о содержании, но и остаточные характеристики тембра исходного исполнителя. Попытки решить эту проблему путем добавления белого шума к скрытым или акустическим характеристикам часто приводят к нежелательным искажениям произношения и снижению качества звука.

В контексте этих вызовов особую актуальность приобретает разработка новых методов и инструментов для создания высококачественных систем моделирования певческого голоса. Значительным шагом вперед в решении этих проблем стало появление инструмента Spleeter, разработанного компанией Deezer [14]. Этот инструмент, использующий глубокие нейронные сети U-Net, позволяет эффективно разделять музыкальные композиции на отдельные стемы, представляющие собой изолированные аудиодорожки различных инструментов и вокала. Архитектура Spleeter, основанная на сверточной нейронной сети и обученная на обширном наборе мультитрековых данных, способна качественно отделять вокал от инструментального сопровождения, что существенно упрощает последующую обработку и моделирование голоса. Такой подход значительно повышает качество конечного результата, минимизируя искажения и артефакты, возникающие при работе с полным миксом. При этом важно отметить, что развитие данной технологии должно сопровождаться пониманием потенциальных рисков её использования, особенно в свете растущей проблемы аудио дипфейков и их возможного применения в целях дезинформации. Таким образом, цель исследования заключалась в разработке и тестировании программного обеспечения для моделирования певческого голоса на основе применения комплексного подхода.

III. Разработка программного обеспечения для моделирования певческого голоса

A. Выбор и обоснование инструментальных средств

В качестве основного метода для моделирования певческого голоса была выбрана нейронная сеть SoftVC VITS, позволяющая обеспечить высокую точность конвертации певческого голоса и сохранить тембральные особенности исходного голоса. Ключевым преимуществом применения данной нейронной сети

является минимальный требуемый объем исходных аудиоматериалов для обучения. SoftVC VITS поддерживает точное моделирование интонационных переходов и использует современные подходы машинного обучения, включая Sequence-to-Sequence архитектуру и вариационный вывод.

Для реализации программного обеспечения был выбран язык программирования Python, обладающий необходимым набором библиотек для работы с машинным обучением и имеющий встроенную

Рисунок 1 — Архитектура программного обеспечения

поддержку нейронной сети SoftVC VITS. Кроме того, данный язык программирования обеспечивает простоту и читаемость кода, при этом поддерживая кроссплатформенную разработку. Python также включает необходимые библиотеки для обработки аудио (Pydub) и разделения треков (Spleeter), что необходимо для реализации полного функционала программы.

Для разработки графического пользовательского интерфейса был выбран фреймворк PyQt6, который обеспечивает создание современного и адаптивного пользовательского интерфейса и обладает высокой производительностью при работе со сложными проектами. PyQt6 поддерживает кроссплатформенную разработку и имеет встроенный инструмент QtCreator для визуального проектирования. Основным преимуществом данного фреймворка является возможность его интеграции с библиотеками машинного обучения на основе Python.

Выбранный стек технологий позволил разработать программное обеспечение для моделирования певческого голоса с помощью SoftVC VITS, выполнить для него удобный графический интерфейс на PyQt6 и обеспечить надежную работу с аудиофайлами через специализированные библиотеки на языке Python. Таким образом, выбранный набор инструментов помог реализовать следующие основные функции программного обеспечения: разделение аудиофайлов на составляющие, обучение голосовых моделей, моделирование певческого голоса, микширование и совмещение аудиотреков.

Архитектура программного обеспечения основана на микросервисной модели и включает четыре основных модуля (рис. 1):

1) модуль для разделения исходного аудиофайла;
2) модуль для моделирования певческого голоса;
3) модуль для обучения модели;
4) модуль для заключительной обработки аудиофайлов.

B. Разработка модуля для разделения аудиофайла

Модуль для разделения музыки реализован при помощи класса SeparationWindow, который представляет собой графический интерфейс для взаимодействия с библиотекой Spleeter. Данный модуль был разработан с учетом необходимости предоставления пользователю удобного и интуитивно понятного интерфейса для разделения музыкальных композиций на составляющие их компоненты. Основными компонентами интерфейса являются кнопки выбора исходного файла, выпадающий список для выбора варианта разделения и элементы навигации.

Функциональность модуля основана на двух ключевых методах: select_song() и save_file(). Первый из них обеспечивает взаимодействие с файловой системой для выбора исходного аудиофайла, при этом была реализована автоматическая обработка имен файлов с заменой пробелов на нижнее подчеркивание для предотвращения потенциальных проблем при обработке. Второй метод отвечает за процессы разделения аудио и сохранения результатов, включая создание временной директории для хранения промежуточных результатов и последующее перемещение файлов в указанную пользователем директорию.

Модуль поддерживает три варианта разделения музыкального файла: разделение на вокал и аккомпанемент (2 дорожки), разделение на вокал, ударные, бас и остальные инструменты (4 дорожки), а также полное разделение на вокал, ударные, бас, фортепиано и остальные инструменты (5 дорожек). Выбор варианта разделения осуществляется при помощи выпадающего списка split_options, значение которого применяется при вызове функции separate().

Интеграция с библиотекой Spleeter позволила повысить качество разделения аудиофайлов. Для обеспечения надежности работы модуля была реализована система обработки ошибок, включающая

проверку на наличие выбранного файла перед началом обработки, корректное создание и удаление временных директорий, а также информирование пользователя о статусе операций через диалоговые окна.

Дополнительно было реализовано управление стилями интерфейса при помощи StyleManager, что позволило сформировать единый визуальный стиль программного обеспечения.

C. Разработка модуля для моделирования голоса

Модуль для моделирования певческого голоса был реализован на основе технологии SoftVC VITS. Причем в качестве основного компонента модуля был использован класс ConversionWorker, наследуемый от QObject и обеспечивающий асинхронное моделирование голоса в отдельном потоке.

Модуль был спроектирован на базе графического интерфейса ModelSelectionWindow, который предоставляет пользователю возможность выбора и настройки параметров для моделирования голоса. Интерфейс включает в себя элементы управления для выбора модели голоса, настройки высоты тона, соотношения звучания и уровня шума.

Ключевой особенностью данного модуля является возможность точной настройки параметров преобразования. Так, пользователь может регулировать высоту тона в диапазоне от -24 до +24 полутонов, что позволяет существенно изменять характер звучания голоса. Также в модуле была реализована возможность настройки соотношения между степенью схожести тембра и артикуляции за счет параметра cluster_ratio, принимающего значения от 0.0 до 1.0. Для контроля качества выходного сигнала предусмотрен параметр noise_scale, позволяющий регулировать уровень шума в диапазоне от 0.0 до 1.0. Для обработки голоса был реализован механизм пакетной обработки аудиофайлов. Входной аудиофайл разделяется на фрагменты, каждый из которых обрабатывается отдельно, что позволяет эффективно работать с длинными аудиозаписями.

Модуль также включает в себя систему обработки ошибок, которая охватывает различные сценарии: отсутствие необходимых файлов моделей, ошибки импорта модулей, проблемы с обработкой файлов. Все исключительные ситуации обрабатываются и предоставляют пользователю понятные сообщения об ошибках через диалоговые окна QMessageBox.

Результаты преобразования сохраняются в формате WAV, что обеспечивает высокое качество выходного аудиофайла. Модуль автоматически создает необходимую структуру каталогов для хранения временных и конечных файлов, используя библиотеку pathlib для кроссплатформенной работы с файловой системой. Разработанный модуль продемонстрировал высокую эффективность при решении задач моделирования голоса, обеспечивая качественный результат при сохранении исходных характеристик певческого голоса и его интонации. Гибкая система настроек позволяет достигать оптимального баланса между качеством моделирования и естественностью звучания певческого голоса.

D. Разработка модуля обучения модели голоса

Модуль для обучения был реализован на основе технологии SoftVC VITS и представляет собой интегрированное решение, обеспечивающее полный цикл подготовки, настройки и проведения процесса обучения данных модели. Структура модуля включает два элемента: ModelSetupWindow для начальной настройки модели и TrainingSetupWindow для управления процессом обучения. Каждый из этих элементов реализует свой собственный набор специфических функций, обеспечивая логическое разделение этапов работы с моделью.

Компонент ModelSetupWindow отвечает за первичную настройку модели и включает функции сбора и валидации базовой информации о модели, включая её наименование, выбор и проверку корректности датасета, установку необходимых начальных файлов модели, а также подготовку конфигурации для последующего процесса обучения. Особое внимание при разработке было уделено автоматизации процесса инициализации, в рамках которого происходит загрузка необходимых предварительно обученных моделей с платформы HuggingFace.

Компонент TrainingSetupWindow осуществляет непосредственно процесс обучения модели и обеспечивает настройку параметров обучения, включая количество эпох и интервал оценки, визуализацию процесса обучения через интеграцию с TensorBoard, мониторинг и логирование процесса обучения в режиме реального времени, управление GPU-ресурсами, а также сохранение и валидацию результатов обучения.

Процесс обучения модели был реализован как последовательность взаимосвязанных этапов, начинающихся с предварительной обработки датасета, включающей ресемплинг аудио до требуемой частоты дискретизации 44.1 кГц, создание конфигурационных файлов и извлечение признаков Hubert и F0. Далее следует этап настройки и валидации параметров обучения, на котором происходят установка количества эпох обучения, определение интервала оценки и конфигурация TensorBoard для визуализации процесса. Завершающим этапом является непосредственное выполнение процесса обучения, которое происходит асинхронно в отдельном потоке с постоянным мониторингом прогресса, обработкой исключительных ситуаций и сохранением промежуточных результатов.

Для обеспечения надежности и безопасности процесса обучения был реализован комплексный набор механизмов, включающий валидацию входных параметров, автоматическое создание и очистку временных директорий, сохранение контрольных точек обучения и обработку исключительных ситуаций при выполнении файловых операций. Результатом работы модуля являются обученные модели, сохраняемые в специализированной структуре каталогов, где G_best.pth представляет собой оптимальную версию генератора, D_best.pth - оптимальную версию дискриминатора, а config.json содержит конфигурационный файл с параметрами обучения. Разработанный модуль

обеспечивает надежный и эффективный процесс обучения моделей певческого голоса, предоставляя пользователю адаптивный интерфейс управления и мониторинга, что делает его ключевым компонентом готового программного обеспечения.

E. Разработка модуля для работы с аудиофайлами

Модуль для окончательной обработки аудиофайлов представляет собой инструмент для совмещения аудио и позволяет пользователям управлять отдельными аудиодорожками и объединять их в единую композицию. Данный модуль был реализован на базе библиотеки Pydub для манипуляции аудиофайлами. Основой модуля является класс AudioProcessor, который инкапсулирует всю логику работы с аудиофайлами. Данный класс реализует механизм кэширования оригинальных аудиофайлов, что позволяет значительно повысить производительность при многократных операциях с одними и теми же файлами. Кэширование осуществляется через словарь original_audio_cache, где ключом является имя файла, а значением – объект AudioSegment, содержащий аудиоданные.

Особое внимание было уделено управлению громкостью отдельных дорожек. Реализованный механизм позволяет регулировать громкость каждой дорожки в диапазоне от -60 до +20 дБ, причем изменения применяются неразрушающим образом - исходный файл остается неизменным, а модификации сохраняются во временных файлах. Для этого был использован отдельный словарь volume_data, хранящий значения корректировки громкости для каждого файла.

Процесс объединения аудиодорожек был реализован при помощи метода merge_tracks(), который принимает список выбранных файлов и выполняет их наложение с учетом установленных уровней громкости. Применение для объединения аудиофайлов метода overlay() библиотеки Pydub обеспечило корректное смешивание аудиосигналов. Результат сохраняется как во временной директории для предварительного прослушивания, так и в указанном пользователем месте для финального сохранения.

Для обеспечения интерактивного взаимодействия с пользователем было разработано диалоговое окно, включающее в себя элементы для управления воспроизведением, для регулировки громкости и визуализации процесса обработки аудио. Особенностью реализации является использование отдельного потока AudioPlayThread для воспроизведения аудио, что предотвращает блокировку интерфейса во время проигрывания файлов.

Для оптимизации производительности и управления ресурсами программы были реализованы механизмы автоматической очистки временных файлов и корректного освобождения системных ресурсов при завершении работы с аудио за счет использования контекстного управления файлами и явного закрытия аудиопотоков.

Тестирование модуля показало его эффективность при работе с различными форматами WAV-файлов, а также высокий уровень стабильности при длительном применении. В процессе исследования модуль успешно справлялся с задачами регулировки громкости и объединения множественных аудиодорожек, обеспечивая при этом высокое качество выходного аудиофайла.

F. Режимы работы программного обеспечения

Готовое программное обеспечение имеет два основных режима работы: создание собственной голосовой модели и моделирование голоса в существующей композиции.

При выборе первого режима процесс начинается со сбора и подготовки обучающих данных. Пользователь предоставляет аудиозаписи голоса, которые в дальнейшем будут использованы для обучения модели. Модуль для обучения производит анализ и обработку предоставленных образцов, после чего осуществляется настройка параметров обучения в соответствии с требованиями пользователя. Процесс завершается формированием уникальной голосовой модели, которая в дальнейшем может быть использована для моделирования голоса.

При выборе второго режима работы процесс начинается с разделения исходного аудиофайла на составляющие его компоненты при помощи технологии Spleeter. Аудиопроцессор выполняет сегментацию композиции, выделяя отдельные инструментальные партии, вокал и другие элементы, в зависимости от выбора пользователя. После успешного разделения программа осуществляет трансформацию вокальной партии с использованием выбранной голосовой модели. На этом этапе происходит моделирование голоса, то есть замена исходного вокала на вокал, сгенерированный при помощи нейронной сети. Финальным этапом является создание аудиотрека, в котором происходит объединение всех заранее подготовленных компонентов: инструментальной составляющей, смоделированного певческого голоса и дополнительных звуковых элементов. При этом происходит окончательная компоновка, результатом которой становится готовый аудиофайл.

Каждый из режимов работы предполагает активное взаимодействие пользователя с программным обеспечением, осуществляемое посредством графического интерфейса, который позволяет контролировать и настраивать параметры обработки на всех этапах и стадиях процесса. Таким образом, программное обеспечение поддерживает поэтапное выполнение операций, предоставляя пользователю возможность корректировки параметров на каждом этапе для достижения оптимального результата.

IV. Исследование программного обеспечения для моделирования певческого голоса

В рамках исследования был проведен комплексный анализ результатов обучения и оценки эффективности работы программного обеспечения. Процесс обучения модели проводился на наборе из 80 аудиозаписей длительностью по 9 секунд каждая, что обеспечило достаточный объем данных для формирования

качественной модели певческого голоса. Техническая база для обучения модели состояла из компьютера со следующими характеристиками: процессор Intel(R) Core(TM) i5-8400 CPU с тактовой частотой 2.80 ГГц, оперативная память объемом 16 ГБ и графический ускоритель NVIDIA GeForce GTX 1060 с 6 ГБ видеопамяти. Полный цикл обучения занял 36 часов, что является приемлемым показателем для модели данной сложности.

Анализ графика суммарных потерь дискриминатора (loss/d/total), представленного на рис. 2, показывает постепенное снижение значений потерь с 2.65 до примерно 2.45 на протяжении 60000 итераций обучения. Причем «зубчатая» структура графика является типовой для процесса обучения генеративно-состязательных сетей (GAN) и отражает конкурентную природу обучения между генератором и дискриминатором. Несмотря на наличие локальных колебаний, общий нисходящий тренд свидетельствует об успешной настройке параметров дискриминатора и его способности эффективно различать реальные и синтезированные образцы певческого голоса.

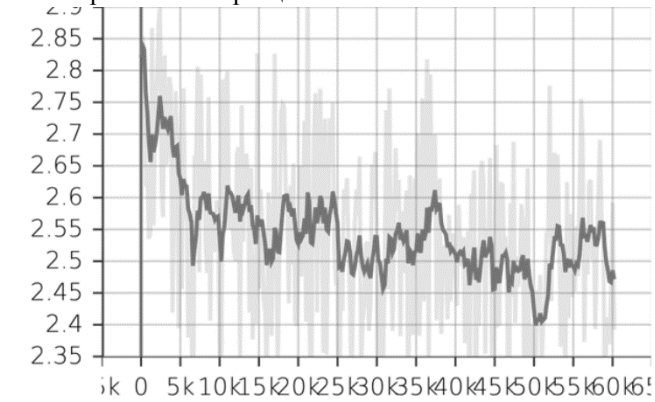

Рисунок 2 – График суммарных потерь дискриминатора (loss/d/total)

Особый интерес представляет график потерь по расстоянию Кульбака-Лейблера (loss/g/kl), показанный на рис. 3. Расстояние Кульбака-Лейблера является мерой расхождения между распределениями реальных и генерируемых данных, и его устойчивое снижение указывает на то, что генератор постепенно учится создавать образцы певческого голоса, все более близкие по своим статистическим характеристикам к реальным образцам [15]. Данный график демонстрирует существенное снижение значений с начального уровня около 1.0 до приблизительно 0.4-0.6 к концу обучения. Особенно показательно, что после 30000 итераций значения потерь стабилизируются в диапазоне 0.5-0.6, что говорит о достижении моделью определенного уровня зрелости в генерации паттернов.

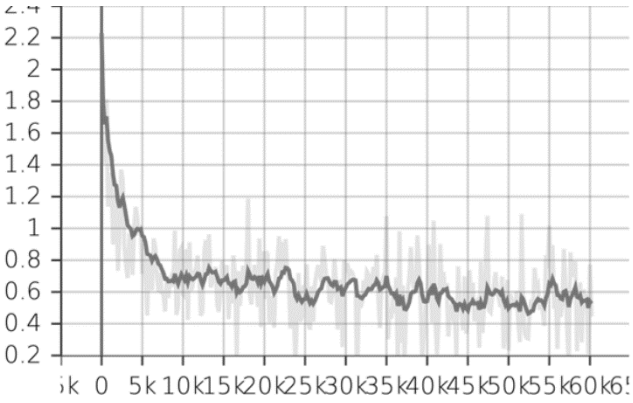

Рисунок 3 - График потерь по расстоянию Кульбака-Лейблера (loss/g/kl)

Анализ графика частотно-модуляционных потерь (loss/g/fm), представленного на рис. 4, показывает колебания значений в диапазоне от 6.5 до 8.5, с заметной тенденцией к повышению волатильности после 30000 итераций. Такое поведение характерно для процесса тонкой настройки частотных характеристик генерируемого сигнала. Увеличение значений потерь на поздних этапах обучения (45000-60000 итераций) может указывать на то, что модель начинает уделять больше внимания детальным частотным характеристикам голоса, пытаясь точнее воспроизвести тонкие нюансы акустического сигнала.

Рисунок 4 - График частотно-модуляционных потерь (loss/g/fm)

Существенным показателем качества обучения является график мел-кепстральных потерь (loss/g/mel), показанный на рис. 5. График демонстрирует устойчивое снижение значений с начального уровня около 17 до финального уровня около 14 единиц. Это снижение особенно заметно в первые 20000 итераций обучения, после чего график выходит на более плавную траекторию снижения. Мел-кепстральные потери являются ключевым индикатором качества синтезируемого голоса, так как они отражают способность модели точно воспроизводить спектральные характеристики голоса в восприятии человеческого слуха [16]. Стабилизация этого показателя на низком уровне к концу обучения свидетельствует о достижении моделью высокого качества синтеза певческого голоса.

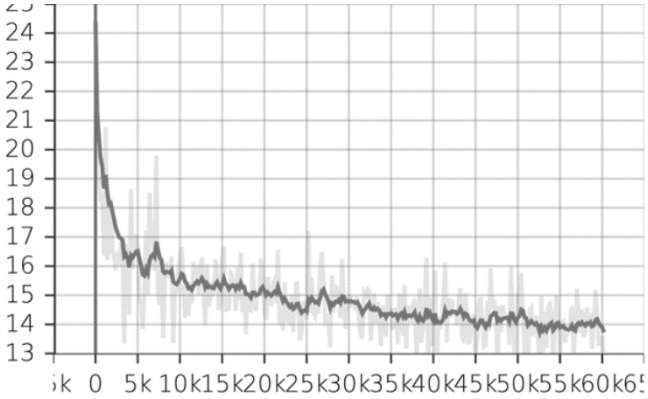

Рисунок 5 - График мел-кепстральных потерь (loss/g/mel)

График суммарных потерь генератора (loss/g/total), представленный на рис. 6, суммирует общую эффективность обучения генеративной части модели. На графике видно, что значения колеблются в диапазоне от 24 до 27 единиц, с общей тенденцией к небольшому снижению и стабилизации после 40000 итераций. Характерные колебания значений отражают сложный процесс балансировки различных компонентов функции потерь генератора. Важно отметить, что, несмотря на локальные скачки, общий тренд указывает на постепенное улучшение качества генерации, а стабилизация значений к концу обучения свидетельствует о достижении моделью устойчивого состояния.

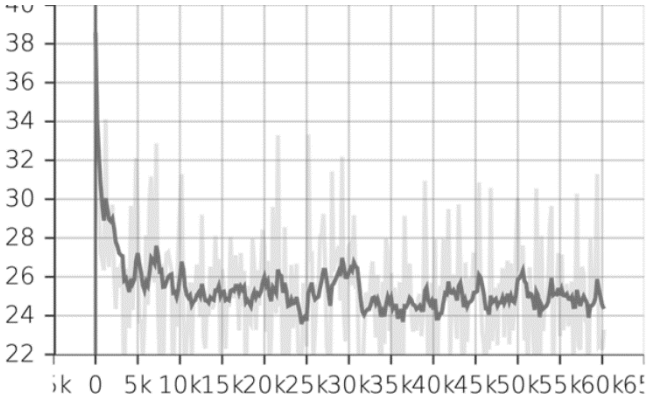

Рисунок 6 - График суммарных потерь генератора (loss/g/total)

Совместный анализ всех графиков потерь показывает, что процесс обучения модели прошел успешно, с достижением оптимального баланса между различными компонентами функции потерь. Особенно важно отметить согласованность динамики различных метрик: в то время как величина мел-кепстральных потерь устойчиво снижалась, значения частотно-модуляционных потерь показывали необходимый уровень детализации, а значения суммарных потерь генератора демонстрировали стабильное поведение. Это указывает на то, что модель успешно научилась воспроизводить как общие характеристики голоса, так и его тонкие спектральные особенности.

В результате проведенного обучения было получено 12 версий модели голоса, каждая из которых демонстрировала стабильные результаты при тестировании на реальных данных. Комплексный анализ всех метрик обучения позволил сделать выводы о том, что финальная версия модели достигла высокого уровня качества в задаче синтеза вокала, что подтверждается как объективными показателями потерь, так и субъективной оценкой качества генерируемых образцов певческого голоса.

V. Заключение

В результате проведенного исследования была подтверждена эффективность выбранного подхода для моделирования певческого голоса на основе технологии SoftVC VITS. Обучение модели проводилось на наборе из 80 аудиозаписей длительностью 9 секунд каждая, что обеспечило достаточный объем данных для формирования качественной модели моделирования вокала. Процесс обучения, занявший 36 часов, продемонстрировал устойчивое улучшение всех ключевых метрик на протяжении 60000 итераций. Анализ графика суммарных потерь дискриминатора показал снижение значений с 2.65 до 2.45, что свидетельствует о повышении способности модели различать реальные и синтезированные образцы голоса. Особенно показательной стала динамика потерь по расстоянию Кульбака-Лейблера, где наблюдалось существенное снижение значений с 1.0 до 0.4-0.6, причем после 30000 итераций значения стабилизировались в диапазоне 0.5-0.6, что указывает на достижение моделью зрелости в генерации речевых паттернов. График частотно-модуляционных потерь продемонстрировал колебания в диапазоне от 6.5 до 8.5 и заметное повышение волатильности после 30000 итераций, что указывает на процесс тонкой настройки частотных характеристик генерируемого сигнала. Мел-кепстральные потери показали устойчивое снижение с 17 до 14 единиц, особенно заметное в первые 20000 итераций обучения, что является ключевым индикатором качества синтезируемого голоса. График суммарных потерь генератора со значениями, колеблющимися в диапазоне от 24 до 27 единиц, продемонстрировал общую тенденцию к стабилизации после 40000 итераций, что свидетельствует о достижении моделью устойчивого состояния. Была отмечена согласованность динамики различных метрик: в то время как мел-кепстральные потери устойчиво снижались, частотно-модуляционные потери поддерживали необходимый уровень детализации, а суммарные потери генератора демонстрировали стабильное поведение. В результате было получено 12 версий модели голоса, каждая из которых показала стабильные результаты при тестировании на реальных данных. Достигнутые показатели свидетельствуют о высокой эффективности выбранного подхода и потенциале его дальнейшего развития в области преобразования певческого голоса.

Библиография

[1] Ren Y. et al. Fastspeech 2: Fast and high-quality end-to-end text to speech // arXiv preprint arXiv:2006.04558. – 2020.

[2] Shen K. et al. Naturalspeech 2: Latent diffusion models are natural and zero-shot speech and singing synthesizers // arXiv preprint arXiv:2304.09116. – 2023.
[3] Qian K. et al. Autovc: Zero-shot voice style transfer with only autoencoder loss // International Conference on Machine Learning. – PMLR, 2019. – P. 5210–5219.
[4] Gu Y. et al. Bytesing: A chinese singing voice synthesis system using duration allocated encoder-decoder acoustic models and wavernn vocoders // 2021 12th International Symposium on Chinese Spoken Language Processing (ISCSLP). – IEEE, 2021. – P. 1–5.
[5] Cui J. et al. Sifisinger: A High-Fidelity End-to-End Singing Voice Synthesizer Based on Source-Filter Model // ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). – IEEE, 2024. – P. 11126–11130.
[6] Hayes B. et al. A review of differentiable digital signal processing for music and speech synthesis // Frontiers in Signal Processing. – 2024. – T. 3. – P. 1284100.
[7] Sun L. et al. Phonetic posteriorgrams for many-to-one voice conversion without parallel data training // 2016 IEEE International Conference on Multimedia and Expo (ICME). – IEEE, 2016. – P. 1–6.
[8] Polyak A. et al. Unsupervised cross-domain singing voice conversion // arXiv preprint arXiv:2008.02830. – 2020.
[9] Liu S. et al. Fastsvc: Fast cross-domain singing voice conversion with feature-wise linear modulation // 2021 ieee international conference on multimedia and expo (icme). – IEEE, 2021. – P. 1–6.
[10] Liu S. et al. Diffsvc: A diffusion probabilistic model for singing voice conversion // 2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). – IEEE, 2021. – P. 741–748.
[11] Li Z. et al. Ppg-based singing voice conversion with adversarial representation learning //ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). – IEEE, 2021. – P. 7073-7077.
[12] Jayashankar T. et al. Self-supervised representations for singing voice conversion // ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). – IEEE, 2023. – P. 1–5.
[13] Zhou Y. et al. VITS-based Singing Voice Conversion System with DSPGAN post-processing for SVCC2023 // 2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). – IEEE, 2023. – P. 1–8.
[14] Hennequin R. et al. Spleeter: a fast and efficient music source separation tool with pre-trained models // Journal of Open Source Software. – 2020. – T. 5. – № 50. – P. 2154.
[15] Delgado-Gutiérrez G. et al. Acoustic environment identification by Kullback–Leibler divergence // Forensic Science International. – 2017. – T. 281. – P. 134–140.
[16] Abdul Z. K., Al-Talabani A. K. Mel frequency cepstral coefficient and its applications: A review // IEEE Access. – 2022. – T. 10. – P. 122136–122158.

Киреенко Михаил Андреевич. Новосибирский государственный технический университет, г. Новосибирск, Россия. Студент бакалавриата факультета автоматики и вычислительной техники. Область научных интересов: глубокое обучение, нейронные сети системы компьютерного зрения.
Антонянц Егор Николаевич. Новосибирский государственный технический университет, г. Новосибирск, Россия. Аспирант факультета автоматики и вычислительной техники. Количество печатных работ: 21. Область научных интересов: машинное обучение, системы компьютерного зрения, информационные сети.
Истратова Евгения Евгеньевна. Новосибирский государственный технический университет, г. Новосибирск, Россия. Кандидат технических наук, доцент кафедры автоматизированных систем управления. Количество печатных работ: 164. Область научных интересов: информационные технологии, информационные сети, системы компьютерного зрения. e-mail: istratova@mail.ru (ответственная за переписку).

Development and research of software for modeling a singing voice based on SoftVC VITS technology

M.A. Kireenko, E.N. Antonyants, E.E. Istratova

***Abstract* — The article presents the results of the development and research of software for modeling a singing voice based on the use of an integrated approach based on the use of neural networks and voice modeling technologies based on search and differentiable digital signal processing. As a result of the comparative analysis of modern solutions for voice modeling, a stack of technologies for software design was selected. The finished complex system includes four interconnected modules for separating audio content based on Spleeter technology, for generating a singing voice model using the SoftVC VITS architecture, for training the generated model, as well as for combining audio files and obtaining the final result. As part of the software research, the model training process was tested using a specially prepared dataset, including 80 audio recordings of 9 seconds each. The discriminator loss values, the Kullback-Leibler distance, as well as the frequency-modulation and mel-cepstral loss values were used as metrics for the study. The quantitative indicators obtained over 60,000 training iterations confirmed the stability of the model convergence. The study noted the ability of the model to preserve the characteristic timbre features of the voice while simultaneously providing high quality of the synthesized singing voice, which is of significant practical importance for various applications in the field of audio content processing. Thus, the results of the study demonstrated the high efficiency of the proposed integrated approach in solving the problems of singing voice modeling.**

***Keywords* — SoftVC VITS, generative adversarial networks, neural networks, audio signal processing, singing voice modeling.**

REFERENCES

[1] Ren Y. et al. Fastspeech 2: Fast and high-quality end-to-end text to speech // arXiv preprint arXiv:2006.04558. – 2020.

[2] Shen K. et al. Naturalspeech 2: Latent diffusion models are natural and zero-shot speech and singing synthesizers // arXiv preprint arXiv:2304.09116. – 2023.

[3] Qian K. et al. Autovc: Zero-shot voice style transfer with only autoencoder loss // International Conference on Machine Learning. – PMLR, 2019. – P. 5210–5219.

[4] Gu Y. et al. Bytesing: A chinese singing voice synthesis system using duration allocated encoder-decoder acoustic models and wavernn vocoders // 2021 12th International Symposium on Chinese Spoken Language Processing (ISCSLP). – IEEE, 2021. – P. 1–5.

[5] Cui J. et al. Sifisinger: A High-Fidelity End-to-End Singing Voice Synthesizer Based on Source-Filter Model // ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). – IEEE, 2024. – P. 11126–11130.

[6] Hayes B. et al. A review of differentiable digital signal processing for music and speech synthesis // Frontiers in Signal Processing. – 2024. – T. 3. – P. 1284100.

[7] Sun L. et al. Phonetic posteriorgrams for many-to-one voice conversion without parallel data training // 2016 IEEE International Conference on Multimedia and Expo (ICME). – IEEE, 2016. – P. 1–6.

[8] Polyak A. et al. Unsupervised cross-domain singing voice conversion // arXiv preprint arXiv:2008.02830. – 2020.

[9] Liu S. et al. Fastsvc: Fast cross-domain singing voice conversion with feature-wise linear modulation // 2021 ieee international conference on multimedia and expo (icme). – IEEE, 2021. – P. 1–6.

[10] Liu S. et al. Diffsvc: A diffusion probabilistic model for singing voice conversion // 2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). – IEEE, 2021. – P. 741–748.

[11] Li Z. et al. Ppg-based singing voice conversion with adversarial representation learning //ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). – IEEE, 2021. – P. 7073-7077.

[12] Jayashankar T. et al. Self-supervised representations for singing voice conversion // ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). – IEEE, 2023. – P. 1–5.

[13] Zhou Y. et al. VITS-based Singing Voice Conversion System with DSPGAN post-processing for SVCC2023 // 2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). – IEEE, 2023. – P. 1–8.

[14] Hennequin R. et al. Spleeter: a fast and efficient music source separation tool with pre-trained models // Journal of Open Source Software. – 2020. – T. 5. – № 50. – P. 2154.

[15] Delgado-Gutiérrez G. et al. Acoustic environment identification by Kullback–Leibler divergence // Forensic Science International. – 2017. – T. 281. – P. 134–140.

[16] Abdul Z. K., Al-Talabani A. K. Mel frequency cepstral coefficient and its applications: A review // IEEE Access. – 2022. – T. 10. – P. 122136–122158.