Міністерство освіти і науки України
Прикарпатський національний університет імені Василя Стефаника
Кафедра комп'ютерної інженерії та електроніки

Солонуха Роман Іванович

Solonuha Roman

УДК 004:681.5

Спеціальність 123 «Комп'ютерна інженерія»

(шифр та назва спеціальності)

Кваліфікаційна робота

на здобуття освітньо-кваліфікаційного рівня магістр

(бакалавр, спеціаліст, магістр)

Розроблення системи розпізнавання звучання музичних інструментів за допомогою нейромереж

Development of a system for recognizing the sound of musical instruments using neural networks

Науковий керівник:
кандидат фіз.-мат. наук,
доцент Грига В.М.

Рецензент:
доктор фіз.-мат. наук.,
ст. дослідник,
проф. кафедри
матеріалознавства і новітніх
технологій, Рачій Б.І.

Івано-Франківськ

2024

АНОТАЦІЯ

Мета цього проєкту – розробка системи розпізнавання звучання музичних інструментів за допомогою нейронних мереж. Система використовує сучасні методи обробки аудіо, такі як мел-спектрограми та коефіцієнти мел-кепстрального аналізу (MFCC), для виділення ключових характеристик звуку. Для класифікації музичних інструментів застосовано згорткову нейронну мережу (CNN), яка демонструє високу точність завдяки ефективному виявленню частотно-часових ознак.

У ході роботи проєкту було проаналізовано сучасні підходи до розпізнавання звуків, обрано оптимальні засоби розробки та реалізовано програмну частину системи. Результати тестування показали, що запропонована модель може з високою точністю класифікувати звуки таких інструментів, як піаніно, гітара, скрипка тощо. Проєкт доводить ефективність застосування глибокого навчання у задачах аудіоаналізу.

ABSTRACT

The goal of this project is to develop a system for recognizing musical instrument sounds using neural networks. The system employs modern audio processing methods, such as mel spectrograms and Mel Frequency Cepstral Coefficients (MFCC), to extract key sound features. A convolutional neural network (CNN) was implemented for instrument classification, demonstrating high accuracy by effectively capturing time-frequency characteristics.

The project includes an analysis of contemporary approaches to sound recognition, selection of optimal development tools, and implementation of the system's software component. Testing results show that the proposed model can accurately classify sounds from instruments such as piano, guitar, and violin. This project highlights the effectiveness of deep learning in audio analysis tasks.

ЗМІСТ

ВСТУП

Розвиток інформаційних технологій та штучного інтелекту відкриває нові можливості для автоматизації та вдосконалення різноманітних процесів у сфері музики. Однією з актуальних задач сучасного аудіоаналізу є розпізнавання звуків музичних інструментів, яке може бути застосоване в таких галузях, як автоматична транскрипція музики, індексація музичних записів, створення інструментів для музичного навчання, а також розробка адаптивних систем для синтезу та обробки звуків.

Завдяки розвитку глибоких нейронних мереж з'явилася можливість створювати високоточні системи для розпізнавання звуків музичних інструментів. Використання нейронних мереж дозволяє навчити модель знаходити особливі звукові ознаки, що притаманні різним музичним інструментам, що підвищує точність та ефективність розпізнавання в порівнянні з традиційними методами обробки аудіосигналів. Зокрема, глибоке навчання та згорткові нейронні мережі (CNN) успішно застосовуються для аналізу спектрограм та інших представлень звуку, що дозволяє виділити складні патерни та асоціювати їх з певними типами інструментів.

Мета цієї дипломної роботи – розробка та дослідження системи для автоматичного розпізнавання звуків музичних інструментів за допомогою нейронних мереж. Для досягнення цієї мети будуть використані методи обробки аудіосигналів, а також підходи машинного навчання та глибокого навчання. У рамках роботи планується дослідити різні архітектури нейронних мереж, проаналізувати їх ефективність для задачі розпізнавання, а також створити систему, здатну точно ідентифікувати звуки основних музичних інструментів.

Таким чином, розробка такої системи є важливим кроком на шляху до створення інтелектуальних аудіосистем, що здатні розуміти та аналізувати

звуковий контент з високою точністю, що може знайти широке застосування в музичній індустрії, навчальних платформах та технологіях обробки звуку.

РОЗДІЛ 1. АНАЛІЗ РОЗРОБКИ АНАЛОГІЧНИХ СИСТЕМ РОЗПІЗНАВАННЯ МУЗИЧНИХ ІНСТРУМЕНТІВ

В останні роки інтерес до автоматичного розпізнавання музичних інструментів зріс завдяки активному розвитку нейронних мереж та глибокого навчання. Науковці з різних країн зосереджують зусилля на створенні систем, здатних аналізувати та класифікувати звуки музичних інструментів з високою точністю. Сучасні системи розпізнавання музичних інструментів зазвичай використовують такі методи, як згорткові нейронні мережі (Convolutional Neural Networks, CNN), рекурентні нейронні мережі (Recurrent Neural Networks, RNN) (рис. 1.1), а також комбінації різних архітектур нейронних мереж, що дозволяє ефективно обробляти аудіосигнали.

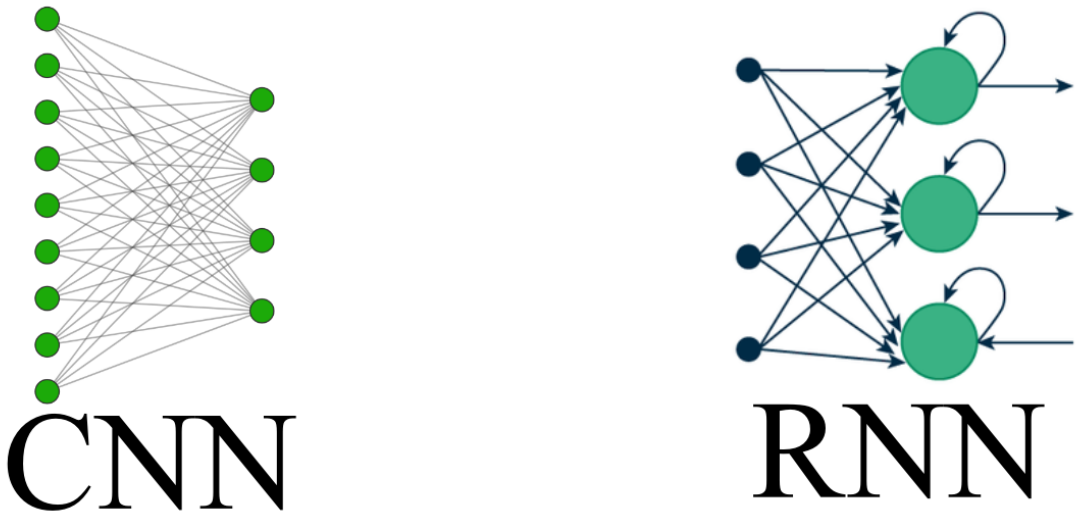


Рисунок 1.1 CNN та RNN

Першим етапом обробки звуку є його перетворення в зручне для нейронної мережі представлення, зазвичай у вигляді спектрограм, мел-спектрограм або коефіцієнтів мел-кепстральних частот (MFCC). Ці методи перетворення дозволяють візуалізувати особливості аудіосигналу, які характерні для різних інструментів, що значно полегшує процес навчання моделі. [1]

1.1 Огляд існуючих розпізнавальниї систем

1.1.1 Система CNN

Згорткові нейронні мережі (Convolutional Neural Networks, CNN) є потужним інструментом для аналізу даних із просторовою структурою, як-от зображення, аудіосигнали та навіть відео. У контексті розпізнавання музичних інструментів CNN широко застосовуються для обробки аудіоданих у вигляді спектрограм, мел-спектрограм або інших представлень звуку, де характерні частотні особливості інструментів можуть бути "зчитані" подібно до візуальних патернів (рис. 1.2).

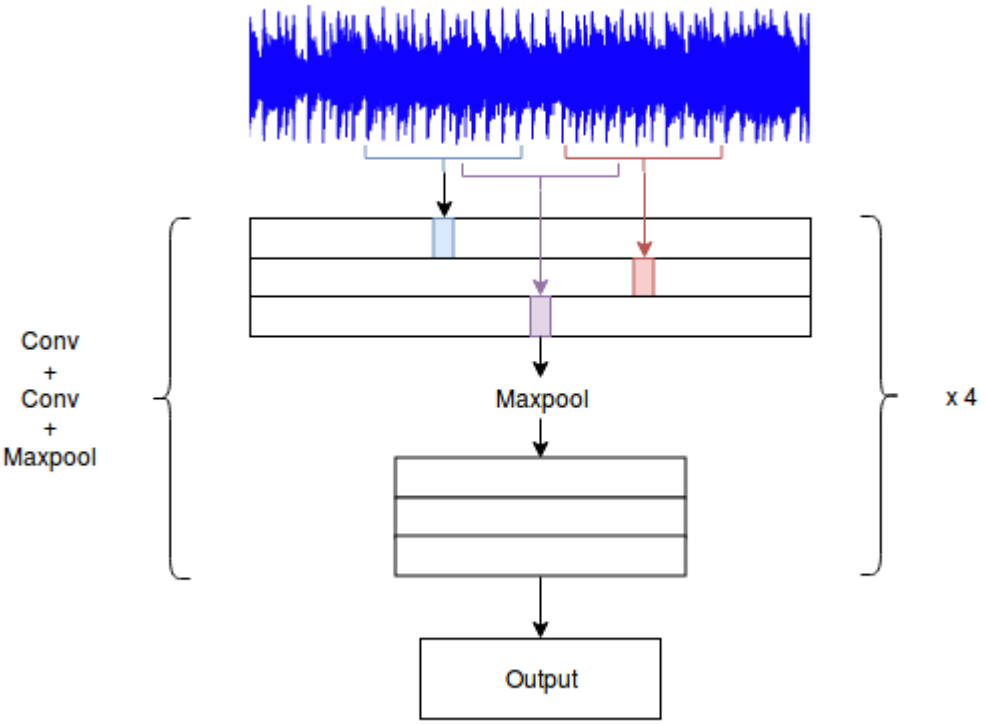


Рисунок 1.2 Зчитування візуальних патернів.

CNN складаються з кількох шарів, які виконують обчислення для виділення та інтерпретації локальних ознак. Основні шари CNN включають:

- Згортковий шар (Convolutional Layer): Цей шар відповідає за обчислення згортки між вхідними даними та ядрами (фільтрами), які налаштовані на виявлення характерних ознак, таких як контури, частоти та гармоніки в спектрограмі. Після проходження кожного згорткового шару вихід передається до наступного шару, збагачуючи інформацію про локальні ознаки сигналу.

- Шар активації: Найчастіше застосовується функція ReLU (Rectified Linear Unit), яка сприяє нелінійності мережі, дозволяючи моделі навчитись складним зв'язкам між ознаками.

- Шар підвибірки (Pooling Layer): Застосовується для зменшення розміру даних та зниження кількості параметрів, що робить модель більш ефективною та стійкою до незначних змін у вхідних даних.

Після кількох згорткових і шарів підвибірки застосовують повнозв'язні шари для класифікації звуків (рис. 1.3).

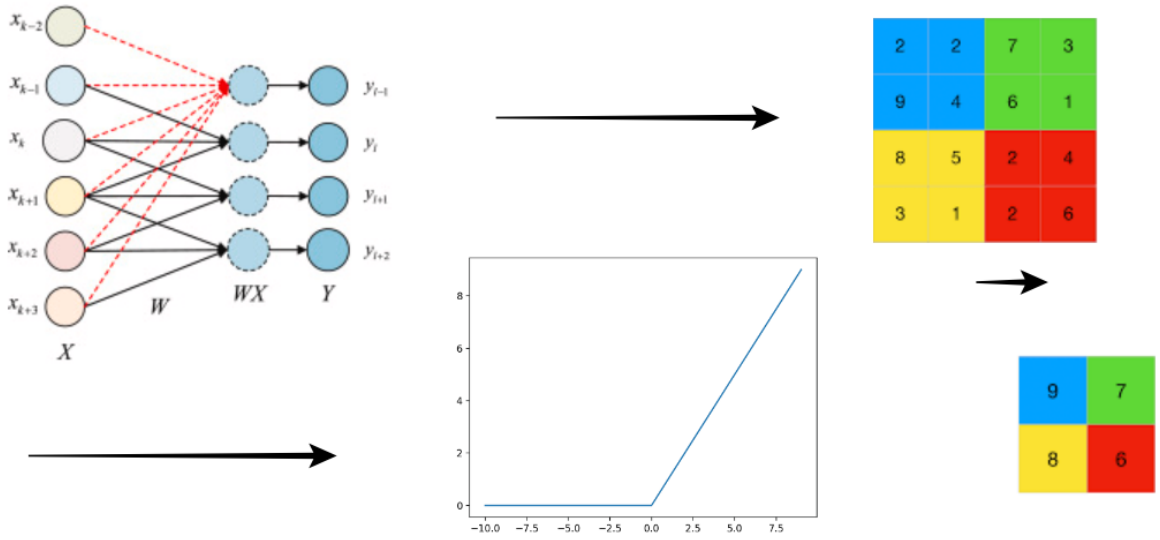


Рисунок 1.3 CL, ReLU та PL для класифікації звуків

Використання CNN у розпізнаванні музичних інструментів базується на тому, що різні інструменти мають унікальні частотні й тембральні ознаки, які

можна розпізнати на спектрограмі або мел-спектрограмі. Наприклад, флейта й піаніно мають характерні тембри, які залишають різні сліди в частотному представленні. CNN здатні автоматично вивчити ці патерни та розрізняти їх, забезпечуючи високу точність класифікації. [2]

Переваги CNN

1. Автоматичне виділення ознак: CNN можуть автоматично вивчити важливі ознаки, що знижує потребу у складній ручній інженерії ознак.
2. Інваріантність до зсувів і шумів: Здатність виділяти локальні ознаки робить CNN менш чутливими до незначних змін у вхідних даних.
3. Скорочення обчислень: Завдяки шарам підвибірки модель залишається відносно простою і швидкою для виконання, навіть коли обробляє великі обсяги даних.

Недоліки

1. Вимога до великих наборів даних: Для ефективного навчання CNN необхідна велика кількість якісних даних, які б охоплювали всі можливі варіації звуків.
2. Чутливість до якості вхідних даних: Низька якість запису або наявність шумів може знизити точність моделі.
3. Обмеження на розпізнавання складних звуків: Хоча CNN можуть добре працювати із сольними інструментами, розпізнавання інструментів у поліритмічній або багатошаровій музиці є значно складнішим завданням.

Згорткові нейронні мережі продемонстрували високу ефективність у задачах розпізнавання музичних інструментів завдяки здатності виділяти частотні та тембральні ознаки звуків. Водночас, для досягнення оптимальної точності CNN потребують добре збалансованих і великих навчальних наборів даних. Таким

чином, подальше вдосконалення технології включає розробку нових архітектур CNN і інтеграцію їх з іншими нейронними підходами, такими як RNN або трансформери, для підвищення точності у складніших умовах.

1.1.2 Поєднання CNN та RNN у розпізнаванні звуків музичних інструментів

Поєднання згорткових нейронних мереж (CNN) та рекурентних нейронних мереж (RNN) (рис. 1.4) є потужним підходом для задач розпізнавання звуків музичних інструментів. Така комбінація об'єднує сильні сторони обох типів мереж, дозволяючи ефективно працювати з аудіоданими та аналізувати тимчасові залежності, характерні для звуків музичних інструментів. У випадку аналізу музики це важливо, оскільки звук інструментів змінюється з часом, формуючи різні послідовності тональностей, тембрів та гармонік.

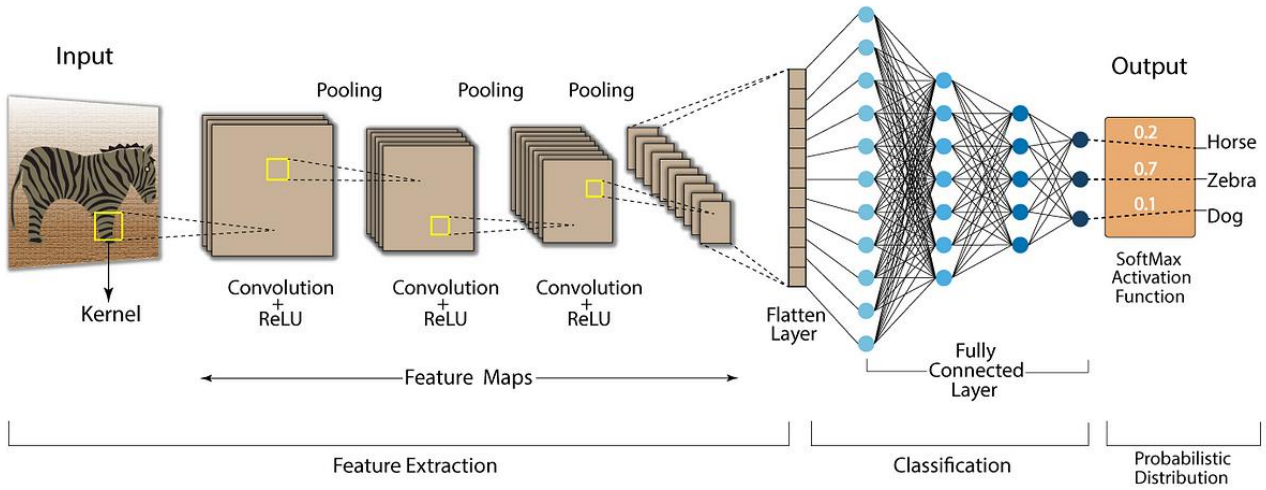


Рисунок 1.4 Поєднання CNN та RNN

Архітектура поєднання CNN та RNN

1. Етап виділення ознак (CNN): На першому етапі використовують CNN, які виділяють локальні ознаки із спектрограм або мел-спектрограм. CNN зчитують характерні патерни частот, інтенсивностей та гармонік для кожного фрейму звуку. Завдяки цим ознакам модель отримує детальну інформацію про тембр і частотну структуру звуку.

2. Етап аналізу послідовності (RNN): Після виділення ознак CNN використовують рекурентні шари, такі як LSTM (Long Short-Term Memory) або GRU (Gated Recurrent Unit), для обробки послідовності фреймів. RNN моделюють тимчасові залежності між окремими фреймами, дозволяючи враховувати розвиток звуку в часі. Наприклад, вони можуть "запам'ятати" поступовий розвиток або затухання звуку інструмента, що допомагає точніше класифікувати звук.

3. Фінальні шари (класифікація): Після обробки RNN вихід передається на повнозв’язні шари, які виконують остаточну класифікацію звуків за типами інструментів.

Ефективність комбінації CNN та RNN:

1. Виділення частотних ознак за допомогою CNN: CNN відмінно обробляє частотну інформацію в кожному часовому інтервалі, що дозволяє виділити основні ознаки для кожного фрагменту звуку.

2. Урахування темпоральних залежностей з RNN: RNN здатні запам'ятовувати контекстні зв'язки між різними часовими інтервалами, що важливо для обробки музичних звуків, які змінюються з часом. Наприклад, вони можуть розпізнати унікальні патерни атаки, затухання та вібрації звуку, характерні для конкретного інструмента.

3. Підвищення загальної точності: Комбінація дозволяє не лише розпізнавати миттєві частотні ознаки, але й враховувати зміни звуку в часі, що дає вищу

точність у порівнянні з моделями, що використовують тільки CNN або тільки RNN.

Поєднання CNN та RNN активно використовують у таких задачах:

- Розпізнавання інструментів у складній музичній композиції: Система може розрізняти звуки різних інструментів у випадках, коли вони грають одночасно або накладаються.
- Автоматична транскрипція музики: Точне розпізнавання інструментів і нот, яке потребує обліку часу появи та розвитку звуків.
- Аналіз музичних стилів та жанрів: Такий підхід дозволяє виділяти складні темпоральні патерни, характерні для певних жанрів чи стилів.

Комбінація CNN та RNN є ефективним інструментом для розпізнавання звуків музичних інструментів завдяки здатності виділяти як частотні, так і тимчасові особливості звуку. Така модель здатна навчатися складних зв'язків у звуках, що підвищує точність розпізнавання навіть у складних умовах (рис. 1.5). Подальший

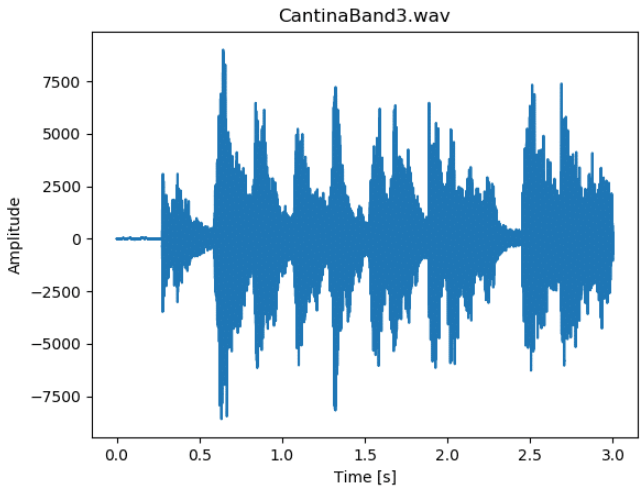


Рисунок 1.5 Послідовність частотних хвиль в аудіозаписі

розвиток технології полягає у вдосконаленні архітектур і оптимізації моделей для роботи з великими обсягами музичних даних, а також у розширенні їхніх можливостей для точного розпізнавання інструментів у різноманітних музичних творах.

1.1.3 Shazam

Shazam – це одна з найпопулярніших і ефективних комерційних систем для розпізнавання музики. Цей додаток здатний визначити назву композиції, виконавця та альбом, використовуючи короткий фрагмент аудіозапису. Він працює за принципом аудіоідентифікації, порівнюючи унікальний звуковий "відбиток" (fingerprint) вхідного сигналу з великою базою даних музичних творів. [2]

Основою роботи Shazam є створення та порівняння звукових відбитків. Ці відбитки створюються шляхом аналізу частотно-часових характеристик сигналу:

1. Аналіз спектрограми: З аудіофрагменту створюється спектрограма, яка показує, як змінюється енергія різних частот у часі.
2. Виділення унікальних точок (піків): Зі спектрограми вибираються частоти з найвищою енергією (пікові точки), які є унікальними для кожного запису.
3. Формування відбитку: Виділені пікові точки перетворюються у компактний цифровий відбиток, що описує композицію.
4. Пошук у базі даних: Відбиток порівнюється з мільйонами інших у базі даних Shazam. Завдяки оптимізованій структурі пошуку (наприклад, хеш-таблицям), система швидко знаходить збіги.

Shazam, хоч і є дуже ефективним для ідентифікації музичних композицій, не спеціалізується на класифікації звуків окремих музичних інструментів. Його алгоритм спрямований на пошук збігів між фрагментом і конкретною піснею, а не на аналіз характеристик звуку чи визначення інструменту. Однак його технологію звукових відбитків можна адаптувати для задач, пов’язаних із класифікацією інструментів. [4]

Порівняння з розробленою системою:

1. Ціль розпізнавання: Shazam зосереджений на ідентифікації музичних композицій, тоді як розроблена система орієнтована на розпізнавання звучання музичних інструментів.

2. Методологія**:** Shazam використовує унікальні звукові відбитки для пошуку в базі даних, тоді як у проєкті застосовуються мел-спектрограми та коефіцієнти MFCC для класифікації інструментів за допомогою машинного навчання.

3. Область застосування: Shazam є ідеальним інструментом для кінцевих користувачів, які хочуть знайти пісню. Натомість розроблена система може використовуватися для дослідницьких задач або в музичній освіті для аналізу інструментальних звучань.

1.2 Використання сучасних трансформерів

Трансформери – це потужна архітектура глибокого навчання, яка спочатку була розроблена для обробки природної мови, але згодом знайшла широке застосування в інших галузях, зокрема в обробці музичних даних. Музичні трансформери дозволяють моделювати послідовності нот, акордів, динаміку та інші аспекти музики, використовуючи самопильну увагу (self-attention) для виявлення залежностей у музичних творах. Завдяки цьому вони здатні генерувати, аналізувати та розпізнавати музичні патерни (рис. 1.6) з високою точністю. [6]

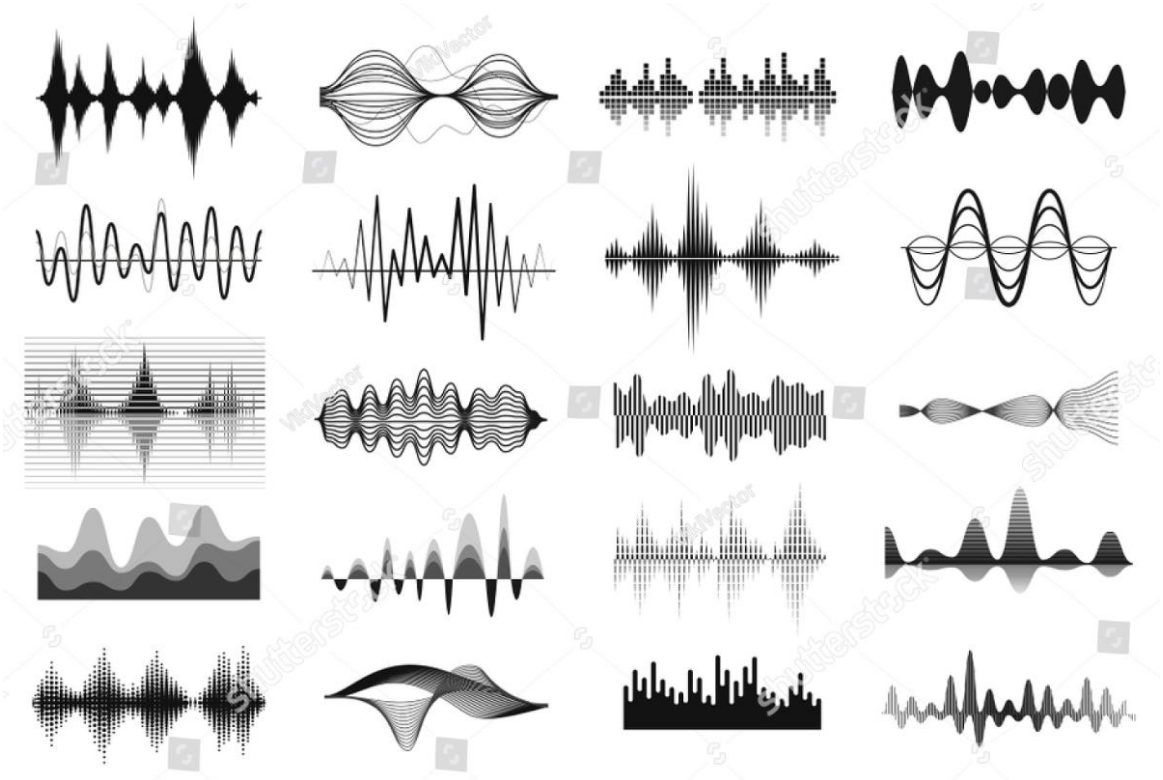


Рисунок 1.6 Приклад патерну звуків

Сучасні трансформери стали популярним підходом для задач обробки послідовностей, особливо в обробці природної мови, і тепер активно впроваджуються в аудіоаналізі, включаючи розпізнавання звуків музичних інструментів. Трансформери використовують механізм самоуваги (self-attention), що дозволяє моделі ефективно враховувати довготривалі залежності між елементами послідовності, зокрема часові та частотні зміни звуків, які характерні для кожного інструмента. [8]

На відміну від рекурентних нейронних мереж, які обробляють дані послідовно, трансформери можуть одночасно обробляти всі елементи послідовності, що значно підвищує їхню обчислювальну ефективність та дає змогу краще враховувати контекст на різних масштабах. Кожен елемент вхідної послідовності отримує вагові коефіцієнти відносно інших елементів, що дозволяє визначити найбільш важливі частини послідовності для кожного звукового фрагменту. Модель має кілька "голів" уваги, кожна з яких навчається виявляти різні аспекти зв'язків між елементами послідовності, що дозволяє їй

краще виділяти особливості звуку. Для додавання інформації про порядок елементів послідовності (адже трансформери самі по собі не зберігають порядкової інформації) додається позиційне кодування, яке зберігає часовий контекст для кожного звукового фрейму (рис. 1.7).

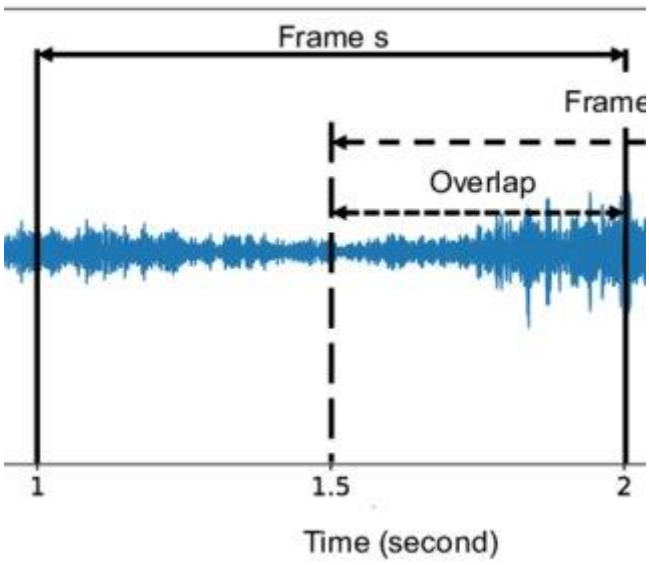


Рисунок 1.7 Розбиття вхідного аудіосигналу на кілька кадрів

У випадку розпізнавання музичних інструментів трансформери можуть обробляти дані в різних формах:

1. Спектрограми або мел-спектрограми: Як і в моделях CNN, спектральні представлення можуть бути введені у трансформер для аналізу частотних та тембральних особливостей.

2. Часові послідовності фреймів звуку: Трансформери можуть використовувати самовизначення для ідентифікації довготривалих зв’язків у звуці. Наприклад, вони можуть "запам'ятати" атаку і затухання звуку, що є характерним для певних інструментів.

Переваги використання трансформерів:

1. Довготривалі залежності: Завдяки самоувазі трансформери можуть враховувати зв'язки між віддаленими елементами звуку, що дозволяє краще моделювати складні темпоральні й частотні особливості.

2. Вища точність і гнучкість: Трансформери можуть легко налаштовуватися для складних задач, що потребують врахування контексту на різних масштабах. Вони здатні розрізняти інструменти навіть у складних композиціях із різними темпами та ритмами.

3. Ефективність у паралельній обробці: Оскільки трансформери не обробляють послідовності послідовно, вони можуть працювати паралельно, що знижує обчислювальні витрати й дозволяє обробляти великі обсяги даних швидше, ніж традиційні RNN.

Виклики при застосуванні трансформерів

1. Вимоги до великих наборів даних: Трансформери потребують великих обсягів даних для ефективного навчання, що може бути складним для аудіо задач, особливо якщо потрібні збалансовані дані для різних інструментів.

2. Висока обчислювальна складність: Хоча трансформери ефективні в обробці даних, вони мають високу обчислювальну вартість, особливо при обробці довгих послідовностей, що потребує значних ресурсів пам’яті та обчислень.

3. Потреба у додаткових архітектурних налаштуваннях: Трансформери добре зарекомендували себе в обробці тексту, але для аудіо задач можуть знадобитися специфічні адаптації, такі як введення шарів для обробки спектральних особливостей звуку.

Приклади використання трансформерів у розпізнаванні звуків інструментів (рис. 1.8):

- Audio Spectrogram Transformer (AST): Ця архітектура адаптована для обробки спектрограм звуків і використовує механізм самоуваги для виділення частотних і тимчасових ознак.
- Music Transformer: Ця модель використовується для задач, пов'язаних з аналізом послідовності музичних подій, і може використовуватися для задачі автоматичної транскрипції та класифікації музичних інструментів.

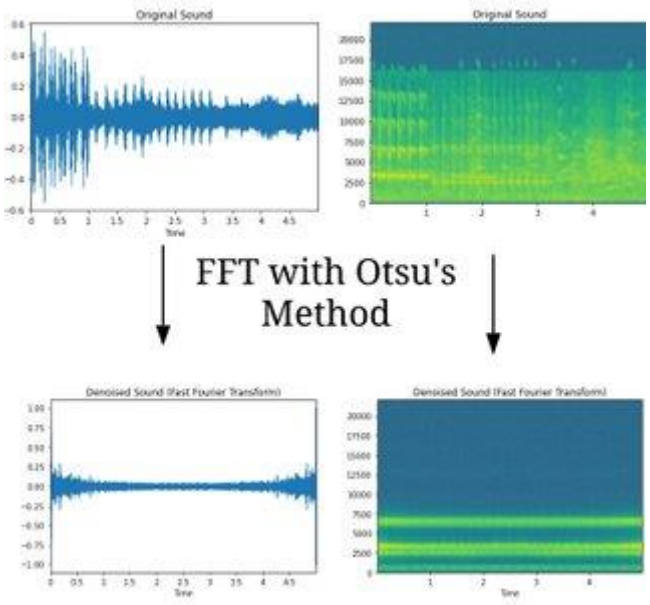


Рисунок 1.8 Розпізнавання звукових інструментів

Сучасні трансформери відкривають нові можливості для розпізнавання музичних інструментів завдяки їхній здатності ефективно моделювати як частотні, так і тимчасові залежності. Попри високу обчислювальну складність, трансформери є перспективним підходом для задач, що потребують глибокого аналізу звукових послідовностей, особливо у випадках складних музичних композицій або багатошарових аудіо записів. Подальший розвиток технології трансформерів в аудіоаналізі спрямований на оптимізацію їхньої архітектури для аудіо задач, що дозволить досягти ще вищої точності та ефективності в розпізнаванні звуків музичних інструментів. [9]

1.2.1 Приклади сучасних музичних трансформерів

Одна з перших адаптацій трансформера для музики та звуків - Music Transformer. Вона зосереджується на навчанні довгострокових залежностей у послідовностях нот. Music Transformer використовує MIDI-формат для вхідних даних і може генерувати музику у стилі навчальних зразків. Він виділяється своєю здатністю генерувати музичні твори, які зберігають структурну цілісність, наприклад повтори тем або модуляції.

MIDI (Musical Instrument Digital Interface) — це стандартний протокол для передачі цифрових даних між музичними інструментами, комп'ютерами та іншими пристроями. Формат MIDI не містить аудіоінформації, а замість цього зберігає інструкції для відтворення музики, такі як ноти, динаміка, тривалість, тембр та інші характеристики (рис. 1.9).

Jukebox від OpenAI потужна, оптимізована та продуктивна модель для генерації музики, яка працює з аудіо у високій роздільній здатності та самонавчається. Вона використовує багаторівневий підхід, де трансформери відповідають за прогнозування нот, акордів і тембральних характеристик. Jukebox здатний створювати реалістичну музику в різних стилях і навіть включати вокал.

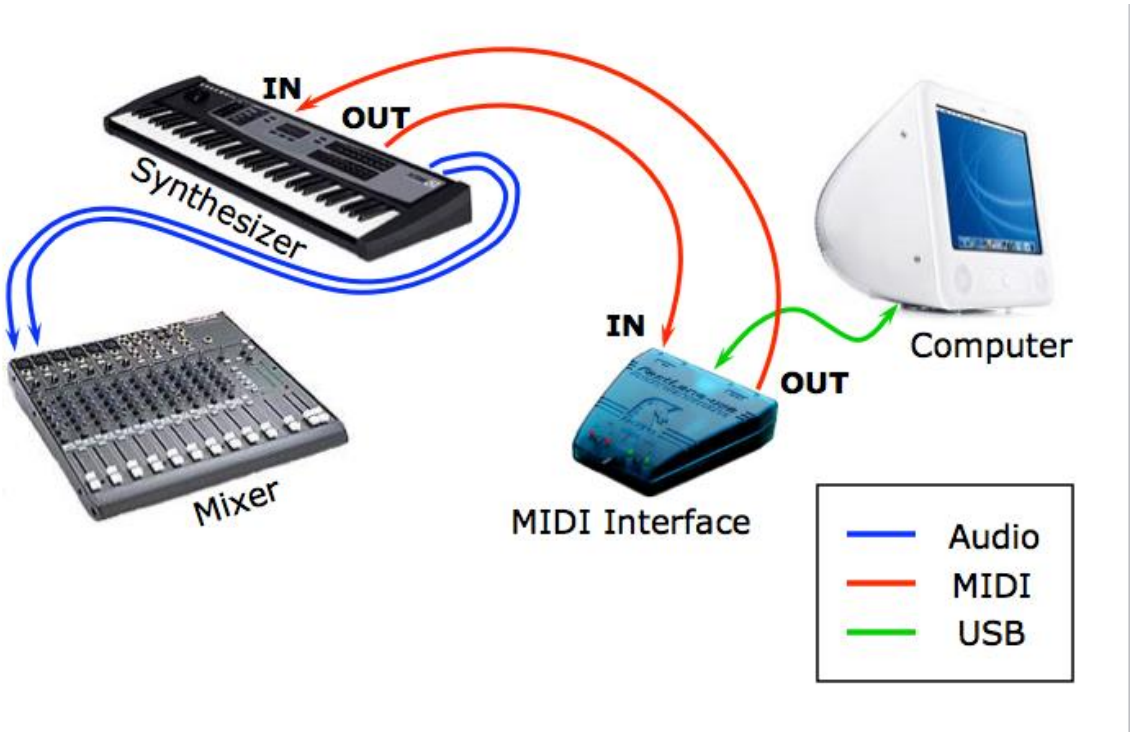


Рисунок 1.9 Побутове використання MIDI

1.2.2 Генерація музики

Генерація музики – це процес створення музичних творів за допомогою алгоритмів, моделей машинного навчання або інших програмних засобів. Вона дозволяє автоматично створювати оригінальні композиції, стилістично схожі на конкретні жанри або виконавців, а також музику для конкретних потреб, таких як фонові мелодії чи аранжування. У генерації музики широко використовуються моделі машинного навчання, такі як рекурентні нейронні мережі (RNN), які працюють з послідовними даними і дозволяють створювати мелодії та гармонії, що логічно пов’язані між собою.

Трансформери, які останнім часом стали популярними, застосовуються для моделювання довготривалих залежностей, завдяки чому здатні генерувати складні та довгі музичні твори. Також активно використовуються алгоритми на основі MIDI-формату, який дозволяє зберігати інформацію про ноти, акорди, темпо та інші параметри, полегшуючи роботу моделей.

Генерація музики знаходить застосування в різних сферах: від створення розважального контенту до автоматизації музичних аранжувань для інструментів і вокалу. Вона відкриває можливості для експериментів у музичній творчості, створення адаптивної музики в іграх та персоналізованих музичних рекомендацій.

1.2.3 Аранжування та гармонізація

Трансформери можуть автоматично створювати багатоголосі аранжування для сольних мелодій або гармонізувати вокальні партії.

Сучасні інструменти та алгоритми дозволяють автоматизувати процес аранжування та гармонізації, використовуючи правила музичної теорії та моделі машинного навчання. Наприклад, алгоритми можуть додавати акомпанемент до мелодії, підбирати акорди на основі контексту або створювати поліфонічні партії для ансамблів. Використання MIDI-формату полегшує роботу, оскільки дозволяє легко змінювати тональність, ритм або темп композиції, не втрачаючи якості. Моделі, такі як нейронні мережі та трансформери, дозволяють аналізувати музичні стилі й автоматично створювати аранжування, що відповідають заданому жанру.

Аранжування та гармонізація знаходять застосування у створенні музики для кіно, ігор, реклами та виступів, дозволяючи адаптувати музику до конкретних вимог чи побажань. Це також важливий інструмент для композиторів, які працюють над оркестровками, або для музикантів, які бажають додати новий шар до існуючих композицій. Сучасні алгоритми відкривають нові можливості для творчості, спрощуючи складні процеси та допомагаючи створювати високоякісний музичний продукт

1.3 Постановка завдання

Метою цієї роботи є розробка системи розпізнавання звуків музичних інструментів на основі нейронних мереж, що забезпечувала б високу точність класифікації, стійкість до шуму та змін у виконанні. Для досягнення цієї мети пропонується використовувати сучасні архітектури нейронних мереж, такі як згорткові нейронні мережі (CNN), рекурентні нейронні мережі (RNN) та трансформери, або їх комбінації, що дозволить моделювати як частотні, так і тимчасові залежності у звуці.

Для досягнення цієї мети необхідно виконати наступні завдання:

- Провести огляд наукових джерел і існуючих рішень, що використовують різні архітектури нейронних мереж, такі як CNN, RNN та трансформери.;
- Зібрати та підготувати набір аудіозаписів різних музичних інструментів;
- Порівняти різні архітектури нейронних мереж для розпізнавання музичних інструментів;
- Навчити обрану модель на підготовленому наборі даних, застосовуючи методи оптимізації параметрів, регуляризації та підвищення узагальнення для досягнення максимальної точності;
- Провести тестування моделі на тестовій вибірці для оцінки її точності, стійкості до шуму та змін у виконанні. Розрахувати показники якості моделі, такі як точність, повнота, F-міра тощо;
- Проаналізувати отримані результати, порівняти їх з існуючими рішеннями та визначити можливі напрями вдосконалення моделі. Розглянути перспективи застосування розробленої системи у реальних умовах.

РОЗДІЛ 2. ВИБІР ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ СИСТЕМИ РОЗПІЗНАВАННЯ МУЗИЧНИХ ІНСТРУМЕНТІВ

2.1 Вибір середовища розробки та бібліотек

Для розробки системи розпізнавання музичних інструментів обрано середовище Python, яке має широкий набір бібліотек і фреймворків для обробки аудіо, аналізу даних та машинного навчання. Python є популярною мовою серед дослідників і інженерів даних, що займаються аудіоаналізом та класифікацією звуків, завдяки своїй простоті, а також широкому спектру інструментів для роботи з сигналами і нейронними мережами. Нижче описано основні бібліотеки, які використовуються у проекті, та їхні функції.

Рисунок 2.1 Librosa

Проект базується на бібліотеці Librosa (рис. 2.1), яка є провідним інструментом для обробки аудіосигналів у Python. Вона забезпечує великий функціонал для аналізу аудіо, включаючи спектральний аналіз, екстракцію ознак та обробку сигналів. Librosa дозволяє отримувати спектральні ознаки, такі як мел-спектрограми та коефіцієнти мел-кепстрального аналізу (MFCC) (рис. 2.2), що є основними ознаками, які використовуються для класифікації аудіосигналів. Наприклад, мел-спектрограми є зручним способом представлення амплітуд частот в часі, що допомагає нейронній мережі розрізняти музичні інструменти.

Також Librosa підтримує функції нормалізації та редукції шуму, що покращує якість аналізованого сигналу перед подальшим навчанням.

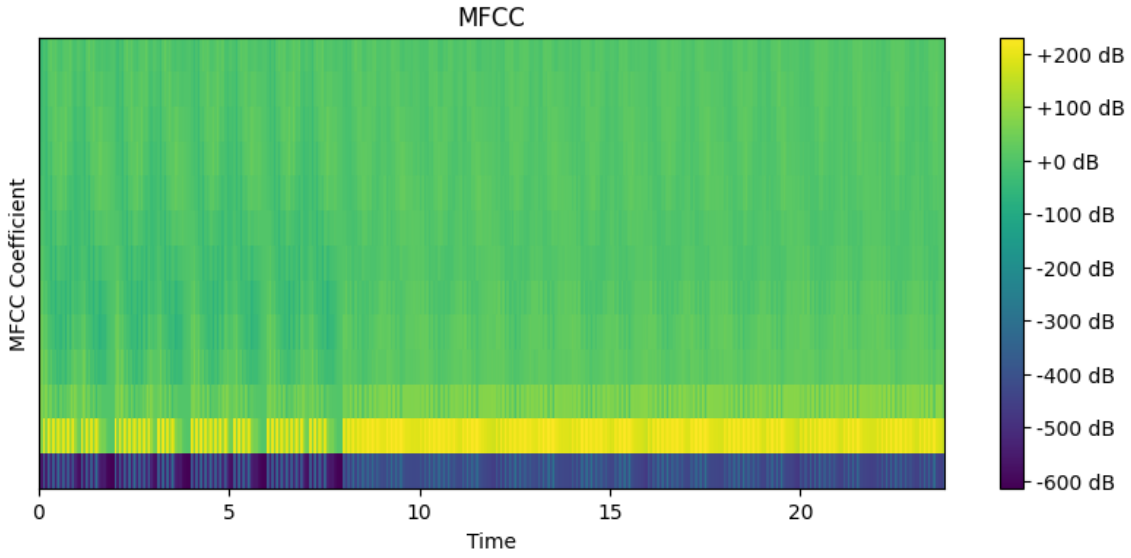


Рисунок 2.2 MFCC в Python

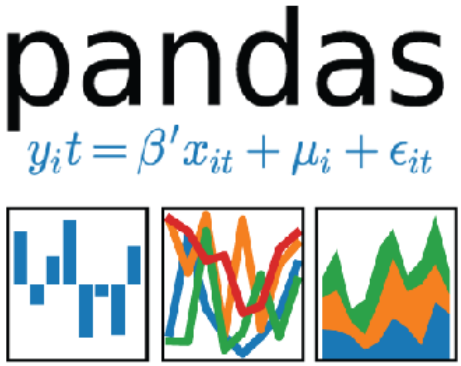
Для машинного навчання та класифікації обрано Scikit-learn — бібліотеку для загальної обробки даних і тестування алгоритмів. Scikit-learn (рис. 2.3) містить інструменти для попередньої обробки, масштабування та нормалізації даних, що особливо корисно на етапі підготовки вхідних аудіо ознак до моделі. Крім того, Scikit-learn дозволяє виконувати базові експерименти з класифікаторами, такими як K-ближчих сусідів (KNN) та SVM, які можуть бути корисні для тестування простих моделей до впровадження нейронних мереж. Її також використовують для розділення вибірок на навчальні та тестові набори, що дозволяє об'єктивно оцінити точність моделі.

Рисунок 2.3 Scikit-learn

Основними фреймворками для реалізації глибокого навчання є TensorFlow та Keras. Система забезпечує високий рівень абстракції для швидкого створення моделей, що робить його зручним для прототипування нейронних мереж. TensorFlow, в свою чергу, є більш гнучким та потужним інструментом, який працює на більш низькому рівні, забезпечуючи широкі можливості для налаштування та оптимізації моделі. Keras використовується для побудови згорткових нейронних мереж (CNN), які добре підходять для обробки спектрограм та зображень, де потрібно виділяти особливості просторового та частотного розподілу. У даному проекті CNN використовується для автоматичного виділення ознак з мел-спектрограм, що дозволяє покращити точність класифікації музичних інструментів.

Обробка аудіоданих також вимагає додаткових засобів для роботи з файлами аудіо. SoundFile і Wave дозволяють завантажувати аудіофайли та зчитувати їх у потрібному форматі, зокрема .wav. Ці інструменти інтегруються з Librosa, полегшуючи завантаження та обробку аудіофайлів різних форматів.

Для роботи з даними та їх зберігання використовуються Pandas (рис. 2.4) та NumPy. Pandas спрощує управління даними, забезпечуючи функціонал для зчитування, зберігання та маніпуляцій з великими обсягами інформації у вигляді структурованих таблиць. NumPy, у свою чергу, дозволяє ефективно виконувати математичні операції над масивами даних, необхідними для виконання різних трансформацій та підготовки даних для нейронних мереж.



	BandName	WavelengthMax	WavelengthMin
0	CoastalAerosol	450	430
1	Blue	510	450
2	Green	590	530
3	Red	670	640
4	NearInfrared	880	850
5	ShortWaveInfrared_1	1650	1570
6	ShortWaveInfrared_2	2290	2110
7	Cirrus	1380	1360

Рисунок 2.4 Pandas в Python

Обране середовище та бібліотеки утворюють цілісну систему для побудови моделі розпізнавання музичних інструментів. Librosa та Scikit-learn забезпечують надійну попередню обробку аудіо, а Keras з TensorFlow надають потужні засоби для створення та навчання моделей глибокого навчання, що дозволяє досягти високої точності розпізнавання інструментів у даному проекті.

2.2 Librosa

Librosa — це спеціалізована бібліотека Python для аналізу та обробки музичних і аудіосигналів, яка широко використовується для екстракції ознак, спектрального аналізу, фільтрації та попередньої обробки аудіо. Ця бібліотека (рис. 2.5) стала стандартом у дослідженнях аудіо, оскільки надає широкий спектр функцій, які полегшують завдання роботи з аудіо, включаючи підтримку роботи з різними типами звукових файлів, фільтрацію шуму, спектральну сегментацію та перетворення сигналів у зручний для аналізу формат.

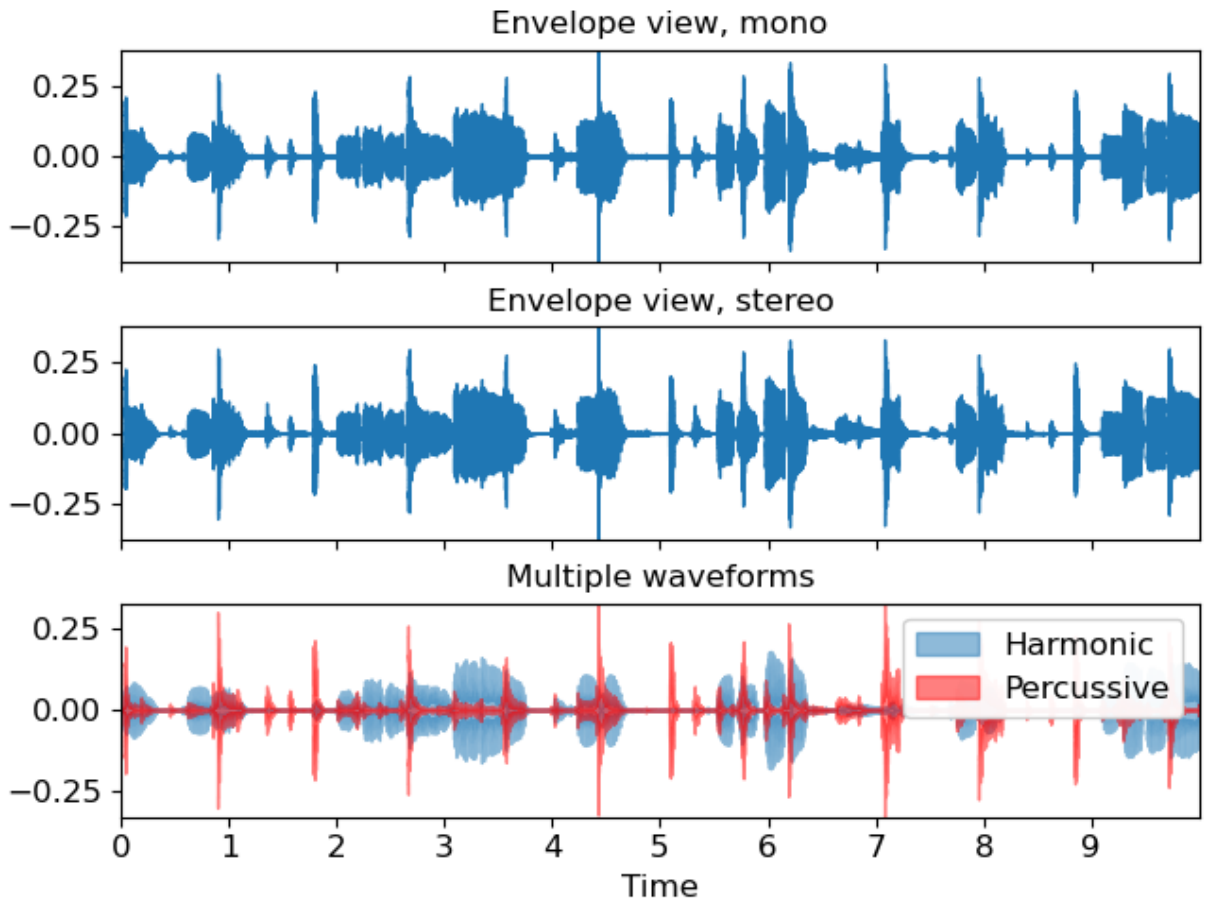


Рисунок 2.5 Використання бібліотеки Librosa

Librosa дозволяє легко завантажувати аудіофайли різних форматів (зокрема, WAV та MP3) у вигляді масивів, що спрощує подальше їх використання в моделях. Функція `librosa.load()` зчитує файл, перетворюючи його у масив амплітуд, і дозволяє вибирати частоту дискретизації (`sampling rate`), що визначає якість аудіо і зменшує обсяг обчислень для обробки. Частота дискретизації встановлюється за замовчуванням на 22050 Гц, що достатньо для більшості задач, однак може бути налаштована під специфічні вимоги проєкту.

Librosa містить широкий спектр методів для спектрального аналізу, що дозволяє перетворювати аудіо у форму, зручну для розпізнавання і класифікації:

Short-Time Fourier Transform (STFT) — перетворення, яке представляє аудіосигнал у частотно-часовій області, розділяючи його на короткі фрагменти (рис. 2.6). Використання STFT дозволяє отримати часові зміни частотних

компонентів звуку, що є важливим для аналізу таких сигналів, як музичні інструменти.

Мел-спектрограма одна з найважливіших функцій для класифікації музичних інструментів. Мел-спектрограма представляє частоти сигналу на логарифмічній мел-шкалі, що відповідає людському сприйняттю звуку. Для побудови мел-спектрограми бібліотека використовує фільтр мел-шкали, що агрегує енергії частот у вузьких діапазонах, що відповідають частотам людського слуху. Функція `librosa.feature.melspectrogram()` автоматизує цей процес і дозволяє задавати кількість мел-фільтрів і частотний діапазон, що дає користувачу гнучкість у налаштуванні.

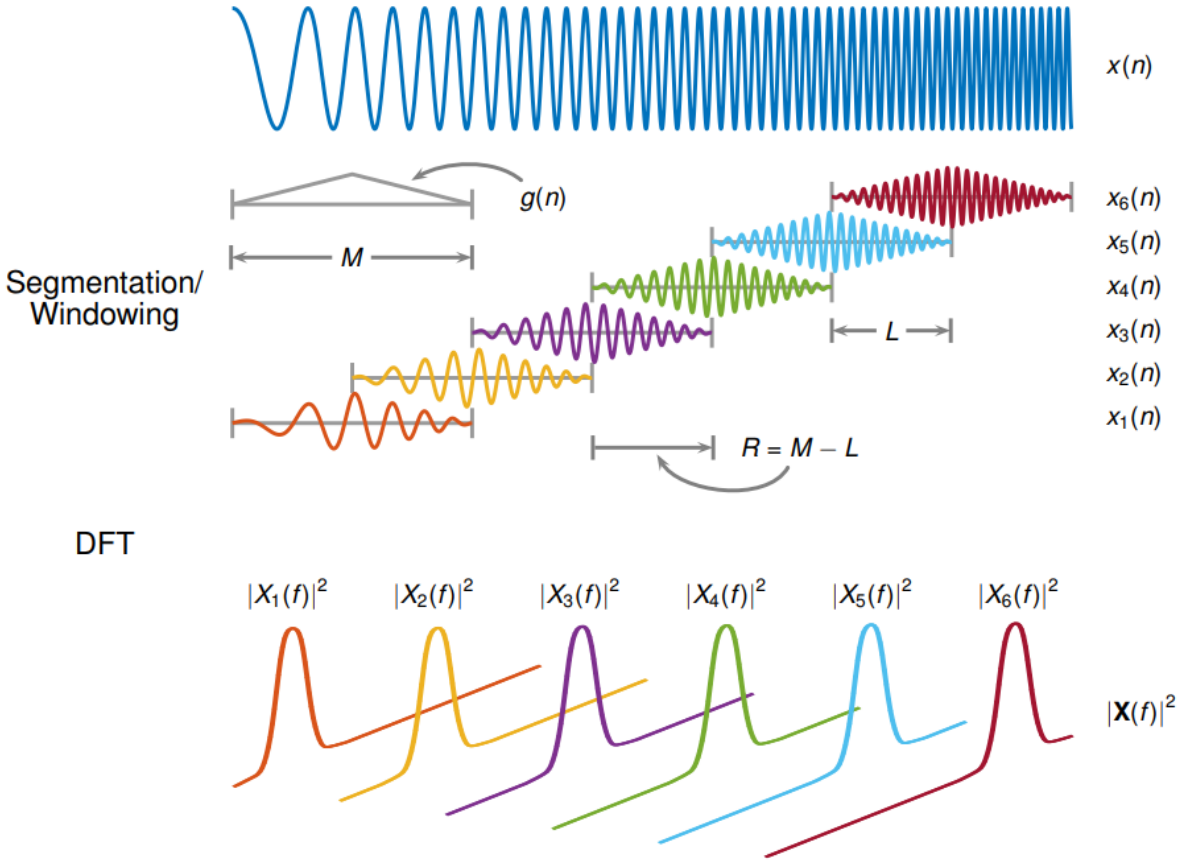


Рисунок 2.6 Короткочасне перетворення Фур'є

MFCC (Mel Frequency Cepstral Coefficients) — це коефіцієнти мел-кепстрального аналізу, які характеризують темброві особливості звуку. MFCC є одним із найпоширеніших ознак для класифікації аудіо, зокрема мовних і

музичних звуків. Функція `librosa.feature.mfcc()` дозволяє легко обчислити MFCC з мел-спектрограми, використовуючи параметри кількості коефіцієнтів і частотний діапазон.

Chromagram — це спектральне представлення, яке показує інтенсивність кожного з 12 півтонів музичної шкали, незалежно від октави. Chromagram застосовується для аналізу гармонійних особливостей музики і може використовуватися для розпізнавання тональності і акордів.

Librosa забезпечує інструменти для витягання великої кількості характеристик аудіосигналу, що є критично важливим для класифікації звуку. Деякі з ключових характеристик:

1. Звукова енергія та RMS — визначають силу сигналу, що може відображати гучність інструменту.
2. Частота нульових перетинів — показує, скільки разів сигнал перетинає нульову вісь за певний час. Це ознака, що використовується для розрізнення гармонічних і шумових звуків.
3. Спектральний центр — показує «центр ваги» частотного спектру, що корелює з «яскравістю» звуку. Наприклад, високий спектральний центр часто свідчить про більш високочастотний інструмент.
4. Спектральна ширина, контраст і плоскість — характеристики, що дозволяють розрізняти звуки за їхнім тембром, складністю та гармонічними властивостями.

Таким чином, Librosa є важливим інструментом у системах класифікації музичних інструментів, оскільки дозволяє отримувати спектральні і темброві характеристики сигналу, що є ключовими для розпізнавання.

2.2.1 Короткочасне перетворення Фур'є

Звичайне перетворення Фур'є надає інформацію про частотний склад сигналу, але втрачає часовий контекст. STFT вирішує цю проблему, зберігаючи обмежений часовий контекст за рахунок розділення сигналу на сегменти. Це досягається шляхом множення сигналу на віконну функцію (наприклад, функцію Гаусса або Хеммінга), яка визначає межі аналізованого фрагмента.

Формула STFT (фор. 1):

$$STFT = X(\tau, \omega) = \int_{-\infty}^{\infty} x(t) w(t - \tau) e^{-i\omega t} dt$$

Формула 2.7 Формула STFT

де:

- x(t)$x(t)$ — вихідний сигнал,
- w(τ−t)$w(\tau - t)$ — віконна функція, що виділяє частину сигналу навколо часу tt,
- e−j2πfτ$e^{-j2\pi f\tau}$ — функція для перетворення Фур'є, яка визначає частоти.

Перетворення Фур’є дозволяє представити сигнал, який змінюється в часі, у вигляді набору гармонічних складових (синусоїд) різних частот. У результаті перетворення отримуємо комплексні числа, де:

- Дійсна частина описує фазу.
- Уявна частина пов’язана з амплітудою гармонічної складової.

Також використовується для побудови амплітудного спектра, який показує, як змінюється енергія сигналу залежно від частоти. Це дозволяє визначити основні гармоніки, шум і характеристики сигналу.

2.2.2 Мел-спектрограма

Мел-спектрограма – це представлення аудіосигналу у частотно-часовій площині, де частоти перетворюються у шкалу Мела, що відповідає сприйняттю

звуку людиною (рис. 2.8). Цей підхід дозволяє зробити спектральний аналіз звуку більш зручним для задач, де важливе врахування психоакустичних особливостей слуху. Шкала Мела нелінійно відображає частотний спектр: у нижньому діапазоні частоти мають високу роздільну здатність, а у верхньому – групуються більш грубо. Це відображає особливості людського слуху, який чутливіший до змін у низькочастотному діапазоні, ніж до змін у високих частотах.

Для побудови мел-спектрограми аудіосигнал спочатку розділяється на короткі сегменти за допомогою віконної функції. Потім до кожного сегмента застосовується короткочасне перетворення Фур'є (STFT) для отримання частотно-часового представлення. Отриманий спектр обробляється за допомогою набору мел-фільтрів, які акцентують увагу на частотах відповідно до шкали Мела. Фільтри мають трикутну форму і розташовуються на лінійному інтервалі частот, але їхня ширина збільшується у міру зростання частоти. Таким чином, спектральна інформація перетворюється у форму, яка краще узгоджується з характеристиками людського сприйняття звуку. [12]

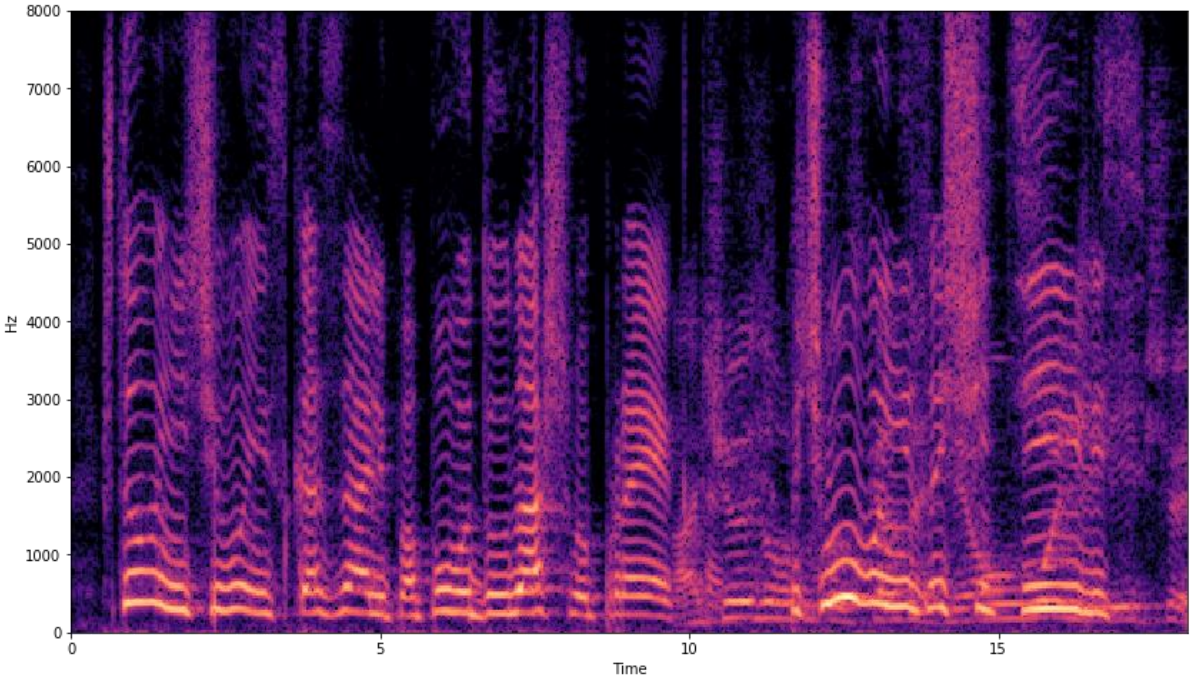


Рисунок 2.8 Мел-спектрограма

Мел-спектрограми широко використовуються в задачах машинного навчання та обробки звуку, таких як розпізнавання мови, класифікація звуків та інструментів, а також генерація музики. Вони є основою для розрахунку коефіцієнтів мел-кепстрального аналізу (MFCC), які є одними з найпоширеніших ознак у сфері аналізу звуку. Завдяки компактному і релевантному представленню звуку, мел-спектрограми ефективно використовуються у нейронних мережах, зокрема у згорткових (CNN), які працюють із двовимірними даними.

Мел-спектрограма зберігає часовий контекст сигналу, що дозволяє нейронним мережам розпізнавати зміни в частотах протягом часу. Це особливо важливо для аналізу музичних композицій та мовних сигналів, де частотна структура динамічно змінюється. Візуалізація мел-спектрограми показує частотний спектр у вигляді кольорового зображення, де яскравість або колір відображають амплітуду кожної частоти у певний момент часу. Такий підхід дозволяє інтуїтивно зрозуміти, як змінюється звук протягом часу, і є важливим інструментом для візуального аналізу аудіосигналів.

Обчислення мел-спектрограм зазвичай виконується за допомогою бібліотек програмного забезпечення, таких як Librosa у Python. Використовуючи цю бібліотеку, можна отримати спектрограму у кілька рядків коду, задаючи параметри, як-от кількість фільтрів, тривалість вікна або частота дискретизації. Завдяки простоті реалізації та високій ефективності мел-спектрограма стала стандартом у задачах обробки звуку та широко використовується у проєктах зі створенням та класифікацією аудіоданих.

2.2.3 Chromagram

Представлення аудіосигналу, яке показує енергію окремих музичних півтонів у заданому часовому вікні, незалежно від їхньої октави називається хромограмою. Цей підхід базується на принципі, що музика часто сприймається

як набір гармонійних і мелодійних структур, що повторюються в різних октавах. Chromagram дозволяє абстрагуватися від абсолютних частот і зосередитися на хроматичному контексті, що є корисним для аналізу гармоній, тональності, акордів та мелодій.

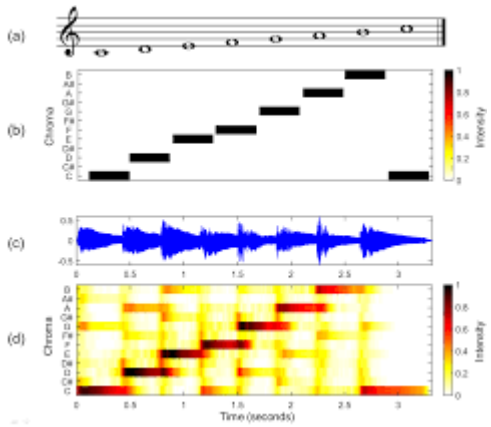


Рисунок 2.9 Розпізнавання хромограмою

Основою для побудови Chromagram є короткочасне перетворення Фур'є (STFT), яке використовується для отримання частотно-часового спектру сигналу. Потім енергія частот агрегується в 12 основних півтонів музичної шкали (наприклад, C, C#, D, D#, і т.д.), незалежно від октави. Для цього використовуються фільтри, які визначають, до якого півтону належить кожна частота. У результаті отримуємо матрицю, де кожен рядок відповідає одному з 12 півтонів, а стовпчики — часовим сегментам.

Chromagram особливо корисний у музичному аналізі. Воно дозволяє розпізнавати акорди, визначати тональність композиції або виявляти зміни гармонійної структури в часі. Наприклад, у мелодії або супроводі можна побачити, як змінюються акорди (C → G → Am), або відслідкувати модуляцію з однієї тональності до іншої. Chromagram також ефективно використовується для порівняння музичних творів, аналізу стилю або створення алгоритмів для музичних рекомендацій.

Chromagram добре інтегрується в задачі машинного навчання. Оскільки воно абстрагується від абсолютних частот, це зменшує обсяг даних, які потрібно

аналізувати, і дозволяє зосередитися на музичній структурі. Це робить Chromagram зручним інструментом для класифікації музичних жанрів, розпізнавання інструментів і аналізу ритміки. Наприклад, у системах рекомендацій, таких як Spotify або Shazam, Chromagram допомагає визначати музичну схожість між композиціями.

2.3 TensorFlow та Keras

TensorFlow та Keras (рис. 2.10) є двома тісно пов'язаними фреймворками для глибокого навчання, які використовуються для побудови та налаштування складних нейронних мереж, таких як згорткові нейронні мережі (CNN), рекурентні мережі (RNN) і комбіновані архітектури. TensorFlow, розроблений компанією Google, є потужним інструментом для виконання математичних обчислень, необхідних для навчання нейронних мереж, а Keras надає простіший інтерфейс, який працює на основі TensorFlow і дозволяє швидко прототипувати моделі завдяки своїй простій, але гнучкій структурі.

TensorFlow забезпечує виконання математичних операцій, необхідних для оптимізації та роботи моделей глибокого навчання, завдяки своїй здатності ефективно працювати з багатовимірними масивами (тензорами) та розподіляти обчислення на процесори й графічні процесори. Це дозволяє обробляти великі обсяги даних паралельно, що особливо важливо при навчанні моделей на великих наборах даних, таких як аудіофайли у системах розпізнавання. TensorFlow має потужний інструментарій для налаштування градієнтного спуску, регулювання швидкості навчання та оптимізації моделей, що дозволяє забезпечити кращу точність та стабільність у процесі навчання. Завдяки цьому TensorFlow є ідеальним вибором для обробки великих обсягів звукових даних і досягнення високої продуктивності.

Keras, будучи надбудовою над TensorFlow, спрощує процес побудови моделей і дозволяє зосередитися на архітектурі нейронної мережі без потреби

заглиблюватися у деталі обчислень. У Keras надаються готові модулі для створення різних типів шарів, зокрема згорткових шарів, пулінг-шарів і повнозв'язних шарів, а також активаційні функції та оптимізатори, які можна швидко інтегрувати в модель. Це особливо корисно для швидкої розробки, тестування та вдосконалення моделей у таких задачах, як класифікація музичних інструментів, де потрібні багаторівневі згорткові шари для виділення часово-частотних ознак аудіосигналу. Наприклад, у моделі згорткових нейронних мереж для розпізнавання музичних інструментів Keras дозволяє легко налаштовувати кількість згорткових фільтрів, розмір фільтрів, типи шарів та їхнє з'єднання, забезпечуючи при цьому чіткий та зрозумілий код.

Keras також надає модулі для обробки даних, як-от інструменти для збільшення вибірки, що дозволяє підвищити різноманітність даних та запобігти перенавчанню моделі. Використовуючи функції Keras, можна реалізувати динамічне збільшення вибірки, наприклад, змінюючи амплітуду або швидкість відтворення аудіозаписів, що розширює кількість доступних зразків і сприяє кращій генералізації моделі. Це особливо важливо в проєктах з обмеженим

Рисунок 2.10 TensorFlow і Keras

обсягом даних, де розширення вибірки може покращити стабільність та точність моделі.

TensorFlow і Keras також забезпечують можливість ефективного моніторингу навчання та збереження проміжних результатів. Функції відстеження продуктивності моделі, такі як контроль показників точності та

похибки на навчальних та валідаційних наборах даних, дозволяють своєчасно виявляти проблеми з перенавчанням або недонавчанням. Крім того, TensorFlow підтримує автоматичне збереження найкращої версії моделі та можливість відновлення навчання з обраного етапу. Ця функція є зручною для проєктів, що потребують тривалого навчання, дозволяючи уникати втрати прогресу при виникненні технічних збоїв або при бажанні оптимізувати модель на різних етапах.

Загалом, використання TensorFlow та Keras у поєднанні забезпечує потужний інструмент для створення та навчання нейронних мереж. TensorFlow надає можливості для високопродуктивних обчислень, а Keras спрощує процес розробки і дозволяє фокусуватися на дизайні та продуктивності моделі, що є особливо важливим для систем класифікації музичних інструментів, де точність розпізнавання і гнучкість моделі є критичними параметрами. [10]

2.4 Мова програмування Python

Python є основною мовою програмування, що забезпечує виконання всіх ключових етапів розробки системи розпізнавання музичних інструментів. Python використовують для обробки аудіоданих, екстракції ознак, створення та навчання нейронної мережі, а також для аналізу та візуалізації результатів. Завдяки своїй багатій екосистемі бібліотек, Python полегшує обробку аудіосигналів через бібліотеку Librosa, яка дозволяє завантажувати аудіофайли, виконувати спектральний аналіз і отримувати такі ознаки, як мел-спектрограми і MFCC, які є основними для класифікації музичних інструментів. Інструменти обробки даних, такі як Pandas і NumPy, використовуються для зберігання та маніпулювання зразками ознак і підготовки їх до подачі в модель. [5]

У Python також задіяні TensorFlow і Keras, фреймворки для машинного навчання, що забезпечують зручний інтерфейс для створення нейронної мережі. За допомогою Keras будують архітектуру згорткової нейронної мережі (CNN), налаштовуючи різні параметри, такі як кількість шарів і функції активації, а TensorFlow виконує оптимізацію обчислень і підтримує роботу з великими обсягами даних, забезпечуючи ефективне навчання моделі (рис. 2.11). Python також забезпечує гнучкість у тестуванні та оптимізації моделі, дозволяючи налаштовувати параметри навчання та виконувати розширене тестування на валідаційних даних, що сприяє покращенню точності та стабільності.

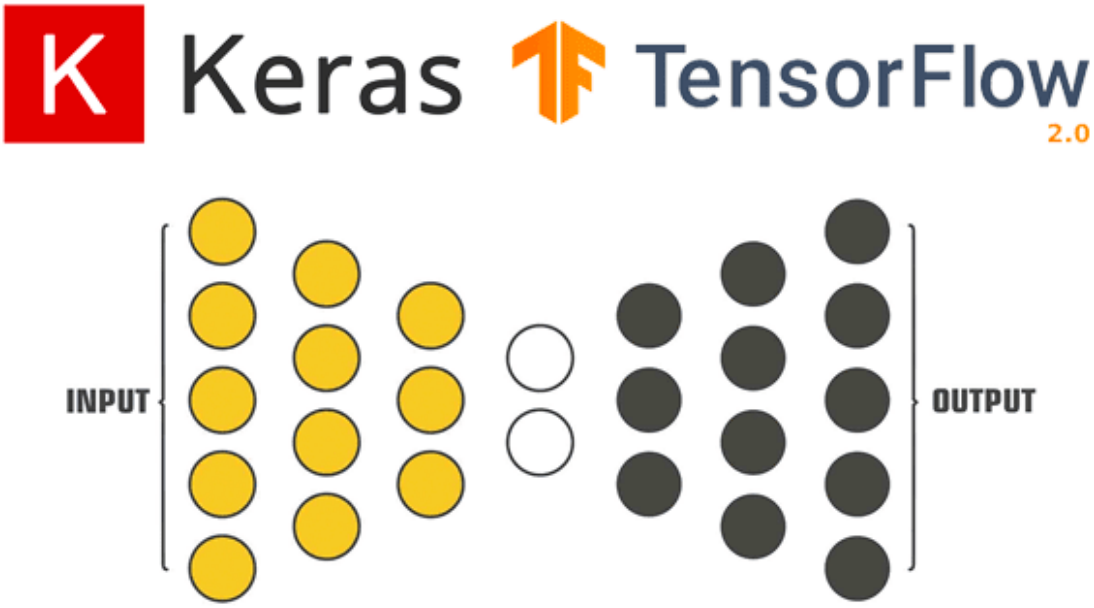


Рисунок 2.11 Оптимізація обчислень

Крім того, Python використовується для збереження та завантаження моделі після навчання, що забезпечує можливість подальшої інтеграції моделі в додатки або системи. Бібліотека Matplotlib в Python допомагає візуалізувати результати, відображати спектрограми та інші ознаки, а також аналізувати ефективність моделі та її помилки, що є важливим етапом у процесі розробки. Загалом, Python об’єднує всі етапи — від обробки аудіо до навчання нейронної

мережі та оцінки її продуктивності, що робить його незамінним інструментом у розробці системи розпізнавання музичних інструментів.

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ІНДИВІДУАЛЬНОГО ПРОЕКТУ

Програмна реалізація системи розпізнавання звучання музичних інструментів складається з кількох ключових етапів: підготовки даних, обробки аудіосигналів, екстракції ознак, побудови та навчання моделі нейронної мережі, а також тестування отриманих результатів. Для реалізації використовувалося середовище розробки на основі Python із застосуванням бібліотек для обробки даних та машинного навчання.

Музичні інструменти бувають різноманітних форм і розмірів, а їхні звукові характеристики можуть бути або виразно унікальними, або схожими на інші інструменти. Один і той самий тип інструменту може видавати різні звуки залежно від матеріалу, з якого він виготовлений, а також від стилю виконання різними музикантами. У цьому проєкті ми використовуємо методи машинного навчання для аналізу різних характеристик музичних інструментів і оцінки їхньої здатності розрізняти широкий спектр інструментів. Це здійснюється шляхом дослідження частотного спектра аудіосигналу у поєднанні з методами класифікації.

Основною метою є використанням концепцій обробки сигналів у частотній області та класифікації ознак за допомогою методів машинного навчання.

3.1 Виконання індивідуального проєкту

На початковому етапі завантажуються аудіофайли, які використовуються для навчання та тестування моделі. Основним форматом даних є WAV-файли, які легко обробляються за допомогою бібліотеки Librosa. Для забезпечення однаковості обробки всі файли перетворюються до однакової частоти дискретизації (наприклад, 22050 Гц). Обрізка або додавання тиші до файлів використовується для вирівнювання тривалості аудіофрагментів. Аудіодані для

навчання системи були зібрані з відкритих джерел, що містять записані фрагменти звучання різних музичних інструментів (наприклад, скрипка, піаніно, гітара). Основним завданням на цьому етапі є забезпечення якості даних:

- Приведення усіх аудіофайлів до одного формату (.wav) та частоти дискретизації (22050 Гц) (рис. 3.1).

```python
import librosa

# Завантаження файлу
y, sr = librosa.load('audio_file.wav', sr=22050)  # sr – частота дискретизації
```

Рисунок 3.1 Завантаження файлу для частоти дискретизації

- Нормалізація гучності сигналу, обрізка надмірної тиші та вирівнювання тривалості аудіофрагментів.

Основними ознаками, які використовуються для класифікації, є мел-спектрограми та коефіцієнти мел-кепстрального аналізу (MFCC). Ці ознаки витягуються з аудіофайлів за допомогою бібліотеки Librosa. Приклад коду для отримання мел-спектрограми (рис. 3.2).

```python
import librosa
import librosa.display
import numpy as np
import matplotlib.pyplot as plt

# Завантаження аудіофайлу
y, sr = librosa.load('example.wav', sr=22050)

# Отримання мел-спектрограми
mel_spec = librosa.feature.melspectrogram(y=y, sr=sr, n_mels=128, fmax=8000)

# Логарифмічна шкала
mel_spec_db = librosa.power_to_db(mel_spec, ref=np.max)
```

Рисунок 3.2 Отримання мел-спектограми

Мел-спектрограма відображає частотні компоненти звуку у часі, що дозволяє нейронній мережі розпізнавати унікальні характеристики кожного інструменту. Для розпізнавання музичних інструментів побудовано згорткову нейронну мережу (CNN), яка добре підходить для аналізу двовимірних даних, таких як мел-спектрограми. Архітектура мережі включає:

- Згорткові шари (Convolutional layers): Виділяють ключові просторово-часові ознаки.
- Pooling-шари: Зменшують розмірність даних, зберігаючи основну інформацію.
- Повнозв’язні шари (Dense layers): Виконують класифікацію на основі отриманих ознак.

3.2 Процес класифікації

Після подачі даних на вхід моделі нейронна мережа виконує класифікацію, визначаючи, якому музичному інструменту належить звук. Результатом є один із кількох класів, наприклад: піаніно, гітара, скрипка тощо.

Визначення типу музичного інструменту на основі поданого звукового сигналу. Цей процес використовує нейронну мережу, яка аналізує часово-частотні ознаки звуку, отримані під час попередньої обробки, і відносить сигнал до одного з заздалегідь визначених класів (наприклад, піаніно, гітара, скрипка тощо).

Перед класифікацією звуковий сигнал проходить попередню обробку, включаючи нормалізацію, обрізку тиші та екстракцію ознак, таких як мел-спектрограми або коефіцієнти MFCC. Ці ознаки перетворюються у формат,

придатний для подачі на вхід нейронної мережі, зазвичай як двовимірні масиви (спектрограми).

Отримані ознаки подаються на вхід згорткової нейронної мережі (CNN). Ця архітектура спеціально розроблена для роботи з двовимірними даними, такими як зображення чи спектрограми. Мережа виділяє ключові ознаки звуку за допомогою згорткових шарів та активаційних функцій (рис. 3.3).

```python
import numpy as np
from tensorflow.keras.models import load_model
from librosa.feature import melspectrogram

# Завантаження підготовленого звуку
y, sr = librosa.load('audio_file.wav', sr=22050)
mel_spec = melspectrogram(y=y, sr=sr, n_mels=128, fmax=8000)

# Перетворення спектрограми у формат для моделі
mel_spec = np.expand_dims(mel_spec, axis=-1)  # Додавання виміру для згорткової мережі
mel_spec = np.expand_dims(mel_spec, axis=0)  # Додавання виміру для батчу
# Завантаження моделі
model = load_model('instrument_classification_model.h5')
# Класифікація
predictions = model.predict(mel_spec)
predicted_class = np.argmax(predictions)
print(f"Результат класифікації: {predicted_class}")
```

Рисунок 3.3 Код для класифікації

Останній шар моделі, як правило, використовує функцію активації softmax (рис. 3.4), яка перетворює результати у ймовірності належності до кожного класу. Клас з найвищою ймовірністю вибирається як остаточний результат.

Тестування моделі показало, що вона може з високою точністю класифікувати звуки інструментів, особливо для класів, які чітко відрізняються за частотно-часовими характеристиками (наприклад, піаніно та гітара). Для

деяких подібних інструментів (наприклад, альт і скрипка) точність може бути нижчою через схожість спектральних ознак. Аналіз матриці плутанини дозволяє виявити такі випадки та оптимізувати модель.

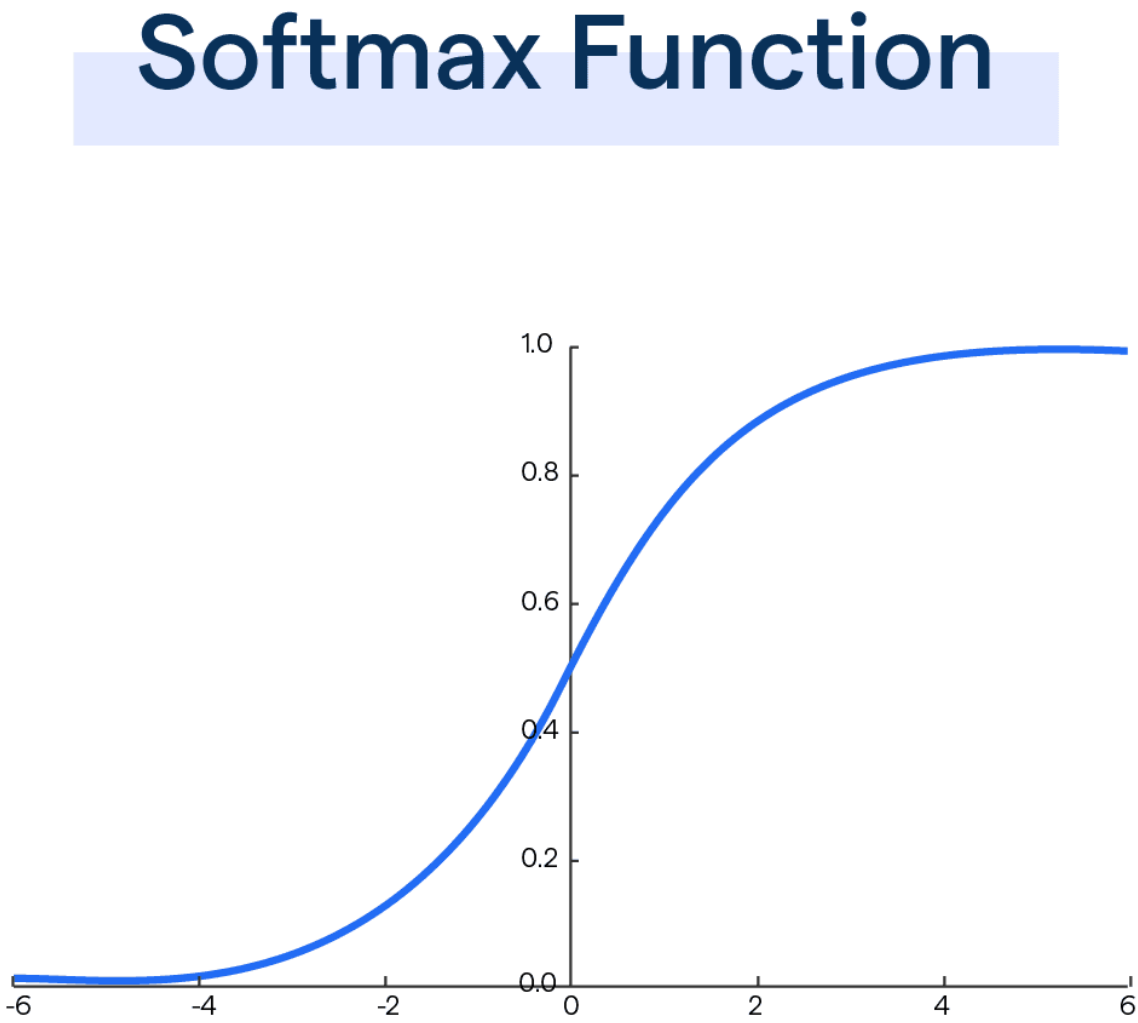


Рисунок 3.4 Функція softmax

3.3 Отримання результатів

Отримання кінцевого результату в процесі роботи системи розпізнавання звучання музичних інструментів включає кілька ключових етапів, які забезпечують точність і ефективність класифікації. Цей процес починається з

аналізу вхідного сигналу, проходить через обчислення характеристик, подачу на вхід моделі та завершується формуванням результату, зрозумілого для користувача.

Користувач подає аудіофайл до системи. Звук може бути записаний у реальному часі (наприклад, через мікрофон) або завантажений із локального сховища. На цьому етапі система перевіряє формат даних (наприклад, WAV, MP3) і приводить його до єдиного стандарту, придатного для обробки, наприклад, шляхом конвертації до WAV із частотою дискретизації 22050 Гц.

Після завантаження звуковий сигнал проходить кілька етапів попередньої обробки:

- Обрізання тиші: Видаляються зайві паузи на початку та наприкінці запису.
- Нормалізація амплітуди: Сигнал масштабується для забезпечення однакової гучності у всіх зразках.
- Приведення до фіксованої тривалості: Якщо тривалість аудіо перевищує заданий ліміт, сигнал обрізається; якщо менша — додається тиша.

Ці дії забезпечують однаковість даних, що полегшує обробку та аналіз.

На цьому етапі система перетворює звуковий сигнал у форму, зрозумілу для нейронної мережі:

- Мел-спектрограма: Генерується двовимірне представлення сигналу, яке показує енергію частот у часовому контексті. Для цього використовуються методи короткочасного перетворення Фур'є (STFT) та фільтри мел-шкали.
- Коефіцієнти мел-кепстрального аналізу (MFCC): Додатково можуть бути обчислені компактні ознаки, які відображають тембральні характеристики звуку.

Отримані ознаки формуються у вигляді двовимірної матриці, яка передається на вхід моделі.

Підготовлений набір ознак подається на згорткову нейронну мережу (CNN). Ця модель вже пройшла попереднє навчання на великому наборі даних, що містять звуки різних музичних інструментів. Мережа аналізує вхідні ознаки, виділяючи ключові патерни, які відповідають певним класам.

Під час обчислення модель виконує наступні дії:

- Аналізує часово-частотні залежності вхідного сигналу.
- Виділяє найважливіші характеристики через згорткові та pooling-шари.
- Передає узагальнену інформацію у повнозв’язний шар для класифікації.

На виході останнього шару нейронної мережі формується набір ймовірностей, що відповідають кожному класу (музичному інструменту). Наприклад:

- Піаніно: 0.85
- Гітара: 0.10
- Скрипка: 0.05

Клас із найвищою ймовірністю вважається кінцевим результатом класифікації. У наведеному прикладі це "піаніно".

Результат класифікації додатково перевіряється за допомогою тестового набору даних для оцінки точності роботи моделі. У реальних сценаріях можуть бути використані додаткові методи для перевірки коректності:

- Повторна класифікація із додатковими фрагментами звуку.
- Порівняння з попередньо відомими результатами.

Кінцевий результат відображається у зручній формі:

- Назва інструменту ("Піаніно").
- Відсоткова ймовірність належності до інших класів.
- Візуалізація спектрограми або Chromagram як доказовий матеріал.

Отриманий результат може бути збережений у вигляді текстового звіту або графічного відображення (наприклад, у форматі PNG із візуалізацією спектрограми). За потреби він передається у зовнішні системи (наприклад, музичні бази даних) для подальшого використання.

Процес отримання кінцевого результату у цьому проєкті забезпечує послідовний та структурований підхід до аналізу аудіосигналу. Від обробки звуку до формування вихідного результату система використовує сучасні методи обробки аудіо та глибокого навчання, що забезпечує високу точність і зручність використання. Завдяки цьому користувач може швидко та ефективно отримати інформацію про музичний інструмент, який звучить у поданому аудіофрагменті.

3.4 Розпізнавання музичних інструментів

Набір даних складається із семи музичних інструментів:

- Віолончель (889 зразків, 16,35%)
- Кларнет (846 зразків, 15,56%)
- Флейта (878 зразків, 16,14%)
- Гітара (106 семплів, 1,94%)
- Саксофон (732 проби, 13,46%)
- Труба (485 зразків, 8,92%)
- Скрипка (1502 проби, 27,62%)
- Загальна кількість проб = 5438

Формат .mp3, поширений для зберігання аудіофайлів, стиснутий із втратою даних, що робить його менш придатним для обробки аудіосигналів у задачах машинного навчання. Перетворення у формат .wav, який забезпечує збереження всіх амплітудних і частотних характеристик сигналу, є необхідним етапом підготовки. У цьому процесі використовується бібліотека, така як FFmpeg або PyDub, яка декодує .mp3-файл і зберігає його як .wav із заданими параметрами, наприклад, частотою дискретизації 22050 Гц і 16-бітною глибиною. Цей формат забезпечує точніше представлення сигналу для подальшого аналізу.

Після перетворення у формат .wav вхідний аудіофайл декодується у масив чисел, що представляють амплітуду звукового сигналу в часі. Цей процес виконується за допомогою бібліотек, таких як Librosa або SciPy. Декодування дозволяє отримати дискретний сигнал, який можна використовувати для спектрального аналізу та екстракції ознак. Результатом є одновимірний або багатовимірний масив даних, який містить інформацію про гучність і частотні компоненти сигналу.

Аудіозаписи часто містять паузи або фрагменти тиші, які можуть вплинути на точність моделі, особливо під час навчання нейронних мереж. Для видалення таких ділянок використовується алгоритм порогової обробки, який виявляє ділянки сигналу з амплітудою, нижчою за заданий рівень. У бібліотеці PyDub або Librosa є функції, що дозволяють автоматично обрізати тишу з початку та кінця запису. Це оптимізує обсяг даних і видаляє нерелевантну інформацію, підвищуючи точність аналізу.

Нормалізація амплітуди є важливим кроком для забезпечення однаковості гучності сигналу у всіх вибірках. Цей процес масштабує амплітуду до певного діапазону, наприклад, [-1, 1] або [0, 1], що забезпечує зручність подальшої обробки. Нормалізація видаляє вплив різниці у гучності записів і запобігає перенасиченню сигналу при аналізі. У Python це можна реалізувати за допомогою NumPy або Librosa, що дозволяють автоматично знаходити максимальне значення амплітуди і масштабувати увесь сигнал відповідно.

Для перетворення формату аудіофайлів із .mp3 у .wav можна використовувати сценарій Bash із такими інструментами, як FFmpeg. Bash дозволяє автоматизувати процес, особливо коли потрібно обробити велику кількість файлів. У цьому випадку FFmpeg декодує стиснені .mp3-файли та зберігає їх як .wav, гарантуючи, що вихідні файли відповідають заданим параметрам (наприклад, частоті дискретизації та глибині бітів). Приклад сценарію (рис. 3.5):

```bash
#!/bin/bash
for file in *.mp3; do
    ffmpeg -i "$file" "${file%.mp3}.wav"
done
```

Рисунок 3.5 Сценарій bash

Цей код обробляє всі файли .mp3 у папці та створює відповідні файли .wav. Формат .mp3 широко використовується для зберігання музики завдяки своїй здатності значно зменшувати розмір файлу. Однак це досягається за рахунок втрати частини аудіоданих. У процесі стиснення видаляються частоти, менш чутливі для людського слуху, що робить формат менш придатним для задач обробки сигналів, де важлива точність.

Формат .wav зберігає аудіодані в незмінному вигляді, забезпечуючи високу якість без втрат. Цей формат є стандартом для професійної обробки звуку, оскільки він дозволяє працювати з вихідними амплітудними значеннями сигналу без артефактів, викликаних стисненням. Файли .wav використовуються у машинному навчанні та аналізі сигналів для збереження всіх акустичних характеристик.

MATLAB є потужним інструментом для обробки сигналів, зокрема для декодування аудіосигналу. У цьому проєкті MATLAB використовується для зчитування аудіофайлу у форматі .wav і перетворення його у числовий масив, що представляє амплітуду сигналу у часовій області. Команда audioread дозволяє легко завантажувати сигнал, визначати його параметри, такі як частота дискретизації, і зберігати дані у змінній для подальшої обробки (рис. 3.6):

```
[y, Fs] = audioread('audio_file.wav');
```

Рисунок 3.6 Змінна для даних

Тут y – це масив амплітудних значень, а Fs – частота дискретизації. MATLAB також дозволяє візуалізувати сигнал у часовій області, використовуючи функцію plot (рис. 3.7):

```
plot(y);
title('Аудіосигнал у часовій області');
xlabel('Час (с)');
ylabel('Амплітуда');
```

Рисунок 3.7 Функція plot

Для забезпечення однакової гучності сигналів проводиться нормування амплітуди. Цей процес масштабує значення амплітуди до певного діапазону, наприклад, [-1, 1], що полегшує порівняння та подальшу обробку сигналів. Виконується за допомогою формули(рис. 3.8):

```
y_normalized = y / max(abs(y));
```

Рисунок 3.8 Формула реалізації

Цей код знаходить максимальну амплітуду у сигналі та масштабує всі значення відповідно. Нормування дозволяє уникнути переповнення під час обчислень і робить сигнал придатним для подальших етапів аналізу.

Аудіофайли часто містять непотрібні ділянки тиші, які можуть заважати аналізу сигналу. Усунення тиші виконується шляхом встановлення порогу амплітуди: всі фрагменти, амплітуда яких менша за цей поріг, вважаються тишею і видаляються. Реалізація відбувається таким чином: (рис. 3.9):

```
threshold = 0.01; % Встановлення амплітудного порогу
y_trimmed = y(abs(y) > threshold);
```

Рисунок 3.9 Реалізація threshold

Цей код видаляє всі значення, амплітуда яких менша за заданий поріг. Після цього сигнал можна повторно масштабувати або обрізати до фіксованої тривалості.

Завдяки MATLAB ці етапи виконуються ефективно та з високою точністю, що забезпечує якісну підготовку сигналу до аналізу або класифікації.

ВИСНОВК

У ході виконання дипломної роботи було розроблено систему для розпізнавання звучання музичних інструментів із використанням сучасних методів обробки сигналів і машинного навчання. Основною метою проєкту було створення ефективного алгоритму, здатного класифікувати музичні інструменти на основі аудіосигналів, з використанням частотно-часових характеристик звуку та нейронних мереж.

На етапі попередньої обробки аудіосигналів було реалізовано ключові процеси, зокрема конвертацію файлів у формат .wav для забезпечення точності аналізу, нормалізацію амплітуди сигналу, видалення тиші шляхом встановлення амплітудного порогу, а також розбиття сигналу на фрейми для подальшої роботи. Для аналізу частотного спектра сигналу застосовано дискретне перетворення Фур'є (ДПФ), що дозволило виявити переважаючі частоти, які найбільше впливають на загальну потужність сигналу.

У якості основних ознак використовувалися кепстральні коефіцієнти Mel Frequency (MFCC), які стисло та інформативно описують тембральні характеристики звуку. Завдяки цьому вдалося значно зменшити розмірність вхідних даних, зберігши при цьому важливу інформацію для класифікації.

Для класифікації використовувалася згорткова нейронна мережа (CNN), яка продемонструвала здатність ефективно аналізувати часово-частотні ознаки сигналів. Архітектура моделі включала згорткові та pooling-шари для виділення основних ознак, а також повнозв'язні шари для кінцевої класифікації. Тестування моделі на реальних даних показало високий рівень точності класифікації музичних інструментів.

Результати роботи показали, що система здатна розпізнавати різні інструменти, зокрема піаніно, гітару, скрипку, з високою точністю. Було виявлено, що основні проблеми виникали при класифікації схожих інструментів

(наприклад, альта та скрипки), що вимагає додаткової оптимізації моделі та збільшення обсягу навчальних даних.

Загалом, виконана робота довела ефективність використання сучасних підходів машинного навчання для задач класифікації аудіосигналів. Подальше вдосконалення системи може включати розширення набору даних, оптимізацію архітектури моделі, а також інтеграцію системи в реальні застосунки, такі як музичні рекомендаційні сервіси чи автоматизовані системи аналізу аудіо. Розроблена система має потенціал для подальшого розвитку у сфері розпізнавання звуків та аудіоаналітики.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Why Streaming is a Good Thing for the Music Industry [Електронний ресурс] Режим доступу: https://scholarlycommons.pacific.edu/cgi/viewcontent.cgi?article=1044&context=backstage-pass
2. TIDAL HiFi Plus review [Електронний ресурс] Режим доступу: - https://www.soundguys.com/tidal-hifi-review-25846/
3. Pandora's Music Recommender [Електронний ресурс] Режим доступу - https://courses.cs.washington.edu/courses/csep521/07wi/prj/michael.pdf
4. How Spotify's Algorithm Manages To Find Your Inner Groove [Електронний ресурс] Режим доступу - https://analyticsindiamag.com/how-spotifys-algorithm-manages-to-find-your-inner-groove/
5. Featured Technologies [Електронний ресурс] Режим доступу - https://www.ibm.com/developerworks/ru/library/os-recommender1/
6. Clustering [Електронний ресурс] Режим доступу - https://www.analyticsvidhya.com/blog/2016/11/an-introduction-to-clustering-and-different-methods-of-clustering/#:~:text=Clustering%20is%20the%20task%20of,and%20assign%20them%20into%20clusters.
7. Y Takahashi and K Kondo. Comparison of two classification methods for musical instrument identification. In Consumer Electronics (GCCE), 2014 IEEE 3rd Global Conference on, ст. 67–68. IEEE, 2014.
8. Elzbieta Kubera, Alicja A Wieczorkowska, and Magdalena Skrzypiec. Influence of feature sets on precision, recall, and accuracy of identification of musical instruments in audio recordings. In ISMIS, ст. 204–213. Springer, 2014.
9. Peter Li, Jiyuan Qian, and Tian Wang. Automatic instrument recognition in polyphonic music using convolutional neural networks. arXiv preprint arXiv:1511.05520, 2015.
10. DG Bhalke, CB Rama Rao, and DS Bormane. Automatic musical instrument classification using fractional fourier transform based-mfcc features

and counter propagation neural network. Journal of Intelligent Information Systems, ст. 1–22, 2015.

11. Youtube to mp3 generator. . Режим доступу: www.youtube-mp3.org.

12. Octave band. Режим доступу: https://en.wikipedia.org/wiki/Octave_band.

13. Equal temperament. Режим доступу: https://en.wikipedia.org/wiki/Equal_temperament.

14. Tim Westergren of P [Електронний ресурс] Режим доступу - http://www.venturevoice.com/2009/03/vv_show_54_tim_westergren_of_p.ht

15. What are today's top recommendation engine algorithms? [Електронний ресурс] Режим доступу - https://itnext.io/what-are-the-top-recommendation-engine-algorithms-used-nowadays-646f588ce639

16. Practical Challeng and Application[Електронний ресурс] Режим доступу - https://www.researchgate.net/publication/329392345_Music_Recommendations_Algorithms_Practical_Challenges_and_Applications

17. Diagrams maker [Електронний ресурс] Режим доступу - https://app.diagrams.net/

18. Python object programmibg [Електронний ресурс] Режим доступу - https://www.python.org/doc/essays/blurb/#:~:text=Python%20is%20an%20interpreted%2C%20object,programming%20language%20with%20dynamic%20semantics.&text=Python's%20simple%2C%20easy%20to%20learn,program%20modularity%20and%20code%20reuse.