



УНИВЕРСИТЕТ
ИСКУССТВЕННОГО
ИНТЕЛЛЕКТА

Обработка аудио

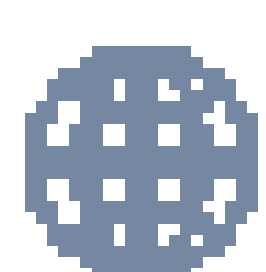
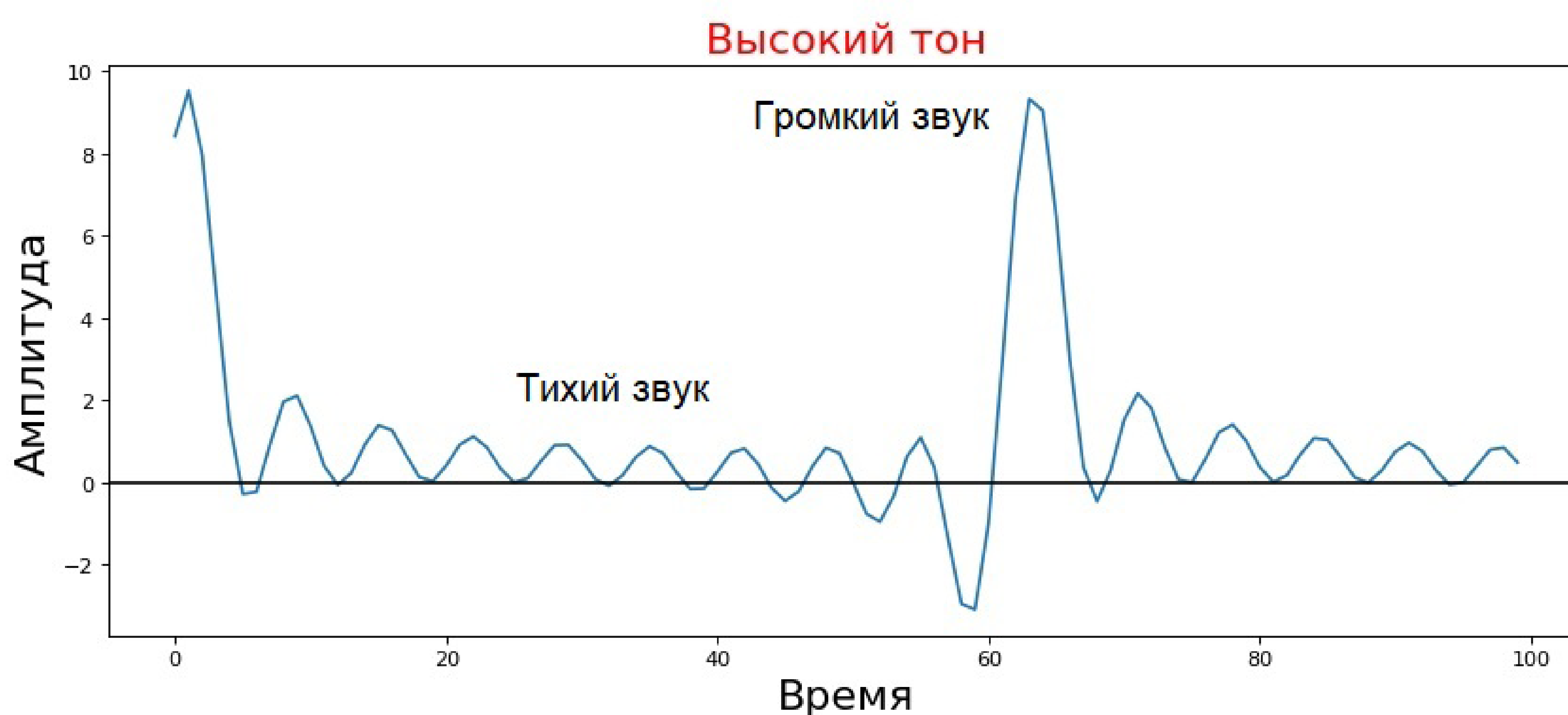




Обработка аудио

Звук – физическое явление, представляющее собой распространение в виде упругих волн механических колебаний в твёрдой, жидкой или газообразной среде.

Как и любая волна, звук характеризуется амплитудой и частотой. Амплитуда характеризует громкость звука. Частота определяет тон и высоту.



Обработка аудио

Чем **больше амплитуда** звуковой волны, тем **громче звук**.

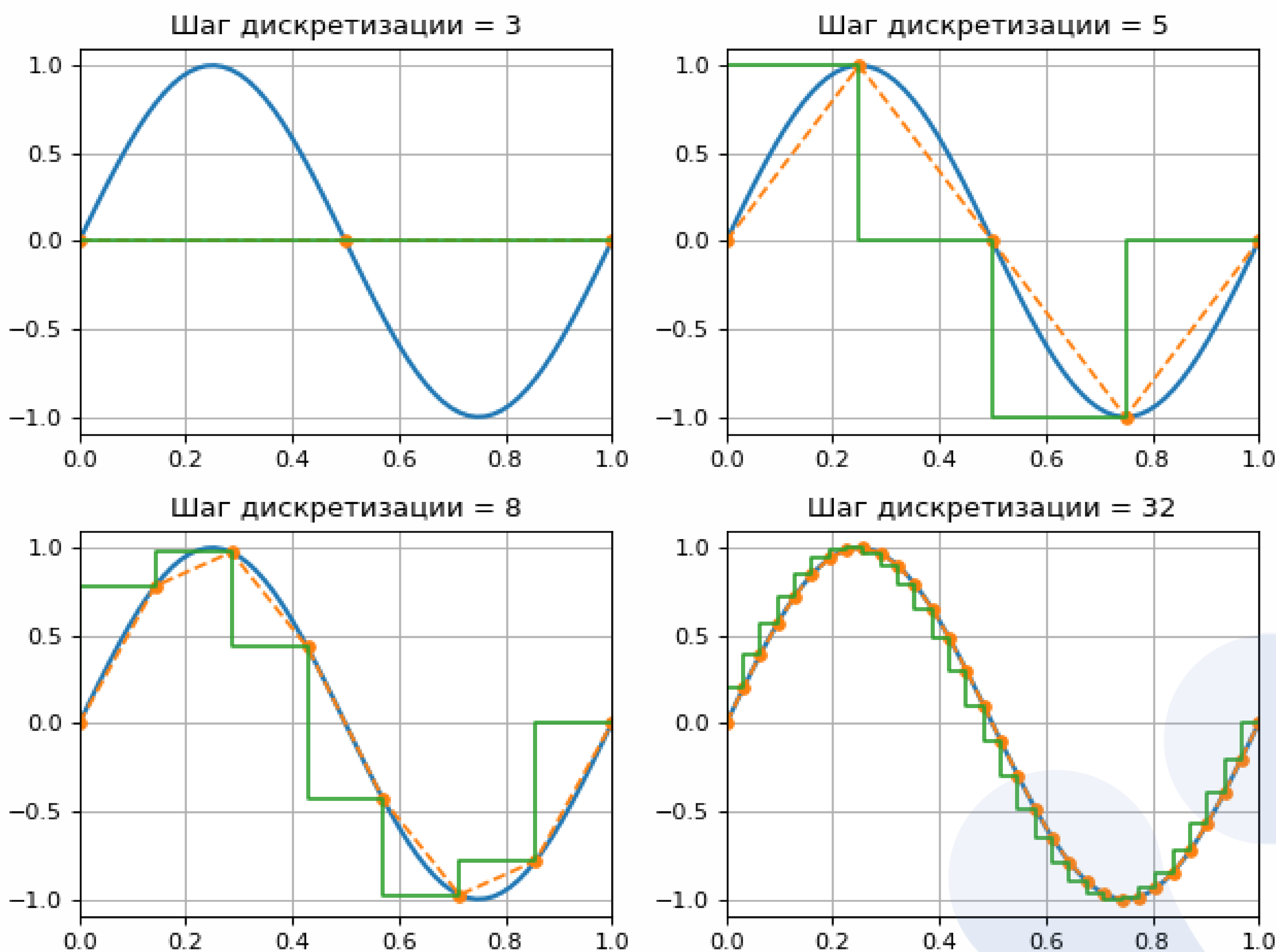
Чем **больше частота** колебаний, тем **выше тон** звука.

Единицы измерения:

- дБ (децибел) – ГРОМКОСТЬ
- Гц (герц) – ЧАСТОТА

Дискретизация

Дискретизация (оцифровка) – процесс перевода звуковых сигналов от непрерывной формы представления к дискретной, цифровой форме, то есть перевод вашей записи, записанной вживую, на цифровой носитель.



Важной характеристикой при кодировании звука является частота дискретизации – количество измерений уровней сигнала за 1 секунду: одно измерение в секунду соответствует частоте 1 Гц; 1000 измерений в секунду соответствует частоте 1 кГц. Частота дискретизации звука – это количество измерений громкости звука за одну секунду.

Обработка аудио

Теорема Котельникова

При кодировании звука важно знать теорему Котельникова.

Теорема Котельникова
(Уиттакера — Найквиста — Шеннона)

Если непрерывный сигнал $x_a(t)$ имеет спектр, ограниченный частотой F_{\max} , то он может быть однозначно и без потерь восстановлен по своим **дискретным** отсчётам, взятым с частотой:

$F_{\text{rADC}} \geq 2 \cdot F_{\max}$,

где F_{\max} — максимальная частота гармонической составляющей в спектре исходного непрерывного сигнала $x_a(t)$.

$$x_a(t) = \sum_{k=-\infty}^{\infty} x(k) \cdot \frac{\sin[(\pi / T) \cdot (t - kT)]}{[(\pi / T) \cdot (t - kT)]}$$

Простыми словами, нужно брать частоту дискретизации таким образом, чтобы она была не меньше двух максимальных ($sr \geq 2 \cdot F_{\max}$) частот взятого аудиосигнала.

Библиотеки для работы с аудио

- **PyAudio** – кроссплатформенная библиотека для форматирования и синтеза аудио информации.
- **Speech Recognition** – библиотека, являющаяся оберткой над многими популярными сервисами/библиотеками распознавания речи.
- **Google Text To Speech (sTTS)** – библиотека для преобразования строки в mp3 файл с речью.
- **LibROSA** – библиотека для анализа аудиоинформации.

Обработка аудио

Мы в основном будем использовать LibROSA. Рассмотрим некоторые инструменты этой библиотеки.

LibROSA – это модуль Python для анализа звуковых сигналов, предназначенный для работы с музыкой. Он включает все необходимое для создания системы MIR (поиск музыкальной информации) и подробно задокументирован вместе со множеством примеров и руководств.

Для работы нейронной сети с аудио, ей нужны какие-то признаки аудиосигнала, чтобы на основе этих признаков построить обучение. Далее мы рассмотрим наиболее используемые и полезные признаки.

1. Спектрограмма (сонограмма).

Чтобы загрузить аудиосигнал, используем библиотеку librosa, функцию **load()**:

```
x, sr = librosa.load('audio_path')
```

x - длина звуковой дорожки,

sr - частота дискретизации.

Далее, чтобы определить спектр сигнала, используем функцию **librosa.stft()**:

```
# Вычисляем спектр сигнала
```

```
X = librosa.stft(x)
```

И выводим при помощи библиотеки matplotlib:

```
# Меняем шкалу на децибелы(для удобного отображения)
```

```
Xdb = librosa.amplitude_to_db(abs(X))
```

```
# Выводим спектрограмму на экран
```

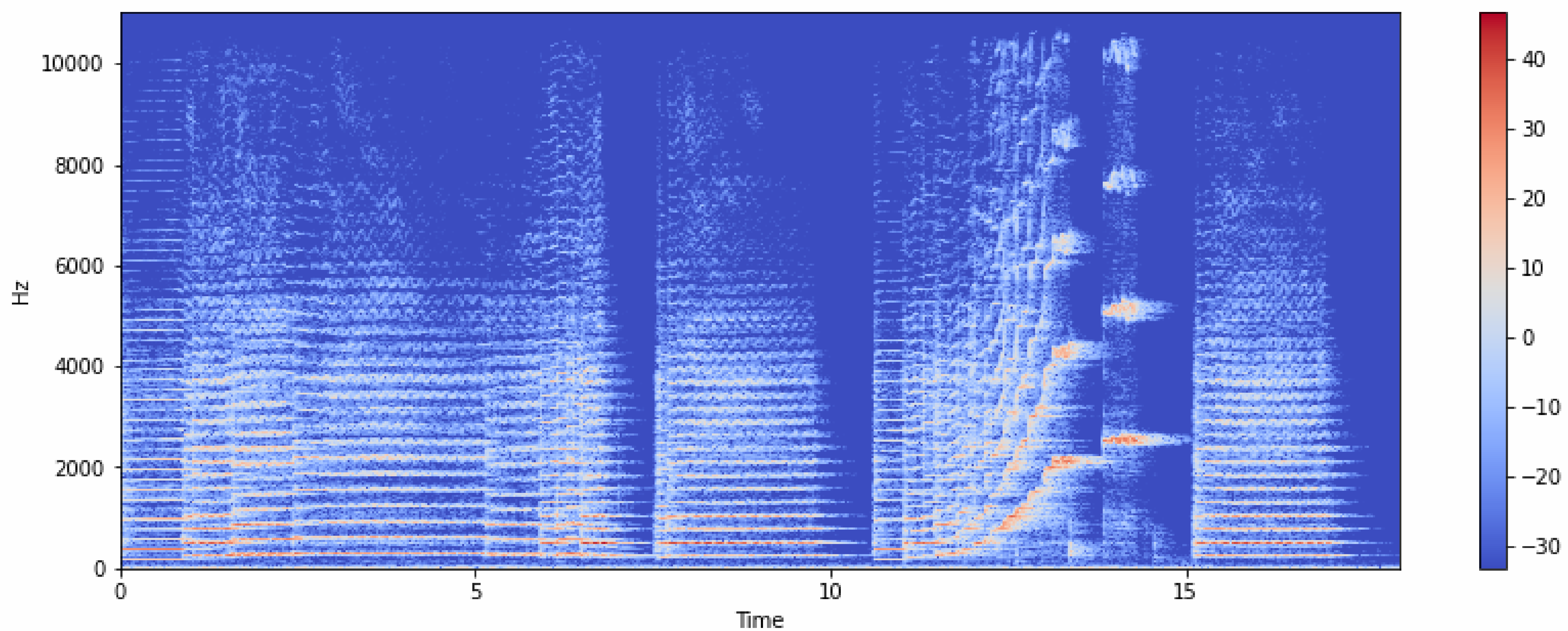
```
plt.figure(figsize=(14, 5)) #задаем размер
```

```
librosa.display.specshow(Xdb, sr=sr, x_axis='Time', y_ axis='Hz') #Отобразим спектрограммы
```

```
plt.colorbar() #выведем цветовую шкалу
```

```
plt.show() #выводим график
```

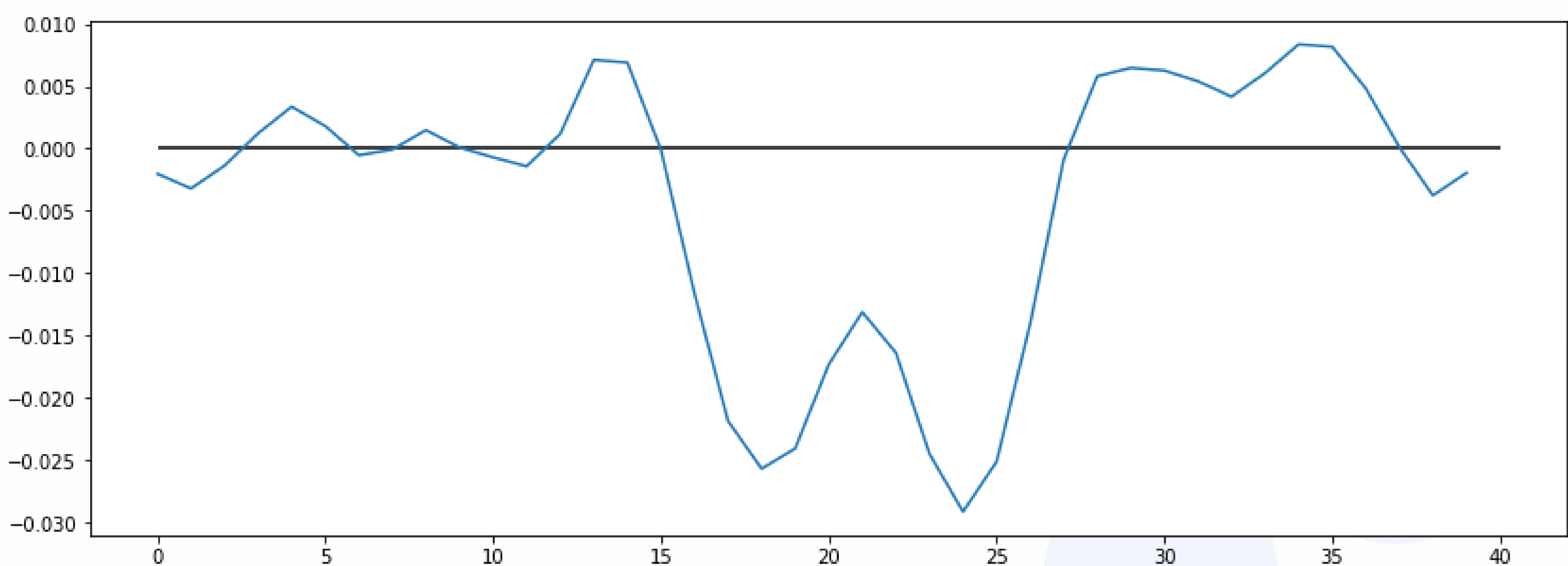

Обработка аудио



Спектрограмма показывает зависимость спектральной мощности сигнала от времени.

2. Частота пересечения нуля (zero crossing rate)

Частота пересечения нуля – это частота изменения знака сигнала, т. е. частота, с которой сигнал меняется с положительного на отрицательный и обратно.



Для подсчетов пересечения нуля используем функцию **zero_crossings()**:

```
# Рассчитываем пересечения нуля
zero_crossings = librosa.zero_crossings(x[0:40],
pad=False)
# Отображаем результаты
print(sum(zero_crossings)) # Суммарное количество
пересечений
```

Обработка аудио

```
print(zero_crossings)          # Наличие пересечения в каждой
точке
-----
8
[False False False  True False False  True False  True
 False  True False True False False  True False False False
 False False False False False False False False False
 True False False False False False False False False
 True False]
```

3. Спектральный центроид

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

Для вычисления спектрального центроида используем функцию **librosa.feature.spectral_centroid()**:

```
import sklearn # Для нормирования

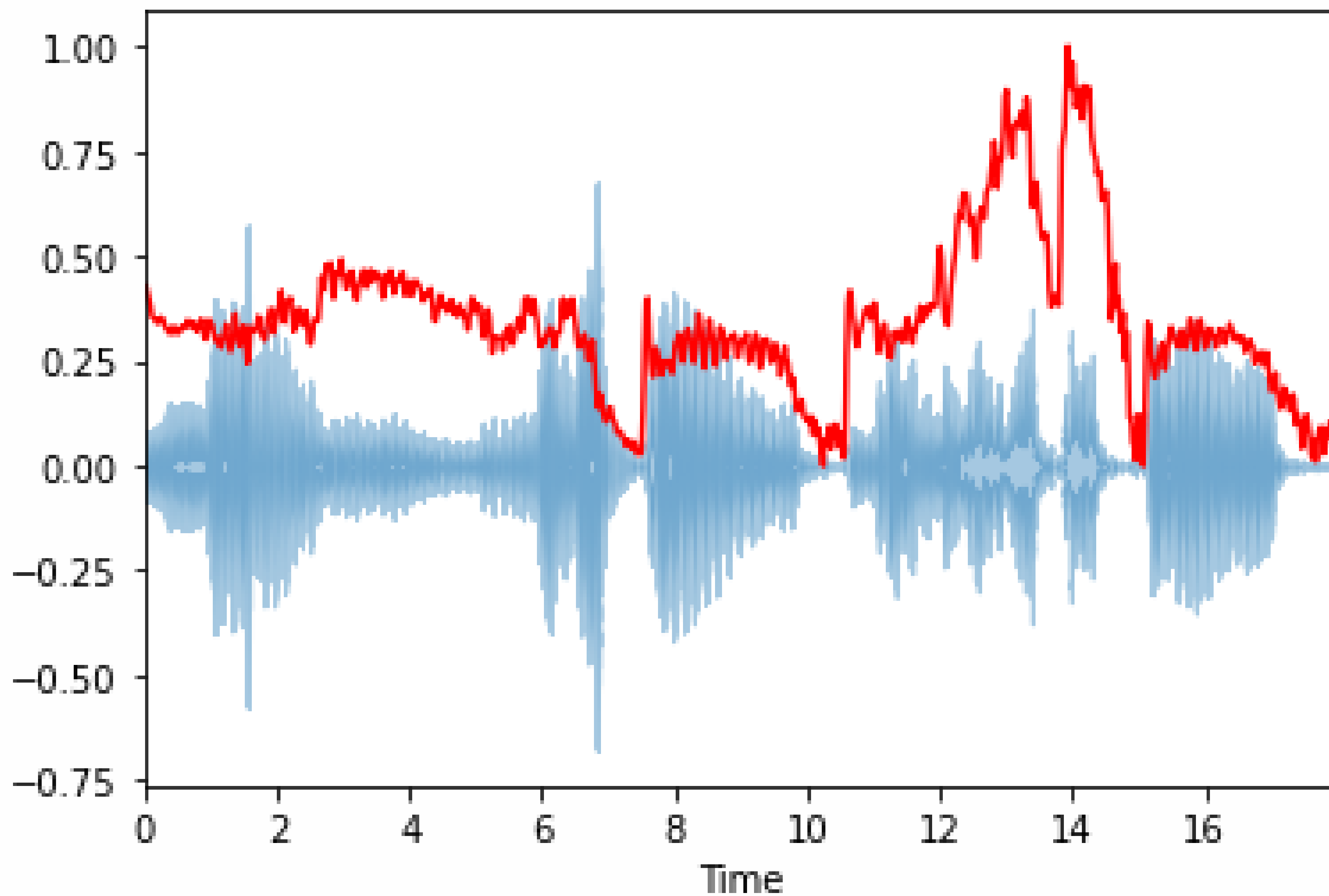
# Вычисляем спектральный центроид
spectral_centroids = librosa.feature.spectral_centroid(x,
sr=sr) [0]

# Вычисляем время для визуализации
frames = range(len(spectral_centroids))
t = librosa.frames_to_time(frames)

# Нормализуем спектральный центроид к отрезку 0-1
def normalize(x, axis=0):
    return sklearn.preprocessing.minmax_scale(x, axis=axis)

# Строим график сигнала и спектрального центроида
librosa.display.waveplot(x, sr=sr, alpha=0.4) # Построим
амплитуду сигнала
plt.plot(t, normalize(spectral_centroids), color='r') #
Построим график нормализованного спектрального центроида
plt.show() # Выводим графики
```


Обработка аудио



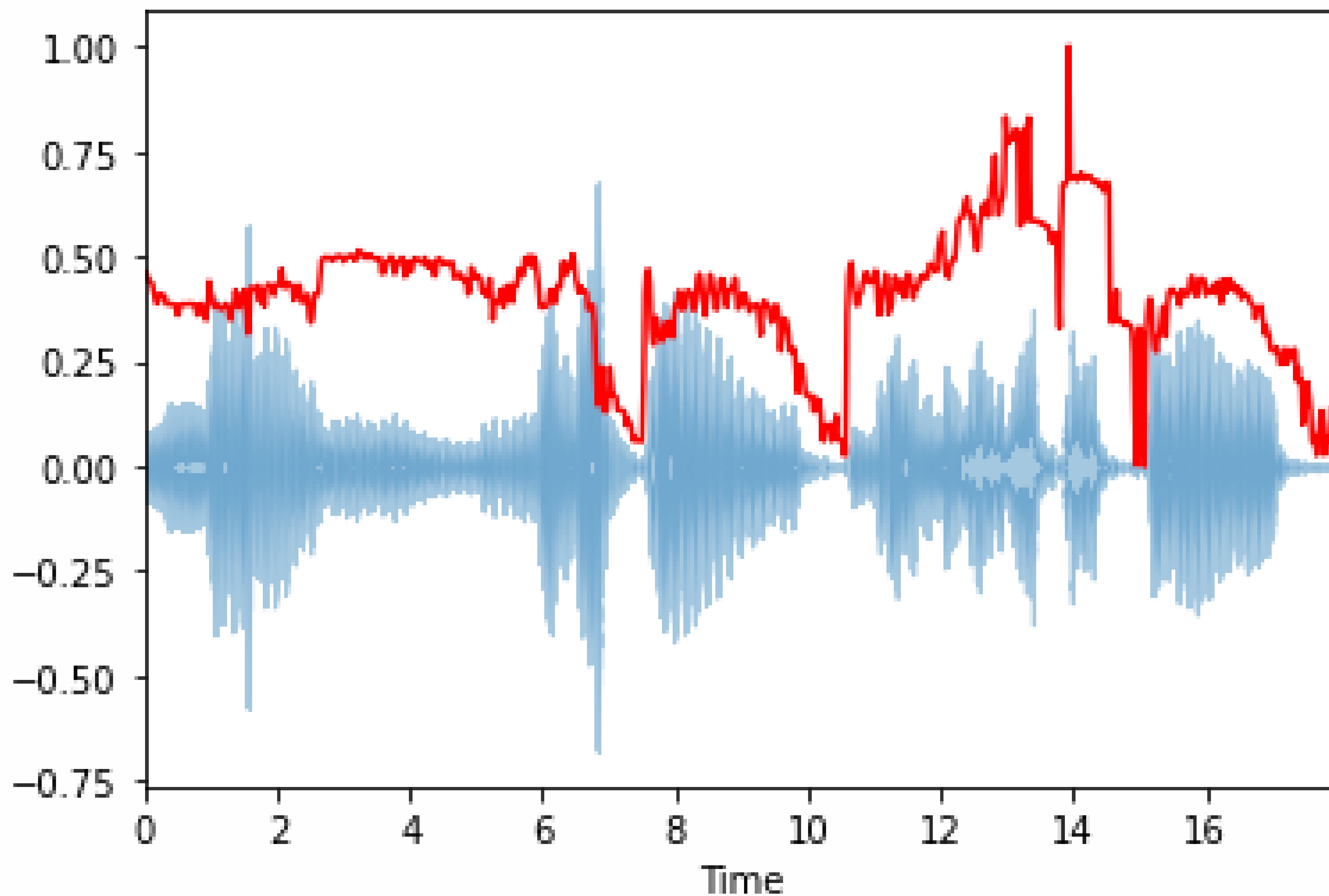
4. Спектральный спад частоты

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

Для вычисления спектрального спада частоты используем функцию **librosa.features.spectral_rolloff()**:

```
# Вычисляем и отображаем спектральный спад частоты
spectral_rolloff = librosa.feature.spectral_rolloff(x,
sr=sr, roll_percent=0.85)[0]
librosa.display.waveplot(x, sr=sr, alpha=0.4) # Построим
амплитуду сигнала
plt.plot(t, normalize(spectral_rolloff), color='r') #
Построим график нормализованного спектрального спада
частоты
plt.show() # Выводим график
```


Обработка аудио



Параметром `roll_percent=0.85` (по умолчанию) мы и указываем этот определенный процент.

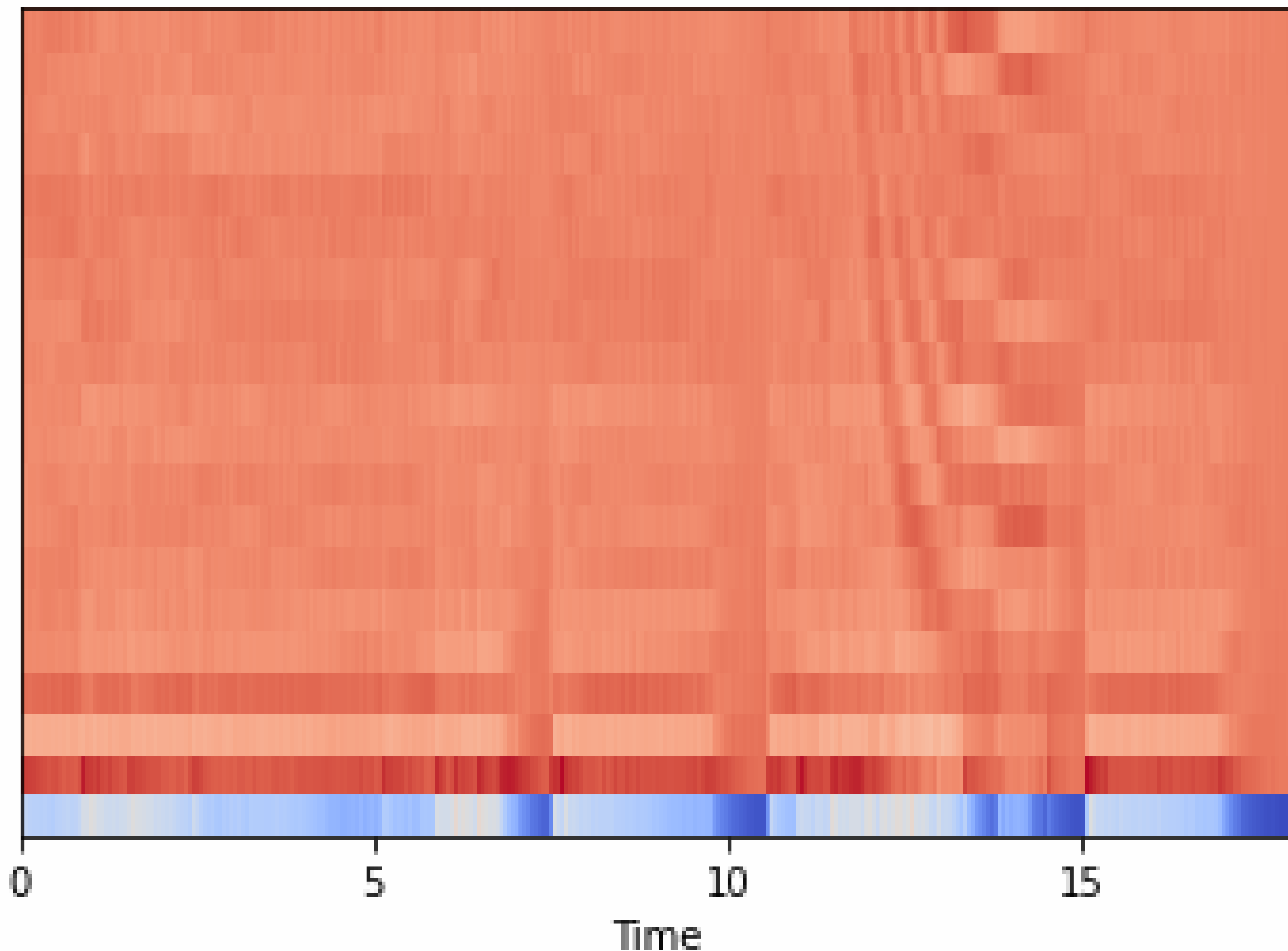
5. Мел-частотные спектральные коэффициенты (mfcc)

Mfcc представляет собой картинку из 20 строк (20 фильтров), где каждым фильтром (с соответствующей частотой) будет пропущен аудиосигнал. Каждая строка – это обработанный звуковой сигнал каким-то частотным фильтром.

Mfcc вычисляется при помощи функции **librosa.feature.mfcc()**:

```
# Вычисляем и отображаем Мел-частотные кепстральные
коэффициенты
mfccs = librosa.feature.mfcc(x, sr=sr)
librosa.display.specshow(mfccs, sr=sr, x_axis='time') #
Отобразим спектрограммы
plt.show() # Выведем график
```


Обработка аудио



Mfcc вносит большой вклад в нейронную сеть (относительно других признаков).

6. Частота цветности

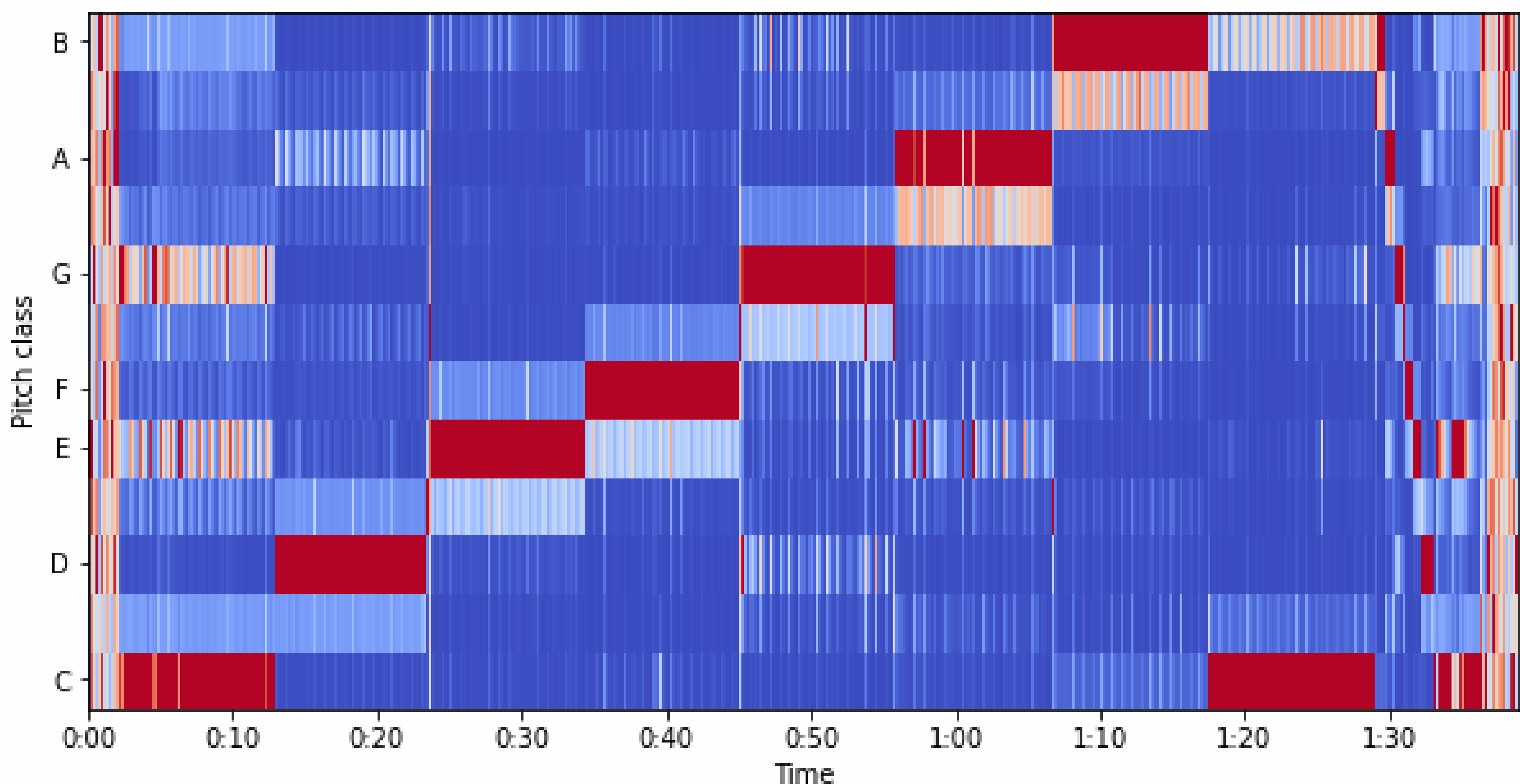
Функция цветности – это представление для музыкального звука, при котором весь спектр звука проецируется на 12 участков, представляющих 12 различных полутонов музыкальной октавы.

Для построения частоты цветности используем функцию **librosa.feature.chroma_stft()**:

```
# Загружаем сигнал
x, sr = librosa.load(audio_path)
hop_length = 512 # Задаем размер отрезка сигнала, по
                  # которому рассчитывается частоты цветности
# Рассчитываем и отображаем частоту цветности
chromagram = librosa.feature.chroma_stft(x, sr=sr, hop_
length=hop_length)
plt.figure(figsize=(10, 5)) # Зададим размер графика
```


Обработка аудио

```
librosa.display.specshow(chromagram, x_axis='time', y_
axis='chroma', hop_length=hop_length, cmap='coolwarm') #
Отообразим спектрограммы
plt.show() # Выведем график
```



A, B, C, D, E, F, G – это ноты и между ними полутона. На картинке представлена запись поочередного звукового сигнала 7 нот, примерно по 10 секунд каждой.

Комбинируя перечисленные признаки, можно выполнять такие задачи, как распознавание речи, цифровая обработка сигналов, а также классификация, тегирование и генерация музыки.