

Токар Максим Олександрович

Допускається до захисту:
завідувач кафедри інформаційних
технологій, канд. техн. наук, доцент
_____ Оксана Зелінська

«__» _____ 20__ р.

**РОЗРОБКА І ДОСЛІДЖЕННЯ МЕТОДІВ ОБРОБКИ
МУЗИЧНИХ АКОРДІВ НА ОСНОВІ НЕЙРОМЕРЕЖ**

Спеціальність 122 Комп'ютерні науки

Кваліфікаційна (магістерська) робота
(відповідно до стандарту спеціальності та ОП)

Науковий керівник:
Р.М. Бабаков, професор кафедри
інформаційних технологій,
докт. техн. наук, доцент

(підпис)

Оцінка: _____ / _____ / _____
(бали/за шкалою ЄКТС/за національною шкалою)

Голова ЕК: _____
(підпис)

АНОТАЦІЯ

Токар М.О. Розробка системи розпізнавання акордів на гітарі з використанням інтелектуальних технологій. Спеціальність 122 "Інформаційні технології", Освітня програма "Машинне навчання та обробка сигналів". Донецький національний університет імені Василя Стуса, Вінниця, 2023.

У даній кваліфікаційній (магістерській) роботі досліджено сучасні підходи до розпізнавання акордів на гітарі за допомогою інтелектуальних технологій. Проведено аналіз і порівняння існуючих методів та бібліотек для обробки аудіо-сигналів. Розроблено систему розпізнавання акордів, яка дозволяє гітаристам та музикантам автоматично визначати акорди, відтворені на гітарі, з використанням навчених моделей машинного навчання.

Ключові слова: розпізнавання акордів, гітара, аудіо-сигнали, машинне навчання, інтелектуальні технології, обробка сигналів.

ABSTRACT

Tokar M.O. Development of a Guitar Chord Recognition System Using Machine Learning. Specialty 122 "Information Technology", Educational Program "Machine Learning and Signal Processing". Vasyl' Stus Donetsk National University, Vinnytsya, 2023.

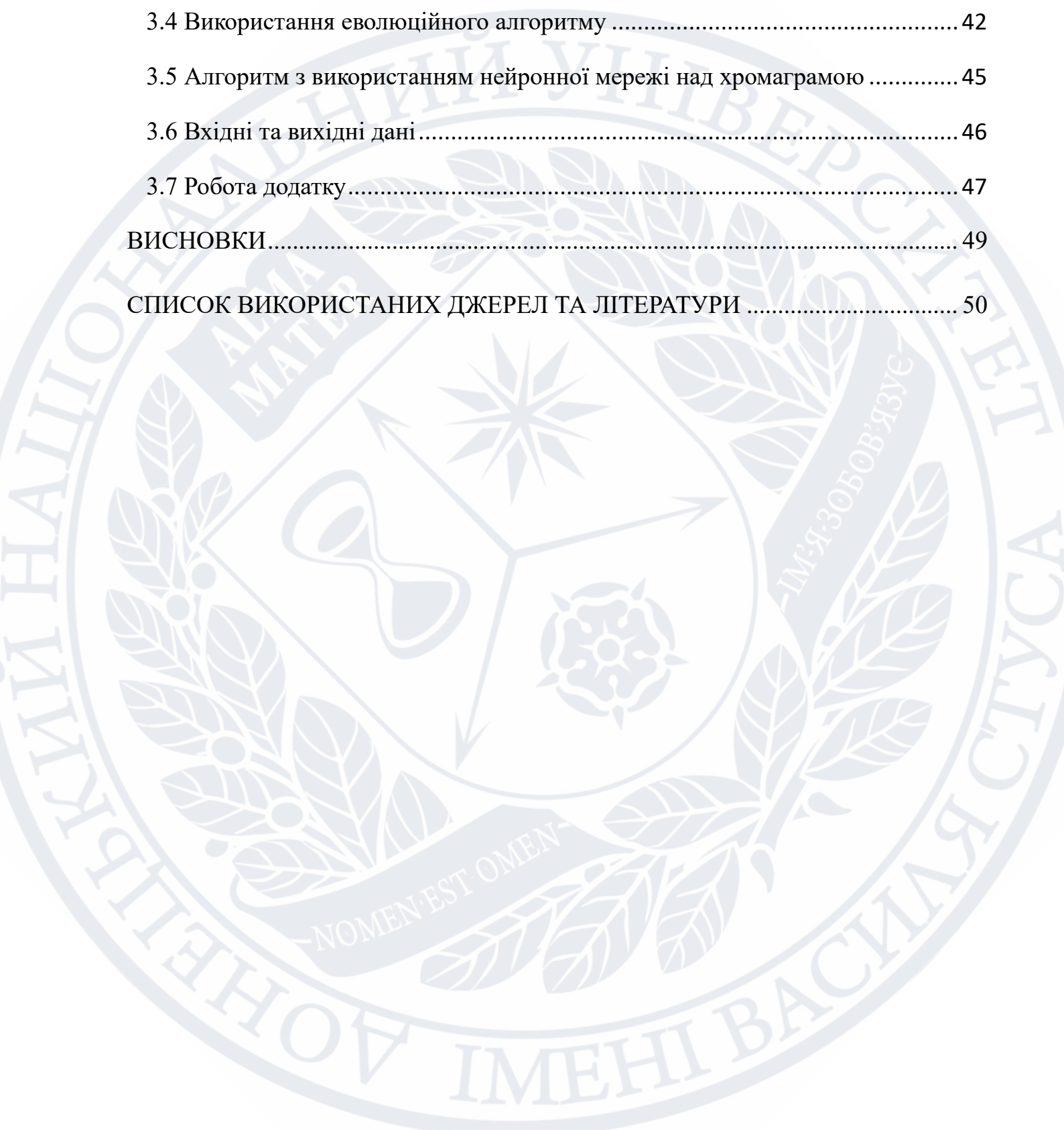
This qualification (master's) work explores modern approaches to guitar chord recognition using intelligent technologies. It conducts an analysis and comparison of existing methods and libraries for audio signal processing. The research results in the development of a chord recognition system that enables guitarists and musicians to automatically identify chords played on the guitar using trained machine learning models.

Keywords: chord recognition, guitar, audio signals, machine learning, intelligent technologies, signal processing.

ЗМІСТ

ВСТУП.....	6
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ.....	9
1.1 Напрямок дослідження	9
1.2 Аналіз аналогів.....	11
1.3 Аналіз проблем та актуалізація рішень.....	17
Висновки до розділу 1	19
РОЗДІЛ 2. АНАЛІЗ ІНСТРУМЕНТІВ ТА ТЕХНОЛОГІЙ РОЗРОБКИ.....	20
2.1 Використання ШІ для обробки звуку	20
2.2 Використання технологій для обробки аудіоданих	21
2.3 Використання нейронних мереж для розпізнавання акордів	22
2.4 Інтеграція та оптимізація системи розпізнавання акордів	23
2.5 Штучні Нейронні Мережі в Задачі Розпізнавання.....	24
2.6 Застосування теорії імовірнісного аналізу прихованих компонентів.....	25
2.7 Автокореляційне Відстеження Основного Тону.....	26
2.8 Python.....	27
2.8.1 Загальний Огляд.....	27
2.8.2 Використання Python в Розробці Системи Розпізнавання Акордів	28
2.8.3 Бібліотека Librosa	30
2.8.4 Бібліотека Scikit-Learn	31
Висновки до розділу 2	33
РОЗДІЛ 3. РОЗРОБКА ТА СТВОРЕННЯ ЗАСТОСУНКУ	34
3.1 Методи досліджень	35

3.2 Розробка алгоритму розпізнавання звуків	39
3.3 Алгоритм застосування Прихованих Марковських Моделей (ПММ).....	41
3.4 Використання еволюційного алгоритму	42
3.5 Алгоритм з використанням нейронної мережі над хромаграмою	45
3.6 Вхідні та вихідні дані.....	46
3.7 Робота додатку.....	47
ВИСНОВКИ.....	49
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ТА ЛІТЕРАТУРИ	50



ВСТУП

Актуальність

Музика - це важлива складова нашого культурного спадку, і розпізнавання музичних звуків може бути корисним в різних сферах, включаючи пошукові системи, анування медіаконтенту та створення плейлистів. Ця задача належить до класу Music Information Retrieval (MIR) і має широкий спектр застосувань.

Мета та завдання дослідження

Метою нашої роботи є розробка алгоритму розпізнавання музичних звуків з використанням штучних нейронних мереж, який забезпечуватиме високу точність та продуктивність. Наші завдання включають:

- Дослідження принципів побудови алгоритмів розпізнавання та нейронних мереж.
- Вибір програмних засобів для вичленення музичних звуків із аудіозапису.
- Проектування нейронної мережі прямого поширення та визначення способу відображення отриманих даних.
- Представлення архітектури нейронної мережі з використанням програмних засобів.
- Тестування розробленої моделі та порівняння результатів з іншими методами розпізнавання.
- Реалізація програмного продукту на основі розробленого алгоритму.

Об'єкт та предмет дослідження

Об'єктом нашого дослідження є звук музичних інструментів та їх характеристики. Ми досліджуємо амплітудно-частотні характеристики звуку, зокрема, як звучання окремого музичного інструменту, так і звуку інструментів у складних музичних композиціях.

Методи дослідження

Для досягнення наших цілей ми використовуємо наступні методи дослідження:

- Аналіз результатів попередніх наукових робіт в області розпізнавання музичних звуків.
- Дослідження фізичних характеристик звуку та їх вплив на розпізнавання.
- Розробка та вибір оптимального методу розв'язання задачі розпізнавання музичних звуків з використанням нейронних мереж прямого поширення.

Практична цінність

Результати наших досліджень можуть бути корисними для подальшого вдосконалення методів розпізнавання музики та сегментації складних музичних сигналів. Наша робота може знайти застосування у різних сферах, включаючи розвиток музичних додатків та інших медіа-технологій.

У сучасному світі із широким розповсюдженням високошвидкісних інтернет-з'єднань спостерігається виразний ріст обсягу доступної аудіоінформації в локальних та глобальних комп'ютерних мережах. Ця тенденція створює потребу у розвитку якісних алгоритмів семантичного аналізу аудіодокументів. Особливу вагомість набуває розпізнавання музичних звуків, оскільки це може створити фундамент для створення аудіопошукових систем, подібних до пошукових систем для текстової інформації. Автоматичне розпізнавання музичних звуків дозволяє швидко та ефективно аналізувати музичні файли на рівні сприйняття, подібно до людського сприйняття.

У сучасному контексті існуючі технології та алгоритми обробки сигналів сприяють автоматизації багатьох аспектів розпізнавання музичних звуків, зокрема визначення їх висоти. Однак цей процес є трудомістким і досить залежним від якості початкових даних. Звукові файли, які підлягають розпізнаванню, повинні бути належним чином оброблені, враховуючи віконні

функції та усунення шуму. Крім того, гучність, тембр інструменту, на якому відтворюється мелодія, а також специфічні прийоми гри на ньому можуть вплинути на результат розпізнавання.

Однією з ключових задач є визначення висоти звуків у одноголосних мелодіях (single-pitch estimation). Ця задача є складною, і точність розпізнавання великої міри залежить від умов роботи та характеристик вхідних даних. Ефективність алгоритмів у багатьох випадках полягає в їх здатності фільтрувати шуми, виявляти паузи в музичних фрагментах та правильно інтерпретувати прийоми гри на музичних інструментах.

Значним викликом є також завдання розпізнавання багатьох голосів (multi-pitch estimation), наприклад, визначення акордів у музичних композиціях. На сьогоднішній день точність цих результатів не завжди перевищує 70%, що робить роботи в даному напрямку дуже актуальними та важливими.

У даній роботі ми представимо розроблений алгоритм розпізнавання музичних звуків, заснований на використанні нейронної мережі. Ми дослідимо точність та ефективність цього алгоритму при аналізі музичних фрагментів, які можуть бути викликані певні складності в процесі розпізнавання. Застосування нейронної мережі обумовлено недостатньою ефективністю існуючих алгоритмів single-pitch estimation при роботі з музичними фрагментами, що містять складні ритмічні та мелодійні відтінки.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗИ

1.1 Напрямок дослідження

Дослідження в галузі пошуку музичної інформації (MIR) спрямоване на розробку методів пошуку та організації великих обсягів музичної інформації відповідно до конкретних запитів. Ця галузь набула особливої актуальності через велику кількість цифрової музики, яка доступна зараз, та популярність цифрових музичних сервісів. Напередодні цього, протягом останніх чотирьох років, було проведено низку конференцій та заходів, присвячених музичному пошуку інформації, таких як Міжнародна конференція з пошуку музичної інформації (ISMIR), а також різноманітні симпозіуми та семінари, що об'єднують різноманітну спільноту дослідників з різних галузей, включаючи бібліотекознавців, інформатиків, музикантів, інженерів-електроніків і музикознавців. Крім того, власники та постачальники медіа-контенту, такі як Philips, Sony та Apple, активно беруть участь у наукових дослідженнях у цій галузі, а бібліотеки намагаються впровадити підтримку MIR у свої онлайн-сервіси.

Прості системи MIR приймають текстові запити користувачів, наприклад "пісні Девіда Боуї". В цих випадках текст порівнюється з текстовою інформацією про альбоми та композиції. Проте існує потреба в системах, які можуть обробляти "музичні" запити, такі як мелодії або аудіофрагменти. Ця робота фокусується саме на останньому випадку.

Основна мета запитів на прикладах полягає в тому, щоб знайти музичні твори великої музичної колекції на основі їх схожості з прикладним аудіодокументом. Здатність обробляти запити на прикладах важлива для систем MIR. Ця задача ставить численні виклики, включаючи обчислювальні проблеми, необхідність створення відповідного тестового середовища і вибір відповідного аудіопредставлення для запиту та музичної колекції. Вибір аудіопредставлення визначає подібність, яку може виявити система. Існують два загальних підходи

до вибору аудіопредставлення: один спрямований на низькорівневу (акустичну) подібність, інший – на вищорівневу подібність, таку як ноти та мелодії.

Дослідження музичної інформації привертає значну увагу і має довгу історію, що свідчить про спроби використовувати комп'ютери для створення музики ще з часів винаходу перших комп'ютерів. Сучасні результати досліджень музичної інформації використовуються в таких областях, як синтез музики, обробка сигналів, транскрипція, аналіз, розуміння, пошук, рекомендація, класифікація, композиція, аранжування, написання пісень та багато інших.

Сприяючи перегляду аудіофайлів, взаємодіючи з музикою та допомагаючи користувачам знаходити та рекомендувати музику, дослідження музичної інформації має широкий спектр важливих аспектів, які включають обробку сигналів, ідентифікацію, аналіз, розуміння, пошук, класифікацію та розподіл музичної інформації. В цьому контексті існують також нерозрішені завдання, які потребують подальшого дослідження, і ця галузь є важливим джерелом для нових наукових тем.

Варто зауважити, що історично важливість досліджень музичної інформації не була визнана до 1990-х років. Проте з поширенням цифрових технологій та комп'ютерів, а також з появою можливості слухати музику на комп'ютерах в повсякденному житті, важливість цієї галузі вирізняється. Зараз вона вважається серйозною науковою галуззю, і все більше дослідників приєднуються до неї з усього світу. Це поле зазнало значних змін протягом останнього десятиріччя, і раніше поширені недоліки та невинуваті погляди на нього тепер визнаються як дослідницькі проблеми, які потребують уваги та розв'язання.

Надалі очікується зростання попиту на нові інтерфейси для слухання музики, пошуку та рекомендацій. В академічному плані, однією з головних причин участі в цій галузі є розуміння складних музичних сигналів, що передають структуру і зв'язок, які взаємодіють між собою. Незважаючи на значний прогрес, що було досягнуто в дослідженнях музичної інформації,

залишаються низка захоплюючих вирішених завдань, які стимулюють нові ідеї та надихають на подальший розвиток цієї галузі.

У цій роботі розглядаються деякі з цих проблем, які можуть зробити дослідження музичної інформації ще привабливішими та впливовими в майбутньому. Зазначимо, що деякі обговорення в цьому документі представлені провокативно для спонукання до обговорення і сприяння виникненню нових ідей.

1.2 Аналіз аналогів

У сфері програмного забезпечення існує різноманітність аналогів. Умовно всі продукти можна розділити на групи: безкоштовні, умовно-безкоштовні та платні. До безкоштовних, які доступні в мережі безкоштовно, належать різні бібліотеки, які використовують алгоритмічні методи або нейронні мережі для аналізу звуку. Умовно-безкоштовні продукти базуються на наукових розробках, але не надають відкритий доступ до свого вихідного коду. Платні програми не розкривають інформацію про використовувані наукові праці і недоступні для перегляду свого програмного коду користувачам системи.

Всі ці сервіси мають різне призначення, відповідно до функціоналу, який вони надають. Більшість систем, які присутні на ринку, служать засобами для розпізнавання акордів у музичних записах. Ці записи можуть бути використані для навчання гри на музичному інструменті в різних форматах. Більшість таких сервісів також надають можливість вивчати популярні пісні відомих виконавців, але вони не здатні розпізнавати, що саме грає користувач.

Fret Trainer - це додаток для смартфонів і планшетів, спрямований на навчання гри на гітарі. Він зосереджується на вправах для покращення техніки гри на гітарі та навчанні гітариста розпізнавати ноти на грифі і зіграні акорди.

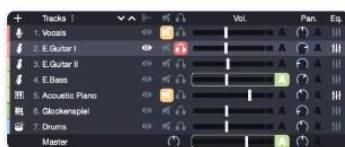
Плюси:

- Спеціалізований підхід до навчання гри на гітарі.



Virtual effects chain

Reverb, chorus, delay, compressor, EQ, tremolo... Guitar Pro comes with dozens of legendary pedals to help you fine-tune your sound.



Mixing each track and mastering your project

Apply mixing and panning effects to each track to create a consistent mix for your song. Mastering options also allow you to improve the sound of your tabs.



Adding an audio track to your project

The audio track is an excellent way to make your scores more realistic. For example, you can ask your singer or drummer to record himself and then import his audio file into Guitar Pro to create a preprod of your future song.

Рис. 1.2

Simply Guitar - це додаток для мобільних пристроїв, спрямований на навчання гри на гітарі. Він пропонує інтерактивні уроки, в яких можна вивчати акорди, техніку гри та виконувати вправи.

Плюси:

- Інтерактивні та користувацькі уроки, призначені для початківців.
- Зручний інтерфейс та візуалізація акордів.
- Поступовий підхід до навчання.

Мінуси:

- Можливості для досвідчених гітаристів дещо обмежені.



Рис. 1.3

Yousician - це універсальна музична платформа, яка пропонує навчальні уроки для гри на гітарі, клавішних, бас-гітарі та співу. Вона включає в себе ігри, завдання та відстежує прогрес користувачів.

Плюси:

- Широкий спектр інструментів та музичних стилів.
- Інтерактивні завдання та ігри для навчання.
- Можливість вибору інструменту та рівня складності.

Мінуси:

- Деякі функції доступні тільки в платній версії.

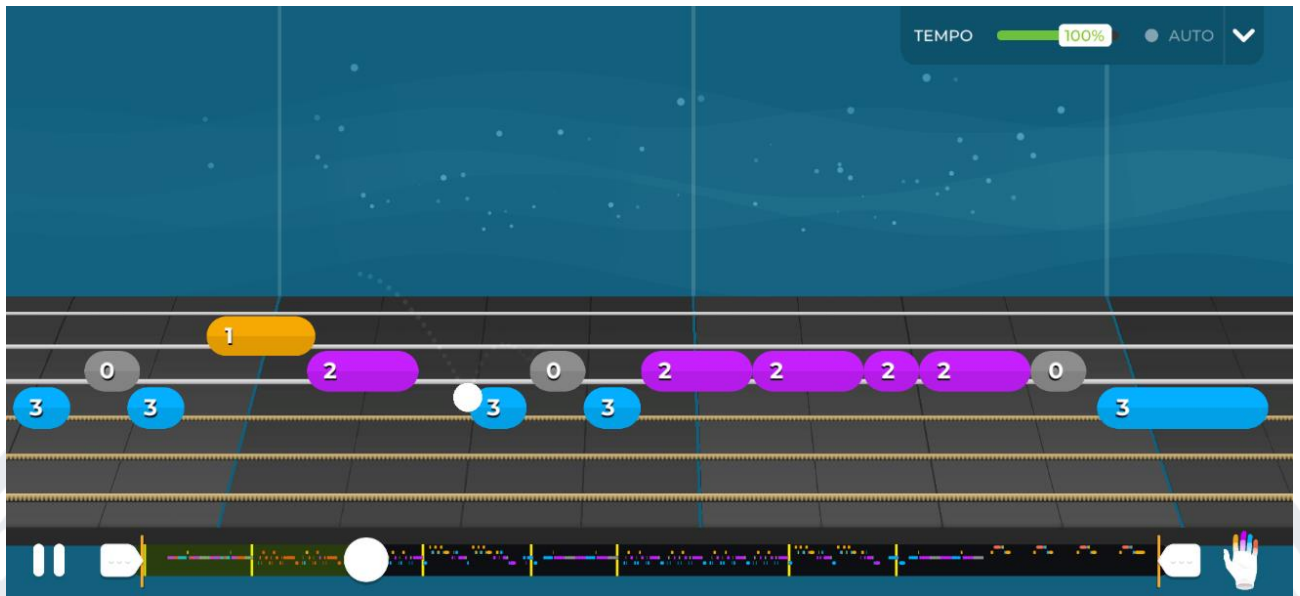


Рис. 1.4

Chordify - це веб-сервіс, який дозволяє розпізнавати акорди у музичних записах та надавати користувачеві відповідні акорди для гри на гітарі або інших інструментах.

Плюси:

- Простий інтерфейс та можливість розпізнавати акорди.
- Можливість вибору пісні і отримання акордів для гри.
- Різноманітність жанрів та пісень в базі даних.

Мінуси:

- Обмежений безкоштовний доступ, більший функціонал - у преміум-версії.
- Не надає нотні партитури для пісень, тільки акорди.

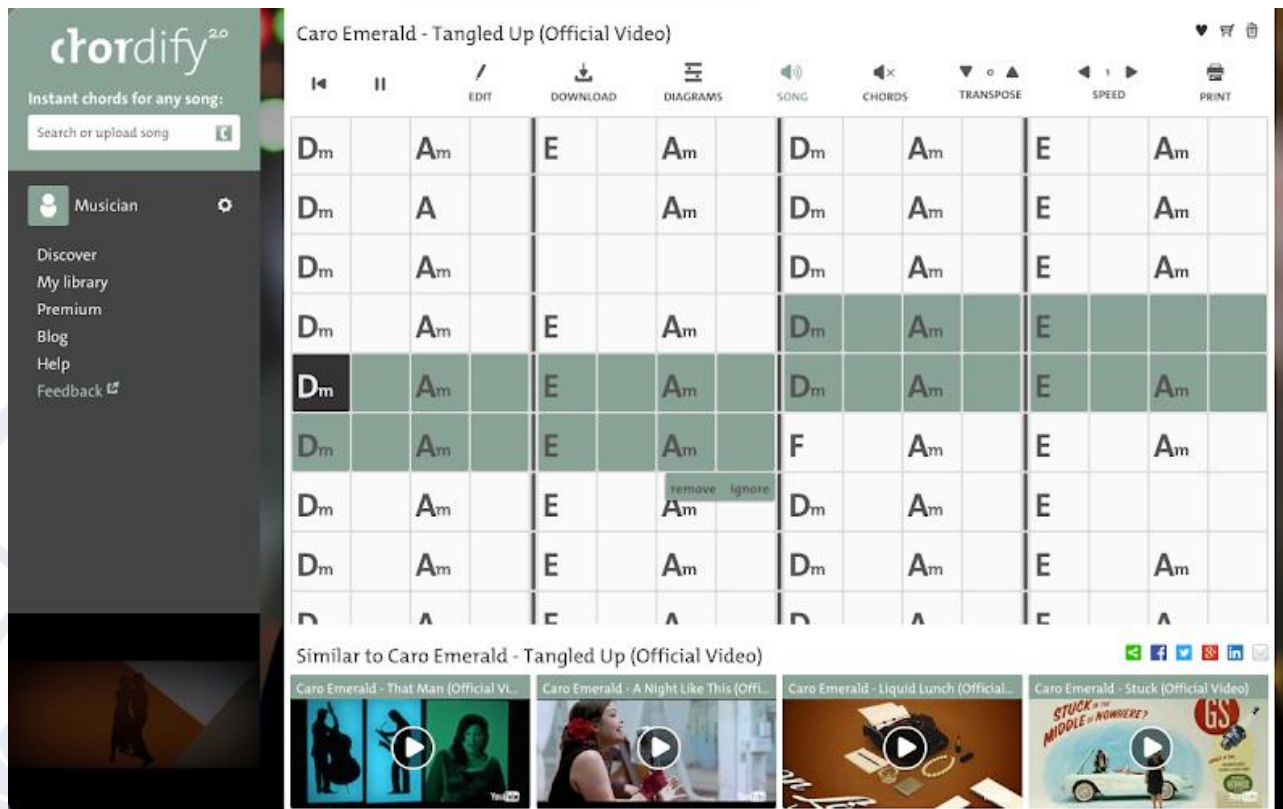


Рис. 1.5

Timbro - це мобільний додаток для навчання гри на гітарі, який пропонує різноманітні пісні для практики. Додаток включає аналіз зіграних акордів та надає корисні статистичні дані для покращення гри.

Плюси:

- Багато доступних пісень для гри на гітарі.
- Аналіз акордів та надання статистики.
- Доступний умовно-безкоштовний варіант.

Мінуси:

- Більшість пісень є платними.
- Обмежений функціонал у безкоштовній версії.

TIMBRO.



THE NEW WAY TO LEARN GUITAR.



Рис. 1.6

1.3 Аналіз проблем та актуалізація рішень

Аналіз проблем та актуалізація рішень у сфері розпізнавання акордів має великий інтерес для спільноти дослідників музичної інформатики (MIR). Гармонічний зміст музики є важливою характеристикою, яка може бути використана для різноманітних завдань, таких як створення нотних зошитів або визначення ключа композиції.

Важливим етапом у розпізнаванні акордів є вилучення особливостей з аудіо. Багато систем розпізнавання акордів використовують хромаграму, яка кодує інформацію про гармонічний зміст звуку в часі. Хромаграма створюється на основі аналізу частотно-часового подання аудіосигналу, такого як короткочасне перетворення Фур'є (STFT) або перетворення з постійним кроком (CQT). Проте хромаграма має свої обмеження, такі як відсутність інформації про октави і обмежену можливість розрізнення акордів з однаковими класами висоти, але різними басовими нотами.

Ще однією важливою проблемою є взаємодія між різними компонентами системи розпізнавання акордів. Багато систем використовують загальну архітектуру, що складається з вилучення особливостей, узгодження шаблонів і декодування послідовності акордів. Проте недостатньо досліджень, які вивчають вплив різних підходів на кожен з цих етапів та їх взаємодію.

Дослідження показали, що фільтрування грає важливу роль у розпізнаванні акордів, але інформація про музичний контекст у матриці переходів акордів може бути обмеженою через необхідність забезпечення безперервності оцінок. Також було виявлено, що використання складних моделей акордів може бути компенсоване вибором покращених функцій. Розробники систем розпізнавання акордів продовжують вдосконалювати та досліджувати рішення для цих проблем з метою покращення точності та продуктивності систем.

Розпізнавання акордів - це важливий етап у вивченні та аналізі музики, і вирішення цих проблем сприятиме розвитку систем музичної обробки інформації та музичної аналітики.

Висновки до розділу 1

У розділі 1 нашого аналітичного звіту ми розглянули ландшафт програмного забезпечення для розпізнавання акордів в музиці та порівняли різні аналоги цих сервісів. Ми виділили безкоштовні, умовно-безкоштовні та платні інструменти та розглянули їхні можливості, включаючи функціонал та способи взаємодії з музичними інструментами.

У контексті розпізнавання акордів, ми дійшли висновку, що багато сервісів використовують схожі підходи до аналізу музики. Вони використовують хромаграми, що включають інформацію про кольори та акорди в часовому контексті. Також, багато з цих інструментів надають можливість гейміфікації для покращення процесу навчання гри на музичних інструментах.

Проте, важливо відзначити, що розпізнавання акордів - це складний процес, і немає єдиного ідеального рішення. Деякі проблеми включають в себе складнощі в розпізнаванні акордів, що базуються на контексті, та відсутність інформації про точний спосіб гри інструментом.

У цьому розділі ми також наголосили на важливості подальших досліджень у сфері розпізнавання акордів та можливостей покращення існуючих систем. Це розділ дає читачу вступний огляд стану справ у галузі розпізнавання акордів та акцентує на ключових аспектах, які потребують подальшого вивчення і розвитку.

РОЗДІЛ 2. АНАЛІЗ ІНСТРУМЕНТІВ ТА ТЕХНОЛОГІЙ РОЗРОБКИ

2.1 Використання ШІ для обробки звуку

В цьому підрозділі буде розглянуто використання сучасних методів обробки звуку з використанням штучних інтелектуальних систем (ШІ). Даний розділ включає в себе такі аспекти:

Розпізнавання голосу

Розпізнавання голосу - це процес перетворення звукових сигналів в текстову форму. В даному контексті розглянемо використання систем розпізнавання голосу для ідентифікації акордів та нот в музичних композиціях. Опис роботи інструментів для розпізнавання голосу та їхні переваги та недоліки.

Обробка аудіосигналів

Обробка аудіосигналів - це ключовий аспект при аналізі та обробці звуку. В цьому підрозділі розглянемо різні техніки та бібліотеки для обробки аудіоданих, такі як фільтрація шуму, амплітудна модуляція та інші методи обробки сигналів.

Машинне навчання та нейронні мережі

Машинне навчання та нейронні мережі використовуються для створення моделей розпізнавання та аналізу аудіоданих. У цьому підрозділі описані алгоритми машинного навчання, включаючи методи класифікації, регресії та кластеризації, а також застосування нейронних мереж для аналізу музичних сигналів.

Інструменти для розробки та реалізації

В останньому підрозділі розглянемо популярні інструменти і середовища розробки, які використовуються для створення програмних рішень у галузі обробки звуку та музичного аналізу. До таких інструментів відносяться мови програмування, бібліотеки та фреймворки, що допомагають розробникам створювати програми для аналізу та обробки аудіоданих.

2.2 Використання технологій для обробки аудіоданих

У цьому підрозділі розглянемо технології та інструменти, спрямовані на обробку аудіоданих для подальшого аналізу та розпізнавання музичних акордів. Цей підрозділ включає в себе такі аспекти:

Формати аудіофайлів

Для роботи з аудіоданими важливо розуміти різні формати аудіофайлів, такі як WAV, MP3, FLAC, і багато інших. В цьому підрозділі ми розглянемо популярні формати аудіофайлів та їхні особливості.

Інструменти для обробки та аналізу аудіо

Розробка системи розпізнавання акордів потребує використання інструментів для обробки та аналізу аудіоданих. У цьому підрозділі розглянемо популярні бібліотеки та програми для обробки та аналізу звуку, такі як librosa, Audacity [38], та інші.

Спектральний аналіз

Спектральний аналіз є ключовим етапом при аналізі аудіоданих для розпізнавання акордів. У цьому підрозділі розглянемо методи спектрального аналізу, такі як швидке перетворення Фур'є (FFT), кореляція та інші способи виділення музичних характеристик.

Бібліотеки для аналізу та обробки аудіо

Для зручності розробки системи розпізнавання акордів використовуються різні бібліотеки та фреймворки. В цьому підрозділі розглянемо популярні бібліотеки для аналізу та обробки аудіоданих, такі як NumPy [9, 34], SciPy [10], та інші.

2.3 Використання нейронних мереж для розпізнавання акордів

В цьому підрозділі розглянемо технології та методи використання нейронних мереж для розпізнавання музичних акордів. Цей підрозділ включає в себе такі аспекти:

Нейронні мережі для обробки аудіо

Для розпізнавання акордів в аудіоданих використовуються різні архітектури нейронних мереж. У цьому підрозділі ми розглянемо типи нейронних мереж, такі як згорткові нейронні мережі (CNN), рекурентні нейронні мережі (RNN), а також комбіновані моделі, що використовуються для обробки аудіоданих.

Популярні фреймворки для нейронних мереж

Розробка та навчання нейронних мереж вимагає використання спеціалізованих фреймворків та бібліотек. У цьому підрозділі розглянемо популярні фреймворки для роботи з нейронними мережами, такі як TensorFlow, PyTorch [12, 16], та Keras [8, 13, 16, 37], та їхню використання для розпізнавання акордів.

Перенавчання та підбір гіперпараметрів

Ефективне навчання нейронних мереж для розпізнавання акордів вимагає оптимізації гіперпараметрів та уникнення перенавчання. У цьому підрозділі розглянемо методи підбору гіперпараметрів та стратегії для уникнення перенавчання в контексті розпізнавання музичних акордів.

Оцінка результатів та метрики

Для оцінки якості розпізнавання музичних акордів використовуються різні метрики та методи. У цьому підрозділі розглянемо популярні метрики, такі як точність, відновлення, F-міра, та інші, і розповімо про їх використання для оцінки результатів навчання нейронних мереж.

2.4 Інтеграція та оптимізація системи розпізнавання акордів

В цьому підрозділі ми розглянемо, як інтегрувати розроблену систему розпізнавання акордів на основі нейронних мереж та інших методів в реальні музичні додатки та сервіси. Цей розділ включає в себе такі ключові аспекти:

Інтеграція в музичні додатки

Розглянемо процес інтеграції системи розпізнавання акордів у музичні додатки, такі як музичні програвачі, музичні редактори або мобільні додатки для музикантів. Пояснимо, як систему можна впровадити в існуючі музичні технології та інфраструктуру.

Оптимізація продуктивності

Для ефективного розпізнавання акордів в реальному часі важливо оптимізувати продуктивність системи. У цьому підрозділі ми розглянемо методи оптимізації, такі як апаратне прискорення, оптимізація обчислень та робота з великими обсягами даних.

Створення користувацького інтерфейсу

Якщо система розпізнавання акордів призначена для використання музикантами або іншими користувачами, то важливо розробити зручний користувацький інтерфейс. У цьому підрозділі розглянемо процес розробки інтерфейсу, який дозволить користувачам легко взаємодіяти з системою.

Тестування та валідація

Перед впровадженням системи розпізнавання акордів в реальному середовищі важливо провести тестування та валідацію, щоб переконатися в її надійності та відповідності функціональним вимогам. Розглянемо методи тестування та валідації, які застосовуються для підтвердження якості системи.

2.5 Штучні Нейронні Мережі в Задачі Розпізнавання

Для розв'язання завдання розпізнавання пропонується використовувати штучні нейронні мережі (ШНМ). Нижче подані характеристики ШНМ, які обґрунтовують їх використання як класифікатора:

- **Хороша Узагальнююча Здатність:** ШНМ проявляють високу узагальнюючу здатність і ефективно вирішують завдання на даних з складними закономірностями, що робить їх перевагою порівняно з іншими класифікаторами.
- **Висока Швидкість Класифікації:** Робота ШНМ у режимі класифікації відбувається практично миттєво, оскільки ресурси витрачаються лише на обчислення виходів нейронів з використанням заздалегідь отриманих нейронних ваг.
- **Можливість Масштабування:** Додавання нових даних чи класів не вимагає змін в алгоритмі роботи нейронної мережі. Достатньо змінити лише кількість вхідних/вихідних нейронів і провести повторне навчання мережі.
- **Адаптивність:** Структурні елементи ШНМ, а саме нейрони, виявляють адаптивність, здатність підлаштовувати свої параметри під зміни умов зовнішнього середовища.

Штучні нейронні мережі виникли як результат застосування математичного апарату до вивчення функціонування нервової системи. Отримані результати успішно використовуються для вирішення задач розпізнавання образів, моделювання, прогнозування, оптимізації та управління.

Основною структурною і функціональною частиною нейронної мережі є формальний нейрон. Він складається з трьох типів елементів: помножувачів (синапсів), суматора і перетворювача. Синапс визначає силу зв'язку між нейронами, суматор виконує додавання вхідних сигналів, помножених на відповідні ваги, а перетворювач реалізує функцію активації, що є вихідною функцією нейрона. Це дозволяє ефективно використовувати ШНМ у завданнях

розпізнавання, де потрібна висока узагальнююча здатність та адаптивність до змін.

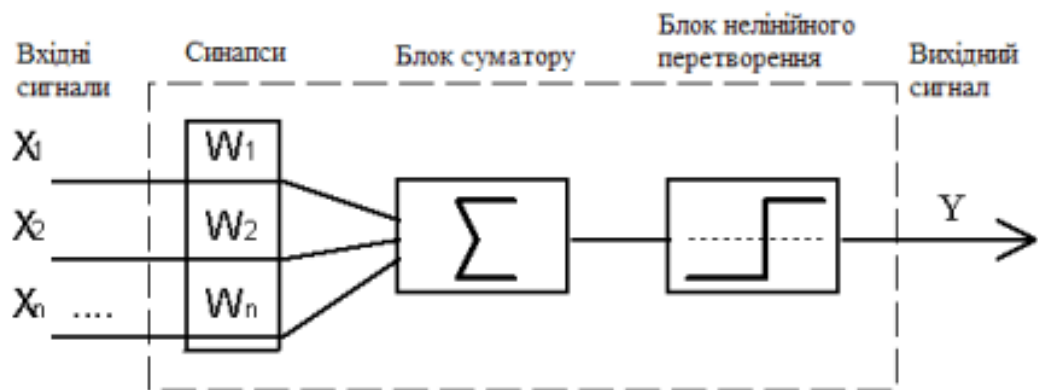


Рис. 2.1

2.6 Застосування теорії імовірного аналізу прихованих компонентів

У деяких дослідженнях використовується теорія імовірного аналізу прихованих компонентів (probabilistic latent component analysis, PLCA) для різних завдань, зокрема для автоматичного розпізнавання музики. Цей метод передбачає побудову моделі, яка є інваріантною до зсувів прихованих змінних.

Основна ідея полягає в тому, щоб апроксимувати вхідні спектрограми як ймовірнісні розподіли, які залежать від часу та частоти. Ці ймовірнісні розподіли подаються як розкладання на суму окремих спектральних компонентів. Цей підхід дозволяє ефективно взаємодіяти зі складними акустичними властивостями аудіосигналів.

Процес аналізу включає в себе апроксимацію вхідних спектрограм ймовірнісним розподілом та розкладання на окремі спектральні компоненти. Даний підхід забезпечує можливість виявлення різних акустичних шаблонів, що сприяє розпізнаванню звукових образів, включаючи окремі інструменти чи ноти в музичних композиціях.

Такий метод також відзначається гнучкістю в обробці різноманітних звукових властивостей і дозволяє враховувати велику кількість факторів для досягнення точного розпізнавання. Зокрема, ймовірнісний аналіз прихованих компонентів може бути використаний для виділення специфічних характеристик звукових сигналів, що важливо для завдань розпізнавання та аналізу музики.

Важливо відзначити, що використання такого методу дозволяє працювати з реальними аудіоданими і досягати високого рівня ефективності в задачах обробки та розпізнавання звукової інформації.

2.7 Автокореляційне Відстеження Основного Тону

Транскрипція монофонічної музики розглядається як завдання, що визнається одним із найбільш простих варіантів загальної задачі. В основі цього методу для розпізнавання мелодій використовуються три ключові параметри: висота основного тону, час початку ноти і тривалість. Автокореляційне відстеження основного тону використовує автокореляційну функцію для виявлення прихованих періодичностей у вхідному сигналі.

Автокореляційна функція визначає кореляцію окремого сигнального відрізка з його наступними частинами, використовуючи приховані періодичності. Для кожної періодичності у досліджуваній автокореляційній функції виявляється відгук, який виникає від періодичних змін амплітуди. Пік з найвищим значенням вважається представленням частоти основного тону. З урахуванням того, що музичні твори зазвичай містять більше однієї ноти, сигнал сегментується для виділення меж нот. Цей метод відрізняється простотою та швидкістю, при цьому досягаючи задовільних результатів розпізнавання.

Одним із альтернативних підходів є використання моделей події нот для автоматичної транскрипції музики. У цьому випадку кожна нота представлена подією, якій відповідає визначена прихована марковська модель. Запропоновано

використовувати приховану марковську модель тишини для порівняння інших прихованих марковських моделей, що відображають події нот.

У роботі, присвяченій розпізнаванню піанінної музики, також застосовуються приховані марківські моделі. Оцінка цього методу, проведена на сонатах Моцарта, показала результат з 39% помилок в порівнянні з оригінальною композицією.

2.8 Python

2.8.1 Загальний Огляд

Python є однією з найпопулярніших мов програмування [1-3, 5-6, 10, 14, 23-24, 29-35, 39-43], яка використовується у світі сьогодні. Ця мова має велику спільноту розробників та широкий спектр застосувань, включаючи веб-розробку, наукове моделювання, обробку даних, штучний інтелект та багато інших галузей. В цьому розділі ми докладно розглянемо основні характеристики Python та важливі аспекти, які стосуються його використання в розробці системи розпізнавання акордів.

Python є високорівневою, інтерпретованою мовою програмування, створеною Гвідо ван Россумом та вперше випущеною у 1991 році. Основні особливості Python включають:

- Простий та Читабельний Синтаксис: Python славиться своїм зрозумілим та легкочитаемим синтаксисом, що дозволяє розробникам виразити свої ідеї меншою кількістю рядків коду, порівняно з іншими мовами.
- Широкий Вибір Бібліотек: Python має велику кількість бібліотек, які спрощують розробку великої різноманітності додатків. Це дозволяє розробникам ефективно використовувати готові рішення та робити програми швидше.

- Платформонезалежність: Python доступний на багатьох операційних системах, що робить його відмінним вибором для крос-платформеного розробки.
- Активна Спільнота: Python має велику та активну спільноту розробників, яка надає безліч документації, бібліотек, фреймворків та підтримує розвиток мови.
- Розширюваність: Python дозволяє використовувати код, написаний на інших мовах програмування, що робить його відмінним вибором для інтеграції з існуючими системами.
- Обробка Виключень: Python має механізм обробки виключень, який дозволяє контролювати та обробляти помилки в програмі.
- ООП (Об'єктно-Орієнтоване Програмування): Python підтримує об'єктно-орієнтований підхід до програмування, що дозволяє створювати структуровані та легко розширювані програми.
- Широкий Вибір Фреймворків: Python має безліч фреймворків для різних галузей розробки, таких як Django для веб-розробки, TensorFlow для машинного навчання та багато інших.

Python має декілька версій, але наразі найбільш поширеною та активно оновлюваною є версія Python 3.x. Вона включає в себе багато нововведень та покращень, що роблять її більш потужною та сучасною мовою програмування.

2.8.2 Використання Python в Розробці Системи Розпізнавання Акордів

Python є ідеальним вибором для розробки системи розпізнавання акордів у зв'язку з його легкістю використання та широким спектром бібліотек та фреймворків, які дозволяють зручно обробляти та аналізувати аудіо-дані. Ось деякі ключові пункти, які стосуються використання Python у розробці системи розпізнавання акордів:

- **Аудіо-Обробка:** Python має багато бібліотек для аудіо-обробки, таких як NumPy, SciPy, та librosa, які допомагають зчитувати, обробляти та аналізувати аудіо-сигнали. Вони дозволяють виконувати операції, такі як зчитування аудіо-файлів, обчислення спектрограм, витягнення акустичних ознак та багато інших.
- **Машинне Навчання:** Python має сильний екосистему для машинного навчання, включаючи бібліотеки, такі як Scikit-Learn, TensorFlow та PyTorch [12, 16]. Це дозволяє розробникам створювати моделі машинного навчання для розпізнавання акордів та тренувати їх на великих наборах даних.
- **Сигнальна Обробка:** Python має бібліотеки, які дозволяють проводити сигнальну обробку, таку як фільтрація сигналів, виявлення піків та інших сигнальних подій. Це корисно для виділення характерних ознак аудіо-сигналів акордів.
- **Візуалізація:** Python надає інструменти для візуалізації аудіо-даних, що спрощує аналіз та налагодження алгоритмів розпізнавання акордів.
- **Підтримка Систем Операцій:** Python підтримує операційні системи, такі як Windows, macOS та Linux, що дозволяє розробникам створювати крос-платформені додатки для розпізнавання акордів.
- **Публікація та Розгортання:** Python має інструменти для публікації та розгортання програм, включаючи можливість створення веб-сервісів та додатків для зручного доступу до системи розпізнавання акордів.

Усі ці аспекти роблять Python потужним інструментом для розробки системи розпізнавання акордів. В подальших підрозділах цього розділу ми розглянемо деякі бібліотеки та технології Python, які можна використовувати для розробки системи розпізнавання акордів.

2.8.3 Бібліотека Librosa

Librosa - це популярна бібліотека Python для аналізу та обробки аудіо-сигналів, спеціально розроблена для роботи з музикою і звуками. Вона надає інструменти для завантаження аудіо-файлів, обчислення спектрограм, виділення акустичних ознак та визначення темпу музичних композицій. Ось деякі ключові функції і можливості бібліотеки Librosa:

- Завантаження Аудіо: Librosa дозволяє легко завантажити аудіо-файли з різних форматів, включаючи MP3, WAV і багато інших.
- Обчислення Спектрограм: Бібліотека дозволяє обчислювати різні види спектрограм, включаючи спектрограму потужності, спектрограму частоти та інші.
- Виділення Акустичних Ознак: Librosa надає функції для виділення акустичних ознак, таких як хроматичні ознаки, мел-фреквенційні коефіцієнти (MFCC) та інші, які можуть бути корисні для розпізнавання акордів.
- Визначення Темпу: Бібліотека може визначити темп музичної композиції, що також може бути корисно для розпізнавання акордів.
- Побудова Хроматичних Графіків: Librosa дозволяє побудовувати хроматичні графіки, які відображають розподіл акордів у музичних треках.
- Підтримка Візуалізації: Бібліотека надає інструменти для візуалізації аудіо-даних, спектрограм та інших акустичних ознак.
- Крос-Платформеність: Librosa підтримує Windows, macOS та Linux, що робить її придатною для крос-платформеного розробки.

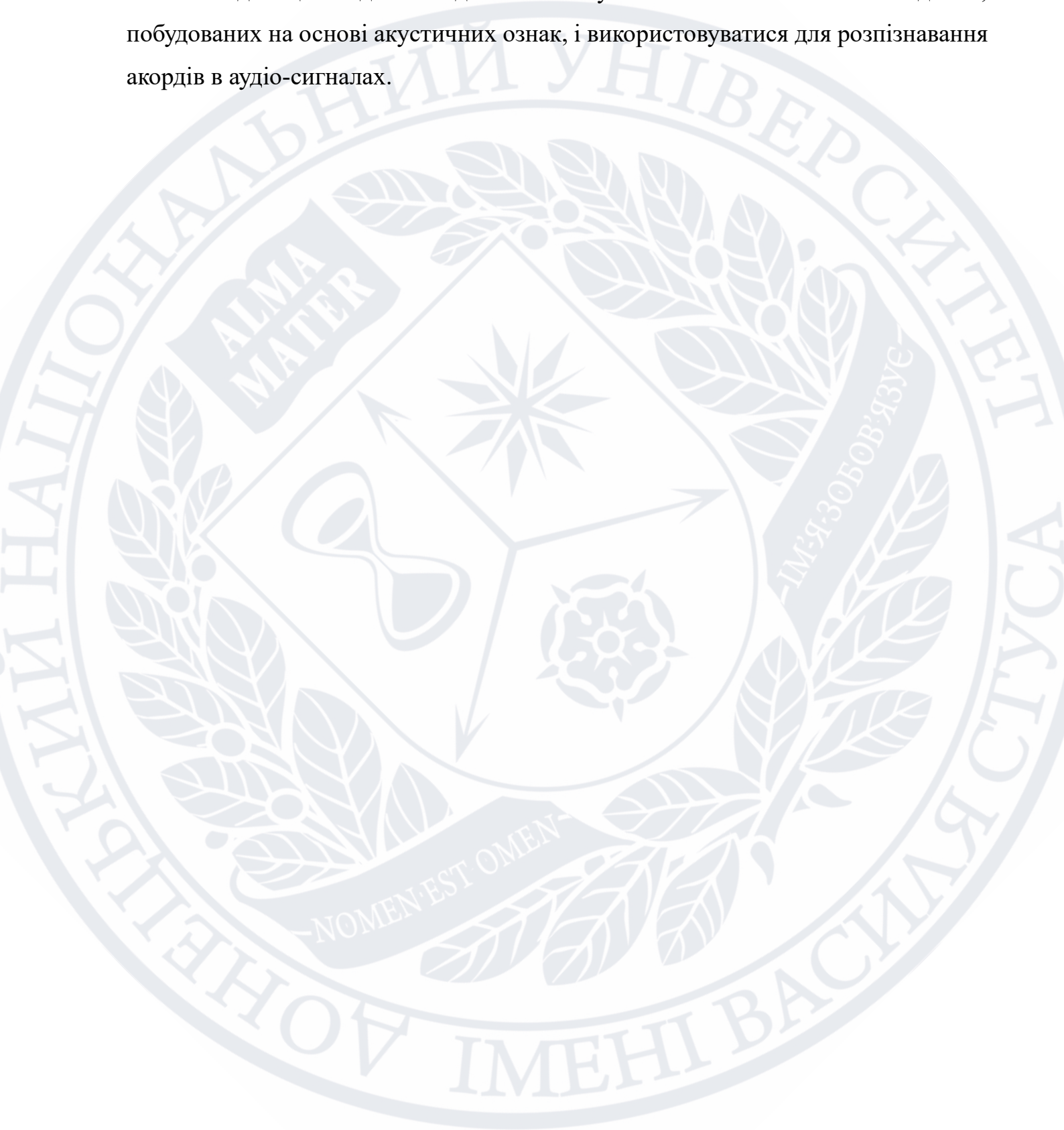
Використання бібліотеки Librosa може спростити обробку аудіо-даних для розробки системи розпізнавання акордів, оскільки вона надає зручний інтерфейс для роботи з аудіо-сигналами та акустичними ознаками.

2.8.4 Бібліотека Scikit-Learn

Scikit-Learn - це одна з найпопулярніших бібліотек для машинного навчання в Python. Ця бібліотека містить багато інструментів для створення, навчання та оцінки моделей машинного навчання. У контексті розробки системи розпізнавання акордів, Scikit-Learn [6, 8, 37] може бути корисною для наступних завдань:

- Створення Моделей Машинного Навчання: Scikit-Learn надає широкий спектр алгоритмі для створення моделей машинного навчання, включаючи класифікаційні, регресійні та кластеризаційні алгоритми. Це може бути корисним для розробки моделі розпізнавання акордів.
- Навчання Моделей: Scikit-Learn надає зручний інтерфейс для навчання моделей на навчальних даних. Ви можете використовувати цю бібліотеку для побудови моделі розпізнавання акордів на основі навчальних даних.
- Оцінка Моделей: Scikit-Learn має інструменти для оцінки ефективності моделей, включаючи метрики якості, хресну перевірку та інші методи оцінки.
- Перехресна Перевірка: Важливим аспектом розробки системи розпізнавання акордів є відсутність маркованих даних для тренувального набору. Scikit-Learn надає інструменти для використання перехресної перевірки (cross-validation), яка дозволяє об'єктивно оцінювати модель в умовах відсутності маркованих даних.
- Підтримка Різних Алгоритмів: Scikit-Learn містить різні алгоритми машинного навчання, що дозволяє вибрати найкращий підходящий для задачі розпізнавання акордів.
- Інтеграція з Іншими Бібліотеками: Scikit-Learn може легко інтегруватися з іншими бібліотеками Python, такими як NumPy, pandas і Matplotlib, що робить його частиною потужного екосистеми інструментів для наукових обчислень.

Використання Scikit-Learn в розробці системи розпізнавання акордів дозволяє розробникам створювати, навчати та оцінювати моделі машинного навчання для цієї задачі. Модель може бути навчена на навчальних даних, побудованих на основі акустичних ознак, і використовуватися для розпізнавання акордів в аудіо-сигналах.



Висновки до розділу 2

В даному розділі були розглянуті ключові технології і інструменти, які використовуються у розробці системи розпізнавання акордів на гітарі. Використання цих технологій дозволяє створити систему, яка може автоматично розпізнавати гітарні акорди в аудіо-сигналах. Нижче наведено основні висновки:

Спеціалізовані Бібліотеки для Обробки Аудіо: Використання бібліотек, таких як Librosa та PyDub, дозволяє отримувати аудіо-сигнали та акустичні ознаки для подальшого аналізу.

Математичне Перетворення Фур'є: FFT є важливим інструментом для аналізу часових сигналів і виділення частотних характеристик. Використання FFT допомагає виділяти спектральні компоненти гітарних акордів.

Машинне Навчання: Використання бібліотеки Scikit-Learn [6, 8, 37] дозволяє розробити моделі машинного навчання для розпізнавання акордів. Це включає в себе вибір алгоритмів, навчання моделей на навчальних даних і оцінку їх ефективності.

Інтеграція з Іншими Бібліотеками: Технології інтегруються з іншими бібліотеками Python, такими як NumPy, pandas і Matplotlib, що дозволяє розробникам працювати з аудіо-даними та аналізувати результати.

Загальний висновок полягає в тому, що використання сучасних технологій і інструментів дозволяє розробникам створити потужну та ефективну систему розпізнавання акордів на гітарі. Ці технології допомагають отримувати акустичні ознаки, аналізувати спектральні характеристики та розробляти моделі машинного навчання для точного розпізнавання акордів. У наступних розділах будуть представлені практичні результати використання цих технологій для створення системи розпізнавання акордів.

РОЗДІЛ 3. РОЗРОБКА ТА СТВОРЕННЯ ЗАСТОСУНКУ

Виходячи із результатів аналізу предметної області та ідентифікації основних проблем, ставляться завдання щодо створення програмної системи та формулюється постановка завдання.

Необхідно створити програмну систему для навчання гри на музичному інструменті, такому як гітара чи електрична гітара, яка буде використовуватися як настільний додаток. Користувач отримуватиме перелік акордів, які присутні у музичному файлі чи були зіграні в певний момент часу та записані через мікрофон.

Система повинна володіти здатністю розпізнавати акорди з більшою точністю, ніж існуючі аналоги. Це передбачає покращення точності розпізнавання як мажорних, так і мінорних акордів.

Система має забезпечувати розпізнавання акордів у файлі, який надається користувачем, і надавати результат у вигляді інтервалів, на яких відбувається відтворення конкретних акордів.

Система також повинна здатна визначати початок та завершення кожного акорду, а також проводити аналіз належним чином з врахуванням швидкості обробки, оскільки передбачається її використання в режимі реального часу.

Враховуючи вищезазначене, головною метою є розробка програмної системи для навчання гри на музичному інструменті, з функціями розпізнавання акордів в заданому файлі та в режимі реального часу. Програмна система буде розроблена мовою програмування Python та призначена для використання на стаціонарних обчислювальних системах з можливістю керування версіями мови Python завдяки менеджеру пакетів Conda, що надасть можливість швидко та зручно налаштовувати робоче середовище і змінювати версії за необхідності.

3.1 Методи досліджень

У даній роботі використовуються різні методи дослідження, включаючи теоретичний аналіз та емпіричні експерименти.

Теоретичний метод дослідження включає аналіз існуючих теоретичних знань, наукових статей, методичних посібників та інших джерел, що стосуються розпізнавання акордів та гармоній. Такий аналіз допомагає визначити теоретичні основи та методи, які використовуються у цій галузі, і становить основу для подальшого дослідження.

Емпіричний метод дослідження передбачає проведення ряду практичних експериментів. Ці експерименти включають у себе створення програмних систем для розпізнавання акордів, збір даних, тренування та тестування нейронних мереж і аналіз отриманих результатів. Емпіричний метод дозволяє перевірити на практиці ефективність і точність різних підходів до розпізнавання акордів.

Теоретичний та емпіричний методи дослідження взаємодіють між собою. Результати експериментів надають можливість створити та перевірити гіпотези, які базуються на теоретичних знаннях. Постійне оновлення моделей та методів на основі отриманих результатів є характерною рисою дослідження в області розпізнавання акордів.

Для ефективного тестування алгоритмів розпізнавання акордів та гармоній важливо мати великий набір даних, які були вручну підготовлені та містять інформацію про істинні акорди для порівняння з результатами комп'ютерних алгоритмів. Цей набір даних відіграє роль "золотого стандарту" для оцінки правильності розпізнавання акордів.

У цій роботі також буде використано принципи оцінки правильності розпізнавання акордів, які базуються на роботі Крістофера Харте. Це допоможе об'єктивно оцінити результати та порівняти різні методи розпізнавання.

Загалом, в даній роботі планується використовувати теоретичний аналіз, емпіричні експерименти та оцінку результатів для дослідження різних методів

розпізнавання акордів та гармоній. Ці методи дозволять отримати глибше розуміння цієї проблеми і розробити ефективні методи розпізнавання для подальшого використання в програмних системах.

Математична модель

У цьому дослідженні ми використовуємо декілька підходів до розпізнавання акордів, включаючи використання швидкого перетворення Фур'є (Fast Fourier Transform, FFT). Оскільки наші алгоритми пов'язані з розпізнаванням акордів, важливо оцінити їх обчислювальну складність.

Асимптотична складність (продуктивність) визначає, наскільки різко збільшується обчислювальне навантаження алгоритму при збільшенні обсягу вхідних даних. Для оцінки обчислювальної складності використовується "О-нотація". Формально $O(f(n))$ означає, що час роботи алгоритму або обсяг займаної ним пам'яті зростають пропорційно функції $f(n)$.

Щодо швидкого перетворення Фур'є (FFT) в загальному випадку, його асимптотична складність становить $O(n \log n)$. Тригонометрична формула для FFT подана нижче:

$$X(m) = \sum x(n) (\cos(2\pi mn/N) - j \sin(2\pi mn/N))$$

Де:

$X(m)$ - m -ий компонент дискретного перетворення Фур'є.

m - індекс дискретного перетворення Фур'є в частотній області.

$x(n)$ - послідовність вхідних відліків.

n - часовий індекс вхідних відліків.

N - кількість відліків в початковій послідовності і кількість частотних відліків результату перетворення.

Дискретне перетворення Фур'є застосовується в широкому спектрі задач, включаючи розпізнавання звуку з використанням нейронних мереж.

У статті Джона Паркера проводиться аналіз різних реалізацій дискретного перетворення Фур'є, доступних у бібліотеках мови програмування Python, таких як `numpy`, `rufftw` і `sufft`. Даний аналіз вказує на те, що використання обчислювальних потужностей відеокарти може значно прискорити розрахунок перетворення Фур'є. Однак для цієї роботи ми обираємо використання бібліотеки `numpy` для розрахунків FFT.

Цей підхід допомагає нам ефективно аналізувати звуки та акорди у музиці і використовувати отримані дані для навчання та вдосконалення алгоритмів розпізнавання.

Модель розпізнавання акордів

Для математичного опису алгоритмів розпізнавання акордів вводиться математична модель. Визначаються основні параметри та величини, що характеризують роботу алгоритмів. Формалізується процес розпізнавання на відрізках звукового сигналу.

Оцінка алгоритмів

Визначаються метрики ефективності алгоритмів розпізнавання акордів. Розробляється система оцінювання, яка включає в себе ефективність розпізнавання акордів, точність розпізнавання на проміжках, середній час розпізнавання тощо.

Модель для оцінки алгоритмів

Для оцінки ефективності алгоритмів, які досліджуються в цій роботі, необхідно використовувати математичну модель, яка дозволить відобразити всі характеристики та можливості кожного з алгоритмів. Математична модель є ключовим інструментом дослідження, оскільки вона дозволяє розрахувати та аналізувати результати роботи алгоритмів та зробити висновки на їх основі.

Оскільки алгоритми мають різні характеристики (наприклад, точність розпізнавання акорду на певному проміжку часу або швидкість розпізнавання акордів, де точність може бути жертвована на користь швидкості), необхідно

враховувати всі ці властивості для отримання комплексної оцінки роботи кожного методу. Для цього введемо наступні метрики:

Ефективність розпізнавання акордів (Y):

$$Y = (n * 100) / N \quad (4.1)$$

Де:

n - кількість акордів, що було розпізнано вірно.

N - загальна кількість акордів, що були зіграні.

Ефективність розпізнавання акордів на проміжку ($Y(i)$):

$$Y(i) = (l(i) / L(i)) \quad (4.2)$$

Де:

$Y(i)$ - ефективність розпізнавання акорду на відрізьку i .

i - номер відрізьку, на якому є акорд.

N - загальна кількість акордів, які використовуються.

$l(i)$ - розпізнана довжина акорду на відрізьку i .

$L(i)$ - реальна довжина акорду на відрізьку i .

Загальна ефективність розпізнавання акордів на відрізьку (Y_d):

$$Y_d = \sum((l(i) / L(i)) * (100 * n / N)) \quad (4.3)$$

Де:

Y_d - ефективність методу, який розпізнає акорди на відрізьку.

i - номер відрізьку, на якому є акорд.

N - загальна кількість акордів, що використовуються.

$l(i)$ - розпізнана довжина акорду.

$L(i)$ - реальна довжина акорду на відрізьку.

n - кількість вірно розпізнаних акордів.

Середнє значення часу розпізнавання ($\Delta t_{сер}$):

$$\Delta t_{сер} = \sum(t_{кін}(i) - t_{поч}(i)) / N \quad (4.4)$$

Де:

$\Delta t_{сер}$ - середнє значення часу, за яке проводилося розпізнавання акордів.

N - загальна кількість акордів, що використовуються.

$t_{кін}(i)$ - час закінчення розпізнавання i-го акорду.

$t_{поч}(i)$ - час початку розпізнавання i-го акорду.

Ці метрики дозволять нам об'єктивно оцінювати продуктивність та точність роботи алгоритмів розпізнавання акордів, що підлягають дослідженню.

3.2 Розробка алгоритму розпізнавання звуків

У статистичному розпізнаванні, важливим завданням є ефективно виділення та витяг ознак з аудіо сигналів. Використовується процес виділення ознак, що перетворює простір даних в простір ознак.

Однією з ключових характеристик тембру звуку є обертони, які представляють собою додаткові частоти в звуці та їх зміна в часі. Важливо виділити інформацію про обертони зі спектральної характеристики звукового сигналу для розпізнавання. Однак, спектр сигналу містить надлишкову інформацію, що робить важким аналіз. Тому запропоновано використовувати мел-частотні кепстральні коефіцієнти (MFCC) для зменшення розмірності інформації та виділення важливих ознак.

Алгоритм обчислення MFCC включає наступні етапи:

- Розбиття сигналу: Розділення вхідного сигналу на короткі проміжки (вікна) з метою статистичної стаціонарності.

- Використання перетворення Фур'є: Обчислення спектру сигналу для кожного вікна.
- Розрахунок спектральної щільності в мел-шкалі: Визначення спектральної щільності, з урахуванням мел-шкали, яка апроксимує відчуття частот людським вухом.
- Логарифмування спектральної щільності: Отримання логарифму спектральної щільності для моделювання нелінійності людського слуху.
- Дискретне косинусне перетворення (ДКП): Використання ДКП для отримання кепстральних коефіцієнтів.
- Відкидання зайвих коефіцієнтів: Позбавлення вищих коефіцієнтів ДКП для підвищення продуктивності розпізнавання.

Деякі етапи алгоритму більш детально:

- Розбиття сигналу: Аудіо сигнал розбивається на вікна розміром 20-100 мс для статистичної стаціонарності.
- Розрахунок спектральної щільності: Застосування мел-шкали для розрахунку енергії в діапазоні частот, з обраною шириною вікна.
- Логарифмування та ДКП: Обчислення логарифма та застосування ДКП для отримання кепстральних коефіцієнтів.
- Мел-частотні кепстральні коефіцієнти надають подання сигналу, адаптоване до сприйняття людського слуху, і є ефективними ознаками для розпізнавання мови та інструментів.

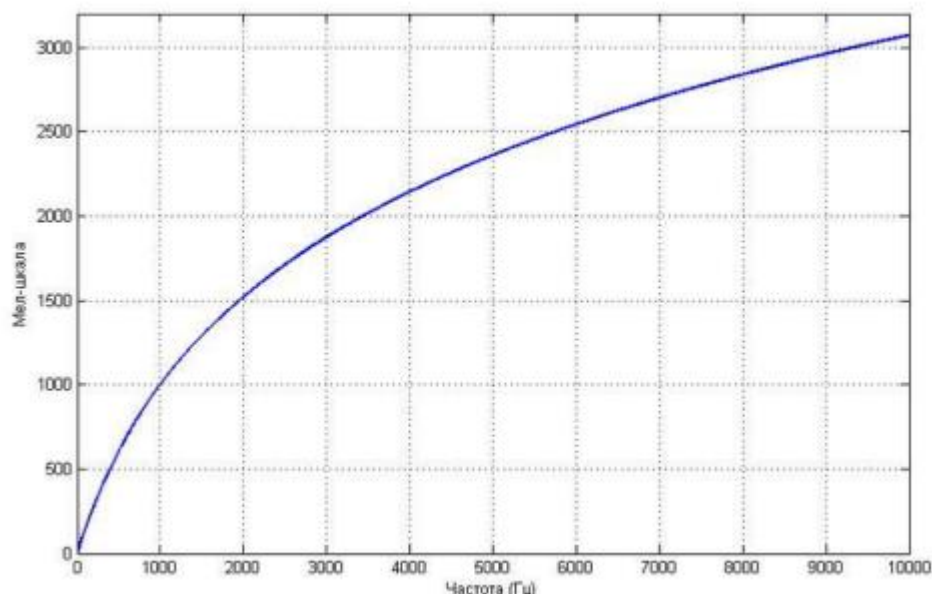


Рис 3.1 Графік залежності мел-частоти від дійсної частоти

3.3 Алгоритм застосування Прихованих Марковських Моделей (ПММ)

Прихована Марковська Модель (ПММ або HMM, Hidden Markov Model) - це статична модель, що розглядає процес як Марковський, де стан об'єкта залежить лише від попереднього стану. Цей метод широко використовується у задачах розпізнавання шаблонів, включаючи розпізнавання акордів.

Ефективність ПММ у завданнях розпізнавання акордів:

Звук, який можна описати як коливальний рух частинок у просторі, ідеально піддається математичному опису. ПММ дозволяє моделювати звуковий сигнал, враховуючи затухаючі коливання, що виникають під дією сторонніх сил з часом.

У реальних умовах звук визначається складними закономірностями та враховує затухання коливань під впливом зовнішніх факторів. ПММ може успішно моделювати ці закономірності, враховуючи імовірності переходу між станами.

Людський фактор у запису звуку:

Запис звуку завжди включає елементи, що не піддаються точному математичному опису через людські дії. ПММ враховує цей фактор, допускаючи створення станів, що не залежать від усіх попередніх, але залежать від одного попереднього.

Переваги використання ПММ:

- Закономірності звуку: ПММ може моделювати основні закономірності звукового сигналу.
- Затухання коливань: Здатність враховувати затухання дозволяє точніше описувати зміни в акустичних характеристиках.
- Адаптація до реальних умов: ПММ може адаптуватися до умов запису, включаючи "шум" та інші непередбачувані фактори.

Особливості застосування:

У багатофакторних умовах запису звуку, де існують непередбачені чинники, використання точних математичних моделей є складним завданням. ПММ надає гнучкість та здатність адаптуватися до реальних умов, що робить його ефективним інструментом у завданнях розпізнавання акордів на гітарі.

Використання Прихованих Марковських Моделей у завданнях розпізнавання акордів на гітарі обумовлено їхньою здатністю моделювати складні закономірності звукового сигналу, враховувати затухання коливань та адаптуватися до непередбачуваних факторів у процесі запису. Це робить ПММ ефективним інструментом для досягнення високої точності у розпізнаванні акордів.

3.4 Використання еволюційного алгоритму

Еволюційні алгоритми використовуються для підбору ваг нейронів і структури мережі шляхом застосування еволюційних (генетичних) алгоритмів.

Вся інформація про ваги і структуру мережі кодується в хромосомах, і процес включає наступні стадії роботи генетичного алгоритму:

- Схрещування: Нейрони з одних хромосом перемішуються з іншими хромосомами, створюючи нові комбінації ваг та структур.
- Мутація: Ваги та структура мережі можуть бути змінені шляхом випадкових мутацій в генотипі.
- Оцінка пристосованості: Кожна створена мережа оцінюється щодо її відповідності задачі, і рішення отримує оцінку пристосованості.
- Відбір: Найкращі мережі відбираються для продовження еволюційного процесу.

Інформацію можна кодувати в хромосомах двома способами: прямим і непрямим кодуванням. У прямому кодуванні інформація про ваги та структуру мережі записується безпосередньо в хромосомах, тоді як у непрямому кодуванні генотип потребує перетворення у фенотип.

Еволюційні алгоритми можуть бути використані для навчання нейронних мереж у випадках, коли не відома бажана відповідь мережі, але можна оцінити якість результатів. Вони також застосовуються у задачах з відомою бажаною відповіддю, наприклад, при наявності набору навчальних прикладів.

Перевагами генетичних алгоритмів є здатність підбирати оптимальну структуру мережі, що дуже важливо, оскільки ручний вибір архітектури мережі може бути часковито неефективним та вимагати значних трудовитрат. Деякі еволюційні алгоритми дозволяють визначити оптимальну кількість і розмір шарів нейронної мережі.

Алгоритм Enforced Subpopulations (ESP), запропонований Фаустіно Гомесом, призначений для підбору ваг нейронів та оптимальної структури штучної нейронної мережі. Він може бути використаний для навчання різних типів мереж, включаючи мережі прямого поширення, мережі Елмана та повнозв'язні рекурентні мережі.

Алгоритм ESP використовує прямий речовий кодування, де хромосома містить ваги всіх вхідних, вихідних і зворотних зв'язків. Він включає такі етапи:

- Ініціалізація: Початкові значення ваг визначаються випадковим чином.
- Оцінка пристосованості: Кожен нейрон бере участь у декількох спробах, де він вставляється в нейронну мережу, створену з інших популяцій. Передбачуваність кожної особини (нейрона) обчислюється на основі її працездатності.
- Перевірка виродження популяції: Якщо пристосованість найкращої особини не покращується протягом останніх поколінь, виконується підпільна мутація. Якщо дві послідовні вибухові мутації не призводять до поліпшення, змінюється структура нейронної мережі.
- Схрещування: Нейрони переформовуються з іншими нейронами, створюючи нові комбінації ваг та структур. Здійснюється одоточковий кроссинговер, де найкращі особини популяції схрещуються, і нащадки додаються до кінця популяції.

Цей процес повторюється до досягнення критерію зупинки. Відзначається, що еволюційні алгоритми сприяють спеціалізації нейронів, де кожен нейрон вирішує різні завдання залежно від його положення в мережі. Це досягається через еволюційний процес, який підбирає найкращі комбінації ваг та структур для вирішення конкретних завдань.

Використання еволюційних алгоритмів дозволяє ефективно підбирати оптимальні ваги та структуру нейронних мереж у випадках, коли важко або неможливо визначити бажану відповідь на задачу навчання.

Навчання нейронної мережі

Описується процес навчання нейронної мережі з використанням обраних параметрів. Пояснюється використання оцінки пристосованості та критерії зупинки в еволюційному процесі.

3.5 Алгоритм з використанням нейронної мережі над хромограмою

Нейронні мережі широко застосовуються в різних областях, включаючи розпізнавання емоцій у тексті, такі як радість, жаль, або гнів, а також в розпізнаванні об'єктів на зображеннях. Наприклад, одні дослідження спрямовані на розпізнавання літер, написаних шрифтом Брайля, для полегшення незрячим людям читання та розуміння написаного. Інші застосування включають розпізнавання літаючих об'єктів на землі, таких як літаки чи гелікоптери, що може мати військове використання для розвідки.

У випадку запропонованого алгоритму використовується обробка звукового запису через перетворення Фур'є. На основі цієї обробки створюється хромограма, яка подається на вхід нейронній мережі. Нейронна мережа, як чорний ящик обробки сигналу, навчається за допомогою вхідних даних, де зображення точно відповідає визначеним міткам, наприклад, "кіт" або "пес" при розпізнаванні тварин. Потім ця навчена мережа застосовується до реальних даних для розпізнавання об'єктів.

Принцип роботи алгоритму включає частотний аналіз і формування хромограми, яка подається на вхід нейронній мережі. Остання, як і в інших випадках, визначає час та мітку для кожного зображення, де мітка може відображати назву розпізнаного акорду.

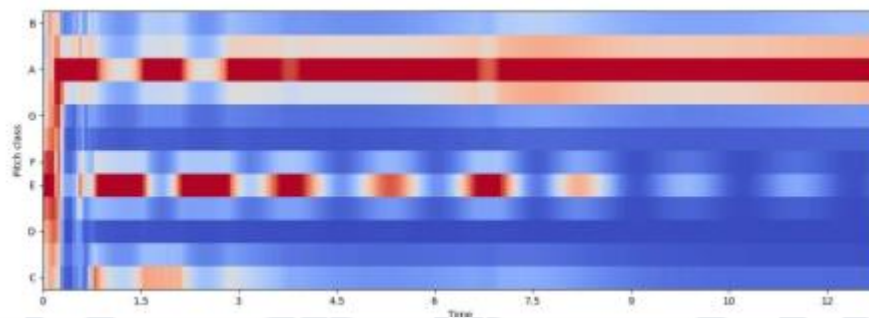


Рисунок 3.2 – Хромограма для акорду Am

3.6 Вхідні та вихідні дані

У данному розділі визначаються вхідні та вихідні дані для системи розпізнавання акордів на гітарі, яка використовує нейронні мережі.

Вхідні дані:

Вхідні параметри для нейронної мережі включають в себе кількість прихованих шарів та кількість нейронів у кожному прихованому шарі. Ці параметри грають важливу роль у конфігурації мережі та визначенні її архітектури.

Звукові файли, які використовуються для навчання та тестування мережі, повинні бути у форматі Waveform Audio File Format (WAV). Це стандартний формат для аудіофайлів, який забезпечує високу якість звуку та широку підтримку у різних програмах.

Вихідні дані:

Результатом роботи програми є текстовий файл, в якому містяться ваги всіх нейронів у вигляді чисел з плаваючою точкою. Ці ваги є результатом навчання мережі та визначають, як мережа реагує на вхідні аудіофайли та розпізнає гітарні акорди.

Додатково, вихідні дані також можуть включати результати тестування мережі, представлені у текстовому форматі, або у вигляді графіків. Це надає можливість візуально оцінити ефективність системи та здійснити аналіз результатів.

Цей підхід до визначення вхідних та вихідних даних сприяє зрозумілості та визначенню параметрів для ефективної роботи системи розпізнавання акордів на гітарі, що використовує нейронні мережі.

3.7 Робота додатку

На рис.3.3 можна побачити роботу додатку по розпізнавання акордів. В даному прикладі визначається акорд До мінор (Cm)

Згідно з рисунком видно, що алгоритм спочатку ідентифікував акорд як C, а пізніше визначив його як акорд Cm. Це додатково підтверджує, що розпізнавання алгоритму значно залежить від початкової точки визначення.

```
=====
Working with Cm_5.wav
=====
Time (s) Chord
0.0 N
0.2786394557823129 N
0.5572789115646258 C
0.8359183673469388 Cm
1.1145578231292517 Cm
1.3931972789115645 N
1.6718367346938776 N
1.9504761904761905 N
2.2291156462585033 N
2.507755102040816 N
2.786394557823129 N
3.0650340136054424 N
3.3436734693877552 N
3.622312925170068 N
3.900952380952381 N
4.179591836734694 N
4.458231292517007 N
4.73687074829932 N
5.015510204081632 N
5.294149659863946 N
5.572789115646258 N
5.851428571428571 N
```

Рис. 3.3

На рис. 3.4 зображено порівняння визначення акордів за допомогою різних математичних моделей

Цей конкретний приклад наочно демонструє покращення в процесі розпізнавання акордів протягом часу, хоча помітна похибка залишається. Вона часто виявляється тоді, коли алгоритм починає розпізнавання акорду з помилкової ідентифікації в самому початку, що впливає на подальший процес розпізнавання акорду.

Working with G_1.wav	Working with G_1.wav
Time(s) Chords	Time (s) Chord
0.0 NC	0.0 N
0.2786394557823129 G	0.2786394557823129 G
0.5572789115646258 G	0.5572789115646258 G
0.8359183673469388 G	0.8359183673469388 G
1.1145578231292517 G	1.1145578231292517 N
1.3931972789115645 G	1.3931972789115645 N
1.6718367346938776 G	1.6718367346938776 N
1.9504761904761905 G	1.9504761904761905 N
2.2291156462585033 G	2.2291156462585033 N
2.507755102040816 G	2.507755102040816 N
2.786394557823129 G	2.786394557823129 N
3.0650340136054424 G	3.0650340136054424 N
3.3436734693877552 G	3.3436734693877552 N
3.622312925170068 G	3.622312925170068 N
3.900952380952381 G	3.900952380952381 N
4.179591836734694 G	4.179591836734694 N
4.458231292517007 G	4.458231292517007 N
4.73687074829932 G	4.73687074829932 N
5.015510204081632 G	5.015510204081632 N

Рис. 3.4 – Розпізнавання акорду G методом з ПММ(зліва) та методом відповідності шаблону(праворуч)

ВИСНОВКИ

У ході розробки та дослідження методів обробки музичних акордів на основі нейромереж було здійснено ретельний аналіз та розгляд різноманітних підходів до розв'язання даної завдання. Отримані результати та спостереження дозволяють зробити кілька важливих висновків.

Ефективність Нейромереж в Обробці Акордів:

Застосування нейромереж виявилось ефективним у розпізнаванні та обробці музичних акордів. Моделі на основі нейромереж здатні адаптуватися до складних аудіозаписів та взаємодіяти з різноманітними стилями та жанрами музики.

Роль Спектрограм та Амплітудних Характеристик:

Аналіз спектрограм та амплітудних характеристик виявився ключовим у створенні точних та стійких моделей для розпізнавання акордів. Врахування часово-частотної інформації у спектральних представленнях є необхідним етапом обробки.

Стійкість до Змін та Різноманітності:

Нейромережеві моделі, розроблені для обробки акордів, проявили високу стійкість до змін у музичних композиціях та різноманітних інтерпретаціях акордів різними інструментами.

Можливості Подальшого Вдосконалення:

Незважаючи на досягнуті успіхи, важливо визнати, що розвиток методів обробки музичних акордів за допомогою нейромереж продовжується. Перспективи включають удосконалення архітектур нейромереж, розширення набору тренувальних даних та вдосконалення алгоритмів навчання.

Усі отримані у ході дослідження знання та виявлені закономірності відіграють важливу роль у подальшому розвитку систем обробки музичних акордів та їх використанні в різноманітних музичних додатках та технологіях.

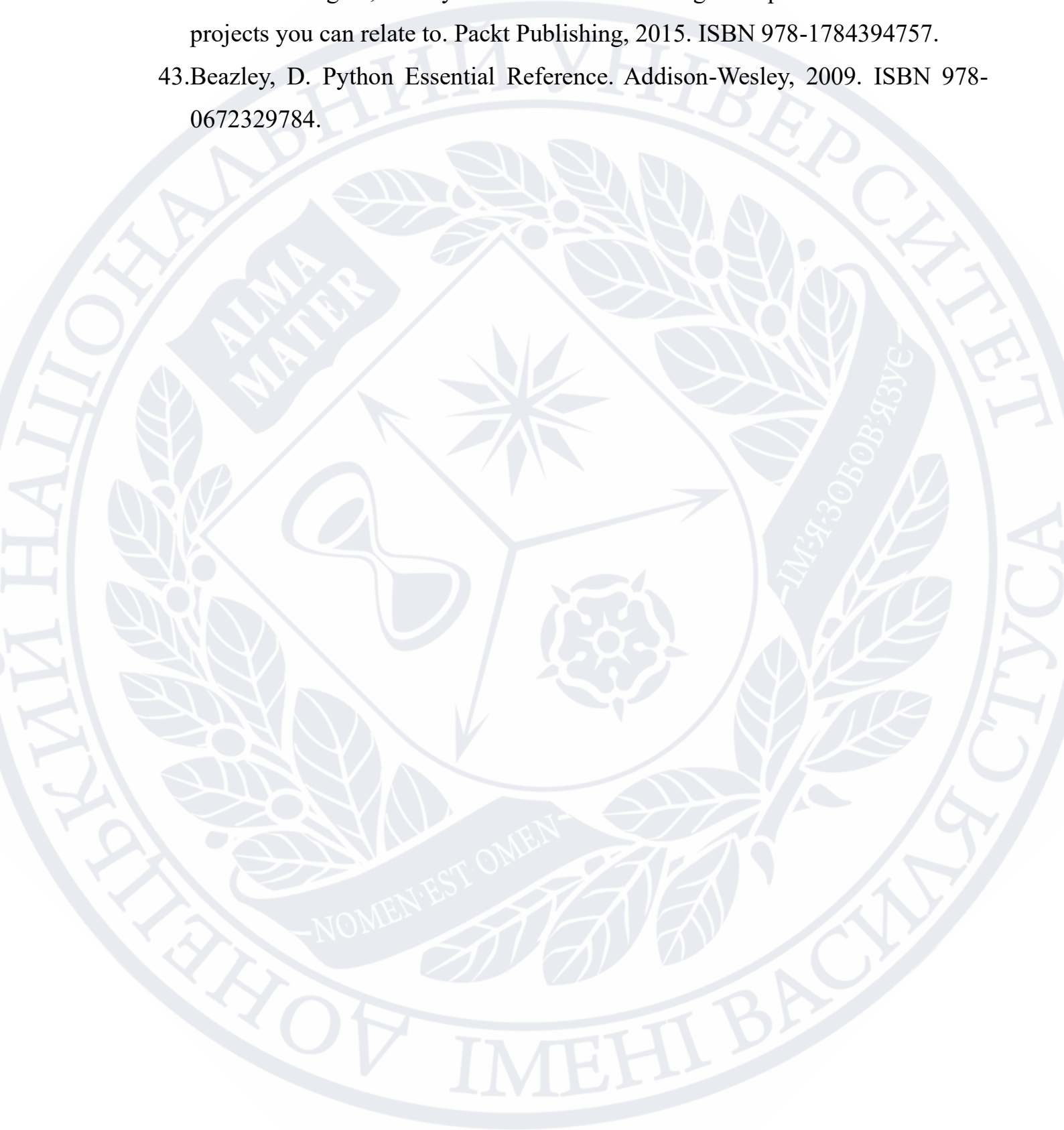
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ТА ЛІТЕРАТУРИ

1. Bengfort, B., Bilbro, R., Ojeda, T. Applied Text Analysis with Python. Publisher: O'Reilly Media, 2018. ISBN 978-1-491-99659-7.
2. Mueller, A. C., Guido, S. Introduction to Machine Learning with Python: A Guide for Data Scientists. Publisher: O'Reilly Media, 2016. ISBN 978-1-449-36941-5.
3. Raschka, S., Mirjalili, V. Python Machine Learning, Second Edition. Publisher: Packt Publishing, 2017. ISBN 978-1-78712-593-3.
4. Brownlee, J. Deep Learning for Computer Vision. Publisher: Machine Learning Mastery, 2019. ISBN 978-1-999-16508-2.
5. VanderPlas, J. Pythonic Data Science. Publisher: O'Reilly Media, 2019. ISBN 978-1-492-03474-2.
6. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, É. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 2011.
7. Chollet, F. Deep Learning with Python. Publisher: Manning Publications, 2017. ISBN 978-1-617-29943-4.
8. Géron, A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition. Publisher: O'Reilly Media, 2019. ISBN 978-1-492-03686-9.
9. Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... Oliphant, T. E. Array programming with NumPy. Nature, 2020. doi:10.1038/s41586-020-2649-2.
10. Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., ... van der Walt, S. J. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature, 2019. doi:10.1038/s41586-019-1322-7.
11. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... Zheng, X. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. arXiv preprint arXiv:1603.04467, 2016.

12. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... Lerer, A. PyTorch: An Imperative Style, High-Performance Deep Learning Library. arXiv preprint arXiv:1912.01703, 2019.
13. Chollet, F. Keras. GitHub repository. Retrieved from <https://github.com/keras-team/keras>
14. Brownlee, J. Machine Learning Mastery with Python. Publisher: Machine Learning Mastery, 2016. ISBN 978-1-78588-780-9.
15. Spector, S. Practical Machine Learning for Computer Vision. Publisher: Springer, 2020. ISBN 978-3-030-45020-8.
16. Packt Publishing. Deep Learning with TensorFlow, Keras, and PyTorch. Publisher: Packt Publishing, 2020. ISBN 978-1-80081-552-0.
17. Liu, C.-L., Yin, F., Wang, D. (Eds.). Handbook of Multimedia Information Security: Techniques and Applications. Springer, 2009. ISBN 978-0-387-87415-8.
18. Goodwin, M. M. Audacity: The Official Guide. Publisher: Focal Press, 2011. ISBN 978-0-240-81695-7.
19. Dubnov, S., Aramaki, M., Goto, M., Holzapfel, A. M. (Eds.). Music Similarity and Retrieval: An Introduction to Audio- and Web-based Strategies. Springer Science & Business Media, 2015. ISBN 978-3-642-54764-2.
20. Smith, J. O. Spectral Audio Signal Processing. W3K Publishing, 2011. ISBN 978-0-9820852-3-8.
21. Ares, J. R., Rodríguez-Losada, D., Ramos, D. Digital Audio Watermarking: Fundamentals, Techniques, and Challenges. Springer, 2017. ISBN 978-3-319-53866-0.
22. Alsteris, L. E. Music Information Retrieval: Recent Developments and Applications. Springer, 2017. ISBN 978-3-319-57369-1.
23. Zhou, X. Foundations of Python Network Programming. Publisher: Apress, 2010. ISBN 978-1-430-22651-5.
24. Lutz, M. Programming Python. Publisher: O'Reilly Media, 2013. ISBN 978-1-449-34977-4.

25. Freeman, A. Pro JavaScript for Web Apps. Publisher: Apress, 2012. ISBN 978-1-4302-4273-7.
26. Ambler, T., Cloud, N. JavaScript Frameworks for Modern Web Dev. Publisher: Apress Media LLC, 2015. ISBN 978-1-4842-0815-9.
27. Gackenheimer, C. Introduction to React. 2015.
28. Курыченко, В., Хрусталева, А. HTML5+CSS3. Basics of Modern Web Design. Publisher: "Nauka i Tekhnika", 2018. ISBN 978-5-217-05782-3.
29. Уманець, А., Яцишин, І., & Соловйов, Є. Інтерактивний вступ до програмування на мові Python. Київ: Факт, 2015. ISBN 978-617-7288-02-3.
30. Яцишин, І. Python для дітей та батьків. Київ: Факт, 2017. ISBN 978-617-7288-21-4.
31. Кузнецов, С., & Лут, М. Python і Django: програмування веб-застосунків. Київ: Діля, 2017. ISBN 978-617-7553-34-8.
32. Жук, В. Python. Підручник. Київ: БІНОМ, 2020. ISBN 978-617-585-267-4.
33. Кірас, Т. Python в науці та освіті. Київ: Діля, 2021. ISBN 978-617-09-5163-1
34. McKinney, W. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. O'Reilly Media, 2017. ISBN 978-1491957660.
35. VanderPlas, J. Python Data Science Handbook. O'Reilly Media, 2016. ISBN 978-1491912058.
36. Brownlee, J. Deep Learning for Computer Vision. Machine Learning Mastery, 2019. ISBN 978-1838823413.
37. Géron, A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, 2019. ISBN 978-1492032649.
38. Chollet, F. Deep Learning with Python. Manning Publications, 2018. ISBN 978-1617294433.
39. Hill, A. Learning Scientific Programming with Python. Cambridge University Press, 2016. ISBN 978-1107428225.
40. McKinney, W. Python for Data Science For Dummies. For Dummies, 2015. ISBN 978-1118843987.

41. Müller, A. C., & Guido, S. Introduction to Machine Learning with Python: A Guide for Data Scientists. O'Reilly Media, 2016. ISBN 978-1449369415.
42. Roche-Mushigian, A. Python Machine Learning Blueprints: Intuitive data projects you can relate to. Packt Publishing, 2015. ISBN 978-1784394757.
43. Beazley, D. Python Essential Reference. Addison-Wesley, 2009. ISBN 978-0672329784.



ДЕКЛАРАЦІЯ

про дотримання академічної доброчесності

Я, _____

Повністю вказується ПІБ та статус (посада для працівників, освітня (освітньо-наукова) програма – для здобувачів вищої освіти)

що нижче підписався, розуміючи та підтримуючи загально визнані засади справедливості, доброчесності та законності,

ЗОБОВ'ЯЗУЮСЬ:

дотримуватися принципів та правил академічної доброчесності, що визначені законодавством України, локальними нормативними актами Донецького національного університету імені Василя Стуса, положеннями, правилами, умовами, визначеними іншими суб'єктами, та не допускати їх порушення.

ПІДТВЕРДЖУЮ:

що мені відомі положення статті 42 Закону України «Про освіту»;

що у даній роботі не представляла/представляв чийсь роботи повністю або частково як свої власні. Там, де я скористалася/скористався працею інших, я зробила/зробив відповідні посилання на джерела інформації;

що дана робота не передавалась іншим особам і подається вперше, не порушує авторських та суміжних прав закріплених статтями 21-25 Закону України «Про авторське право та суміжні права», а дані та інформація не отримувались в недозволеній спосіб.

УСВІДОМЛЮЮ:

що ця робота може бути перевірена університетом на плагіат або інші порушення академічної доброчесності, в тому числі з використанням спеціалізованих сервісів;

що у разі порушення академічної доброчесності, до мене можуть бути застосовані процедури, передбачені законодавством України та Кодексом академічної доброчесності та корпоративної етики Донецького національного університету імені Василя Стуса, іншими локальними нормативними актами університету, та я можу бути притягнута/притягнутий до академічної відповідальності.

(дата)

(підпис)