

ЦИФРОВАЯ ОБРАБОТКА ИЗОБРАЖЕНИЯ И ЗВУКА НА PYTHON

Касимова Ш.Т.

*Доцент Ташкентского университета информационных технологий
имени Мухаммада ал-Хоразми
shoista.kasimova@mail.ru*

Аннотация: Обработка изображений (обычно называемая «Цифровая обработка изображений») и Computer Vision - это области компьютерных наук с множеством алгоритмов, которые работают с изображениями или видео для получения какой-либо информации. В то время как обработка изображений имеет дело с преобразованиями изображения в изображение, т. е. вход и выход обработки являются изображениями, компьютерное зрение является междисциплинарной областью, касающейся того, как можно заставить компьютер понимать или получать информацию или информацию высокого уровня как цифровое изображение или видео.

Алгоритм обработки изображений принимает изображение или видео в качестве входных данных, обрабатывает их, и в результате обработки остается изображение или видео. Но алгоритм компьютерного зрения берет изображение или видео, обрабатывает его и создает из него явные и содержательные описания.

Все знают что звук (в узком смысле) — это волны сжатий и разрежений, распространяющиеся в воздухе. Запись звука, например в wav-файле, представляет собой последовательность значений амплитуды (физически она соответствует степени сжатия, или давлению).

Но мы не воспринимаем эти колебания частоты непосредственно, а слышим звуки разной частоты и тембра. Поэтому часто используется другой способ визуализации звука — спектрограмма, где на горизонтальной оси представлено время, на вертикальной — частота, а цвет точки обозначает амплитуду.

Ключевые слова: цифровое изображение, цифровой звук, аудиоанализ, спектрограмма, фильтр, частота цветности

ВВЕДЕНИЕ

Компьютерная обработка изображений (Computer Vision) - это область искусственного интеллекта, которая занимается анализом, интерпретацией и пониманием изображений и видео. Эта технология позволяет компьютерам

"видеть" и понимать мир вокруг себя, а также принимать решения на основе визуальной информации.

Примеры приложений компьютерного зрения включают в себя:

1. Распознавание объектов и лиц: определение и классификация объектов на изображениях, а также идентификация лиц.
2. Автоматическая проверка качества: анализ качества продукции на производстве с помощью компьютерного зрения.
3. Медицинская диагностика: автоматическое обнаружение и анализ медицинских изображений, например, рентгеновских снимков.
4. Автономные транспортные средства: использование компьютерного зрения для навигации автономных транспортных средств.
5. Распознавание жестов и движений: интерпретация жестов или движений человека на видео.

Для решения задач компьютерного зрения используются различные методы и алгоритмы, такие как нейронные сети, методы машинного обучения, обработка изображений и другие. Компьютерная обработка изображений играет важную роль во многих областях, включая медицину, промышленность, безопасность, автомобильную промышленность и другие(1,14).

Компьютерная обработка звука (Audio Signal Processing) - это область науки и технологии, которая занимается анализом, синтезом, улучшением и преобразованием звуковых сигналов с помощью компьютерных алгоритмов и методов. Эта технология позволяет компьютерам обрабатывать звуковую информацию, как музыку, речь или другие звуковые сигналы.

Примеры приложений компьютерной обработки звука включают в себя:

1. Аудиообработка и редактирование: изменение громкости, улучшение качества звука, удаление шумов, эффекты и фильтры.
2. Распознавание речи: преобразование речевых сигналов в текст с помощью алгоритмов распознавания речи.
3. Музыкальная обработка: создание и обработка музыкальных композиций, синтез звуков, автоматическое распознавание музыки.
4. Аудиоаналитика: анализ звуковых данных для выявления паттернов, трендов или аномалий.
5. Звуковое моделирование: моделирование и воспроизведение звуковых сцен для виртуальной реальности или игр.

Для решения задач компьютерной обработки звука применяются различные методы и алгоритмы, такие как цифровая фильтрация,

спектральный анализ, вейвлет-преобразование, машинное обучение и другие. Компьютерная обработка звука играет важную роль во многих областях, включая медиаиндустрию, медицину, телекоммуникации, аудиоинженерию и другие.

2. Обработка изображений

Что такое изображение? На этот вопрос можно ответить по-разному. Самое простое и широкое определение этого понятия: изображение – это то, что мы видим(5-7). Другое определение: изображение – это информация, пригодная для визуального восприятия. В зависимости от происхождения условно можно выделить следующие *типы изображений*:

1. Рисованное или печатное (художник, полиграфия, принтер).

2. Оптическое (распределение интенсивности электромагнитного поля создаваемое оптическим прибором в некоторой области пространства (области локализации) например, на сетчатке глаза, на экране при проецировании, в плоскости приемника объектива фотоаппарата).

3. Фотографическое (оптическое изображение, зарегистрированное на фотоматериале в результате химического процесса).

4. Электронное или цифровое (оптическое изображение, зарегистрированное с помощью электронного приемника, например, ПЗС-матрицы, сканера, микроденситометра). Электронным также называют изображение, отображаемое на экране монитора.

Как видим, деление условное. Изображение из одного типа сразу переходит в другой. Цепочка этих преобразований в большинстве случаев заканчивается изображением на сетчатке глаза и образом в мозге человека. Существуют ли понятие изображения в отдельности от глаза человека? Попытаемся дать изображению формализованное определение, которое позволит нам описать этот объект математически и манипулировать им для достижения определенных целей. Эти манипуляции назовем обработка изображений.

Обработка изображений может производиться в различных целях:

- Изменение (искажение) изображения с целью достижения каких-либо эффектов (художественное улучшение). Эти преобразования не будут рассматриваться в рамках данного курса;

- Image Processing – визуальное (заметное глазом) улучшение качества изображения (коррекция яркости и контраста, цветокоррекция и т.п.);

объективное улучшение качества изображения (устранение искажений типа дисторсия, смаз, расфокусировка и т.п.);

- Image Analysis – проведение измерений на изображении (анализ интерферограмм, гартманогамм, ФРТ и т.п.);

- Image Understanding – распознавание образов (распознавание символов, отпечатков пальцев, лиц, приборы наведения и т.п.).

Изображение может быть определено как двумерная функция $f(x, y)$, где X также Y -пространственные (плоские) координаты и амплитуда любой пары координат (x, y) называется интенсивностью или уровнем серого изображения в этой точке. Когда (x, y) и амплитуда f являются конечными дискретными значениями, мы называем изображение цифровым изображением.

Что такое видео?

Основное определение видео - это не что иное, как изображения, расположенные вдоль временной оси. Видео может характеризоваться соотношением сторон, частотой кадров, чересстрочной или прогрессивной разверткой, цветовой моделью, методом сжатия и т. д.

Учитывая, что фрагмент видео $M, X[n]$ является первым кадром, а $X[n + M - 1]$ является последним кадром, а $Y[n]$ является вектором, укладываемым все их вместе, чтобы сформировать фрагмент (часть) все видео.

2.1. Обработка изображений с помощью Python

Библиотека изображений Python, или **PIL (Python Imaging Library)** нужна для обработки графики в Python. Обратите внимание на то, что PIL и Pillow не могут быть установлены одновременно(10).

Установка Pillow в Python

Установить Pillow в Python можно через [pip](#) или `easy_install`. Обратите внимание, что при работа на Linux или Mac может потребоваться запускать команду через `sudo` т.е. от имени администратора.

Открытие изображения в Python с Pillow

Через Pillow можно легко **открыть изображение** и отобразить его на экране через внешнюю программу. Метод `show()` в основном используется для отладки.

На Windows изображение сохранится во временном файле BMP и откроется через простую программу вроде Paint.

Получение информации об изображении через Pillow

С помощью Pillow также можно получить подробную информацию об изображении.

Обрезка изображений через Pillow (crop)

Pillow также можно использовать для обрезки изображения. Это довольно легкий процесс, хотя он постигается методом проб и ошибок. Попробуем обрезать нашу картинку через `Image.crop()`:

Обратите внимание, здесь просто нужно открыть изображение, после чего вызвать метод `crop()`. Потребуется передать координаты x/y того, что нужно обрезать, например $(x1, y1, x2, y2)$.

В Pillow пиксель 0 является верхним левым уровнем. С увеличением значения x идет смещение вправо. С увеличением значения y идет смещение вниз.

Для быстрого получения правильных координат можно использовать Gimp или Photoshop. Он поможет определить координаты для следующего обрезки.

Программа обрезает картинку, после чего новая версия сохраняется на диск. Метод `crop()` принимает кортерж с четырьмя элементами, которые представляют координаты пикселей (левый верхний, правый верхний, левый нижний, правый нижний).

Поворачивание изображения — метод `rotate()` Pillow

`Image.rotate()` возвращает развернутую копию изображения.

Данная программа поворачивает изображения на 180 градусов и [сохраняет файл](#) результат в виде новой картинки на диск.

Отображение картинки в Tkinter Python

`ImageTk` является изображением, совместимым с Tkinter. Может использоваться везде, где Tkinter ожидает получения объекта изображения.

Изображению передаются параметру `image` виджета `Label`.

Размер окна совпадает с размером изображения.

Загрузка изображения из URL в Pillow

Картинка создается из ответного объекта `response`.

Создание рисунка в Pillow

У Pillow есть базовые возможности для создания 2D графики. Модуль `ImageDraw` предоставляет простую 2D графику для объектов `Image`. Мы можем создавать новые изображения, аннотации к ним, ретушировать существующие фото, а также сразу генерировать графику для веб.

ImageFont — Пишем текст на изображение используя Pillow

Текст наносится через метод `text()`. По умолчанию цвет шрифта белый.

2.2. Использование фильтров для обработки изображения

Pillow позволяет использовать множество различных фильтров для обработки изображения(11-12). Они являются частью модуля **ImageFilter**.

Программа берет определенное изображение, создает на его основе размытую картинку, используя `ImageFilter.BLUR`, и сохраняет полученный результат на диск с помощью метода `save()`.

Размытая картинка

Однако в подавляющем большинстве случаев размывать изображение нет нужды, наоборот — требуется увеличить резкость. Pillow **меняет резкость картинки**.

Кроме того, для увеличения резкости фотографий в Python можно использовать модуль **ImageEnhance**.

Можно использовать и другие фильтры — `DETAIL`, `EDGE_ENHANCE`, `EMBOSS`, `SMOOTH` и так далее. В коде для одного изображения также можно одновременно использовать несколько фильтров.

Конвертирование из JPG в PNG

В Python Pillow метод `save()` позволяет конвертировать изображение в другой формат.

Программа считывает изображение JPG и конвертирует его в PNG формат. Это делается в следующей строчке:

```
1 tatras.save('tatras.jpg', 'png')
```

Второй параметр метода `save()` нужен для уточнения итогового формата изображения.

Создаем черно-белое изображение GrayScale в Pillow

С помощью метода `Image.convert()` можно сделать оригинальное изображение черно-белым.

Изменение размера изображения в Pillow- `resize()`

Можно изменить длину и ширину изображения при помощи метода `resize()`. В данном примере будут показаны три примера изменения размера:

- Изменение размера изображения имея ширину и высоту;
- Изменение ширины с учетом пропорций для высоты;
- Изменение высоты пропорционально ширине.

2.3. Использование возможностей OPENCV

OpenCV (Open Source Computer Vision) - широко используемая библиотека для задач компьютерного зрения и машинного обучения. Оригинальная реализация OpenCV находится на C ++. Python медленнее по сравнению с C / C ++, но, с другой стороны, его можно легко расширить с помощью этих языков. Эта особенность позволила нам создавать вычислительно-интенсивные коды в C ++, но обертки в Python, чтобы их можно было использовать в качестве модулей Python. [OpenCV-Python](#) это реализация OpenCV на Python. Библиотека [кросс-платформенная](#) и может быть использована под открытым исходным кодом.

3. Обработка звука

Развитие компьютерных технологий существенно изменило многие сферы деятельности человека, создав новые возможности и повысив эффективность традиционных видов работ. Одна из областей, в которых компьютер произвел настоящую революцию – это аудиотехника и, в частности, музыкальная деятельность в ее многообразных приложениях(14).

Даже первые персональные компьютеры отличались от калькуляторов и больших ЭВМ тем, что они могли издавать звуки с помощью маленького динамика, установленного в их корпусе. И хотя акустические возможности РС были более чем скромными, уже на заре компьютерной эры появились музыкальные редакторы, с помощью которых можно было создать "звуковой файл" для подключения к той или иной программе, написанной пользователем.

С появлением звуковых карт перед пользователями открылись новые возможности. И дело даже не в том, что на порядок улучшилось качество звука. Появилась новая (*звуковая*) *подсистема* — комплекс программно-аппаратных средств, предназначенный для:

- **записи звуковых сигналов**, поступающих от внешних источников, например, микрофона или магнитофона. В процессе записи входные аналоговые звуковые сигналы преобразуются в цифровые и далее могут быть сохранены на винчестере РС;

- **воспроизведения записанных ранее звуковых данных** с помощью внешней акустической системы или головных телефонов (наушников). При воспроизведении звуковой сигнал считывается с носителя информации, преобразуется из цифрового в аналоговый и направляется к акустической системе;

- **микширования** (смешивания) при записи или воспроизведении сигналов от нескольких источников;

- **одновременной записи и воспроизведения** звуковых сигналов;

- **обработки** звуковых сигналов: редактирования, объединения или разделения фрагментов сигнала, фильтрации, изменения его уровня и т. п.;

Звуки - это изменяющиеся во времени сигналы в реальном мире, и, действительно, все их значение связано с такой изменчивостью времени. Поэтому интересно разрабатывать методы звукового анализа, позволяющие уловить хотя бы некоторые отличительные особенности изменяющихся во времени звуков, чтобы облегчить задачи понимания, сравнения, модификации и ресинтеза.

3.1. Аудиоанализ

Аудиоанализ — область, включающая автоматическое распознавание речи (ASR), цифровую обработку сигналов, а также классификацию, тегирование и генерацию музыки — представляет собой развивающийся поддомен приложений глубокого обучения. Некоторые из самых популярных и распространенных систем машинного обучения, такие как виртуальные помощники Alexa, Siri и Google Home, — это продукты, созданные на основе моделей, извлекающих информацию из аудиосигналов(15-16).

Аудио фрагменты представлены в формате *.wav*. Звуковые волны оцифровываются путем выборки из дискретных интервалов, известных как частота дискретизации (как правило, 44,1 кГц для аудио с CD-качеством, то есть 44 100 семплов в секунду).

Каждый семпл представляет собой амплитуду волны в определенном временном интервале, где глубина в битах (или динамический диапазон сигнала) определяет, насколько детализированным будет семпл (обычно 16 бит, т.е. семпл может варьироваться от 65 536 значений амплитуды).

В обработке сигналов семплинг — это преобразование непрерывного сигнала в серию дискретных значений. Частота дискретизации — это количество семплов за определенный фиксированный промежуток времени.

Высокая частота дискретизации приводит к меньшей потере информации, но к большим вычислительным затратам.

Приложения по обработке звука

К ним можно отнести:

- Индексирование музыкальных коллекций согласно их аудиопризнакам.
- Рекомендация музыки для радиоканалов.
- Поиск сходства для аудиофайлов (Shazam).
- Обработка и синтез речи — генерирование искусственного голоса для диалоговых агентов.

3.2. Обработка аудиоданных с помощью Python

Звук представлен в форме аудиосигнала с такими параметрами, как частота, полоса пропускания, децибел и т.д. Типичный аудиосигнал можно выразить в качестве функции амплитуды и времени.

Некоторые устройства могут улавливать эти звуки и представлять их в машиночитаемом формате. Примеры этих форматов:

- wav (Waveform Audio File)
- mp3 (MPEG-1 Audio Layer 3)
- WMA (Windows Media Audio)

Процесс обработки звука включает извлечение акустических характеристик, относящихся к поставленной задаче, за которыми следуют схемы принятия решений, которые включают обнаружение, классификацию и объединение знаний. Библиотеки Python помогают решить эту задачу.

3.3. Аудио библиотеки Python

Мы будем использовать две библиотеки для сбора и воспроизведения аудио:

1. Librosa

Librosa может работать с любыми звуковыми сигналами, но ориентирована в основном именно на музыку. Она позволяет создать полноценную систему извлечения музыкальной информации (MIR). Модуль прекрасно документирован, кроме того, существует множество руководств по использованию.

Можно также установить модуль ffmpeg со множеством готовых решений для конвертации аудиосигналов.

2. IPython.display.Audio

`IPython.display.Audio` позволяет воспроизводить аудио непосредственно в Jupyter Notebook.

Частота дискретизации – это количество семплов (колебаний) звука, передаваемого в секунду, измеренное в Гц или кГц.

Для воспроизведения аудио используется `IPython.display.Audio`.

Используя `librosa.display.waveplot`, можно визуализировать массив аудиоданных.

3.4. Обработка спектограмм на python

Спектрограмма – это визуальное представление спектра частот звуковых или других сигналов, изменяющихся со временем. Иногда их также называют сонограммами. На двумерных графиках по первой оси задается частота, по второй – время.

Для создания спектрограммы на Python используем `librosa.display.specshow`.

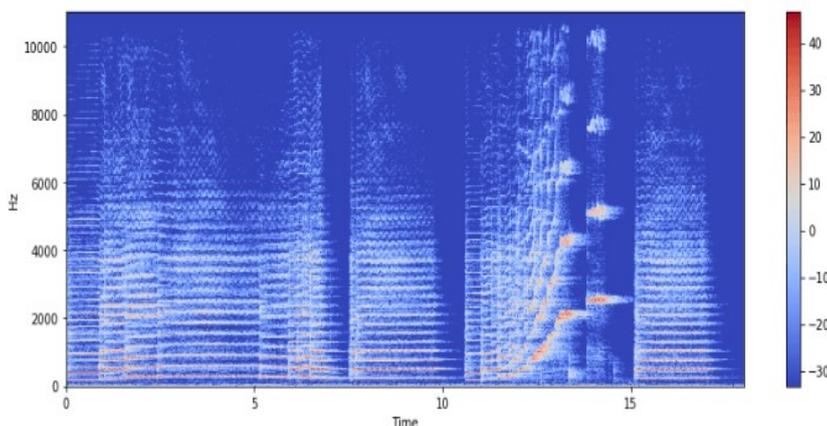


Рис.1.1. Спектограмма аудиофайла

Вертикальная ось – это частоты (от 0 до 10 кГц), а горизонтальная – время клипа. Поскольку все значимые изменения происходят в нижней части спектра, частотную ось можно преобразовать в логарифмическую.

Извлечение сущностей

Каждый звуковой сигнал имеет множество характеристик, из которых следует отобрать нужные. Процесс извлечения информации для анализа называется извлечением объектов или извлечением сущностей (feature extraction)(17-19).

Частота перехода через нуль

Частота пересечения нуля (zero crossing rate) – это частота изменения знака сигнала, т. е. частота, с которой сигнал меняется с положительного на отрицательный и обратно. Эта функция широко используется как для распознавания речи, так и для извлечения музыкальной информации. Для металла и рока этот параметр обычно выше, чем для других жанров, из-за большого количества ударных.

Рассчитаем частоту перехода через ноль для нашего примера на Python:

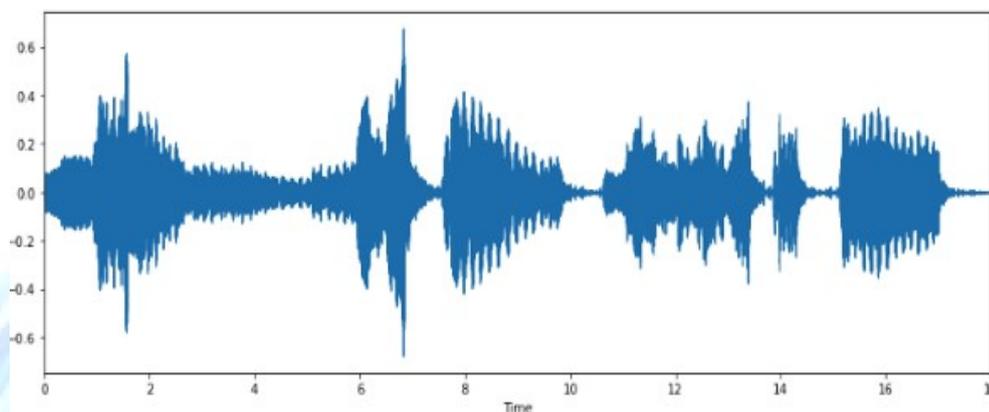


Рис.1.2. Определение частоты перехода через ноль

Спектральный центроид

Спектральный центроид указывает, где расположен "центр масс" звука, и рассчитывается как средневзвешенное значение всех частот.

В блюзовых композициях частоты равномерно распределены, и центроид лежит где-то в середине спектра. В металле наблюдается выраженное смещение частот к концу композиции, поэтому и спектроид лежит ближе к концу спектра.

Вычислим спектральный центроид для каждого фрейма с помощью `librosa.feature.spectral_centroid`:

Спектральный спад частоты

Это мера формы сигнала, представляющая собой частоту, ниже которой лежит определенный процент от общей спектральной энергии, к примеру, 85%.

`librosa.feature.spectral_rolloff` вычисляет спад частоты для каждого фрейма.

Мел-частотные кепстральные коэффициенты

Мел-частотные кепстральные коэффициенты (MFCC) сигнала – небольшой набор характеристик (обычно около 10-20) которые сжато

описывают общую форму спектральной огибающей. Этот параметр моделирует характеристики человеческого голоса.

Мы также можем выполнить масштабирование таким образом, чтобы каждое измерение коэффициента имело нулевое среднее и единичную дисперсию.

Частота цветности

Цветность (chroma features) – это интересное и мощное представление для музыкального звука, при котором весь спектр проецируется на 12 контейнеров, представляющих 12 различных полутонов музыкальной октавы.

Для вычислений используем **librosa.feature.chroma_stft**:

После знакомства со структурой акустического сигнала и особенностями процесса извлечения музыкальной информации, рассмотрим Python библиотеки для работы со звуком.

Попробуем смоделировать жанровый классификатор музыки. Он пригодится, если понадобится разобрать кучу неизвестных mp3-файлов.

Жанровый классификатор музыки

Перед обучением модели классификации нужно преобразовать необработанные данные из звуковых выборок в более осмысленное представление. Преобразуем клипы в формат **wav**, чтобы Python мог с ними работать, с помощью модуля SoX.

Теперь извлечем из аудиофайлов всю необходимую информацию:

- мел-частотные кепстральные коэффициенты,
- спектральный центроид,
- частоту перехода через нуль,
- частоты цветности,
- спектральный спад частоты.

Все эти функции сохраним в **.csv**-файле.

Можно использовать существующие алгоритмы классификации для распределения песен по жанрам или использовать непосредственно спектрограммы, либо извлечь сущности и применять модели классификации на них.

Жанровая классификация – лишь одна из многих прикладных отраслей извлечения музыкальной информации.

РЕЗУЛЬТАТЫ

Для начала зададимся вопросом: кому вообще нужно распознавать рекламу на радио? Это полезно рекламодателям, которые могут отслеживать

реальные выходы своих рекламных роликов, ловить случаи обрезки или прерывания; радиостанции могут мониторить выход сетевой рекламы в регионах, и т.п. Та же задача распознавания возникает, если мы хотим отследить проигрывание музыкального произведения, или по небольшому фрагменту узнать песню.

Более строго задача формулируется так: у нас есть некоторый набор эталонных аудио-фрагментов (песен или рекламных роликов), и есть аудио-запись эфира, в котором предположительно звучат какие-то из этих фрагментов. Задача — найти все прозвучавшие фрагменты, определить моменты начала и длительность проигрывания. Если мы анализируем записи эфира, то нужно чтобы система в целом работала быстрее реального времени.

В первую очередь нужно преобразовать аудиофайлы в изображения формата PNG (спектрограммы). Затем из них нужно извлечь значимые характеристики: MFCC, спектральный центроид, скорость пересечения нуля, частоты цветности, спад спектра.

После извлечения признаки можно добавить в файл CSV, чтобы ANN можно было использовать для классификации.

Конвертируем файлы аудиоданных в PNG или извлекаем спектрограмму для каждого аудио.

Преобразование аудиофайлов в соответствующие спектрограммы упрощает извлечение функций.

Создание заголовка для файла CSV.

Извлекаем признаки из спектрограммы: MFCC, спектральный центроид, частоту пересечения нуля, частоты цветности и спад спектра.

Выполняем предварительную обработку данных, которая включает загрузку данных CSV, создание меток, масштабирование признаков и разбивку данных на наборы для обучения и тестирования.

ЗАКЛЮЧЕНИЕ

Аудиообработка и редактирование — это одна из основных областей компьютерной обработки звука, которая включает в себя широкий спектр методов и техник для улучшения и изменения звуковых сигналов. Вот некоторые основные задачи и методы аудиообработки и редактирования:

1. Изменение громкости: это одна из самых простых операций аудиообработки, которая позволяет увеличить или уменьшить громкость звукового сигнала.

2. Улучшение качества звука: включает в себя различные методы для устранения и уменьшения шумов, искажений и других артефактов, которые могут повлиять на качество звука.

3. Удаление шумов: это процесс фильтрации и удаления нежелательных шумов из аудиозаписи, таких как фоновый шум, электрические помехи, щелчки и т. д.

4. Применение эффектов и фильтров: это включает в себя добавление различных звуковых эффектов, таких как реверберация, эквалазация, задержка, хорус, флэнжер и другие, чтобы изменить звучание аудиозаписи.

Для выполнения этих операций используются различные алгоритмы обработки сигналов, цифровые фильтры, спектральный анализ, многополосная обработка и другие методы. Программное обеспечение для аудиоредактирования, такое как Adobe Audition, Audacity, Pro Tools, Ableton Live и другие, предоставляет широкий набор инструментов для выполнения различных задач по обработке и редактированию аудиозаписей.

Задачу поиска фрагмента в эфире можно разбить на две части: сначала найти среди большого числа эталонных фрагментов кандидаты, а затем проверить, действительно ли кандидат звучит в данном фрагменте эфира, и если да, то в какой момент начинается и заканчивается звучание. Обе операции используют для своей работы «отпечаток» фрагмента звучания. Он должен быть устойчивым к шумам и быть достаточно компактным. Этот отпечаток строится так: мы разбиваем спектрограмму на короткие отрезки по времени, и в каждом таком отрезке ищем частоту с максимальной амплитудой (на самом деле лучше искать несколько максимумов в различных диапазонах, но для простоты возьмем один максимум в наиболее содержательном диапазоне). Набор таких частот (или индексов частот) и представляет собой отпечаток. Очень грубо можно сказать, что это «ноты», звучащие в каждый момент времени.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Trevor Cox, *The Sound Book: The Science of the Sonic Wonders of the World*, ООО «Издательская Группа «Азбука-Аттикус», 2018.
2. Rocchessho Davide. *Sound processing*, 2012, 244
3. Burger W., Burge M. J. *Digital Image Processing: An Algorithmic Introduction using Java*. –New York: Springer, 2008. –564 p.
4. Гонзалес Р., Вуде Р. *Цифровая обработка изображений*. Издание 3-е, испр. и доп. – М.: Техносфера, 2012. – 1104 с.

5. Гашников М. В., Глумов Н. И., Ильясова Н. Ю. и др. Методы компьютерной обработки изображений. Изд. 2. Под ред. В.А. Соифера. - М.: Физматлит, 2003. -784 с.
6. Абламейко С.В., Лагуновский Д.М. Обработка изображений: технология, методы, применение. - Мн.: Амалфея, 2000.
7. Яне Б. Цифровая обработка изображений. - М.: Техносфера, 2007. – 584 с.
8. Грузман И.С., Киричук В.С., Косых В.П. и др. Цифровая обработка изображений в информационных системах. – Новосибирск: НГТУ, 2002. – 352 с.
9. Визильтер Ю.В. Обработка и анализ изображений в задачах машинного зрения. – М.: Физмат книга, 2010. – 672 с.
10. Pratt W.K. Digital Image Processing. – New York: Wiley, 2001. – 792 p
11. Jahne B. Digital Image Processing. - New York: Springer, 2005. – 585 p.
12. Chi Zh., Yan H., Pham T. Fuzzy algorithms : with applications to image processing and pattern recognition. New Jersey: World Scientific, 1996. – 230 p.
13. Gonzalez R., Woods R. Digital Image Processing. 3rd Edition - New Jersey: Prentice-Hall, 2008. – 954 p
14. Загуменнов А. Компьютерная обработка звука. – М.: Радио и связь, 2008. – 496 с.
15. Olli Niemitalo. Digital sound processing tutorial for the braindead!-2012,-562 p.
16. Кинтцель Т. Руководство программиста по работе со звуком. Пер. с англ. – М.: ДМК Пресс, 2000. – 432 с.
17. Лоянич А.А. Запись и обработка звука на компьютере–М.: Горячая линия , 2014.- 462 с.
18. Секунов Н.Ю. Обработка звука на РС. – СПб.: БХВ-Петербург, 2001. – 1238 с.
19. Furui S. Digital Speech Processing, Synthesis and Recognition. New York: Marcel Dekker, 2001. - 452 p.