

ОЛИМПИАДА ШКОЛЬНИКОВ «ШАГ В БУДУЩЕЕ»
ПО ПРОФИЛЮ «ИНЖЕНЕРНОЕ ДЕЛО»

38089

регистрационный номер

Компьютерные системы

название секции

**Алгоритм сглаживания посторонних шумов в аудиозаписи для улучшения
качества восприятия передаваемой в ней информации**

название работы

Автор:

Рамазанов Абдул Хаирбегович

фамилия, имя, отчество

ГБОУ Школа №1155, 10

наименование учебного заведения, класс

Научный руководитель:

Юсупова Кристина Олеговна

фамилия, имя, отчество

ГБОУ Школа №1155

место работы

учитель информатики

звание, должность

подпись научного руководителя

Алгоритм сглаживания посторонних шумов в аудиозаписи для улучшения качества восприятия передаваемой в ней информации

Аннотация

Создание программы для сведения к минимуму слышимости шума, мешающего восприятию информации и звука в аудиофайле, является важной задачей в области цифровой обработки сигналов и аудиотехники. Нормализация звука требуется во многих ситуациях: при передаче специального сигнала, в аудио-производстве, для удобства в прослушивании личных голосовых сообщений. Целью данного проекта является разработка программы для снижения уровня лишнего шума в аудиозаписи, который мешает восприятию информации и ухудшает качество основного звука в аудиофайле.

Созданная программа получает на вход запись голоса в формате .wav (при необходимости, можно настроить загрузку файла и другого формата) и производит дальнейшее преобразование аудио сигнала в цифровой вид. Далее с помощью специального алгоритма выполняется определение речевого сигнала и дальнейшее подавление сторонних шумов, отличающихся своими свойствами от первого. Главным достижением проекта является возможность программным образом устранить, кроме статического шума, еще и нестационарные шумы. Это шумы более сложной структуры, импульсные, прерывистые, которые длятся короткое промежутки времени. Программа после обработки обеспечивает сохранение обработанного аудиофайла в нужном формате.

Реализация программы выполнена на языке Python в среде разработки PyCharm Community Edition 2021.3.2. На основе разработанного алгоритма был также создан Telegram-бот для использования функционала пользователями при работе с некачественными голосовыми сообщениями.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
ОСНОВНАЯ ЧАСТЬ.....	5
1 Представление характеристик записи звука	5
2 Алгоритм сглаживания шума и его программная реализация.....	8
3 Тестирование программы	12
ЗАКЛЮЧЕНИЕ	15
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ	16
Приложение А. Отзыв.....	17

ВВЕДЕНИЕ

Создание программы для сведения к минимуму слышимости шума, мешающего восприятию информации и звука в аудиофайле, является важной задачей в области цифровой обработки сигналов и аудиотехники. Нормализация звука требуется во многих ситуациях: при передаче специального сигнала, в аудио-производстве, для удобства в прослушивании личных голосовых сообщений. В работе создана программа, которая получает на вход запись голоса в каком-либо формате (в проекте представлена работа с wav-файлами) и производит дальнейшее преобразование аудио сигнала в цифровой вид, а далее с помощью специального алгоритма выполняется определение речевого сигнала и дальнейшее подавление сторонних шумов, отличающихся своими свойствами от первого. Главным достижением проекта является возможность программным образом устранить, кроме статического шума, еще и нестационарные шумы. Это шумы более сложной структуры, импульсные, прерывистые, которые длятся короткие промежутки времени. Программа после обработки обеспечивает сохранение обработанного аудиофайла в нужном формате.

Реализация программы выполнена на языке Python в среде разработки PyCharm Community Edition 2021.3.2. На основе разработанного алгоритма был также создан Telegram-бот для использования функционала пользователями при работе с некачественными голосовыми сообщениями.

Целью данного проекта является разработка программы для снижения уровня лишнего шума в аудиозаписи, который мешает восприятию информации и ухудшает качество основного звука в аудиофайле.

Для достижения поставленной цели были выполнены следующие задачи:

1. изучение различных методов и алгоритмов подавления шума в аудиофайлах, включая фильтрацию, спектральный анализ, адаптивные алгоритмы и машинное обучение;
2. анализ аудиофайла и идентификация основного голоса с помощью программы Audacity;

3. разработка алгоритма фильтрации шумов с учетом особенностей человеческого голоса;
4. тестирование алгоритма на различных аудиофайлах с различными типами шумов;
5. реализация алгоритма на языке программирования Python с использованием соответствующих библиотек;
6. обеспечение сохранения обработанного аудиофайла в нужном формате;
7. создание Telegram-бота для мгновенного доступа пользователя к функционалу алгоритма и получению результата.

ОСНОВНАЯ ЧАСТЬ

1 Представление характеристик записи звука

Входным файлом для будущей программы является запись звука в формате .wav (при желании, можно выбрать любой другой формат). Рассмотрим структуру содержащегося в файле звукового сигнала. Для анализа и визуализации различных аспектов звука, таких как речь, музыка или шум, используется спектрограмма. Спектрограмма представляет собой графическое представление спектра звукового сигнала в зависимости от времени. Она является визуальным способом представления того, как меняется частотный состав аудиосигнала в течение времени.

Для визуальной оценки характеристик сигнала, таких как амплитуда, длительность, частота и форма волны, будем пользоваться волноформой. Волноформа (или временная область) представляет собой график амплитуды звукового сигнала в зависимости от времени. Это простое графическое представление аудиосигнала, где по оси абсцисс откладывается время, а по оси ординат — амплитуда звукового давления. Каждая точка на волноформе представляет собой амплитуду сигнала в определенный момент времени [1].

На рисунке 1 покажем спектрограмму и волноформу голоса человека, записанного в отсутствие шума.

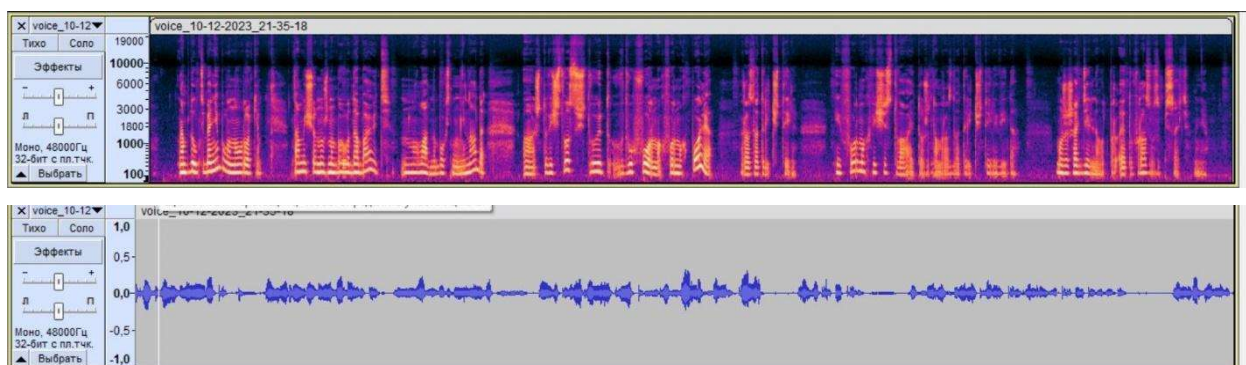


Рисунок 1 — Спектрограмма и волноформа голоса человека

Как видно по рисунку 1 спектрограмма голоса человека содержит четкие полосы частот, соответствующие акустическим характеристикам, которые являются резонансными пиками, создаваемыми речевыми органами. Изменения в высоте тона, тембра и интенсивности визуально выделены в спектрограмме. Волноформа голоса человека содержит периодические колебания, связанные с высокочастотными колебаниями голосовых складок в гортани. Амплитуда звуковых колебаний на волноформе изменяется в зависимости от интенсивности и громкости речи. Форма волны содержит информацию о звуковых характеристиках, таких как согласные, гласные и паузы в речи [2].

Рассмотрим теперь визуализацию голоса человека на записи с присутствием шума (рис. 2).

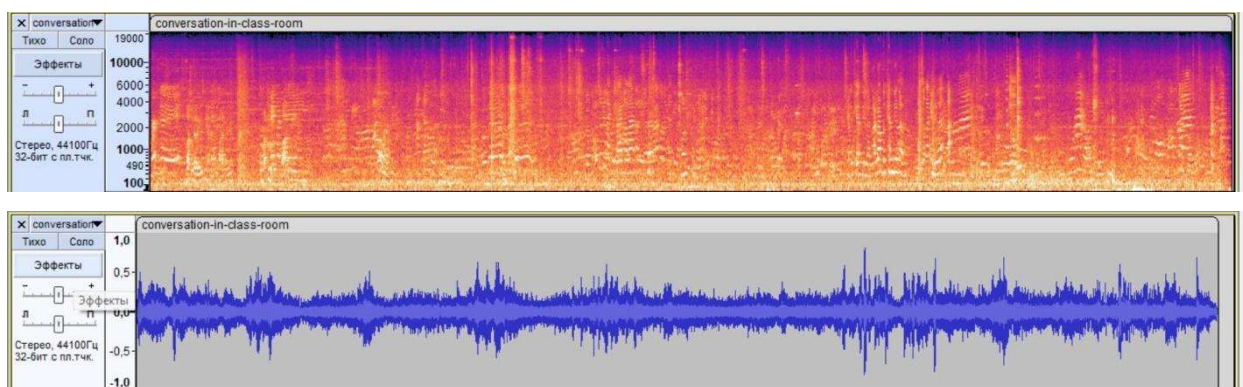


Рисунок 2 — Спектрограмма и волноформа голоса человека с шумом

На спектрограмме рисунка 2 отчетливо видны частотные полосы, соответствующие голосовым характеристикам человека. Это могут быть форманты (резонансные пики, создаваемые речевыми органами), интонационные особенности и другие особенности речи. В спектрограмме также видны частотные компоненты, характерные для шума. Например, если шум содержит низкочастотные резкости или высокочастотные шумы, они будут выражены на спектрограмме в виде дополнительных полос или пятен [2].

В волноформе же видны периодические колебания, связанные с голосовыми складками человека. На волноформе можно наблюдать добавочные случайные колебания, которые представляют шум в аудиосигнале.

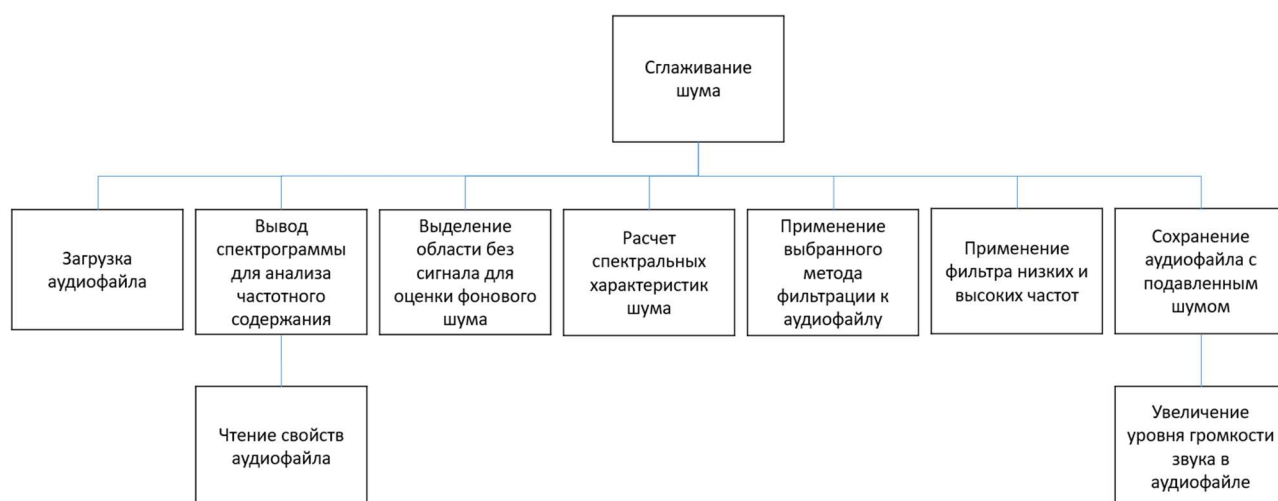


Рисунок 3 — Структурная схема алгоритма сглаживания шума

Перечисленные выше изображения зависимостей для звука являются вспомогательным инструментом при анализе работоспособности созданной программы, которая подавляет шум в аудиозаписи. На результаты вывода спектрограммы и волноформы до и после обработки алгоритмом, мы ориентировались в процессе корректировки алгоритма. В итоге получили быстрый инструмент, способный улучшить качество аудиозаписи. На рисунке 3 представлена структурная схема программы. На схеме видно, что был реализован авторский подход по устранению шума из аудиофайла, в котором применяются как известный набор способов фильтрации, так и точечный анализ

звука на наличие как статического (шум города, пылесоса, ветра), так и нестационарного шума (резкий сигнал автомобиля, крик птицы).

2 Алгоритм сглаживания шума и его программная реализация

В проекте также разработана реализация алгоритма на языке Python с помощью следующего набора вспомогательных библиотек.

Librosa — это библиотека Python для анализа музыки и аудио, которая помогает разработчикам программного обеспечения создавать приложения для работы с музыкальными файлами. Библиотека очень проста в использовании и может выполнять как базовые, так и сложные задачи, связанные с обработкой аудио и музыки. Библиотека имеет открытый исходный код и находится в свободном доступе по лицензии ISC.

Matplotlib — это библиотека на языке программирования Python для создания графиков и визуализации данных. Она предоставляет широкий спектр функций для создания различных типов графиков, включая линейные, точечные, столбчатые, гистограммы, круговые диаграммы и многое другое. Matplotlib часто используется в научных и инженерных приложениях, а также в области анализа данных и машинного обучения [3].

Ryudub — это библиотека на языке Python, предназначенная для удобной обработки аудиофайлов. С ее помощью можно выполнять различные операции над звуковыми данными, такие как чтение и запись аудиофайлов, нарезка, склейка, изменение громкости, преобразование форматов и многое другое [4].

SciPy — это библиотека Python с открытым исходным кодом, предназначенная для решения научных и математических проблем. Она построена на базе NumPy и позволяет управлять данными, а также визуализировать их с помощью разных высокоуровневых команд. При импорте SciPy отдельно импортировать NumPy не нужно.

Noisereducer — это библиотека алгоритмов шумоподавления на Python, которая использует методы обработки сигналов, такие как фильтры и статистические методы, для удаления шума из аудио. Эти методы могут

варьироваться в зависимости от реализации, но общая идея заключается в том, чтобы изолировать компоненты аудио, которые представляют собой шум, и уменьшить их влияние на остальной сигнал.

Программная реализация алгоритма начинается с загрузки аудиофайла с помощью `wave.open()`. Далее происходит вывод спектрограммы и волноформы исходного аудиофайла

Прежде чем построить график волноформы, нам нужно рассчитать время каждого сэмпла (небольшой оцифрованный звуковой фрагмент). Другими словами, это просто общая длина аудиофайла в секундах, деленная на количество сэмплов. Мы можем использовать `linspace` от `numpy`, чтобы создать массив временных меток. Построение графика выполняется с помощью класса `pyplot` от `matplotlib`. Для простоты мы строим график сигнала только с одного канала. Покажем на рисунке соответствующий фрагмент программы.

```
27 sample_freq = wav_obj.getframerate()#График волноформы
28 n_samples = wav_obj.getnframes()
29 t_audio = n_samples/sample_freq
30 n_channels = wav_obj.getnchannels()
31 signal_wave = wav_obj.readframes(n_samples)
32 signal_array = np.frombuffer(signal_wave, dtype=np.int16)
33 l_channel = signal_array[0::2]
34
35 times = np.linspace( start= 0, n_samples/sample_freq, num=n_samples)
36
37 plt.figure(figsize=(15, 8))#График спектрограммы |
38 plt.plot( *args: times, l_channel)
39 plt.title('Left Channel')
40 plt.ylabel('Signal Value')
41 plt.xlabel('Time (s)')
42 plt.xlim( *args: 0, t_audio)
43 plt.show()
```

Рисунок 4 — Программное построение графиков

На первых этапах обработки применяется фильтр скользящей средней (рис. 5). Математически скользящее среднее определяет среднее значение сигнала за определенный промежуток времени и, таким образом, представляет собой метод сглаживания сигнала, накопленного за это время, поэтому его можно рассматривать как пример фильтра нижних частот, используемого при обработке сигналов. Фильтр нижних частот — это фильтр, эффективно

пропускающий частотный спектр сигнала ниже некоторой частоты (частоты среза) и подавляющий частоты сигнала выше этой частоты. Этот фильтр со скользящим средним является базовым фильтром, и он довольно прост в использовании. В данной программе для фильтра была установлена частота среза 350 Гц [2].

```
58 cutOffFrequency = 350.0
59
60 1 usage
61 def running_mean(x, windowSize):
62     cumsum = np.cumsum(np.insert(x, 0, values: 0))
63     return (cumsum[windowSize:] - cumsum[:-windowSize]) / windowSize
64 1 usage
65 def interpret_wav(raw_bytes, n_frames, n_channels, sample_width, interleaved = True):
66     if sample_width == 1:
67         dtype = np.uint8
68     elif sample_width == 2:
69         dtype = np.int16
70     channels = np.fromstring(raw_bytes, dtype=dtype)
71     if interleaved:
72         channels.shape = (n_frames, n_channels)
73         channels = channels.T
74     else:
75         channels.shape = (n_channels, n_frames)
76     return channels
77 with contextlib.closing(wave.open(fname, mode: 'rb')) as spf:
78     sampleRate = spf.getframerate()
79     ampWidth = spf.getsampwidth()
80     nChannels = spf.getnchannels()
81     nFrames = spf.getnframes()
82
83     signal = spf.readframes(nFrames*nChannels)
84     spf.close()
85     channels = interpret_wav(signal, nFrames, nChannels, ampWidth, interleaved: True)
86
87     freqRatio = (cutOffFrequency/sampleRate)
88     N = int(math.sqrt(0.196196 + freqRatio**2)/freqRatio)
89     filtered = running_mean(channels[0], N).astype(channels.dtype)
90     wav_file = wave.open(outname, mode: "w")
91     wav_file.setparams((1, ampWidth, sampleRate, nFrames, spf.getcomptype(), spf.getcompname()))
92     wav_file.writeframes(filtered.tobytes('C'))
93     wav_file.close()
```

Рисунок 5 — Реализация алгоритма фильтра нижних частот

В программе также используется фильтр верхних частот из библиотеки Pydub (рис. 6). Далее же с помощью фильтра из библиотеки noisereducer убирается оставшийся шум. Данный фильтр первые пять секунд аудиофайла использует для оценки шума. Такой подход подразумевается в алгоритме фильтра потому, что зачастую в аудиозаписях в начале может наблюдаться

отсутствие речи или другие фоновые звуки, которые можно считать шумом. Этот шум используется в дальнейшем для создания профиля шума.

```
99 muffled_song = audio.high_pass_filter(1500)
100
101 audio_data, sample_rate = librosa.load("C://Users/Muhammed/PycharmProjects/pythonProject6/sample-12s.wav")
102
103 reduced_noise = nr.reduce_noise(y=audio_data, sr=sample_rate)
104
105 sf.write( file: "C://Users/Muhammed/PycharmProjects/pythonProject6/sample-15s.wav", reduced_noise, samplerate: 22200, subtype: 'PCM_24')
106
107 wav_obj1 = wave.open( f: "C://Users/Muhammed/PycharmProjects/pythonProject6/sample-15s.wav", mode: 'rb')
```

Рисунок 6 – Применение фильтра верхних частот

После того, как шум определен, он уменьшается по всему аудиофайлу. И после применения перечисленных фильтров основная информация в аудиофайле становится заметно тише. И для улучшения слышимости в программе настраивается громкость звука – увеличивается на 25 децибел. В финальной части кода в программе предусмотрен вывод спектрограммы и волноформы отфильтрованного звука для дальнейшего анализа и проверки работоспособности программы.

3 Тестирование программы

Покажем на примерах эффективность работы программы. С помощью спектрограмм и волноформ аудиофайла до обработки и после произведем сравнение этих файлов, визуально выявим отличия. На рисунке 7 представлены волноформа и спектрограмма для исходного аудиофайла, в котором присутствует монотонный статический шум без резких коротких сигналов.

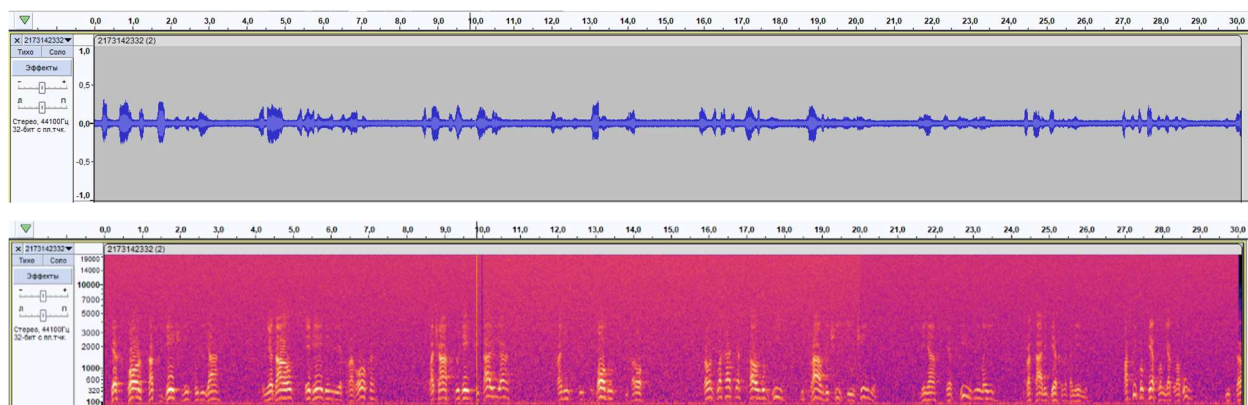


Рисунок 7 – Волноформа и спектрограмма для аудио с монотонным шумом

После обработки данного аудиофайла программа сохраняет полученный файл, в котором уже шум сведен к минимуму, и очень отчетливо слышно основной голос в аудиозаписи без посторонних звуков. На спектрограмме визуально можно передать разницу между этими файлами (рис. 8).

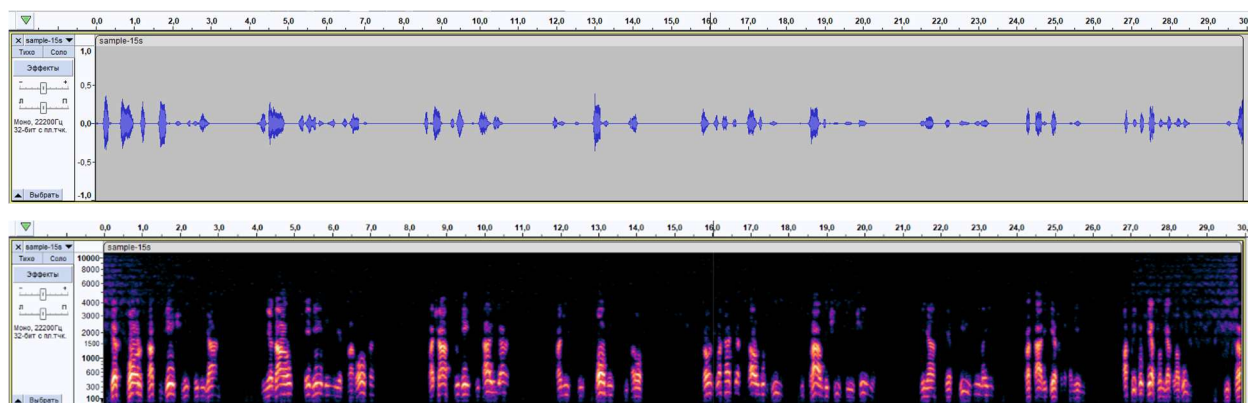


Рисунок 8 – Волноформа и спектрограмма обработанного аудио

Рассмотрим другой вариант аудиофайла, когда по ходу воспроизведения присутствует несколько типов шумов, явно выраженные звуки от нескольких сторонних источников (рис. 9).

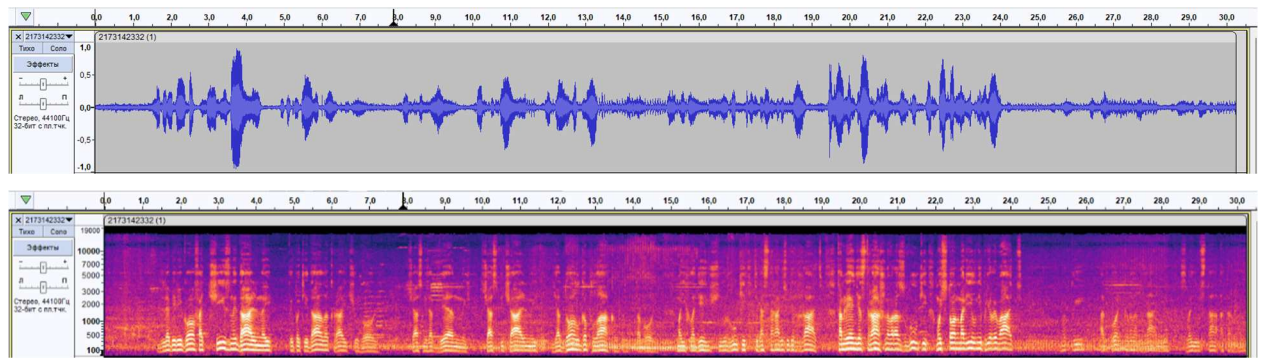


Рисунок 9 – Волноформа и спектрограмма аудио с несколькими типами шумов

Программа выполняет качественную обработку и в таких случаях. На рисунке 10 можем видеть характеристики результата работы программы.

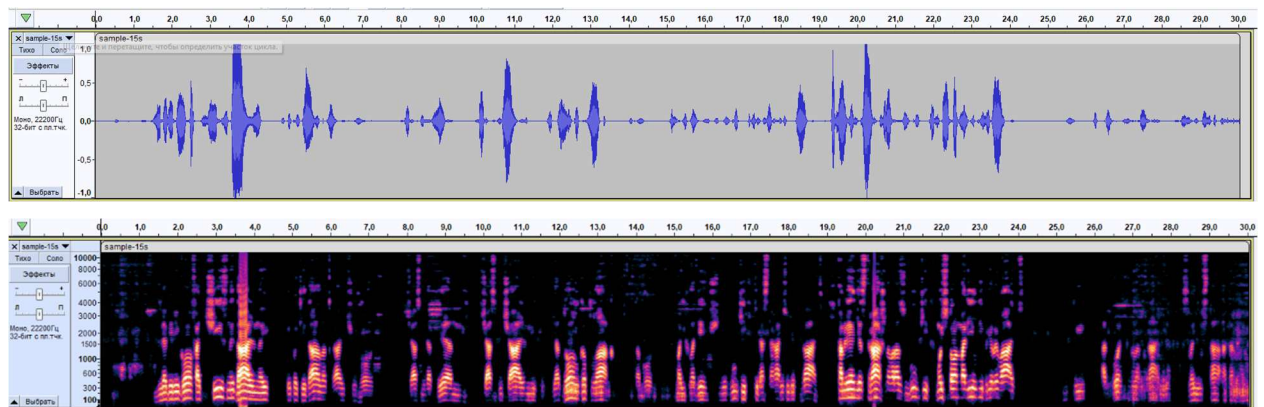


Рисунок 10 – Волноформа и спектрограмма после удаления нескольких шумов

В данном проекте, также, был создан Telegram-бот, который работает на основе разработанного алгоритма (рис. 11). Он позволяет быстро произвести шум из отправленного ему файла и вернуть пользователю тот же файл, но без шума, не искажая при этом основной сигнал с информацией.

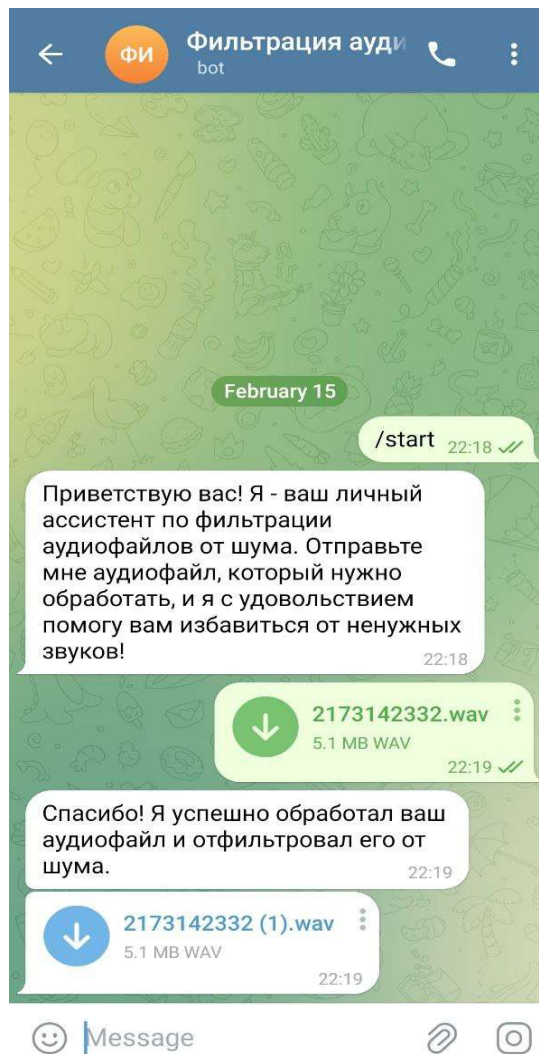


Рисунок 11 – Telegram-бот

Telegram-бот нужен, например, в ситуации, когда человеку было отправлено голосовое сообщение в шумной обстановке. Чтобы распознать основную информацию из сообщения, необходимо избавиться от мешающих этому шумов. Тогда пользователь может просто переслать сообщение в этот бот и сразу же, через 1–3 секунды, получить от него то же самое голосовое сообщение с четким голосом говорящего. Результаты тестирования программы: <https://drive.google.com/drive/folders/1ug82a0-7wSY6T6GXTAzslTTuObUkkKi2?usp=sharing>

ЗАКЛЮЧЕНИЕ

В данной работе создан алгоритм удаления шума из аудиозаписи с применением нескольких фильтров: известных фильтров из специальных библиотек Python и написанные автором. Работоспособность программы по удалению шума продемонстрирована на нескольких примерах. Данный алгоритм удобно применять в ситуациях, когда нет под рукой компьютера со специальным аудиоредактором, а достать информацию из аудиофайла нужно. Самая частая из подобных ситуаций – это распознавание речи из голосового сообщения. Поэтому представленный алгоритм лег в основу Telegram-бота, которому всегда можно быстро переслать сообщение и также быстро получить ответ в виде этого же голосового сообщения без шума.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Фильтрация шума сигнала [Электронный ресурс] URL: <https://habr.com/ru/articles/588270/> (дата обращения: 13.12.2023)
2. Каламбет Ю.А., Козьмин Ю.П., Самохин А.С. Фильтрация шумов. Сравнительный анализ методов // Аналитика: электрон. науч. журн. № 5/2017(36). С. 88 – 101. URL: https://www.j-analytics.ru/files/article_pdf/6/article_6306_795.pdf (дата обращения: 04.01.2024)
3. Руководство по Matplotlib [Электронный ресурс] URL: https://indico-hlit.jinr.ru/event/151/attachments/340/492/Project_school_Matplotlib_original.pdf (дата обращения: 10.12.2023)
4. Pydub API Documentation [Электронный ресурс] URL: <https://www.codetd.com/article/4409537> (дата обращения: 19.12.2023)

Приложение А. Отзыв



Федеральное государственное
бюджетное образовательное учреждение
высшего образования
«Национальный исследовательский
университет «МЭИ» (ФГБОУ «МЭИ»)
111250, г. Москва,
вн.тер.г. муниципальный округ Лефортово,
ул. Красноказарменная, д. 14, стр. 1
Тел.: (495) 362-75-60, факс: (495) 362-89-38
E-mail: universe@mpei.ac.ru
<http://www.mpei.ac.ru>

№ _____

« ____ » _____ 20 ____ г.

ОТЗЫВ

на проектную работу
(проектную / исследовательскую)

по теме «Алгоритм сглаживания посторонних шумов в аудиозаписи для улучшения качества
восприятия передаваемой в ней информации»
(наименование работы)

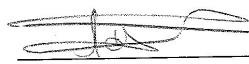
учащегося 10 класса ГБОУ Школа №1155
(класс) (наименование образовательной организации)
Рамазанов Абдул Хаирбекович
(фамилия, имя, отчество (при наличии), заполняется на каждого участника)

В проектной работе «Алгоритм сглаживания посторонних шумов в аудиозаписи для улучшения качества восприятия передаваемой в ней информации» автор реализовал алгоритм, который позволит внести изменения в свойства аудиофайла для более качественного восприятия слушателем информации, содержащейся в нем.

Работоспособность нового алгоритма показана в проекте на множестве примеров: приведены качественные иллюстрации с подробным объяснением новой идеи по минимизации шума в голосовом сообщении. Автор в работе продемонстрировал умение определять проблему в рассмотренной тематике, выбирать способы и средства ее решения, творчески подходить к реализации данного решения, а также подбирать подходящий для этого набор программных средств. В проекте также показано преимущество предложенного метода над некоторыми известными стандартными приемами устранения шума в аудиозаписях, что позволяет данный метод рекомендовать к практическому применению.

Цели работы автором достигнуты, актуальность представленных в работе достижений и их практическая значимость подтверждены. Разработанная в рамках проекта программа найдет широкое применение в повседневной жизни, например, при интеграции в любой мессенджер для коррекции личных голосовых сообщений пользователей.

доцент
(должность)


(подпись)



Лямасов А.К.
(И. О. Фамилия)