

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук
(повна назва)

Кафедра _____ Програмної інженерії
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти _____ другий (магістерський)

Дослідження методів розпізнавання акордів для навчання гри на
музичному інструменті

(тема)

Виконав:

Випускник 2 курсу, групи ІПЗМ-19-1

Яровий М. В.

(прізвище, ініціали)

Спеціальність 121 – Інженерія програмного
забезпечення

(код і повна назва спеціальності)

Тип програми Освітньо-наукова

(освітньо-професійна або освітньо-наукова)

Керівник доцент, к.т.н. Назаров О. С.

(посада, прізвище)

Допускається до захисту

Зав. кафедри

_____ (підпис)

З.В. Дудар

(прізвище, ініціали)

2021р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
 Кафедра Програмної інженерії
 Рівень вищої освіти другий (магістерський)
 Спеціальність 121-Інженерія програмного забезпечення
 (код і повна назва)
 Тип програми освітньо-наукова програма
 Освітня програма Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:
 Зав. кафедри _____
 (підпис)
 « ____ » _____ 2021 р.

ЗАВДАННЯ
 НА КВАЛІФІКАЦІЙНУ РОБОТУ

студента Ярвого Микити Володимировича
 (прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів розпізнавання акордів для навчання гри на музичному інструменті

затверджена наказом університету від "26" березня 2021р. № 385

заповнюється вручну після отримання наказу

2. Термін подання студентом роботи до екзаменаційної комісії _____ 2021р.

3. Вихідні дані до роботи: розробити програмну систему для розпізнавання акордів. Використовувати: мову програмування Python, менеджер пакетів Conda.

4. Перелік питань, що потрібно опрацювати в роботі мета роботи, аналіз проблемної галузі і постановка задачі, огляд методів аналізу музичної інформації, методи розпізнавання акордів.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, слайдів, ілюстрацій: слайди презентації, рисунки та таблиці.

6 Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Спецчастина			

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка *
1.	Аналіз предметної галузі	26 березня 2021 р.	виконано
2.	Огляд існуючих методів	31 березня 2021 р.	виконано
3.	Розробка алгоритмів, проектування та розробка ПЗ	25 квітня 2021 р.	виконано
4.	Підготовка пояснювальної записки	06 травня 2021 р.	виконано
5.	Спецчастина	06 травня 2021 р.	виконано
6.	Підготовка презентації та доповіді	06 травня 2021 р.	виконано
7.	Попередній захист	13 травня 2021 р.	виконано
8.	Нормоконтроль, рецензування	07 травня 2021 р.	виконано
9.	Занесення диплома в електронний архів	10 травня 2021 р.	виконано
10.	Допуск до захисту у зав. кафедри	13 травня 2021р.	виконано
* заповнюється вручну після виконання чергового пункту			

Дата видачі завдання _____ 2021 р.

Студент _____
(підпис)

Керівник роботи _____ доцент, к.т.н. Назаров О.С.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Кваліфікаційна робота магістра містить: 61 с., 32 рис., 5 табл., 18 джер.

PYTHON, AUDIO, SOUND, CHORDS, RECOGNITION, KERAS

Метою роботи є методи, алгоритми аналізу, проектування та розробка програмної системи для розпізнавання акордів у сфері пошуку музичної інформації, ПМІ.

Методи розробки базуються на інструментах розробки програмних систем на мові програмування Python, використанням Ubuntu 20.04 у якості операційної системи та Visual Studio Code - засіб для створення, редагування та підтримки сучасних програмних систем. Для створення нейромережі для аналізу даних використовувалась відкрита нейромережева бібліотека Keras.

В результаті роботи було розглянуто методи та алгоритми аналізу, проектування та розробки програмної системи для розпізнавання акордів, яка представляє собою програмну систему на мові Python.

PYTHON, AUDIO, SOUND, CHORDS, RECOGNITION, KERAS

The aim of the work is methods, algorithms of analysis, design and development of a software system for chord recognition in the field of music information retrieval, MIR.

Development methods are based on tools for developing software systems in the Python programming language, using Ubuntu 20.04 as an operating system and Visual Studio Code - a tool for creating, editing and maintaining modern software systems. The Keras open neural network library was used to create a neural network for data analysis.

As a result, the methods and algorithms of analysis, design and development of a software system for chord recognition, which is a software system in Python, were considered.

Я, Яровий Микита Володимирович, студент гр. ІІЗм-19-1, здобувач вищої освіти на другому (магістерському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Дослідження методів розпізнавання акордів для навчання гри на музичному інструменті», що буде представлена в екзаменаційну комісію для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу EIAr KhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений (а) з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

РЕФЕРАТ / ABSTRACT.....	3
ЗМІСТ.....	5
ВСТУП.....	7
1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ.....	10
1.1 Напрямок дослідження.....	10
1.2 Аналіз аналогів.....	12
1.3 Аналіз проблем та актуалізація рішень.....	15
2 ПОСТАНОВКА ЗАДАЧІ.....	18
3 МЕТОДИ ДОСЛІДЖЕННЯ.....	19
4 МАТЕМАТИЧНА МОДЕЛЬ.....	21
4.1 Базовий алгоритм.....	21
4.2 Модель для оцінки алгоритмів.....	22
5 ПРОВЕДЕННЯ ЕКСПЕРИМЕНТУ.....	25
5.1 Методологія експерименту.....	25
5.2 Алгоритм на відповідність шаблону.....	26
5.3 Алгоритм з використанням ПММ.....	32
5.4 Алгоритм з використанням нейронної мережі над хромаграмою.....	35
5.5 Покращений алгоритм з використанням нейронної мережі.....	39
6 ОПИС РОЗРОБЛЕННОЇ ПРОГРАМНОЇ СИСТЕМИ.....	40
7 РЕЗУЛЬТАТИ РОБОТИ.....	41
ВИСНОВКИ.....	43
ПЕРЕЛІК ПОСИЛАНЬ.....	44

ДОДАТОК А Перелік посилань відповідно до наукових досліджень кафедри
..... **Ошибка! Закладка не определена.**

ДОДАТОК Б Звіт результатів перевірки на унікальність тексту в мережі
інтернет та базі ХНУРЕ **Ошибка! Закладка не определена.**

ДОДАТОК В Наукова апробація результатів роботи **Ошибка! Закладка не
определена.**

ДОДАТОК Г Слайди презентації **Ошибка! Закладка не определена.**

ДОДАТОК Д Електронні матеріали до роботи на СДО **Ошибка! Закладка не
определена.**

ВСТУП

Світ навколо нас наповнений різноманіттям. Все це різноманіття ми можемо відчувати за допомогою певних «сенсорів», які дані нам від природи. Над природою більшості явищ ми не задумуємось. Вони буденні, ми звикли до них. Явища, що оточують нас багаті на різноманітні звуки, ці звуки розпізнаються людиною та всіма живими істотами зі слухом і служать для розпізнавання позиції тіла у просторі та для спілкування між собою. Ми не задумуємось над тим, що таке звук, ми ніби всі знаємо що це.

Терміном звук описують коливальний рух частинок середовища, що поширюється у вигляді хвиль у речовинах в різних фізичних станах. Зазвичай, цим терміном визначають коливання, які сприймаються живими організмами з спроможністю їх відчувати [1]. В цьому випадку маються на увазі збурення, що поширюються в речовині, зокрема в повітрі. Органи слуху людини сприймають звуки у певному частотному діапазоні, який у порівнянні з іншими частотними діапазонами є доволі вузьким. Слуховий апарат багатьох тварин сприймає звуки в інтервалі частот, який часто ширше за людський. В загальному випадку цим терміном визначається процес поширення збурень в різних за фізичними властивостями середовищах, в яких існує відновлювальна сила, що намагається повернути збурену частинку в положення рівноваги, це сила пружності в більшості випадків. Вивченням закономірностей генерації, поширення та сприйняття звуків в різних середовищах займається така наукова дисципліна як акустика.

Сприйняття цих звукових коливальних рухів частинок повітря слуховим апаратом людини зумовлює поділ вушним равликом звуків на приємні, милозвучні, гармонійні (наприклад, спів пісень, звуки музичних інструментів, спів пташок) та звуки зі специфічним спектральним наповненням, часто небажані та дратівні, які визначаються як шум, а також крик, писк.

Зміщення в просторі частинок середовища у звукових збуреннях описуються величиною під назвою «амплітуда коливань». Для опису швидкості коливань

використовується така характеристика як частота. Звук, що сприймається органами слуху людини та різних тварин це лише невелика частка коливань, що присутні в навколишньому світі. Людське вухо може сприймати лише звуки в певних межах інтервалах амплітуд та частот.

Для багатьох випадків практичного використання звуків велике значення мають звуки, в яких основна частина енергії зосереджена в області частот, що не сприймаються людським вухом. В зв'язку з цим окремо існують крайні випадки типу збурень (високих та низьких частот відповідно - ультразвук та інфразвук).

Термін звук використовується для характеристики процесів поширення коливань в матеріальних середовищах (не важливо рідина це чи тверда речовина), що мають пружні властивості. В таких середовищах на будь-яку частинку, зміщену з положення рівноваги, діє протидійна сила пружності, яка намагається повернути її в супер позицію, а саме початкове положення. Характер руху частинок середовища при цьому може бути різним. При коротких збуреннях (постріл, падіння чогось на підлогу та ін.) здійснюють короткочасні рухи, після яких частинки повертаються до початкового рівноважного стану дуже швидко. При дії довготривалих, частинки середовища здійснюють коливальні рухи доволі довго поблизу положення рівноваги чим.

Звук реєструють за допомогою приладів запису звуку, а саме мікрофонів — приладів, що перетворюють коливання у середовищі на електричні. Зареєстровані коливання середовища можна передати на віддаль різними засобами телекомунікації — за каналом зв'язку, наприклад по інтернету, або в записі. Переданий або записаний звук відтворюється за допомогою гучномовців, які перетворюють електричні коливання у звукові хвилі. В основному систему можна вважати чорною скринькою. Ви подаєте деяку кількість вхідних даних у систему, і вона повертає деякий результат. Те, що подається в систему, найчастіше не просто одне значення або набір значень, ні, це потік даних. Такий залежний від часу вхідний потік називається сигналом [2].

При записі звуку розуміють запис дискретного сигналу. Дискретний сигнал (лат. *Discretus* - «переривчастий», «розділений») - сигнал, який є переривчастим (на

відміну від аналогового) і який змінюється в часі і приймає будь-яке значення зі списку можливих значень. Список можливих значень може бути безперервним або квантованим [3].

Таким чином метою роботи є реалізація програмної системи для аналізу існуючих методів розпізнавання акордів, яка потім може бути використана для вивчення гри на музичному інструменті.

Під основним інструментом, який буде використовуватися для навчання слід розуміти електрогітару. Запис звуку проводився з звукознімачів розміщених на електрогітарі.

Середою розробки було обрано Visual Studio Code з використанням мови програмування Python та менеджера пакетів Conda.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Напрямок дослідження

Пошук музичної інформації (MIR) передбачає пошук та організацію великих колекцій музики або музичної інформації відповідно до їх відповідності конкретним запитам. Це особливо актуально, враховуючи величезну кількість музичної інформації, доступної у цифровому форматі, та популярність цифрових послуг, пов'язаних з музикою. За останні чотири роки було представлено велику серію конференцій на цю тему, наприклад, міжнародну конференцію з пошуку музичної інформації (ISMIR), а також численні симпозиуми та семінари, що збирають все більшу спільноту багатoproфільних дослідників, включаючи бібліотекарів, інформатиків, музикантів, серед інженерів-електронників та музикознавців. Крім того, зважаючи на очевидну комерційну привабливість, більшість власників та розповсюджувачів медіа-контенту (наприклад, Philips, Sony, Apple) беруть активну участь у наукових дослідженнях у цій галузі, тоді як численні бібліотеки прагнуть включити певну форму підтримки MIR у свої онлайн-цифрові послуги.

Прості системи MIR отримують дані відповідно до текстового запиту, введеного користувачем, наприклад «Герої Девіда Боуї». У цих випадках текст порівнюється з текстовими даними, пов'язаними з альбомами та композиціями, завдяки чому система по суті не відрізняється від будь-якої текстової пошукової системи (наприклад, Google, Yahoo). Однак, враховуючи характеристики вмісту, що отримується, існує потреба в системах, які можуть приймати «музичні» запити, такі як партитури, заспівані мелодії (запит гудінням) або записані аудіо сегменти (запит на прикладі). Ця пропозиція стосується останнього випадку.

Метою запиту на прикладі є отримання музичних творів із великої колекції цифрового музичного вмісту за їх схожістю з прикладом аудіодокумента. Вміння виконувати запити на прикладах є важливою вимогою до систем MIR. Це ставить численні проблеми, включаючи обчислювальні та складні питання, дизайн

відповідного тестового стенду та вибір належного представлення звуку в запиті та музичній колекції. Вибір звукового представлення диктує схожість, яку може виявити система. Існуючі підходи до вибору репрезентації можна умовно розділити на такі, що намагаються виявити низькорівневу (акустичну) подібність, і ті, що мають на меті кількісно визначити схожість високого рівня (наприклад, ноти, мелодії тощо). Дослідження музичної інформації привертає велику увагу. Це має давню історію, про що свідчать спроби використовувати комп'ютер для складання музики з часів винаходу комп'ютера. Результати досліджень музичної інформації широко поширилися по всьому суспільству включаючи синтезатори, які стали важливими для виробництва популярної музики, та послуги з розповсюдження музики через мобільні телефони. Сфера досліджень музичної інформації охоплює всі аспекти музики та всі аспекти музичної діяльності людей і пов'язана з різними темами, такими як обробка сигналів, транскрипція, сегрегація джерел звуку, ідентифікація, аналіз, розуміння, пошук, рекомендація, класифікація, розподіл, синхронізація, перетворення, обробка, узагальнення, композиція, аранжування, написання пісень, виступ, супровід, розпізнавання партитур, синтез звуку, синтез співу, генерація, допомога, кодування, візуалізація, взаємодія, користувальницькі інтерфейси, бази даних, анотації та соціальні теги до музики.

Однак важливість досліджень музичної інформації не визнавалася до 1990-х років. Це різко змінилося після 2000 року, коли широка громадськість почала слухати музику на комп'ютерах у повсякденному житті. Зараз він широко відомий як важлива наукова сфера, і нові дослідники постійно приєднуються до цієї галузі у всьому світі. Хоча дослідження музичної інформації іноді потребували аргументів, щоб визнати їх серйозними дослідженнями, а не дослідженнями задля розваги більше десяти років тому, такі помилкові уявлення залишаються в минулому. Ця зміна була спричинена тим фактом, що широка громадськість усвідомлює, що вся музика врешті-решт буде оцифрована, створена, розповсюджена, використана, спільно використовувана тощо.

Подальший попит буде на нові інтерфейси прослуховування музики, пошук та рекомендації. В академічному плані одна з причин, за якою багато дослідників

беруть участь у цій галузі, полягає в тому, що основною невирішеною проблемою є розуміння складних музичних звукових сигналів, що передають вміст, утворюючи тимчасову структуру, тоді як множинні звуки взаємопов'язані. Крім того, досі існують привабливі невирішені проблеми, які ще не зачіпались, і це поле є скарбницею дослідницьких тем. У цій роботі розглядаються деякі важливі проблеми, які можуть у подальшому збільшити як привабливість, так і соціальний вплив досліджень музичної інформації в майбутньому. Зверніть увагу, що деякі обговорення в цьому документі навмисно провокують, щоб викликати суперечливі дискусії та стимулювати нові ідеї.

1.2 Аналіз аналогів

У сфері програмного забезпечення існує багато аналогів. Умовно всі продукти можна поділити на групи: безкоштовні, умовно-безкоштовні та платні. До безкоштовних, які знаходяться в вільному інтернет доступі можна віднести ряд бібліотек, які аналізують звук використовуючи алгоритмічні методи та/або нейронні мережі. Умовно-безкоштовні використовують наукові роботи, але не пропонують код для перегляду. Платні не повідомляють які наукові праці використовують у своїх програмах, код програм також недоступний для користувачів систем.

Всі сервіси мають різне використання, яке відповідає функціоналу, який вони пропонують. Більшість систем, які існують на ринку служать засобами для розпізнавання акордів у записі. Цей запис використовується для навчання гри на музичному інструменті в тому чи іншому вигляді. Більшість сервісів також пропонують вже проаналізовані пісні відомих музикантів, але вони не мають змоги розпізнавати що саме зіграв користувач сервісу.

Серед аналогів можна виділити: Yalp.io, ChordU, Chordify, Rocksmith, Timbro. Наведемо їх опис в таблиці 1.1.

Таблиця 1.1 – Список сервісів для навчання

Назва	Платформа	Короткий опис
Yalp.io	Web, mobile	Платформа надає акорди до більшості відомих пісень, також надає змогу отримати акорди для завантаженого аудіо файлу. Також дозволяє спостерігати «перехід» від одного акорду до іншого при програванні файлу (тільки в преміум акаунті).
ChordU	Web	Сайт пропонує користувачу завантажити аудіо файл та отримати перелік акордів, які граються в записі (тільки в преміум акаунті)..
Chordify	Web	Ця можливість з'являється при виборі пісні, якої користувач хоче навчитись. Присутній перехід від одного акорду до іншого в процесі гри музичного файлу (тільки в преміум акаунті)..
Rocksmith	Desktop	Rocksmith позиціонує себе як гра для навчання гри на гітарі. Є платним програмним забезпеченням, яке поширюється через платформу Steam. Дозволяє в реальному часі навчатись гри на гітарі. Платформа аналізує, що зіграв користувач.
Timbro	Mobile	Сервіс надає змогу навчатись гри на гітарі на фіксованих піснях, перелік з часом доповнюється. Є умовно-безкоштовним, більшість з пісень є платними.

Сформуємо таблицю для порівняння аналогів та їх функцій. Серед порівнюваних сервісів виділимо найбільш значимі функціональні властивості, їх зазначимо в таблиці 1.2. Незважаючи на те, що цілі в них схожі, сервіси надають все ж таки різні можливості для взаємодії. Всі наведенні вище сервіси мають за

основний намір – навчити користувача грі на музичному інструменті, але не всі з них використовують всі можливості для цього.

Таблиця 1.2 – Порівняння сервісів

Назва сервісу	Присутній перехід від одного акорду до іншого	Дозволяє аналізувати свої файли для навчання	Аналізує те, що грає користувач	Надає можливість переходу між півтонами	Розпізнавання оффлайн
Yalp.io	+	+	-	+	-
ChordU	+	+	-	+	-
Chordify	+	+	-	+	-
Rocksmith	+	-	+	-	+
Tibmro	+	-	+	-	+

Слід зауважити, що засобів для прослуховування музичного інструменту, зокрема гітари або електрогітари в зазначених сервісах можна виділити декілька.

Таблиця 1.3 – Засоби для взаємодії з музичним інструментом

Назва сервісу	Без підключення	Через мікрофон	З підключенням через шнур	Власна розробка для підключення
Yalp.io	+	-	-	-
ChordU	+	-	-	-
Chordify	+	-	-	-
Rocksmith	+	+	+	+
Tibmro	-	+	+	-

Як зазначено в Rocksmith існує можливість використовувати всі зазначені способи підключення, а також існує їх власна розробка, яка повинна покращити взаємодію з грою. Загалом розробка виглядає так само як звичайний шнур, але як зазначають розробники, надає змогу збільшити точність розрахунків та точності розпізнавання зіграних на інструменті нот, акордів, хаммер-онів та інших прийомів, які специфічні для гітари.

Також важливо зазначити, що Rocksmith також використовує засоби гейміфікації, що дозволяє доволі швидко почати навчання гри на гітарі. Гейміфікація – це використання ігрових підходів для неігрових процесів. Гейміфікація допомагає достукатися до аудиторії, яка звикла грати в комп'ютерні ігри і спілкуватися в соціальних мережах. Саме вона, гейміфікація, давно і вдало використовується в маркетингу і керуванні персоналом, та є найбільш цінною в освіті [4]. Прийоми гейміфікації навіть застосовують у школах [5].

1.3 Аналіз проблем та актуалізація рішень

Розпізнавання акордів відповідає великому інтересу у спільноті MIR (дослідження музичної інформації), оскільки гармонічний вміст - це описова риса середнього рівня (західної) музики, яка може бути використана безпосередньо (наприклад, для створення нотних зошитів для музикантів) та як основа для вищих рівнів задач таких, як ідентифікація кавер пісні, виявлення ключа або аналіз гармонік.

Більшість систем розпізнавання акордів мають спільну архітектуру, що складається з двох основних етапів: вилучення особливостей та узгодження шаблонів та двох необов'язкових підкаскадів: попередньої та подальшої фільтрації. Розуміння взаємодії між цими базовими компонентами дуже важливо не тільки для досягнення оптимальних показників, але й для оцінки потенціалу та обмежень

системи. На жаль, немає досліджень, які б достатньо оцінювали ефекти різних підходів до кожного етапу обробки та взаємодії між цими етапами. У цій роботі ми намагаємось усунути цей недолік шляхом систематичного оцінювання, що охоплює широкий спектр методів, що використовуються для кожного етапу обробки. У нашому дослідженні ми виявили, що фільтрування має значний вплив на продуктивність, але надання інформації про музичний контекст у матриці переходів стає спірним через необхідність забезпечення безперервності оцінок. Було виявлено, що переваги використання складних моделей акордів можуть бути в значній мірі компенсовані відповідним вибором функцій. Крім того, початковий розрив продуктивності між різними характеристиками не був повністю компенсований будь-якими наступними стадіями обробки.

Більшість систем розпізнавання акордів дотримуються загального конвеєру вилучення ознак, узгодження шаблонів та декодування послідовності акордів (також зване постфільтрацією). У цій роботі ми зупинимось на першому кроці в цій трубопроводній програмі: вилученні особливостей [6].

Два спостереження приводять нас до вивчення кращих можливостей розпізнавання акордів:

— Можливості моделей акордів для розпізнавання шаблонів обмежені. У роботі [6] Чо і Белло дійшли висновку, що відповідні риси значною мірою викупувають переваги складних моделей акордів.

— Можливості постфільтрації обмежені. В [6, 7] показано, як методи постфільтрації корисні, оскільки вони забезпечують безперервність окремих акордів, а не надають інформацію про переходи акордів. Включення такої інформації не суттєво покращило результати розпізнавання в обох дослідженнях. Дослідник в [7] також кількісно зазначав, що у популярній музиці «прогрес акордів менш передбачуваний, ніж здається», і, отже, знання історії акордів не значно звужує можливості для наступного акорду. Враховуючи ці очевидні обмеження етапів узгодження зразків та етапів фільтрації, не дивно, що вони лише частково компенсують розрив між характеристиками. Тому нам доводиться обчислювати кращі характеристики, якщо ми хочемо покращити розпізнавання акордів.

Найпопулярнішою функцією, яка використовується для розпізнавання акордів, є Хромаграма. Хромаграма містить часовий ряд векторів кольоровості, які представляють вміст гармоніки в певний час в аудіо як $s \in \mathbb{R}^{12}$. Кожен s_i означає клас висоти, а його значення вказує на поточну популярність відповідного класу висоти. Вектори кольоровості обчислюються, застосовуючи банк фільтрів до частотно-часового подання звуку. Це подання є результатом або короткочасного перетворення Фур'є (STFT), або перетворення з постійним q (CQT), причому останнє є більш популярним через більш тонку роздільну здатність частоти в області нижчих частот.

Хромаграми - це стислі описи гармонік, оскільки вони кодують якість тону і нехтують висотою тону. Теоретично це обмежує їх репрезентативну силу: без октавної інформації не можна розрізнити напр. акорди, що складаються з тих самих класів висоти, але мають іншу басову ноту (наприклад, G проти G / 5 або A: sus2 проти E: sus4).

Тут важливо зазначити в чому, власне, різниця між певними акордами. Призупинений акорд (або сус акорд) - це музичний акорд, у якому (мажорний чи мінорний) третій опускається і замінюється ідеальним четвертим або, рідше, мажорним другим. Відсутність мінорної або мажорної третини в акорді створює відкритий звук, тоді як дисонанс між четвертою і п'ятою або другою і коренем створює напругу. При використанні символів популярної музики вони позначаються символами "sus4" та "sus2". Наприклад, підвішені четвертий та другий акорди, побудовані на C (C – E – G), записані як Csus4 та Csus2, мають висоти звуку C – F – G та C – D – G відповідно [8].

2 ПОСТАНОВКА ЗАДАЧІ

За результатами проведеного аналізу предметної області та виявивши найбільш проблемні місця, були сформовані вимоги щодо створюваної програмної системи та сформована постановка задачі.

Необхідно створити систему для навчання гри на музичному інструменті, наприклад гітарі або електричній гітарі, яка буде використовуватись як настільний застосунок. Користувач буде отримувати перелік акордів які присутні в музикальному файлі або які були зіграні в певний момент часу та записані на мікрофон.

Система повинна мати змогу розпізнавати акорди з більшою точністю ніж існуючі аналоги. Це означає, зокрема, більшу точність для розпізнавання мажорних та мінорних акордів.

Система повинна мати змогу розпізнавати акорди в файлі, який надається користувачем, та надавати як результат проміжки, на яких звучить той чи інший акорд.

Система повинна мати змогу, розпізнавши той чи інший акорд надати час його початку та кінця, також дуже важливо щоб система проводила аналіз доволі швидко, оскільки планується її використання в реальному часі.

Все вище зазначене підводить нас до головного – необхідно створити програмну систему для навчання гри на музичному інструменті, наприклад гітарі, з функціями: розпізнавання акордів в заданому файлі, розпізнавання акордів в реальному часі.

Програмна система має бути написана на мові програмування Python, як програмна система для стаціонарної обчислювальної системи з використанням менеджера пакетів Conda, що дозволить у разі необхідності використання різної версії мови Python швидко та зручно налаштувати середу розробки та переключатись між ними у разі необхідності.

3 МЕТОДИ ДОСЛІДЖЕННЯ

Методи дослідження, які використовуються в даній роботі – теоретичний, емпіричний на основі ряду експериментів.

Теоретичний метод дослідження є аналіз існуючих теоретичних відомостей, посібників, статей, наукових праць за темою розпізнавання акордів. Сюди також можна віднести збір всіх вхідних даних, необхідних для проведення експерименту та проектування моделі.

Теоретичний та емпіричний методи дуже тісно пов'язані між собою. На основі результатів експерименту створюється гіпотеза, яку можна спростувати чи підтвердити наступними експериментами. У випадку постійного повторювання результатів експерименту, можна казати про наявність залежності.

У нашому випадку, в процесі моделювання програмної системи ми постійно стикаємось з необхідністю оновлення моделі або створення нової. Під час тренування нейронних мереж, дослідник постійно стикається з процесом «навчання» нейронної мережі на певних тестових даних. У випадку коли точність результатів стає доволі високою та достатньою для певних цілей, дослідник може почати використовувати модель для проведення експерименту на реальних даних, які можуть мати випадкові або зайві дані. Система, основана на нейронних мережах, повинна розпізнавати з зайві дані та зменшувати їх вплив на результат вимірів.

Для того, щоб забезпечити ретельне тестування алгоритмів розпізнавання акордів та гармоній, необхідно мати великий набір даних, переписаний вручну, який може бути використаний як інформація про основну істину для порівняння з результатами комп'ютерного алгоритму. У своїй праці [9], Кристофер Харте, описує три методи для розпізнавання акордів без використання нейронних мереж, а також наводить ряд принципів для оцінки правильності розпізнавання акордів. Також в цій праці надається опис того, яким саме чином було створено та описано

набір даних, який є ніщо інше як 12 альбомів групи Beatles з детальним описом акордів, які використовувались в конкретній пісні в конкретний проміжок часу.

Ми не будемо використовувати цей датасет у своїй роботі, оскільки проводити аналіз методів, які навчались на цьому датасеті є не ефективним і не покаже дійсного стану проблеми.

Загалом, в нашій роботі ми плануємо використовувати ті ж принципи для виділення акордів та оцінки вірності «розпізнаного» та реального акорду, який був використаний в конкретний момент часу.

Основним з методів дослідження буде експериментальний, оскільки необхідно порівняти як існуючі алгоритми ведуть себе в заданій задачі розпізнавання акордів. Буде проведено порівняння алгоритмів з використанням нейронних мереж та без їх використання.

4 МАТЕМАТИЧНА МОДЕЛЬ

4.1 Базовий алгоритм

У цьому дослідженні буде використано декілька підходів до використання конкретного алгоритму, а саме FFT (Fast Fourier Transform). Оскільки річ йде про алгоритми, які використовуються для розпізнавання акордів, то необхідним для оцінки алгоритму необхідна оцінка обчислювальної складності.

Асимптотична складність (продуктивність) визначається функцією, яка вказує, наскільки погіршується робота алгоритму з ускладненням поставленого завдання. Таку функцію записують у круглих дужках, випереджаючи великою літерою O [10].

Використання великої літери O (або так звана O -нотація) прийшло з математики, де її застосовують для порівняння асимптотичної поведінки функцій. Формально $O(f(n))$ означає, що час роботи алгоритму (або обсяг займаної пам'яті) росте в залежності від обсягу вхідних даних не швидше, ніж деяка константа, помножена на $f(n)$.

Для швидкого перетворення Фур'є в загальному випадку характерна оцінка алгоритму: $O(n \log(n))$. Тригонометричну формулу можна побачити нижче:

$$X(m) = \sum_{n=0}^{N-1} x(n) \left(\cos \frac{2\pi n m}{N} - j \sin \frac{2\pi n m}{N} \right),$$

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 початкової послідовності, відповідно і кількість частотних відліків результату перетворення.

Дискретне перетворення Фур'є може бути застосоване в широкому спектрі задач. Нас же цікавить використання цього перетворення в задачах розпізнавання звуку з використанням нейронних мереж [11].

В своїй статті [12] Джон Паркер проводить аналіз використання різних реалізацій дискретного перетворення Фур'є. Загалом, він порівнює три реалізації, присутні в бібліотеках мови програмування Python: `numpy`, `pyfftw`, `cufft`. Цей аналіз дає нам зрозуміти, що використання потужностей відеокарти може значно прискорити розрахунок перетворення Фур'є, яке нам знадобиться в подальшому. Але в рамках цієї роботи, ми будемо використовувати лише версію `numpy`.

4.2 Модель для оцінки алгоритмів

Для відображення ефективності алгоритмів, які будуть розглянуті в цій роботі, необхідно надати математичну модель, якої буде достатньо, щоб відобразити всі характеристики та можливості кожного з алгоритмів. Математична модель є одним з основних інструментів дослідження, оскільки за допомогою неї можна розрахувати та проаналізувати результати роботи алгоритмів яких ми досліджуємо, та отримати результати на основі яких зробити певні висновки.

Оскільки алгоритми, які будуть розглядатись далі мають різні властивості (як то розпізнавання акорду з певною точністю на проміжку часу або розпізнавання акордів, яке проводить розпізнавання швидше, але з певними втратами у точності) то необхідно врахувати всі ці властивості для отримання однієї комплексної оцінки роботи методу. Для цього введемо наступні метрики. Основна метрика яка буде відображати наскільки алгоритм справляється з поставленою задачею (Y):

$$Y = \frac{n*100}{N} \quad (4.1)$$

де n – кількість акордів, що було розпізнано вірно;

N – загальна кількість акордів, що були зіграні.

Оскільки деякі алгоритми дозволяють отримати також довжину відрізка на якому було зіграно певний акорд, будемо використовувати для таких методів наступну метрику ($Y(i)$)

$$Y(i) = \frac{l(i)}{L(i)} \quad (4.2)$$

де $Y(i)$ – ефективність розпізнавання акорду на відрізку;

i – відповідно номер відрізка, на якому є акорд;

N – загальна кількість акордів, які використовуються;

$l(i)$ – розпізнана довжина акорду;

$L(i)$ – реальна довжина акорду на відрізку.

Вищезазначені метрики (4.1), (4.2) можна об'єднати для того, щоб отримати більш точну оцінку для методів, які можуть розпізнавати акорди в записі, з вказанням початку та кінця акорду. Таким чином отримаємо:

$$Y_d = \sum_{i=1}^N \frac{l(i)}{L(i)} * \frac{100n}{N} \quad (4.3)$$

де Y_d – ефективність методу, який розпізнає акорди на відрізку;

i – відповідно номер відрізка, на якому є акорд;

N – загальна кількість акордів, які використовуються;

$l(i)$ – розпізнана довжина акорду;

$L(i)$ – реальна довжина акорду на відрізку;

n – кількість вірно розпізнаних акордів.

В рамках даного дослідження також дуже важливо розрахувати значення часу, яке використовується для розпізнавання акордів, оскільки одна з цілей дослідження – знаходження ефективного підходу для навчання гри на гітарі, а це в свою чергу означає розпізнавання акордів в реальному часі.

Для цього вводимо просту метрику виміру часу, яка присутня в більшості систем з оцінки роботи алгоритму, нас цікавить саме середнє значення.

$$\Delta t_{\text{сер}} = \frac{\sum_{i=1}^N t_{\text{кін}}(i) - t_{\text{поч}}(i)}{N} \quad (4.4)$$

де $\Delta t_{\text{сер}}$ - середнє значення часу, за яке проводилося розпізнавання акорду;

N – загальна кількість акордів, які використовуються;

$t_{\text{кін}}(i)$ - час закінчення розпізнавання i -го акорду;

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

Таким чином були введені метрики для даного дослідження, за цими метриками буде проведено оцінку працездатності та якості розпізнавання акордів алгоритмами, що підлягають дослідженню.

5 ПРОВЕДЕННЯ ЕКСПЕРИМЕНТУ

5.1 Методологія експерименту

За допомогою методології експерименту можна визначити його структуру, постановку та послідовність дій до виконання, кожне з яких ґрунтується на сфері застосування. Суть експерименту в цій роботі полягає в порівнянні існуючих методів, поясненні їх основних принципів та порівнянні їх загальної ефективності в поставленій задачі. Мета експерименту віднайти найбільш ефективний підхід для розпізнавання акордів для подальшого його використання для навчання гри на музичному інструменті.

В експерименті будуть приймати участь 4 найбільш актуальні методи, які базуються на швидкому перетворенні Фур'є. Ці методи будуть тестуватись на одному наборі даних, які були спеціально підготовані для експерименту.

В якості музичного інструменту використовувалась електрогітара, дані з якої зчитувались безпосередньо за допомогою звуко-зчитувачів, які є на електрогітарі, та записувались через зовнішню звукову карту.

Для проведення експерименту було попередньо записано 80 акордів. Серед яких присутні мажорні, мінорні та їх сус версії. Таким чином для всіх алгоритмів буде використовуватись один і той же підхід та набір даних. На цих акордах ми протестуємо роботу кожного з алгоритмів, попередньо пояснивши принципи роботи кожного з алгоритмів.

Далі буде підведено підсумки експерименту та всі дані будуть занесені для порівняння в таблицю. Дані для порівняння будуть відображувати значення метрик, які були наведені в попередньому пункті, а саме: ефективність розпізнавання акорду того, чи іншого алгоритму, а також час, витрачений на його розпізнавання.

Після оцінки ефективності розпізнавання акордів кожним з запропонованих алгоритмів буде проведено аналіз проведеного експерименту для підведення підсумків та розуміння які підходи в реалізації алгоритмів можна об'єднати або

вдосконалити для покращення результату. Також буде дано оцінку алгоритмів у розрізі використання для розпізнавання акордів у реальному часі.

5.2 Алгоритм на відповідність шаблону

Базовий алгоритм, який використовується в даному підході до аналізу звуку полягає у наступному:

- подаємо сигнал на аналіз (під сигналом слід розуміти звуковий потік);
- проводимо спектральний аналіз для отримання частотних компонентів сигналу;
- використовуємо перетворення Фур'є для того, щоб перетворити сигнал в спектрограму (хромограму);
- проводимо частотну фільтрацію (видаляємо звук нижче певного порогу);
- знаходимо верхні значення, так звані локальні максимуму;
- проводимо обчислення контрольної частоти;
- проводимо співвідношення висоти звуку до розрахованої контрольної частоти. Таким чином проводимо визначення класу висоти тону за значенням частоти;
- нормалізація властивостей за кадрами (тут зазвичай рекомендують збільшувати максимальне значення вікна для того, щоб усунути залежність від загальної гучності).

Метод, який було використано в дослідженні описано в [13]. Запропонована реалізація працює за принципом розпізнавання акорду за кадрами, що означає, що в алгоритм кожен кадр аналізує незалежно від попереднього, таким чином втрачає результат початкового розпізнавання.

Під час експерименту над цим алгоритмом це було помітно від самого початку. Що важливо, так це те, що в тестових даних на вхід подається один акорд, який лунає до кінця звукової доріжки. Таким чином, можна побачити, що

ефективність цього алгоритму дуже залежить від розміру рамки, для якої застосовується метод. Також даний алгоритм залежить від того, з якої точки на звуковій доріжці він почав свою роботу (що знову ж таки підкреслює важливість використання дещо більшого значення для рамки аналізу. Наводимо приклад розпізнавання акорду Em на рисунку 5.1.

```

=====
Working with em_7.wav
=====
Time (s) Chord
0.0 N
0.6501587301587302 Em
1.3003174603174603 N
1.9504761904761905 N
2.6006349206349206 N
3.250793650793651 N
3.900952380952381 N
4.551111111111111 N
5.201269841269841 N
5.851428571428571 N
6.501587301587302 N
7.151746031746032 N
7.801904761904762 N
8.452063492063493 N

```

Рисунок 5.1 – Приклад тестування методом відповідності шаблону

Загальна оцінка, яку було отримано для даного алгоритму при початковій ширині вікна 4096 складає:

$$Y = 11\%$$

$$Y_d = 5\%$$

$$\Delta t = 505ms$$

Метрику Y пораховано, беручи за увагу моменти, коли алгоритм розпізнав акорд, незважаючи на те, що не розпізнав повністю його тривалість (в такому випадку все, що було після вдалого розпізнавання акорду було знехтуване).

При збільшенні вікна до значення 8192 точність дещо зростає:

$$Y = 12\%$$

$$Y_d = 7\%$$

$$\Delta t = 244ms$$

Метрику Y пораховано так само, як і в попередньому випадку.

Таким чином, можна побачити ріст двох метрик, в залежності від того, який розмір вікна для розпізнавання акорду було використано, разом з тим трошки зменшувалось і середнє значення часу, яке було використано на розпізнавання, оскільки зростала ширина проміжку, який аналізується. Чому зросло Y_d ? Ріст цієї метрики пов'язаний з тим, що при збільшенні вікна, алгоритм покриває більший проміжок часу, тим самим зменшуючи кількість кадрів в записі. Тобто, на початку, наприклад, було 10 кадрів шириною 8192, при збільшенні вдвічі ширини кадру, ми зменшили їх загальну кількість вдвічі, що вплинуло на збільшення метрики.

Спробуємо збільшувати ширину кадру і далі, аж до 65536 наведемо результати в загальній таблиці 5.1.

Таблиця 5.1 – Результати проведення експерименту для алгоритму

	Ширина вікна				
Метрика	4096	8192	16384	32768	65536
Y	11%	12%	14%	10%	7%
Y_d	5%	7%	10%	22%	12%
Δt	505ms	244ms	157ms	108ms	87ms

Що ж, результати доволі несподівані, хоч ми і помічали певний ріст метрик по мірі збільшення значення ширини вікна, але в піковому значенні отримали доволі несподіваний результат. Так само, як і в метриках для ширини вікна 32768.

Що це означає? Це означає наявність залежності метрики від «часу» в якому було почато розпізнавання акорду. Це було помітно вже на початкових значеннях ширини вікна, при 4096.

Для демонстрації цього ефекту наведемо приклад розпізнавання акорду на значенні ширини вікна 4096, а саме розпізнавання акорду Cm (див. рис. 5.2).

```

=====
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

```

Рисунок 5.2 – Приклад розпізнавання акорду Cm

Як можна побачити на рисунку 5.2, алгоритм спочатку розпізнав акорд як C, а вже потім як акорд Cm, що ще раз доказує наявність залежності алгоритму від початкової точки для розпізнавання.

Слід зазначити, що в експерименті були використані також сус акорди. Для сус акордів було проведено дещо інше дослідження, оскільки було дуже цікаво, яким чином даний алгоритм поведе себе при розпізнаванні сус акордів.

Впродовж всього експерименту, даний алгоритм також створював РСР зображення, яке і відображує який саме акорд розпізнано. В якості сус акордів було використано сус акорди для Em акорду, а саме Emsus, Emsus2, Emsus3.

Наводимо приклади цікавих моментів, що були помічені під час досліду на сус акорди на рисунку 5.3.

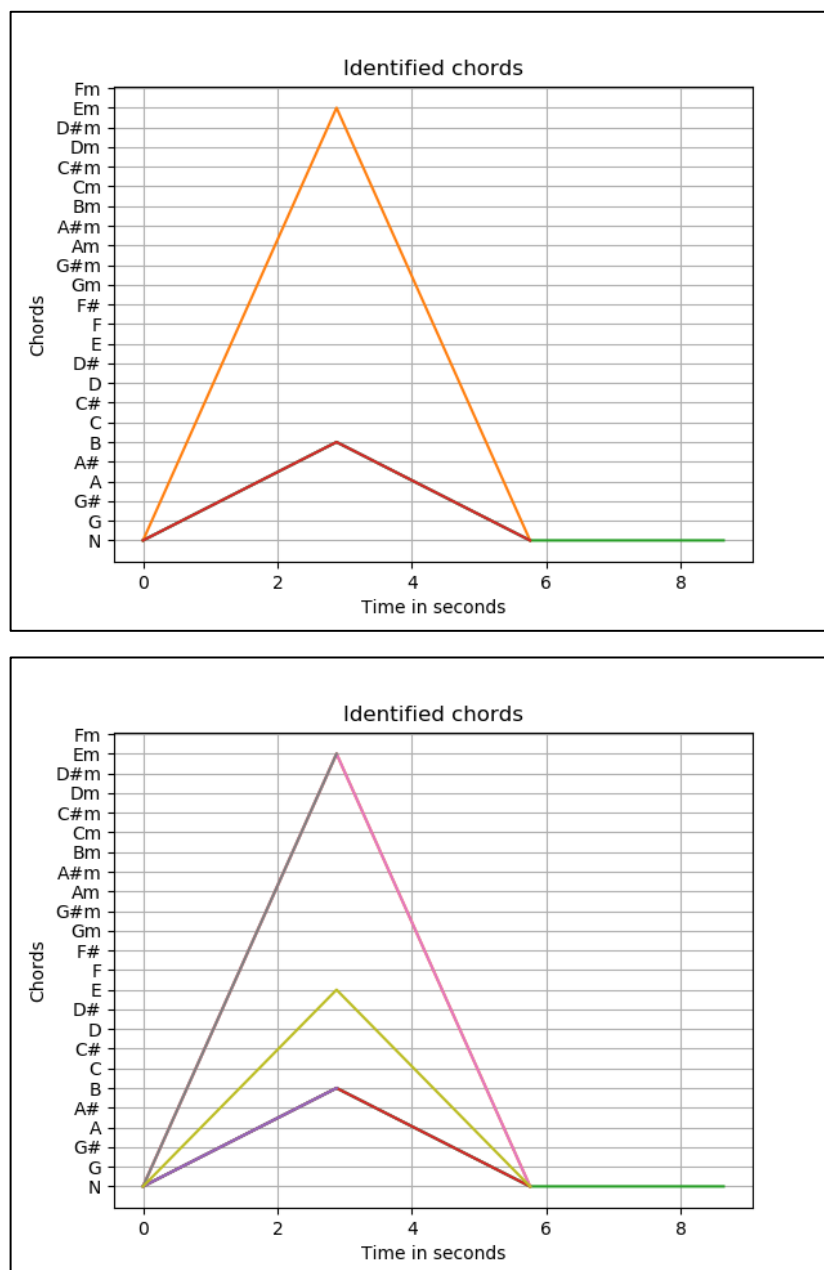


Рисунок 5.3 – Профіль класу висоти тону для Em(зверху) та Emsus(знизу)

Як можна побачити, алгоритм розпізнає деяку зміну в звичному нам Em акорді, у вигляді додаткової ноти E. Загалом можна було б сказати, що алгоритм

може розпізнавати і сус акорди, але для Emsus акорду характерні такі ноти: E-A-B, тобто навіть цю складову алгоритму не вдається розпізнати точно.

Аналогічна ситуація і для акордів Emsus2 та Emsus3, їх профіль класу висоти тону наведено на рисунках 5.4.

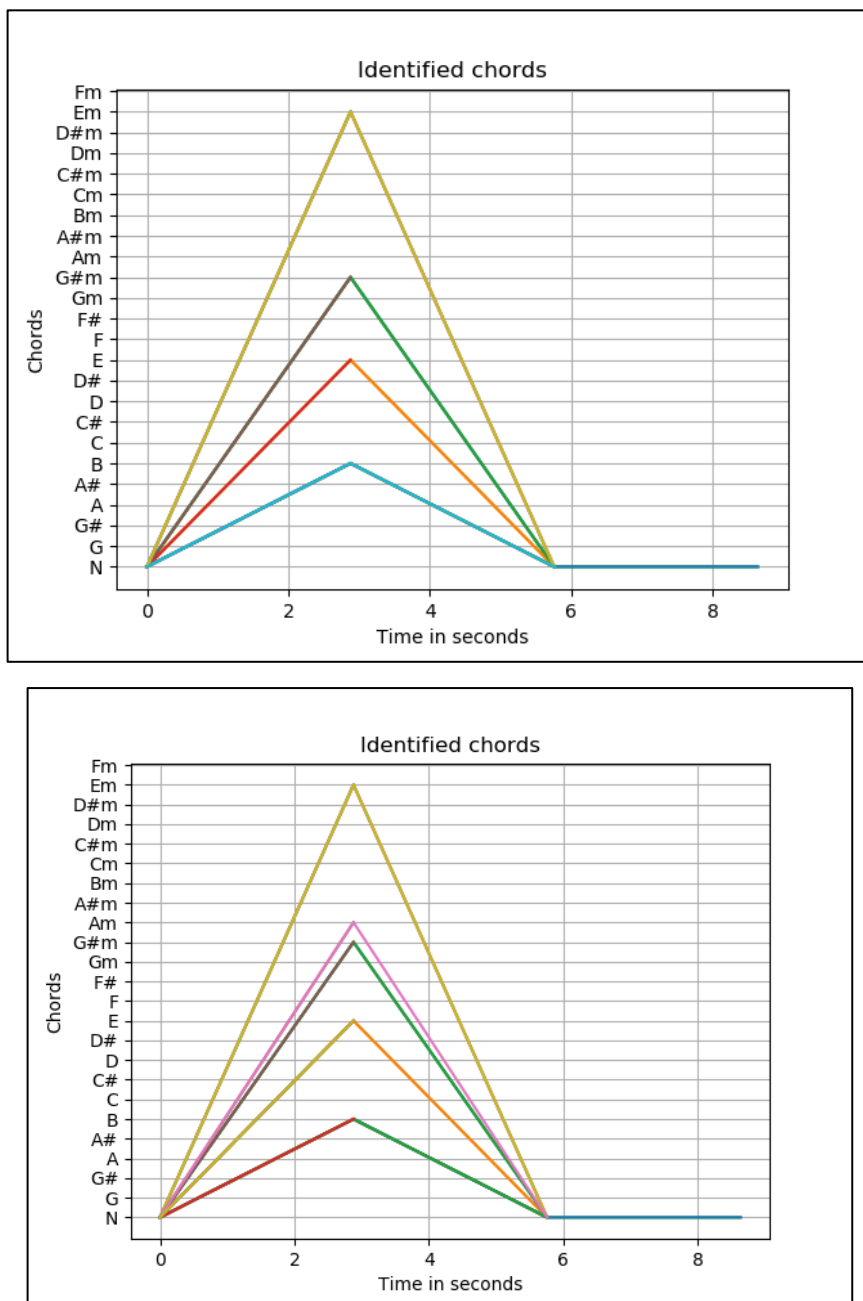


Рисунок 5.4 – Профіль класу висоти тону для Emsus2 та Emsus3

Для Emsus2 характерні ноти E-Gb-B, а для Emsus3 – так само як і для Emsus (E-A-B), різниця між ними в тоні. Але, як можна побачити для Emsus3 у порівнянні

з Emsus алгоритм показує зовсім різний профіль класу, хоча з нотної грамоти ми очікуємо побачити хоча б однаковий стан.

Загалом, цей алгоритм показує достатньо точні результати у випадку, коли він застосований точно на початку акорду, саме там де його почали грати, в інших випадках він дає значну кількість невірних результатів.

5.3 Алгоритм з використанням ПММ

ПММ – Прихована Марковська модель (в англійській літературі HMM), певна статична модель, яка розглядає процес як Марковський процес (коли стан певної характеристики об'єкту залежить від попереднього, але не залежить від того, що був до нього, тобто стан в момент часу t залежить від $t-1$, але не залежить від $t-2$) [14][15].

Чому ця модель може бути ефективна в задачах розпізнавання акордів? Все дуже просто, як ми все знаємо звук описується як коливальний рух частинок у просторі, що можна описати формулами в теорії, також важливо розуміти, що у реальному світі звук це ще й затухаючі коливання, які під дією сторонніх сил зменшують кількість коливань з часом.

Знаючи, що звук вже описується певними закономірностями та формулами, ми можемо точно стверджувати, що прихована Марковська модель буде ефективна.

Чому ж тоді ми не шукаємо модель, яка б в точності описувала залежність поточного стану від усіх попередніх? Доволі проста відповідь – людський фактор. А якщо точніше, то уявіть процес запису звуку на диктофон. Ви граєте певну пісню, і в цей момент, у вас може дригнути рука, змінити кут натиску палець (у випадку з грою на гітарі). Таким чином, людина створює багато станів звуку, які не залежать від усіх попередніх, але залежать від одного попереднього.

Якщо ж проста відповідь здається вам недостатньою, то ось більш розгорнута: кожен звук, який було записано залежить від багатьох факторів при

яких умовах цей звук було записано, не говорячи вже про «шуми», які навіть в ідеальних умовах для запису звуку можливі, причому в таких випадках, саме звукозаписуюча техніка дає шум, оскільки вона залежить щонайменше від електроенергії. Будь-яка техніка, що залежить від електроенергії у свою чергу має певну «шумівну» здатність. Що мається на увазі? Ви певну чули, як під час сильного навантаження на процесор, останній починає нагріватись, а значить виділяти тепло, що в свою чергу створює умови для коливального руху частинок навколо нього, а це і є звук, який в свою чергу може бути доволі тихим або доходити до писку, що вже відчутно людським слухом. А тепер уявіть скільки всього електричних частин в звукозаписній техніці взагалі є. Тепер зрозуміло, чому на будь-якій техніці присутній шум, який доволі часто не можна розрахувати точно, як і позбутись від нього.

А тепер, коли ми прояснили належність використання даної моделі, ми можемо приступити до експерименту над використанням цієї моделі в задачах розпізнавання акордів.

Так само, як і для попереднього алгоритму, будемо змінювати ширину вікна перетворення Фур'є, щоб проаналізувати залежність цього алгоритму від ширини вікна за метриками вказаними в пункті 4.2.

Для тестування алгоритму було обрано ті ж розміри вікон: 4096, 8192, 16384, 32768 та 65536. Результати експерименту над даним алгоритмом можна побачити в таблиці 5.2.

Таблиця 5.2 – Результати проведення експерименту для алгоритму

Метрика	Ширина вікна				
	4096	8192	16384	32768	65536
Y	11%	10%	11%	10%	3%
Y_d	39%	41%	29%	16%	10%
Δt	384ms	166ms	81ms	38ms	17ms

Можна побачити, що в даного методу результати схожі з результатами попереднього, хоча і є сильне покращення в розпізнаванні акорду на відріжку. Річ у тім, що даний алгоритм «запам'ятовує» стан на попередньому проміжку часу, таким чином, якщо він розпізнав акорд правильно від самого початку, то він його буде розпізнавати і до його повного згасання.

Це підтверджують результати експерименту. Для розуміння про що йде мова, наводимо приклад розпізнавання акорду G даним алгоритмом у порівнянні з попереднім при ширині кадру 4096 на рисунку 5.5.

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

Рисунок 5.5 – Розпізнавання акорду G методом з ПММ(зліва) та методом відповідності шаблону(праворуч)

Цей приклад якомога краще показує покращення в розпізнаванні акорду на проміжку часу, хоча похибка завжди присутня. Вона найчастіше проявляється у випадках, коли алгоритм розпізнав акорд не правильно від самого початку, тим

самим впливаючи на подальше розпізнавання акорду. Приклад описаної поведінки наведено на рисунку 5.6.

```

=====
Working with Cm_7.wav
=====
Time(s) Chords
0.0 C
0.2786394557823129 C
0.5572789115646258 C
0.8359183673469388 C
1.1145578231292517 G#
1.3931972789115645 G#
1.6718367346938776 G#
1.9504761904761905 G#
2.2291156462585033 G#
2.507755102040816 G#
2.786394557823129 G#
3.0650340136054424 G#
3.3436734693877552 G#
3.622312925170068 G#
3.900952380952381 G#
4.179591836734694 G#

```

Рисунок 5.6 – Розпізнавання акорду Cm при ширині кадру 4096

Як можна побачити, при початковому розпізнаванні акорду як C, наступні результати розпізнавання були скомпрометовані, тобто вони пішли по курсу, який зовсім не схожий на акорд, який був насправді, а саме Cm.

5.4 Алгоритм з використанням нейронної мережі над хромаграмою

Нейронні мережі мають широке використання в повсякденному житті. Так існують нейронні мережі для розпізнавання відтінків в тексті, як то сум, жаль,

радість, турбота або злість, так існують і нейронні мережі, які розпізнають певні об'єкти на зображеннях.

Наприклад, в [16] об'єкт, що розпізнається, – літера, записана шрифтом Брайля. Таке розпізнавання допомогло б незрячим людям в системі з розпізнавання та озвучення того, що написано на тому чи іншому об'єкті, який використовується незрячою людиною.

В [17] розпізнають літаючі об'єкти, що знаходяться на землі в даний момент часу. До останнього можна віднести літаки, гелікоптери. Дану систему можна використовувати у військових цілях, для розвідки.

В нашому випадку алгоритм побудовано таким чином, що початкові дані, а саме цифровий запис звуку, проходить обробку перетворенням Фур'є, далі, на базі цієї обробки формується зображення, що називається хромаграмою і вже це зображення передається на обробку нейронній мережі.

Принцип нейронних мереж можна пояснити доволі просто, як певну систему з обробки сигналу, найчастіше представлену чорним ящиком. На вхід передаються дані для навчання, ті в яких зображення точно відповідає певним міткам, як-то при розпізнаванні котів чи собак, кожному зображенню ставиться у відповідність мітка «кіт» або «пес», далі навчену з певною вірогідністю розпізнавання нейронну мережу застосовують на більш реальних даних. Що мається на увазі під реальними даними? Тут слід розуміти, що в реальному житті коти не завжди дивляться у камеру, навіть більше, вони доволі верткі, через що важко зловити потрібний кадр, так от такі «верткі» кадри і можуть передаватись нейронній мережі в якості тесту, щоб зрозуміти що на зображенні.

В запропонованому алгоритмі, так само як і в попередніх двох, на певному етапі використовується частотний аналіз, який представлений у вигляді хромаграми (приклад хромаграми можна побачити на рисунку 5.7). Хромаграми в даному методі формуються для подальшого використання нейронною мережею, яка так само, як і в попередніх випадках, буде йти певним «вікном» від початку зображення до його кінця. В своїй роботі науковці описують принцип побудови глибинного створення хромаграми. Нейронна мережа аналізує хромаграму таким

чином, що кожному зображенню ставиться у відповідність час та мітка, у якості мітки використовується назва акорду, що зображено на хромаграмі.

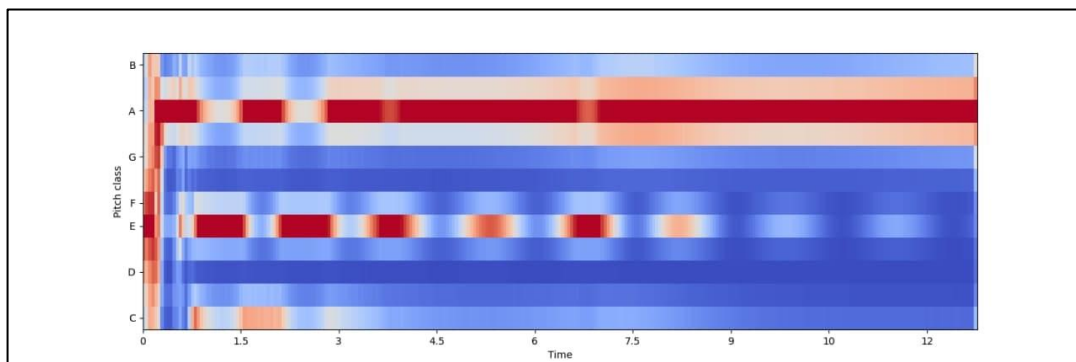


Рисунок 5.7 – Хромаграма для акорду Am акорду

Підхід з використанням нейронної мережі аналізує кожне більш детальне зображення хромаграми на наявність певних скритих для людського ока закономірностей. Згідно з цим підходом, використовуються три рівні аналізу таких «скритих» закономірностей, які в решті решт приводять спектрограму до вигляду простої хромаграми (див. рис. 5.8) [18].

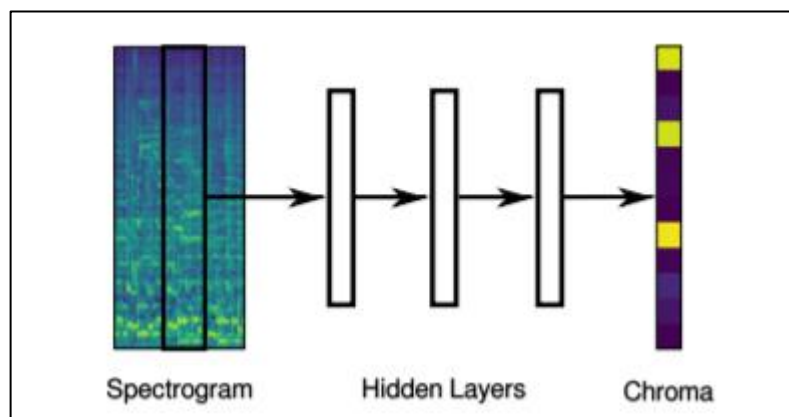


Рисунок 5.8 – Обробка спектрограми в хромаграму

В даному підході не існує можливості задавати розміри вікна для проведення частотного аналізу, оскільки частотний аналіз проводиться нейронною мережею, за допомогою цих трьох «скритих» рівнів. Певно це зроблено заздалегідь для того,

щоб уникнути проблем з пошуком «початку» акорду, які були присутні в перших двох методах.

Наводимо результати роботи цього методу:

$$Y = 52\%$$

$$Y_d = 41\%$$

$$\Delta t = 11ms$$

В даному випадку, можна побачити, що нейронна мережа справляється з поставленою задачею значно краще всіх своїх попередників, хоч і не може якісно розпізнати все ті ж сус акорди (див. рис. 5.9).

```

=====
Working with Emsus2_3.wav
=====
Start Time End time Chords
(0., 5.4, 'D#:min')
(5.4, 6.5, 'D#:maj')

```

Рисунок 5.9 – Розпізнавання акорду Emsus2

Можна помітити, що даний метод також виділяє початок та кінець відповідних акордів. Зрозуміло, що початок та кінець акорду вимірюються в секундах на звуковій музичній доріжці. Також трошки змінилась форма розпізнаного акорду, тут слід зазначити, що всі версії акордів, як то наприклад, акорд A, це ніщо інше як A:maj. Це може трохи збентежити на перший погляд, але так насправді і є, просто зазвичай ніхто не використовує «мажорну» приставку для мажорних акордів, використовують лише мінорну, так і пишуть Am – A:minor.

Однак розпізнавання акордів на музичній звуковій доріжці в даного методу найвище в порівнянні з його попередниками, також слід зауважити, що час витрачений на розпізнавання акордів сягнув 11ms, що приближує нас до

використання останнього методу на задачах розпізнавання акорду в реальному часі.

5.5 Покращений алгоритм з використанням нейронної мережі

Попередній алгоритм можна покращити, використавши умовні випадкові поля. Умовні випадкові поля описують клас методів для статистичного аналізу, на відміну від попереднього алгоритму, використання УВП дозволить враховувати контекст, по аналогії з ПММ, коли ми змогли врахувати попередній стан при розпізнаванні акорду.

Наводимо результати експерименту для покращеного алгоритму:

$$Y = 60\%$$

$$Y_d = 65\%$$

$$\Delta t = 3s\ 461ms$$

Результат двох метрик зріс, але ми втратили багато позицій за останньою метрикою і в порівнянні з попереднім алгоритмом ми отримали збільшення часу на розпізнавання акордів аж в 331 раз.

Чи помітно різницю при розпізнаванні акордів? Так, оскільки зросла точність розпізнавання акордів на відрізку, що демонструється на рисунку 5.10.

<pre>===== Working with A_3.wav ===== Start Time End time Chords (0., 7.1, 'A:maj')</pre>	<pre>===== Working with A_3.wav ===== Start Time End time Chords (0., 0.2, 'N') (0.2, 4.3, 'A:maj') (4.3, 7.1, 'N')</pre>
---	---

Рисунок 5.10 – Порівняння розпізнавання акордів двох останніх методів

6 ОПИС РОЗРОБЛЕННІ ПРОГРАМНОЇ СИСТЕМИ

Було розроблено програму для проведення аналізу існуючих алгоритмів розпізнавання акордів. Програма складається з двох модулів, в кожному з модулів присутні методи для аналізу двох алгоритмів розпізнавання акордів, тобто кожен з модулів може провести аналіз двох алгоритмів.

Кожний з модулів аналізує музичні дані, попередньо записані за допомогою сторонніх програм з використанням електрогітари у якості музичного інструменту. Всі музичні файли записані при частоті 44100Гц, зовнішньою звуковою картою, яка зчитує звук з електрогітари та передає його до комп'ютеру. Така звукова карта розпізнається як мікрофон.

В кожному з модулів присутнє використання швидкого перетворення Фур'є результати якого потім використовуються для розпізнавання акордів.

Останній модуль також має розширений функціонал для розпізнавання акордів, в якому використовується умовно випадкове поле.

Кожний з модулів використовує мову програмування Python, але різної версії. Для використання різних версій мови та більш якісного використання робочого простору, було створено середовища Conda, які були налаштовані саме для запуску того, чи іншого модуля.

Перший модуль називається Chord-Recognition-HMM, в ньому є перший на другий алгоритми, що були проаналізовані. Кожен із методів може бути викликаний командою «python <назва_файлу>». Для запуску цього модуля необхідно мати налаштований Python 2.7 з усіма сторонніми модулями, включаючи pandas, numpy та scipy.

Другий модуль має назву neural-networks, для його запуску потребується Python 3.8. Для правильної роботи цього модулю слід також встановити необхідні додаткові сторонні модулі, які зокрема включають в себе pandas, matplotlib, numpy.

7 РЕЗУЛЬТАТИ РОБОТИ

За результатами роботи, було проаналізовано 4 методи для розпізнавання акордів. Три з них базуються на частотному аналізі, який в свою чергу базується на швидкому перетворенні Фур'є.

Для методів, які не використовують нейронні мережі можна виділити найбільш ефективні параметри вікна розпізнавання акордів.

Для першого алгоритму, що проводить розпізнавання базуючись на зіставленні відповідності шаблону найбільш ефективним значенням ширини вікна є значення 16384. Але, хоч це і може показатись найбільш ефективним значенням, ми рекомендуємо використовувати таку ширину вікна, яка менше ніж сам проміжок. Як зрозуміти яке значення вам необхідно в такому випадку? Все просто, при частоті запису 44100Гц, одна секунда рівноцінна проміжку довжиною 44100 одиниць. На цьому проміжку найбільш ефективним значенням вікна буде 32768, оскільки в такому випадку, найбільша частина відрізка буде покрита першим же аналізом. У випадку, коли ви впевнені, що на цьому короткому відрізку присутні декілька акордів, краще використовувати значення менше, наприклад 8192. В такому випадку буде проаналізовано більше частей відрізка окремо одне від одного. Загалом даний алгоритм підійде для систем, де ми можемо гарантувати, що саме з початку звукозапису йде акорд, який ми хочемо розпізнати. Такий алгоритм підійде там, де не має можливості використовувати нейронні мережі через їх залежність від ресурсів.

Другий алгоритм ефективний в тих умовах, що і перший, але його краще не використовувати на записах, в яких присутні декілька акордів послідовно, оскільки через свою специфіку, алгоритм дасть не вірний результат на розпізнаванні акордів, які йдуть одне за одним.

Третій алгоритм підходить для більшості реальних задач розпізнавання акордів, але існує певна залежність від ресурсів, які потребуються для нейронної мережі.

На жаль, в цій роботі ми не проводили експерименти з алгоритмами швидкого перетворення Фур'є, але ми впевнені, що найбільш ефективним перетворенням було б те, що використовує графічну пам'ять для обчислення. Що і підтверджувалось деякими дослідниками.

Останній алгоритм підходить для випадків, коли вам дуже важливо точно розуміти де починається та закінчується акорд. Він ще більше залежить від ресурсів, а також часу. Для використання цього підходу вам слід мати на увазі, що швидкість розпізнавання одного акорду значно вище ніж реалізації з використанням хромаграми та її аналізу нейронною мережею.

Всі методи, що використовувались в цій роботі, виявились не здатні розпізнавати суц акорди, через що, нам здається, що тема розпізнавання акордів буде ще довго актуальною, оскільки із усіх методів, що існують в вільному доступі не існує того, що міг би достовірно розпізнавати суц або «зменшенні» акорди.

ВИСНОВКИ

В результаті роботи було проведено аналіз найбільш актуальних алгоритмів для розпізнавання акордів. Підхід до аналізу через створення власних, незалежних початкових даних показав, що деякі методи, що при їх проектуванні тестувались на інших даних і давали результати вище 60%, можуть бути не зовсім вірними.

Для проведення дослідження було створено програмну систему, яка на одних даних проводить аналіз існуючих методів за трьома характеристиками: кількість розпізнаних акордів, кількість розпізнаних акордів на відрізок та час використаний на їх розпізнавання.

За результатами дослідження можна сказати, що найбільш ефективним методом для розпізнавання акордів є метод з використанням умовно випадкового поля, який жертвує часом розпізнавання і тим самим приводить до збільшення ефективності розпізнавання акордів.

Через свій недолік у часі, метод з використанням нейронних мереж для аналізу хромаграми здається найбільш ефективним, хоч і з трохи меншою точністю розпізнавання акордів. Даний підхід можна використовувати в реальних системах для розпізнавання акордів в реальному часі.

Для всіх зазначених методів характерний один важливий недолік, всі ці методи погано розпізнають сус акорди, не говорячи вже про «зменшенні» акорди. Зменшенні акорди не були розглянуті в даній роботі, то ж це ще одна можливість для подальшого вдосконалення аналізу підходів для розпізнавання акордів.

Для подальшого дослідження необхідно знайти метод, який зміг би розпізнавати сус або «зменшенні» акорди, але вже остаточно зрозуміло, що підходи з використанням нейронних мереж будуть більш ефективними, оскільки вони більше здатні адаптуватись під вхідні дані ніж аналогічні підходи без використання нейронних мереж. Нерідко нейронні мережі знаходять властивості, які не бачить людське око.

ПЕРЕЛІК ПОСИЛАНЬ

1. Звук / Матеріал з вікіпедії — вільної енциклопедії. URL: <https://uk.wikipedia.org/wiki/Звук> (дата звернення: 02.02.2021)
2. Taco Walstra, Jose Lagerberg, Rein van den Boomgaard. Digital Signal Processing v0.1. Systems and Signals URL: <https://staff.fnwi.uva.nl/r.vandenboomgaard/SP20162017/SystemsSignals/index.html> (дата звернення: 02.02.2021)
3. Дискретный сигнал / Материал из википедии — свободной энциклопедии. URL: https://ru.wikipedia.org/wiki/Дискретный_сигнал (дата звернення: 02.02.2021)
4. Gamification / Wikipedia material . URL: <https://en.wikipedia.org/wiki/Gamification> (дата звернення: 05.02.2021)
5. Назаров А.С., Сушинський І.К. Геймификация обучения с использование VR. The World of Science and Innovations, London, United Kingdom.- 14-16 October 2020.
6. T. Cho and J. P. Bello. On the Relative Importance of Individual Components of Chord Recognition Systems. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 22(2):477–492, Feb. 2014
7. R. Chen, W. Shen, A. Srinivasamurthy, and P. Chordia. Chord recognition using duration-explicit hidden Markov models. In Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR), Porto, Portugal, 2012.
8. Suspended Chord / Wikipedia material. URL: https://en.wikipedia.org/wiki/Suspended_chord (дата звернення: 15.02.2021)
9. Christopher Harte. Towards Automatic Extraction of Harmony Information from Music Signals. 2010.
10. Rod Stephens. Essential Algorithms. A practical approach to computer algorithms, 28-29, 2016

11. Назаров О.С., Яровий М.В. Частотний аналіз в задачах розпізнавання звуку з використанням нейронних мереж. Сучасні аспекти та перспективи розвитку науки. 16 квітня 2021 м. Кропивницький, Україна, с. 48-50.
12. John Parker, Phd. FFT performance using NumPy, PyFFTW, and cuFFT. URL: https://www.jparker.me/blog/fft_2d_performance (дата звернення 24.04.2021).
13. Kyogu Lee. Automatic Chord Recognition from Audio Using Enhanced Pitch Class Profile. Center for Computer Research in Music and Acoustics. 2014
14. Прихована марковська модель / Матеріал вікіпедії. URL: https://uk.wikipedia.org/wiki/Прихована_марковська_модель (дата звернення: 01.05.2021)
15. Daniel Jurafsky, James H. Martin. Speech and Language Processing. December 30, 2020.
16. Kirill Smelyakov, Dmytro Yeremenko, Anton Sakhon, Vitalii Polezhai, Anastasiya Chupryna. Braille character recognition based on neural networks // IEEE Second International Conference on Data Stream Mining & Processing (DSMP), August 21-25. – 2018. – P. 509-513.
17. Ruban I., Smelyakov K., Martovytskyi V., Pribylnov D., Bolohova N. Method of neural network recognition of ground-based air objects // IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT), 24-27 May. – 2018. – P. 589-592.
18. Filip Korzeniowski, Gerhard Widmer. Feature learning for chord recognition: the deep chroma extractor. 15 Dec 2016. URL: <https://arxiv.org/pdf/1612.05065v1.pdf> (дата звернення 02.05.2021)