

Вінницький національний технічний університет

(повне найменування вищого навчального закладу)

Факультет інтелектуальних інформаційних технологій та автоматизації

(повне найменування інституту, назва факультету (відділення))

Кафедра комп'ютерних наук

(повна назва кафедри (предметної, циклової комісії))

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Інформаційна технологія тонічного аналізу музичних творів»

Виконав: студент 2-го курсу, групи 1КН-21м
спеціальності 122 «Комп'ютерні науки»

(шифр і назва напрямку підготовки, спеціальності)

Гульчак Р.С.

(прізвище та ініціали)

Керівник: к.т.н., доцент каф. КН

Барабан С.В.

(прізвище та ініціали)

« 15 » 12 2022 р.

Опонент: д.т.н., професор, в.о. зав. каф. КСУ

Ковтун В.В.

(прізвище та ініціали)

« 15 » 12 2022 р.

Допущено до захисту

Завідувач кафедри КН

д.т.н., проф. Яровий А.А.

(прізвище та ініціали)

« 16 » 12 2022 р.

Вінниця ВНТУ - 2022 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних наук
Рівень вищої освіти II-й (магістерський)
Галузь знань – 12 «Інформаційні технології»
Спеціальність – 122 «Комп'ютерні науки»
Освітньо-професійна програма – «Системи штучного інтелекту»

ЗАТВЕРДЖУЮ

Завідувач кафедри КН
д.т.н., проф. Яровий А.А.

15.09 2022 року

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Гульчак Роман Сергійович

(прізвище, ім'я, по батькові)

– Тема роботи Інформаційна технологія тонічного аналізу музичних творів

керівник роботи к.т.н., доцент кафедри КН Барабан С. В.

затверджені наказом вищого навчального закладу від "14" 09 2022 року № 203

– Строк подання студентом роботи 18 листопада 2022 року

– Вихідні дані до роботи:

вхідні дані – використання об'єктно-орієнтованої мови програмування, операційна система – Windows, підтримка основних форматів аудіофайлів (.mp3, .wav, .ogg, .acc, .flac).

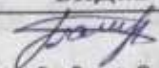
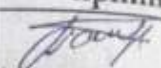
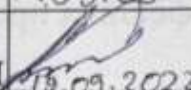
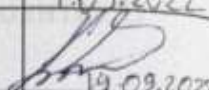
– Зміст текстової частини:

Вступ, аналіз предметної області та огляд відомих рішень, розробка алгоритмів функціонування основних модулів, програмна реалізація модулів системи, приклади роботи розроблених модулів, висновки, список використаних джерел, перелік використаних джерел, додатки

– Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень)

Графічне представлення ІКМ-сигналу, схема перетворення дискретного аудіосигналу при представленні на вхід нейронної мережі, діаграма прецедентів роботи додатку, головне вікно додатку, графічне представлення способу генерування навчальної вибірки, результат роботи автоматичного транскрибування.

- Консультанти розділів роботи


Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	виконання прийняв
1-3	Барабан С. В., к.т.н., доц. каф. КН	 14.09.2022	 14.09.2022
4	Нікіфорова Л. О., к. е. н., доц. каф. ЕПВМ	 15.09.2022	 19.09.2022

- Дата видачі завдання 14.09 2022 року

КАЛЕНДАРНИЙ ПЛАН

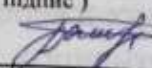
№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи
1	Аналіз предметної області та огляд відомих рішень. Постановка задач дослідження	14.09.2022р - 04.10.2022р
2	Розробка алгоритмів та моделей нейронних мереж	02.10.2022р - 16.10.2022р
3	Програмна реалізація модулів з використанням розроблених моделей	17.10.2022р - 02.11.2022р
4	Підготовка економічної частини	08.11.2022р - 21.11.2022р
5	Апробація та/або впровадження результатів дослідження	23.11.2022р - 01.12.2022р
6	Оформлення пояснювальної записки, графічного матеріалу та презентації	02.12.2022р - 14.12.2022р

Студент


(підпис)

Гульчак Р. С.

Керівник роботи


(підпис)

Барабан С. В.

АНОТАЦІЯ

УДК 004.8

Гульчак Р.С. Інформаційна технологія аналізу музичних творів. Магістерська кваліфікаційна робота зі спеціальності 122 – комп'ютерні науки, освітня програма - комп'ютерні науки. Вінниця: ВНТУ, 2022. 89 с. На укр. мові. Бібліогр.: 21 назв; рис.: 22; табл. 7.

Дана магістерська кваліфікаційна робота присвячена розробці інформаційної технології аналізу музичних творів. Дана технологія дає змогу спростити користувачеві процес транскрибування музичних творів.

В ході роботи було проведено аналіз предметної області транскрибування музичних творів, було розглянуто та проаналізовано системи аналогів. Розроблено підхід до аналізу вхідного цифрового аудіо-сигналу. Було проаналізовано декілька парадигм штучних нейронних мереж та обґрунтовано доцільність використання багатошарового перцептрона для вирішення поставленої задачі. Для розробки було обрано мову програмування TypeScript. Розроблено модель багатошарового перцептрона, що має 65 входів, один прихований шар з 8 нейронів та один вихідний нейрон. Було проведено тестування розробленого додатку, що показало, що усі працюють коректно.

Графічна частина складається з 6 плакатів.

У економічному розділі розраховано суму витрат на розробку та виготовлення нового технічного рішення, яка складає 1119805 гривень, спрогнозовано орієнтовану величину витрат по кожній з статей витрат, розраховано чистий прибуток, термін окупності витрат для виробника 0,93 роки та економічний ефект для споживача при використанні даної розробки.

Ключові слова: музичні твори, транскрибування, нейронна мережа, багатошаровий перцептрон, зворотнє поширення помилки

ABSTRACT

The master's thesis is devoted to the development of information technology for tonal analysis of musical works. The technology makes it easier for a user to transcribe musical works.

I

h

a

k

Keywords: musical works, transcription, neural network, multilayer perceptron, inverse error propagation

f

w

p

r

k
t

o

t

n

h

a

e

l

a

a

n

n

a

a

l

l

y

y

s

f

i

s

s

o

e

ЗМІСТ

Вступ.....	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОРІВНЯННЯ ВІДОМИХ ТЕХНОЛОГІЙ ТОНІЧНОГО АНАЛІЗУ МУЗИЧНИХ ТВОРІВ.....	10
1.1 Аналіз предметної області транскрибування музичних творів.....	10
1.2 Автоматичне транскрибування музичних творів.....	12
Огляд відомих реалізацій тонічного аналізу музичних творів.....	13
1.4 Висновок до розділу 1.....	18
2 МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ТОНІЧНОГО АНАЛІЗУ МУЗИЧНИХ ТВОРІВ.....	19
2.1 Математична модель тонічного аналізу вхідного дискретного аудіосигналу.....	19
2.2 Використання нейронних мереж для вирішення задачі тонічного аналізу м	
2.3 Вибір типу штучної нейронної мережі для автоматичного транскрибування музичних творів.....	24
2.4 Розробка загальної структурної схеми інформаційної технології тонічного аналізу.....	31
2.5 Висновок до розділу 2.....	33
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ТОНІЧНОГО АНАЛІЗУ МУЗИЧНИХ ТВОРІВ.....	34
3.1 Обґрунтування вибору мови програмування для програмної реалізації інформаційної технології тонічного аналізу музичних творів.....	34
3.2 Програмна реалізація модуля перетворення вхідного дискретного аудіосигналу.....	36
3.3 Програмна реалізація інтерфейсу користувача.....	38
3.4 Формування навчальної вибірки та навчання моделі штучної нейронної мережі для тонічного аналізу музичних творів.....	40
3.....	
3.6 Висновок до розділу 3.....	49
5	

ЕКОНОМІЧНА ЧАСТИНА.....	50
4.1 Комерційний та технологічний аудит науково-технічної розробки.....	50
4.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи.....	53
4.4 Висновок до розділу 4.....	65
ВИСНОВКИ.....	66
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	68
ДОДАТКИ	70
Додаток А (обов'язковий) Результат перевірки на плагіат в онлайн-системі	
Додаток Б (обов'язковий) Лістинг програми.....	72
Додаток В (обов'язковий).....	78
Додаток Г (довідниковий) Інструкція користувача.....	85
Додаток Д Довідка про впровадження	88

ВСТУП

Актуальність теми дослідження. Стрімкий розвиток комп'ютерної техніки у різних сферах призвели до масової комп'ютеризації всіх галузей людської діяльності, у тому числі освіти. В сучасних умовах комп'ютер став необхідним і надійним засобом розв'язання навчальних завдань, знаряддям наукової діяльності, застосування якого якісно змінює можливості нагромадження і застосування нових знань, значно розширює межі пізнання.

З розвитком мережі Інтернет у користувачів з'явився доступ до необмеженої кількості навчальних матеріалів. Сьогодні будь-хто маючи доступ до мережі може почати опановувати нові навички та знаходити нові хобі перебуваючи в дома. Один з популярних варіантів є ознайомлення з музичним інструментом.

Навіть при тому, що в мережі розміщена величезна кількість нот доступних для вивчення, далеко не завжди вдається знайти документацію або ноти до улюблених композицій не виходить. В такому випадку доводиться самотужки, використовуючи власний слух, або використовуючи послуги спеціалістів-транскрипторів підбирати ноти обраної композиції, намагаючись відтворити оригінальну композицію, іншими словами – транскрибувати обрану композицію.

Транскрипція музичної композиції для середньостатистичної людини є досить складним процесом, займає багато часу та, без додаткових засобів, потребує базової нотної грамоти та гарного музичного слуху, на розвиток якого необхідні роки практики.

Тому тема цієї магістерської кваліфікаційної роботи “Інформаційна технологія тонічного аналізу музичних творів”, що дозволить спростити процес транскрибування музичних творів, є актуальною темою для дослідження.

Зв'язок роботи з науковими програмами, планами, темами.

Магістерська робота виконана відповідно до напрямку наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету 22 К1 «Моделі, методи, технології та пристрої інтелектуальних інформаційних систем управління, економіки, навчання та комунікацій» та плану наукової та навчально-методичної роботи кафедри.

Метою роботи є розширення функціональних можливостей програмного забезпечення тонічного аналізу музичних творів для спрощення процесу транскрибування музичних творів.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- обґрунтувати актуальності розробки тонічного аналізатора.
- провести порівняння існуючих технологій тонічного аналізу музичних творів.
- удосконалити математичну модель підходу до аналізу музичних творів.
- розробка алгоритмів функціонування модулів тонічного аналізатора.
- розробка загальної структурної схеми тонічного аналізатора.
- Здійснити програмну реалізацію інформаційної технології тонічного аналізу музичних творів.
- провести тестування розробленого програмного додатку.
- обґрунтувати економічну доцільність розробки.

Об'єктом дослідження є процес тонічного аналізу музичної композиції.

Предметом дослідження є інформаційна технологія тонічного аналізу музичних творів.

Методи дослідження. Для досягнення поставленої мети було застосовано методи моделювання UML, інтелектуального аналізу даних, теорію штучних нейронних мереж а також методи об'єктно-орієнтованого програмування для реалізації інформаційної технології тонічного аналізу музичних творів.

Наукова новизна одержаних результатів полягає в наступному: удосконалено інформаційну технологію тонічного аналізу музичних творів, яка відрізняється від відомих удосконаленою математичною моделлю розбиття вхідного аудіо сигналу на складові, що дає можливість розширити функціонал програмного забезпечення автоматизованого транскрибування музичних творів.

Практичне значення одержаних результатів полягає у тому, що розроблений програмний додаток, що реалізує інформаційну технологію тонічного аналізу музичних творів, що полегшує користувачу транскрибувати музичний твір за рахунок автоматичного транскрибування та можливості корегувати результат.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується строгістю постановки задач, коректним застосуванням різноманітних методів під час доведення наукових положень, строгим виведенням аналітичних співвідношень, порівнянням результатів з відомими та збіжністю результатів математичного моделювання з результатами, що отримані під час впровадження розроблених програмних засобів.

Особистий внесок здобувача. Усі результати даної магістерської кваліфікаційної роботи були отримані здобувачем самостійно. У тезах, опублікованих у співавторстві, здобувачу належать такі результати: розробка удосконаленого методу розбиття вхідного аудіосигналу; розробка алгоритму

роботи тонічного аналізатора; розробка удосконаленого методу автоматичного транскрибування музичного твору [1].

Апробація та публікація: результати дослідження доповідались на науково-технічній конференції викладачів та студентів ВНТУ та викладені у тезі, опублікованій в репозиторії ВНТУ [1].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОРІВНЯННЯ ВІДОМИХ ТЕХНОЛОГІЙ ТОНІЧНОГО АНАЛІЗУ МУЗИЧНИХ ТВОРІВ

1.1 Аналіз предметної області транскрибування музичних творів

Транскрибування музичного твору — це процес створення нотних записів використовуючи як вхід лише аудіо запис. Створені нотні записи при цьому повинні максимально точно передавати виконавцеві інформацію про те, як правильно відтворити оригінальний запис. Нотні записи зазвичай створюються на основі існуючої композиції (найчастіше з запису виступу), рідше — навпаки. Транскрипцію іноді називають аранжуванням, хоча насправді транскрипції є вірними адаптаціями, тоді як аранжування містять змінену версію оригіналу [2].

Термін «транскрипція» походить від латинського слова «transcribo» і перекладається як «переписування» (trans— — «через, пере-» та scribo — «пишу») [2].

Транскрипції можуть бути створені як людиною так і комп'ютерною програмою. Створені транскрипції мають бути доступними для того, щоб їх прочитала людина (у вигляді нотного стану, гітарної табулатури і т.д.) або машина (у вигляді MIDI (Musical Instrument Digital Interface) файлу чи інших форматів) [4]. Коли транскрипція у доступному для людей вигляді, вона може бути використаною як інструкція для виконавців, але у етномузикології (дисципліні, присвяченій систематизації та дослідженню народної музики) транскрипції часто використовуються у ході аналізу невідомих до цього музичних творів.

В етномузикології як і в музикознавстві транскрипція довгий час відігравала ключову роль. На початку це був єдиний спосіб стиль та настроїв

твору читачам, які ніколи його не чули. Але згодом записи виконань перейняли цю функцію і основною задачею транскрипції стала ілюстрація при аналізі композицій.

В часи, коли людям ще не були доступні засоби цифрової транскрипції музиканти були змушені прослухувати запис десятки разів, при цьому, часто сповільнюючи запис щоб мати змогу розкласти швидкі паттерни композиції. Проблема підходу сповільнення полягає в тому, що музикант, сповільнивши запис, також зменшує висоту всіх нот цього запису пропорційно до сповільнення. Наприклад, сповільнивши запис у два рази, музикант, закінчивши транскрипцію, змушений підняти ноти на октаву вище, тобто збільшити частоти нот у два рази, щоб висота співпадала з оригінальним записом. Також, зі зменшенням висоти нот людському слуху стає дедалі важче їх розрізнати. Частково цю проблему вирішує програмне забезпечення для сповільнення запису без втрати висоти тону, але при цьому часто з'являються артефакти в аудіо та погіршується якість запису, що також додає проблем при транскрибуванні [3].

Деяке комерційне програмне забезпечення може визначити висоти нот в мелодіях. Але таке сканування не точне та на виході дає велику кількість зайвих нот. Це викликано наявністю у записі кількох інструментів, а також тим, що більшість музичних інструментів генерують не тільки чистий основний тон, а ще й велику кількість вищих гармонік основного тону. Користувачу здебільшого доводиться корегувати нотні записи, отримані в результаті роботи таких програмних засобів, іноді навіть використовуючи інші засоби. Саме тому програмне забезпечення для транскрибуванні творів зазвичай використовується лише як помічник.

1.2 Автоматичне транскрибування музичних творів

Можливість транскрибувати, тобто перетворювати музичний твір в нотний запис, є гарним прикладом здібностей людського інтелекту. Транскрибування вимагає від людини перцепції (аналізу складних зовнішніх сигналів), представлення знань (розпізнавання музичних елементів), відтворення (формування паттернів) та формування висновків.

Автоматичне транскрибування музичних творів — це спосіб транскрибувати музичний твір використовуючи програмний засіб, що конвертує звукові сигнали в певну форму нотних записів. Таким чином, що відтворення виконавцем нот отриманим в результаті транскрибування дає в результаті наближення до оригінальної композиції. Розробка такого способу є досить складною задачею в області обробки медіа та штучного інтелекту [5]. Ця задача складається з набору підзадач, включаючи оцінку тонів вхідного сигналу, визначення тривалостей, розпізнавання інструментів, трекінг ритму та інтерпретацію динаміки. Через саму природу, звуковий сигнал може містити одночасно кілька джерел звуку (ноти музичних інструментів, голос, і т.д.), які несуть з собою велику кількість інформації доступну для аналізу. Але задача автоматичної транскрипції все ще вважається відкритою і досить складною, через те, що зазвичай музичні композиції містять велику кількість одночасних нот та інструментів з різноманітними частотними характеристиками.

Огляд відомих реалізацій тонічного аналізу музичних творів

Після проведення пошуку аналогів, вдалося знайти три додатки схожих за функціоналом системи-аналоги які можуть вирішити поставлену задачу транскрипції музичних творів:

AnthemScore є гарним програмним забезпеченням для транскрипції музики. AnthemScore може перетворити mp3, wav та інші аудіо формати у ноти. Додаток дозволяє записувати ноти поверх візуалізації та прослуховувати записані ноти поверх оригінальної композиції. Додаток включає в себе можливість автоматично згенерувати ноти за вхідним аудіо, після чого користувач може їх редагувати. Додаток має 3 різних за ціною і функціоналом версії починаючи з тридцяти доларів США закінчуючи на ста доларах США як разова оплата. Перша з яких дозволяє лише переглядати згенеровані ноти без візуалізації та прослуховування, тоді як третя надає повний функціонал. Додаток доступний для користувачів як настільний додаток для Windows, Mac або Linux [6].

Описаний додаток має певні недоліки:

- Додаток не є браузерним, а отже вимагає встановлення на пристрій користувача, слідкування за виходом версій та розповсюдження на невелику кількість платформ;
- Відсутність можливості змінити параметри автоматичної генерації нот вхідного аудіо файлу;
- Відсутність можливості налаштувати параметри візуалізації;
- Відсутність фільтрації візуалізації;

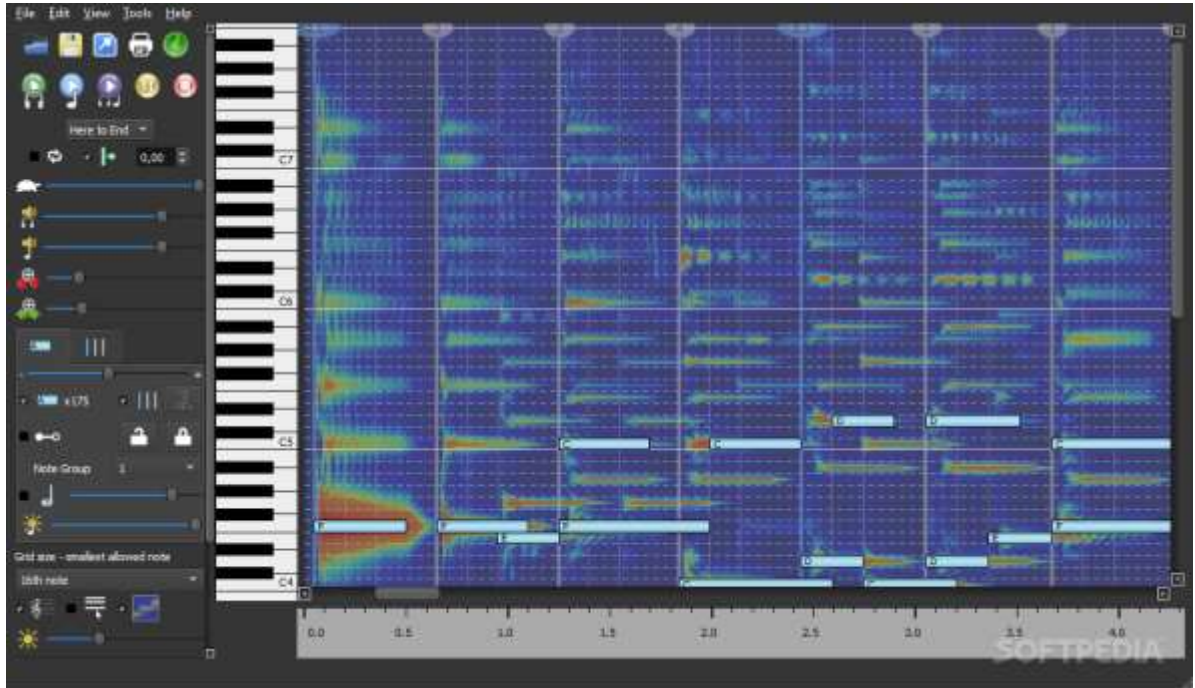


Рисунок 1.1 – Головне вікно програми AnthemScore

Notation програма що дозволяє створити нотні записи для різних інструментів, імпортувати ноти з інших форматів нотних записів та експортувати їх в PDF та MIDI. Додаток не містить функціоналу для автоматичного транскрибування записів а також не надає можливості користувачу завантажувати записи для ручного транскрибування запису на фоні запису що є його основним недоліком. Додаток розповсюджується у двох платних версіях починаючи з дев'яноста доларів США закінчуючи на ста п'ятидесяти доларах США – версія для музикантів (notation musician) та для композиторів (notation composer) [7].

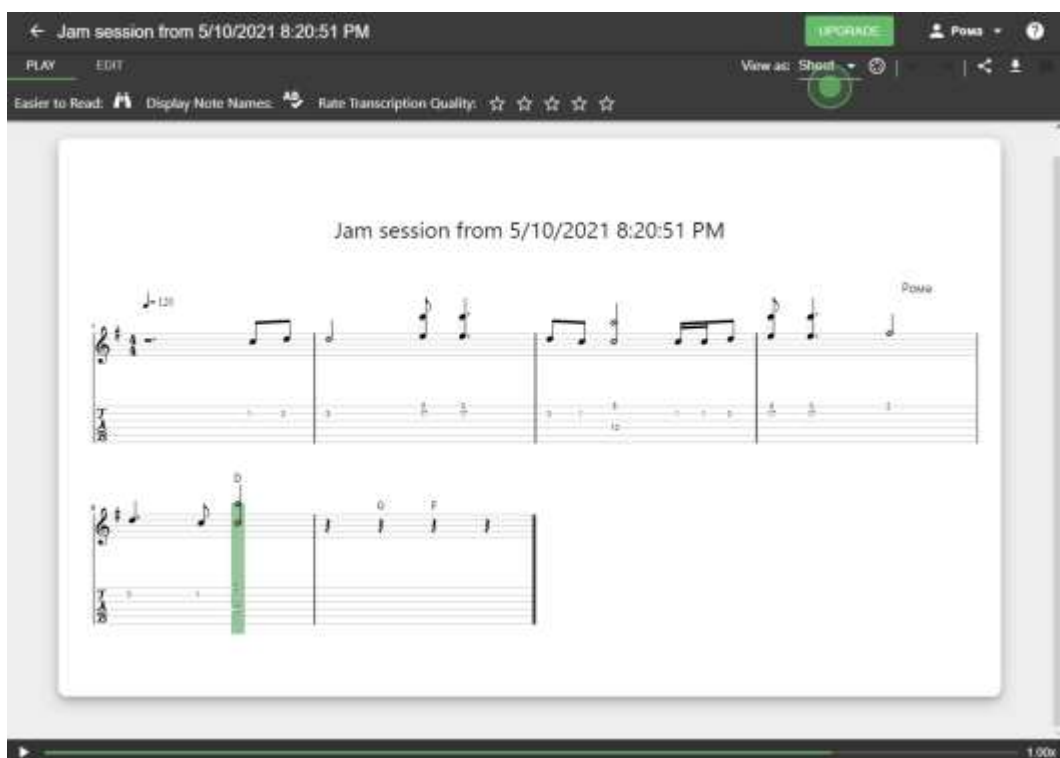


М
е
л
о
д
у
Scanner браузерний додаток для автоматичної транскрипції творів з
Те, що додаток є вебсайтом, робить його доступним на всіх платформах
завантажених аудіо файлів. Додаток дозволяє завантажити вхідну композицію
що підтримують нові стандарти мови JavaScript. Додаток не надає додаткових
з мікрофону, як аудіо файл з файлової системи або як посилання на відео з
візуалізації чи можливості відтворення оригінального аудіо поверг записаних
стан, гітарна табулатура та piano roll. Також додаток має вбудовані інструменти
нот. Автоматична транскрипція вхідних даних відбувається на серверній
для редагування нотних записів, згенерованих з вхідних даних.
сторони, що в деяких випадках надає перевагу, а в деяких є недоліком. Це
пов'язано з тим, що серверна обробка дає змогу працювати з додатком на
малопотужних платформах, але накладає обмеження на потужні настільні

Р
и
с
у
н
о
к
1
к
н
с

комп'ютери, тому що об'єм вхідних даних обмежується для збереження продуктивності серверів і обмежуються шляхи використання проміжних даних користувачем.

Додаток має дві версії — безкоштовну з сильно урізаним функціоналом та повну у вигляді підписки на місяць або на рік [8].



Р

и

с

у

н

о

к

Таблица 1.1 – порівняння відомих систем-аналогів

Додаток	AnthemScore	Notation Composer	Melody Scanner
Характеристика			
Тип додатка	Додаток для настільного комп'ютера	Додаток для настільного комп'ютера	Вебсайт
Вартість	28 – 99\$		0
Автоматична транскрипція			
Ручна транскрипція			-
Редагування нотних записів	+		+
Надання додаткової інформації для ручного транскрибування			-
Налаштування відображення			
Відтворення оригіналу поверх транскрипції		-	-

1.4 Висновок до розділу 1

У першому розділі було проведено аналіз предметної області “Транскрибування музичних творів”. Розглянуто проблеми розробки сучасного програмного додатку для спрощення процесу транскрибування музичних творів, а також розглянуто існуючі (на момент написання МКР)

рішення задачі транскрипції музичних творів та виконано їх порівняльний аналіз. Знайдено невелику кількість рішень задачі. Вони не вирішують поставлену задачу в повній мірі та мають відносно велику ціну.

Було сформульовано основні вимоги до роботи, в яких відображається основний функціонал (в рамках МКР) майбутнього програмного додатку.

МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ТОНІЧНОГО АНАЛІЗУ МУЗИЧНИХ ТВОРІВ

2.1 Математична модель тонічного аналізу вхідного дискретного аудіосигналу

Переважає кількість популярних стандартів аудіо-файлів кодуєть аудіосигнал з метою зменшення об'єму файлу при збереженні чи передачі на інші пристрої. Таких стандартів існує дуже велика кількість, але, на щастя, для всіх них вже існують готові рішення для декодування та представлення аудіоформату у вигляді імпульсно-кової модуляції (далі ІКМ) з якою зручно працювати, тому це і буде вхідними даними при подальшій обробці.

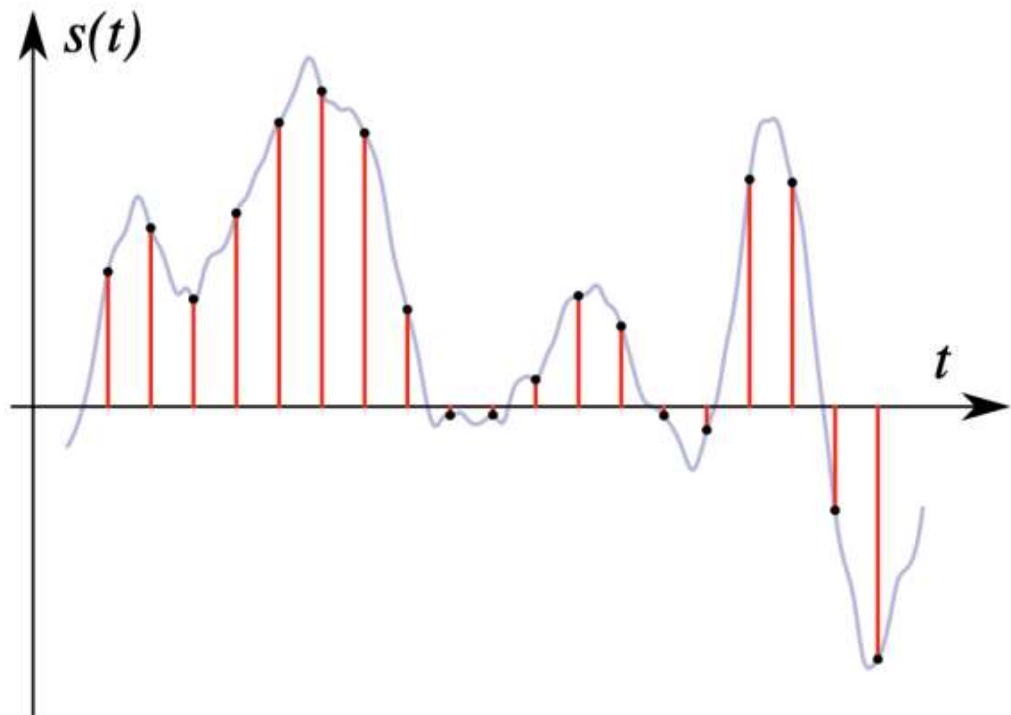


Рисунок 2.1 – Графічне представлення ІКМ-сигналу

Основним інструментом, що використовується в аналізі цифрових аудіо-сигналів є дискретне перетворення Фур'є (далі ДПФ). ДПФ – це математична процедура, що використовується для визначення гармонічного, або частотного, складу дискретних сигналів. ДПФ є однією з найбільш розповсюджених і потужних процедур цифрової обробки сигналів. ДПФ дозволяє аналізувати, перетворювати і синтезувати сигнали такими способами, які неможливі при неперервній (аналоговій) обробці [10].

Для сигналу, заданого вигляді дискретної послідовності $x(n)$, пряме ДПФ та, його тригонометрична форма, мають вигляд

$$S(k) = \sum_{n=0}^{N-1} x_n e^{-i\frac{2\pi}{N}kn} =$$

де

$$e^{-i\frac{2\pi}{N}kn} = \cos\left(\frac{2\pi}{N}kn\right) - i \sin\left(\frac{2\pi}{N}kn\right)$$

та Bk – реальна та уявна складова комплексного спектру сигналу відповідно. Амплітудна $C(k)$ і фазова $\varphi(k)$ складові сигналу відповідно визначаються формулами

$$C(k) = \frac{1}{N} \sum_{n=0}^{N-1} x_n \cos\left(\frac{2\pi}{N}kn\right)$$

$$\varphi(k) = \arctan\left(\frac{\sum_{n=0}^{N-1} x_n \sin\left(\frac{2\pi}{N}kn\right)}{\sum_{n=0}^{N-1} x_n \cos\left(\frac{2\pi}{N}kn\right)}\right)$$

Фазова складова сигналу необхідна здебільшого для точного відновлення початкового дискретного сигналу. Оскільки для генерації спектрограми достатньо лише амплітудної складової, в подальшому будемо використовувати тільки її.

Особливістю в розпізнаванні музичних патернів в порівнянні з іншими звуками є те, що форма частотної характеристики ноти майже повністю зберігається зі зміною висоти ноти. Цю особливість можна використати в подальшому для автоматичної транскрипції твору, тому важливо зберігати інтенсивність та точність максимально схожими для всіх частот при розбитті аудіо.

Для генерації спектрограми, прийнятної для подальших кроків при аналізі, результат перетворення має бути нормалізованим. Також немає необхідності в пошуку складових сигналу на усій площині можливих частот, достатньо лише значення в наперед визначеному наборі частот f_k

де k – номер частоти, d – кількість частотних позицій на ноту, k_0 –

Для нормалізації запропоновано регулювати розмір фрагменту аудіо сигналу для кожної частоти, так щоб кількість періодів була однаковою. Можемо скласти формулу для визначення довжини фрагменту для аналізу

П
О
Ч
а
Т
К
О
В
а
Н
О

=

—

≡

1

де df – частота дискретизації, s – кількість періодів. Враховуючи це, можемо вивести формулу для кожної складової сигналу

Застосувавши отриману формулу для генерації спектрограми отримаємо зображення



Рисунок 2.2 – Представлення результату перетворення вхідного сигналу за допомогою запропонованого методу у вигляді зображення

Використання нейронних мереж для вирішення задачі тонічного аналізу музичних творів

Машинне навчання є сімейством методів за допомогою яких можливо передбачити відповідь базуючись на досвіті, отриманому в процесі навчання. На стадії навчання програма отримує навчальну вибірку даних, в багатьох випадках навчальні дані представлені навчальною парою, а саме входом для нейронної мережі та очікуваним від неї результатом. Вхідними даними можуть

бути будь-які дані (картинка, аудіо, текст), нормалізовані під вхід до нейронної мережі в залежності від її типу та моделі. Наприклад входом до багато шарового персептрона є багатовимірний вектор, з фіксованою кількістю вимірів, а для згорткової нейронної мережі це може бути двовимірне зображення фіксованого розміру [11].

Процесом навчання нейронної мережі називається процес підбору параметрів моделі нейронної мережі, таким чином, що результатом роботи нейронної мережі є рішення задачі. В залежності від типу нейронної мережі, у навчанні можуть бути застосовані різні підходи до її навчання. При досягненні успіху при навчанні, модель нейронної мережі здатна давати правильні передбачування для вхідних наборів, які були відсутні при навчанні, що є основною ціллю навчання нейронних мереж.

Одними з найпопулярніших підходів до навчання є навчання з вчителем та навчання без вчителя. При навчанні з вчителем навчальна вибірка формується як набір навчальних пар, тобто вхід нейронної мережі і бажаний результат. При такому типі навчання на вхід моделі передається вхід навчальної пари, отримується відповідь нейронної мережі, порівнюється вихід моделі з бажаним результатом, вираховується помилка та корегуються ваги в залежності від похибки. Такий алгоритм навчання нейронної мережі з вчителем називається алгоритмом зворотнього розповсюдження помилки. Такий підхід широко застосовується для на навчання багат шарових персептронів та згорткових нейронних мереж [12].

Для кластеризації масиву даних використовується метод навчання без вчителя. Навчальна вибірка при цьому формується без вказання бажаного результату. При навчанні ваги корегуються таким чином що елементи вибірки, які мають схожі ознаки розміщені близько одне до одного в багатовимірному просторі утворюючи кластери об'єктів в багатовимірному просторі [12].

В задачі автоматизації транскрибування музичних творів, методи машинного навчання мають широке застосування, оскільки результати роботи навчених моделей нейронних мереж дають високу точність в транскрибуванні. Недоліком цього підходу є низька швидкодія моделей, особливо на етапі навчання, але з розвитком обчислювальної техніки продуктивність середнього користувацького комп'ютера збільшилась достатньо, щоб можна було ігнорувати цей недолік.

2.3 Вибір типу штучної нейронної мережі для автоматичного транскрибування музичних творів

Перед тим, як обрати тип нейронної мережі, необхідно визначитись з способами подання інформації на вхід нейронної мережі. При транскрибуванні музичних творів на вході маємо цифровий звуковий сигнал, отже необхідно обрати спосіб його представлення нейронній мережі. Передати аудіосигнал нейронній мережі можливо багатьма способами, розглянемо кілька з них.

Найбільш інтуїтивним способом є передавання на вхід мережі фіксований фрагмент дискретного аудіосигналу без додаткової обробки. Проблемою в цьому підході є те, мережа повинна навчитись розпізнавати частоти, коли це можливо зробити на попередньому етапі, тому цей збільшить час необхідний на навчання нейронної мережі.

Поширеною формою подання інформації нейронній мережі аудіосигналу є сигнал перетворений у частотний простір як двовимірне зображення або набір частот в певний момент часу. В цьому випадку можемо використати спосіб перетворення вхідного дискретного аудіосигналу в частотний простір використовуючи математичну модель, описану в розділі 2.1.

На рисунку 2.3 зображена схема представлення частини аудіо сигналу на вхід нейронної мережі.

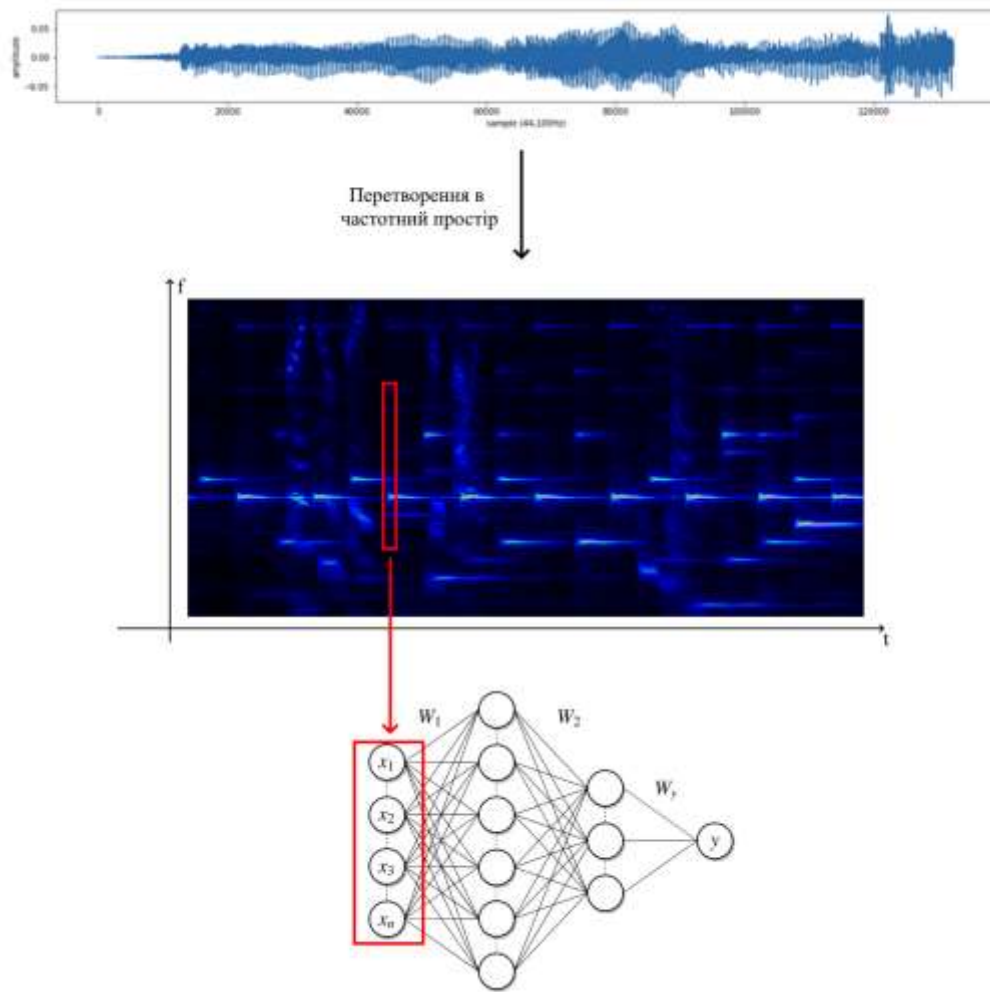


Рисунок 2.3 – Схема перетворення дискретного аудіосигналу при представленні на вхід нейронної мережі

Перевагою багатошарового перцептрона є те, що його просто проектувати, використовувати та навчати. Для навчання багатошарового перцептрона використовується метод зворотнього розповсюдження помилки. Багатошаровий перцептрон є повнозв'язною мережею прямого поширення, тобто складається з набору шарів нейронів, де кожен шар може мати різну кількість нейронів та кожен шар є повнозв'язним з наступним шаром, таким чином інформація, що подається на вхід проходить по черзі по кожному шарі аж до виходу. На вхід багатошарового перцептрона зазвичай передається вектор фіксованої довжини[13]. Вхідний вектор можемо сформулювати взявши значення

з частотного простору опираючись на висоту ноти та момент часу. Таким чином нейронна мережа буде застосована на для кожної ноти в кожен момент часу. Схематичне зображення багатошарового перцептрону зображено на рисунку 2.4.

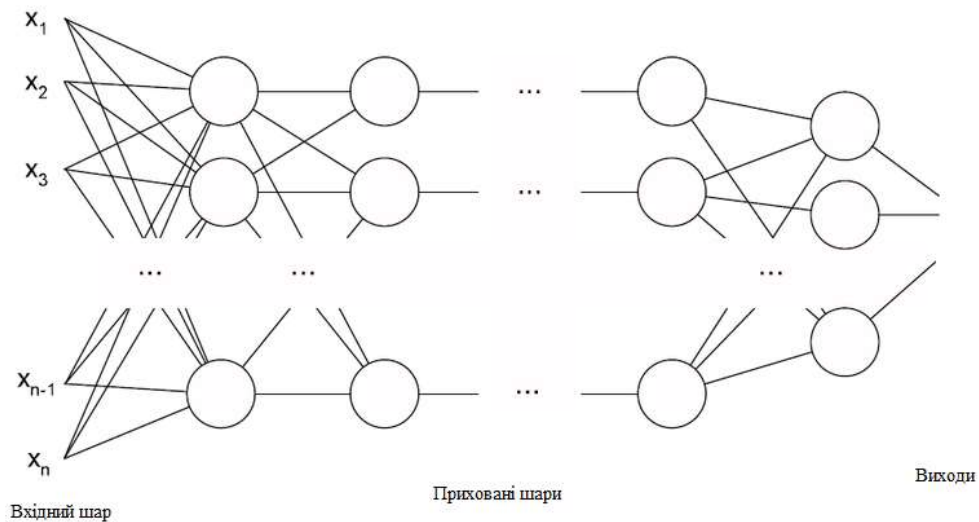


Рисунок 2.4 – Загальна схема структури багатошарового перцептрону

Якщо розглядати результат перетворення дискретного аудіосигналу в частотний простір як двовимірне зображення, то застосувати згортковий тип нейронних мереж. Зображення, отримане в цьому випадку, має фіксований розмір, не може бути повернутим та інвертованим, що пов'язане з тим, що аудіосигнал не може бути симетричним.

Згорткові нейронні мережі аналізують зображення частинами, а не все зображення за раз. Наприклад, класична композиція має спільні деталі, як і інші класичні композиції, в той час як рок-пісня має свої унікальні деталі, і саме ці деталі аналізуються згортковими нейронними мережами. Головна ідея згорткових мереж полягає у класифікації малих деталей.

В основі згорткових нейронних мереж є теорія обробки зображень, а саме, згорткові фільтри (рисунок 2.5). В таких мережах шари досить сильно відрізняються від класичних багат шарових перцептронів. Зазвичай кожен шар може бути згортковим шаром, агрегуючим шаром або повнозв'язним, що зазвичай використовуються в кінці.

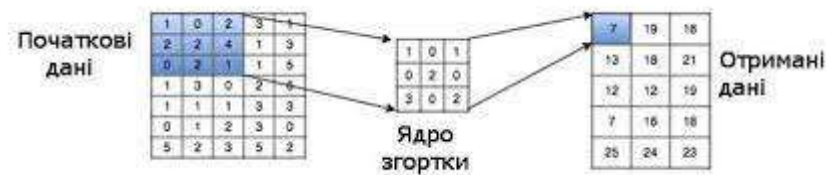


Рисунок 2.5 – Схема роботи згорткових фільтрів

Згортковий шар працює як фільтр згортки, ядром якого є нейрон. Таким чином згортковий нейрон знаходить на зображенні певну ознаку та перетворює вхідне зображення на карту цієї ознаки. Як правило на кожному шарі використовується одразу кілька нейронів, кожен з яких генерує свою карту ознак. Наступні згорткові шари генерують в свою чергу знаходять на картах ознак, отриманих від попереднього шару, більш специфічні та складні карти ознак.

Агрегуючий шар необхідний для спрощення карти ознак, шляхом зменшення розширення карти. Спосіб яким відбувається це зменшення може бути різним як наприклад пошук найбільшого значення в кожному блоці по 4 значення. В цьому випадку карта ознак розміром 16x16 зменшиться до карти ознак розміром 8x8. Оскільки агрегуючий шар не містить жодних динамічних параметрів, його конфігурація не змінюється під час навчання, тому такий тип шару згорткової мережі використовується для збільшення швидкості навчання мережі та її швидкодії.

Повнозв'язні шари використовуються в кінці мережі, де на вхід повнозв'язної частини приходить результат згорткових та агрегуючих шарів, а саме набір карт ознак. Так як на вхід повнозв'язного шару має приходити вектор, карти ознак розкладаються та перетворюються в один великий вектор, що є входом для повнозв'язного шару. Повнозв'язні шари використовуються для формування відповіді згорткової нейронної мережі. На рисунку 2.6 зображено загальну схему згорткових мереж.

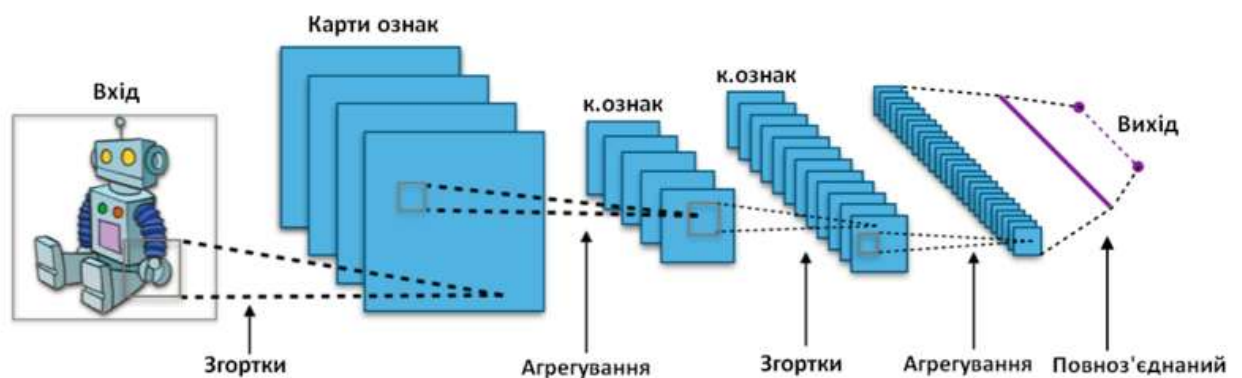


Рисунок 2.6 — Загальна структурна схема згорткової нейронної мережі

Ще однією парадигмою нейронних мереж, що широко застосовується в аналізі сигналів є спайкінгові нейронні мережі. За задумом мережа даного типу проектується в аналоговому вигляді, як набір інтегральних схем, але з розвитком цифрової техніки почали проектуватись цифрові аналоги мереж цього типу. Особливістю спайкінгових мереж полягає у тому, що при роботі враховується часова складова. Штучні нейрони цієї мережі суттєво відрізняються від описаних в інших типах штучних нейронних мереж. Різниця полягає у тому, що на вхід нейрону замість значення надходять імпульси. Імпульс пройшовши через зв'язок з певною вагою може стати активуючим або гальмівним та надходить в нейрон, де накопичується заряд. При досягненні порогового значення створюється вихідний імпульс, а накопичений заряд скидається. Графік заряду спайкінгового нейрону зображено на рисунку 2.7.

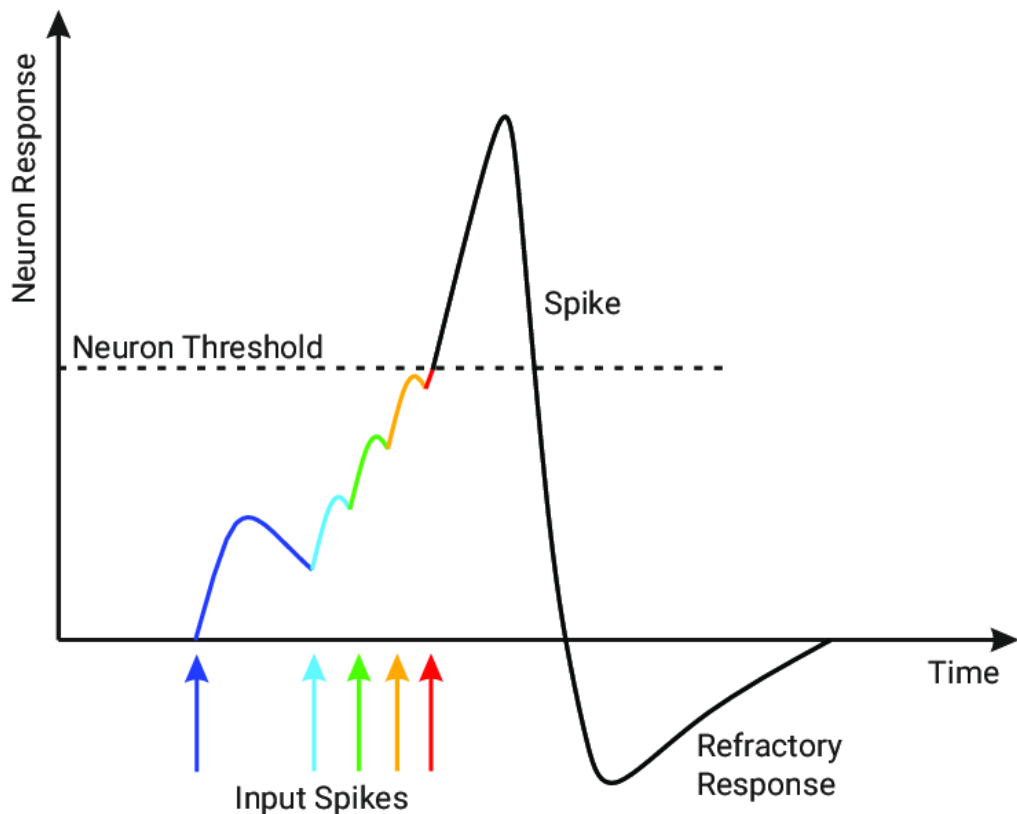


Рисунок 2.7 – Графік заряду спайкінгового нейрону під час роботи

Спайкінгові нейрони організовуються в окремий шар, де зв'язки утворюються довільним чином наприклад, кожен спайкінговий нейрон з'єднаний з 20% інших нейронів в мережі. Певний відсоток виходів спайкінгових нейронів з'єднується з наступним шаром, який вже є звичайним багатошаровим перцептроном. Але так, як вигід спайкінгового нейрону є аналоговий сигнал, для передачі сигналу на вхід багатошарового перцептрону використовується перетворення у вигляді значення. Наприклад сумуються імпульси отримані за одиницю часу.

Навчання спайкінгової нейронної мережі здійснюється навчанням лише повнозв'язних шарів на виході. Таким чином багатошаровий перцептрон навчається розпізнавати патерни, що утворюються в спайкінговому шарі під

впливом певних вхідних імпульсів. Загальний вигляд спайкінгової нейронної мережі зображено на рисунку 2.8.

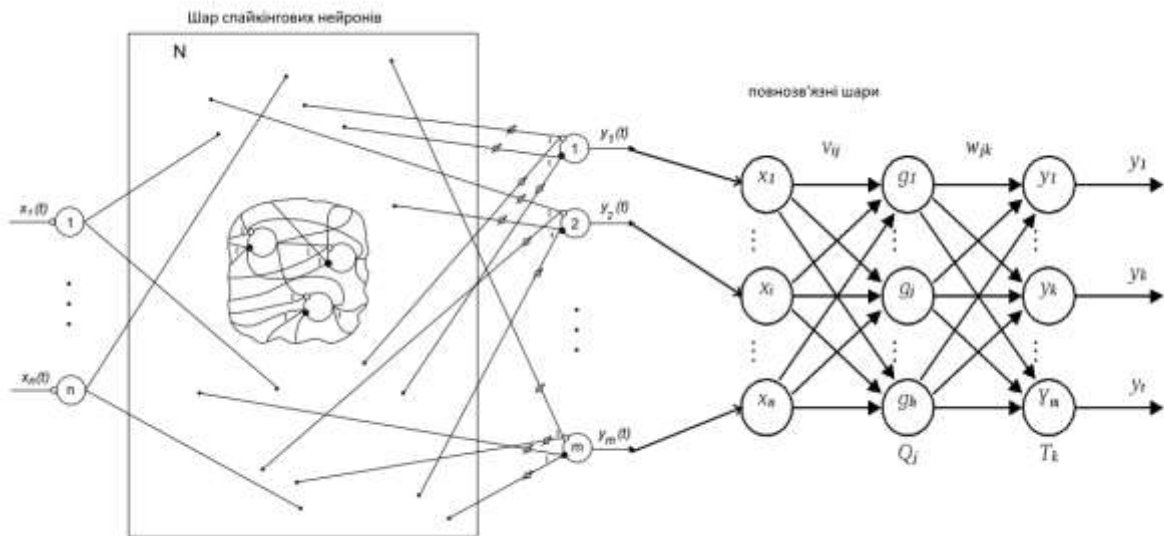


Рисунок 2.8 – загальний вигляд спайкінгової нейронної мережі

Для формування входу для спайкінгової нейронної мережі можемо перетворити інтенсивність частоти в імпульси, наприклад більша інтенсивність — більша частота імпульсів на вхід спайкінгової нейронної мережі.

Отже, для розробки модуля автоматичного транскрибування було розглянуто три парадигми штучних нейронних мереж, та вирішено використати багатошаровий перцептрон, через те, що його просто проектувати, реалізувати та провести навчання методом зворотнього розповсюдження похибки.

Розробка загальної структурної схеми інформаційної технології тонічного аналізу

Основною ціллю програмного додатку, що розробляється в межах магістерської кваліфікаційної роботи, є спрощення процесу транскрибування музичних творів користувачеві, отже додаток повинен мати відповідний набір інструментів а також зручний та зрозумілий інтерфейс. Для досягнення цілі вирішено реалізувати можливість відкриття файлу з аудіозаписом, який буде розбито на складові та відображено на часовій шкалі. Під складовими аудіозапису мається на увазі входження певної ноти у запис. При цьому має бути можливість керувати відтворенням аудіозапису. Основним функціоналом, що забезпечує спрощення процесу транскрибування є можливість записувати ноти поверх відображення складових, можливість автоматично транскрибувати частину запису та можливість експортувати нотні записи у файл .midi.

Для вирішення поставленої задачі спершу необхідно побудувати структурну схему додатку. Під час проектування важливу роль займає діаграма прецедентів. Діаграми прецедентів дають змогу візуалізувати складну поведінку системи. Вони використовуються користувачами з метою зрозуміти способи використання системи, а розробники – з метою імплементувати відповідний елемент системи. Діаграма прецедентів дає можливість відображувати різні типи користувачів системи а також різні випадки її використання. Можливі способи взаємодії з системою відображені колами або еліпсами, стрілками – зв'язки акторів та прецедентів а також можливі переходи між прецедентами. На рисунку 2.7 зображено діаграму прецедентів додатку, що реалізує інформаційну технологію тонічного аналізу музичних творів.

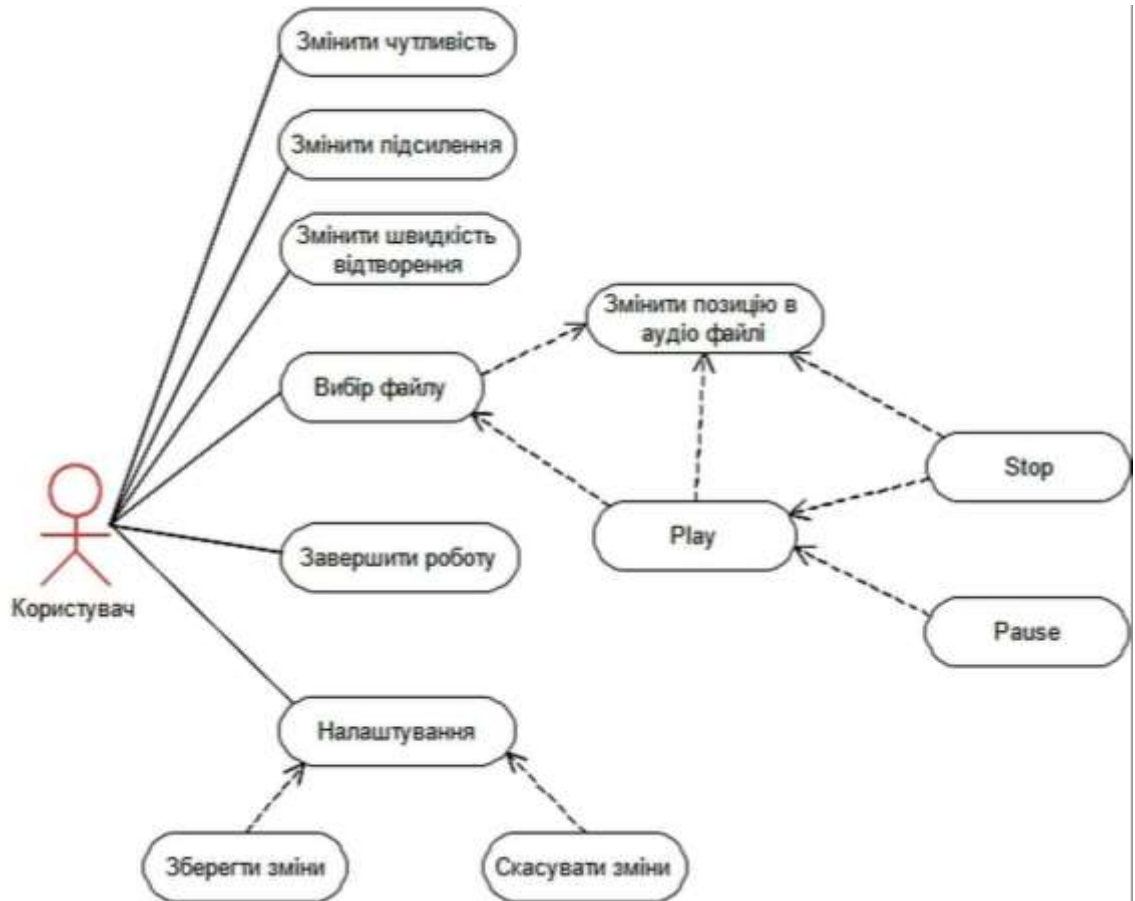


Рисунок 2.7 – Діаграма прецедентів інформаційної технології тонічного аналізу музичних творів

Обробка необхідна обробка аудіозапису є досить дорогою по часу виконання задачею для виконання комп'ютером, особливо якщо середовище виконання має малу швидкодію. Оскільки користувач може керувати відтворенням аудіофайлу, постійне переобчислення результату може негативно вплинути на швидкодію додатка, тому розглянемо можливість збереження всіх результатів обчислення в оперативній пам'яті комп'ютера. Додаток повинен мати можливість гнучкої зміни всіх параметрів. Звідси впливає, що обсяги зайнятої пам'яті в процесі роботи додатку будуть відрізнятися. Тому будемо найгірші за використанням пам'яті налаштування. Людський слух не здатний розрізняти часові відхилення коротші ніж 2мс, тому максимальна кількість часових поділок у вихідних даних на секунду є 500. Максимальним діапазоном

нот є 10(120нот) октав, оскільки це також є обмеженням середньостатистичного людського слуху. Враховуючи мінімальний крок між частотам (1/8 ноти), маємо 960 частотних позицій. Обмеженням по тривалості аудіозапису є 15хв, що покриває переважну більшість музичних композицій. Отже, на збереження обчислень в оперативній пам'яті, найбільшу кількість пам'яті, яку може використати додаток в процесі роботи є ~1.6 Гб, що є цілком прийнятним для сучасних комп'ютерів.

5 Висновок до розділу 2

В даному розділі магістерської роботи було визначено спосіб виведення оброблених даних користувачу, розроблено загальну структуру додатку. Було сформульовано основні вимоги до інтерфейсу користувача, описано основні компоненти та принцип їх роботи. Наведено UML діаграму прецедентів роботи додатку. На діаграмі зображено можливі способи взаємодії користувача з інтерфейсом програми.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ТОНІЧНОГО АНАЛІЗУ МУЗИЧНИХ ТВОРІВ

3.1 Обґрунтування вибору мови програмування для програмної реалізації інформаційної технології тонічного аналізу музичних творів

У наші дні неабияку популярність набула веб розробка додатків. Величезна кількість додатків Які були розроблені як настільні додатки з часом перейшли у веб. Це викликано великою кількістю переваг які пропонує веб розробка, наприклад можливість не встановлювати додаток перед запуском та кросплатформенність “з коробки”.

Коли мова заходить про веб розробку, на думку завжди приходять мова програмування JavaScript. JavaScript – легка інтерпретована або just-in-time спрямована на розробку під веб, але після набуття великою популярності вийшла за межі розробки під веб і використовується для вирішення широкого спектру задач. JavaScript – прототипоно-орієнтована, мультипарадигмальна мова програмування з динамічною типізацією, що підтримує кілька стилів програмування, як об’єктно-орієнтоване програмування, імперативне та декларативне програмування [14].

Хоч і мова JavaScript має велику кількість прихильників, і є широко застосованою, вона має низку недоліків, а саме динамічне типування та низька швидкодія. Динамічне типування не завжди є недоліком, але в складних додатках розробнику доводиться тримати в голові більше деталей про код, який вже був написаний. Це приводить до можливості допускання помилок, які забирають значну частину часу при розробці.

Для вирішення цього недоліку було створено нову мову програмування синтаксис доповнюючи можливістю описувати тип змінних, класи, інтерфейси та інші структури. Дана мова є компільованою мовою, результатом компіляції є звичайні JavaScript скрипти, а отже, TypeScript можна використовувати скрізь, де і JavaScript. мова стрімко набуває популярності та впроваджується у нові проекти замість JavaScript [15].

Т
у
р
е
S
с
r
і
р
t
t
перейняв частину недоліків від JavaScript, а саме низьку швидкість та розробнику створювати додатки з високою швидкістю на будь-якій мові неможливість під час роботи додатку достовірно визначати належність одного типу до іншого.

Вирішити проблему з продуктивністю покликана технологія Web безпечного виконання в середині браузеру. Особливістю цього стандарту є його низький рівень роботи з даними, що надає можливість розробляти додатки під веб з високою швидкістю. Код на Web Assembly сам по собі не є зрозумілим для людини, і розробники не пишуть на ньому свої додатки безпосередньо. Натомість файли з кодом Web Assembly є результатом компіляції з будь-якої іншої мови програмування. Такий підхід дозволяє

В решті решт технологія Web Assembly знайшла певну аудиторію поціновувачів, але і близько не наблизилась до JavaScript. Все це тому, що Web зі DOM сторінки (Document Object Model). Тому, дана технологія зазвичай використовується для розробки модулів, які вимагають високої швидкодії, але для роботи з інтерфейсом користувача все рівно використовується JavaScript.

Отже, для подальшої розробки було обрано мову програмування частину логіки по обробці дискретного аудіо-сигналу в модуль Web Assembly

3.2 Програмна реалізація модуля перетворення вхідного дискретного аудіосигналу

Після відкриття аудіо файлу з диску користувача, інтерфейс для роботи з аудіо в JavaScript повертає масив чисел з плаваючою точкою, в якому кожне значення є дискретною позицією аудіосигналу в кожен момент часу. В більшості випадків кількість значень на одну секунду аудіосигналу рівна 48000 або 44100 значень. Для виконання алгоритму обробки сигналу необхідно для кожної аналізованої частоти в кожен момент часу знайти відхилення середнього значення в полярних координатах. Для цього розробимо клас `AnalyzingPoint`. Розроблений клас містить метод `next`, який викликається для кожного значення вхідних даних і має саме значення як параметр. Значення отримане на вході перетворюється в полярні координати і заноситься в буфер. Для отримання результату клас містить метод `calculateValue`, що повертає середнє відхилення на основі фрагменту хвилі записаної у буфері. Нижче описані методи `next` та `calculateValue` відповідно.

```
public next(value: number){
    this.bufferX[this.position] = value * Math.cos(this.cyclePos);
    this.bufferY[this.position] = value * Math.sin(this.cyclePos);
    this.position = (this.position + 1) % this.length;
    this.cyclePos = (this.cyclePos + this.cycleStep) % (Math.PI * 2);
}

public calculateValue(): number {
```

```

let sumX = 0;
let sumY = 0;
for(let i = 0; i < this.length; i++){
    sumX += this.bufferX[i];
    sumY += this.bufferY[i];
}
sumX /= this.length;
sumY /= this.length;
const res = Math.sqrt(sumX * sumX + sumY * sumY);
return res;
}

```

Набір екземплярів класу `AnalyzingPoint` ініціюються та в подальшому використовуються у композитному класі `Analyzer`. Його роль полягає у отриманні частини аудіоданих проведенні підрахунків для усіх необхідних частот та поверненні масиву значень частот для кожного моменту часу. Метод що виконує це обчислення має назву `analyzeNext`, та приймає значення розміру аудіофрагменту, який необхідно обробити. Далі наведено лістинг конструктору цього класу та методу `analyzeNext`

```

constructor(public settings: AnalyzerOptions, private audioBuffer:
AudioBuffer){
    super();
    this.tickLength = settings.tickLength;
    this.length = audioBuffer.length / settings.tickLength;
    this.tickRate = audioBuffer.sampleRate / settings.tickLength;
    const notes = generateNotes(settings.notes).reverse();
    this.pointsNumber = notes.length;
    this.points = notes.map(note => new
AnalyzingPoint(getNoteFrequency(note, settings.notes.tuning),
audioBuffer.sampleRate, settings.cyclesCount));
}

```

```

}

public analyzeNext(count: number): boolean {
    const data = this.audioBuffer.getChannelData(0);
    const startTick = this.position;
    const endTick = this.position + count;
    for(let t = this.position; t < endTick; t++){
        for(let i = 0; i < this.tickLength; i++){
            const val = data[i + this.tickLength * t] * 0.5 + 0.5;
            this.points.forEach(p => p.next(val));
        }
        this.ticks[t] = this.points.map(p => p.calculateValue());
    }
    this.position = endTick;
    this.emit({from: startTick, to: endTick});
    return this.position < this.length;
}
}

```

3.3 Програмна реалізація інтерфейсу користувача

Так як від додатку не вимагається зберігання інформації, вирішено всю логіку роботи додатку винести на клієнтську сторону додатку, отже, веб застосунок в результаті розробки буде набором статичних файлів вебсайту.

Для розробки клієнтської сторони обрано технологію React. React – це мінімалістична JavaScript бібліотека, що дозволяє швидко розробляти SPA(Single Page Application) застосунки, які легко підтримувати та розширювати. Технологія має стрімкий розвиток, велику кількість прихильників та детальну документацію. Розробка інтерфейсів на React базується на розробці компонентів – частин інтерфейсу, які можуть бути використані повторно в інших компонентах. Це дозволяє ефективно розбивати

інтерфейс на фрагменти та гнучко ними маніпулювати. Окрім зручності та швидкості розробки додатки розроблені на React мають високу швидкодію. Це досягається за рахунок того, що React працює з легким віртуальним DOM, а потім лише зміни у віртуальному DOM точково переносяться у реальний DOM

Нещодавні оновлення React надають можливість розробнику створювати функціональні компоненти у декларативний спосіб. Використання функціональних компонентів у більшості випадків зменшує кількість коду необхідного для реалізації компонента а також спрощує читання коду для розробника [18].

На рисунку 3.1 зображено головне вікно додатку, розроблене за допомогою React



Рисунок 3.1 – Головне вікно додатку

Формування навчальної вибірки та навчання моделі штучної нейронної мережі для тонічного аналізу музичних творів

Штучні нейронні мережі, виконують роботу базуючись на навчальній вибірці, на якій та була навчена, отже для якісного навчання штучної нейронної мережі необхідно мати відповідну навчальну вибірку. Для багатьох різних задач у мережі інтернет можливо знайти відповідну навчальну вибірку, але далеко не для всіх задач. Це обумовлено тим, що в залежності від поставленої задачі вхідні дані можуть мати певну специфіку, а тому сформувати усі можливі навчальні вибірки для усіх можливих задач не є можливим. В задачі транскрибування музичних творів існують готові навчальні вибірки, але такі вибірки строго зав'язані на вигляді вхідних даних, якщо ж ми маємо намір змінити\покращити формат вхідних даних, то існуючих вибірок немає і доводиться створювати власну.

Формування навчальної вибірки вручну є трудомістким завданням, від виконання якого залежить подальша якість навчання штучної нейронної мережі. Існують різні шляхи оптимізації генерування вибірки. У випадку транскрибування, у мережі Інтернет хоч і немає потрібно вибірки, але є велика кількість нотних записів у вигляді .midi – файлів. Їх можна використати для формування навчальної вибірки шляхом перетворення їх на аудіо за допомогою програмного забезпечення для синтезування аудіо. Таким чином можемо з формувати навчальну вибірку використовуючи набір нот з мережі бажану відповідь, а синтезоване аудіо подаючи на вхід. Для покращення результату можемо використовувати різні синтезатори аудіо, для синтезування різних інструментів.

Сформована навчальна вибірка у такий спосіб представляє собою тьох-годинний аудіо файл з записом гри фортепіано, органу та електрогітари. До аудіо файлу представлений один .midi файл що містить оригінальні ноти запису. При навчанні на вхід передається перетворений вхідний аудіофайл, а для

проведення навчання алгоритмом зворотнього розповсюдження помилки як очікуваний результат беруться значення з нотного файлу. На рисунку 3.2 зображено графічне представлення способу генерування навчальної вибірки.

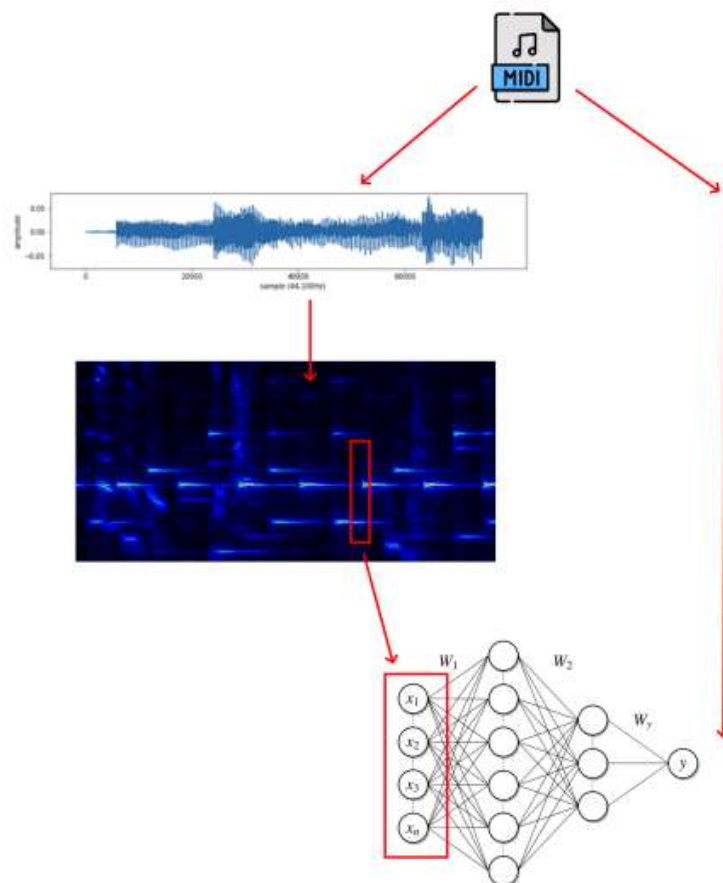


Рисунок 3.2 – Графічне представлення способу генерування навчальної вибірки

У ході навчання різних моделей багатошарового перцептрона, було виявлено що найкращих результатів досягає модель багатошарового перцептрона, що має 65 входи, має один прихований шар з 8-ма нейронами що мають логістичну функцію активації, та одним нейроном на виході з ступінчастою функцією активації. На 64 входи до нейронної мережі передаються значення частотних складових у певний момент часу, та на один

вхід передається значення результату попередньої ітерації. На рисунках 3.3 та 3.4 зображено схему спроектованої моделі штучної нейронної мережі та графік її навчання відповідно.

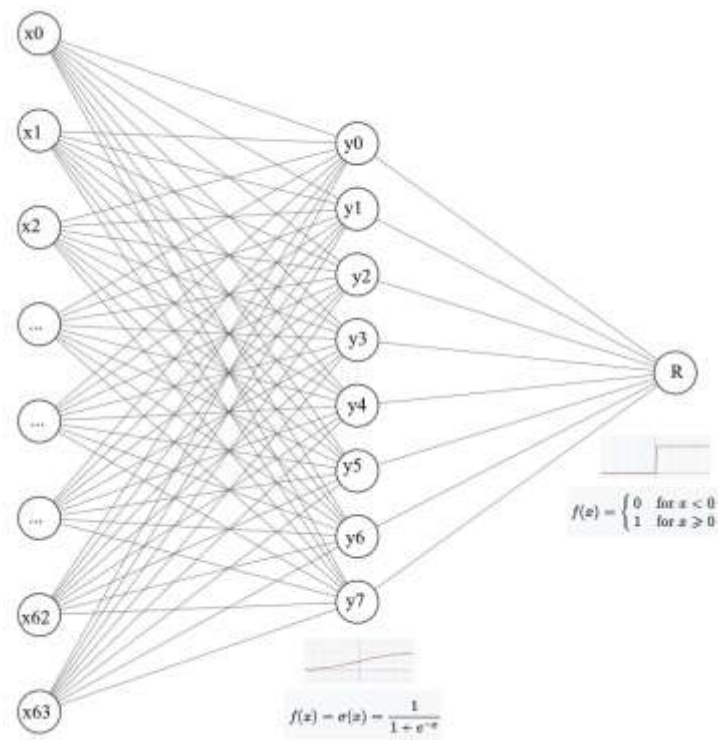


Рисунок 3.3 – Графічне представлення спроектованої моделі багатошарового перцептрона

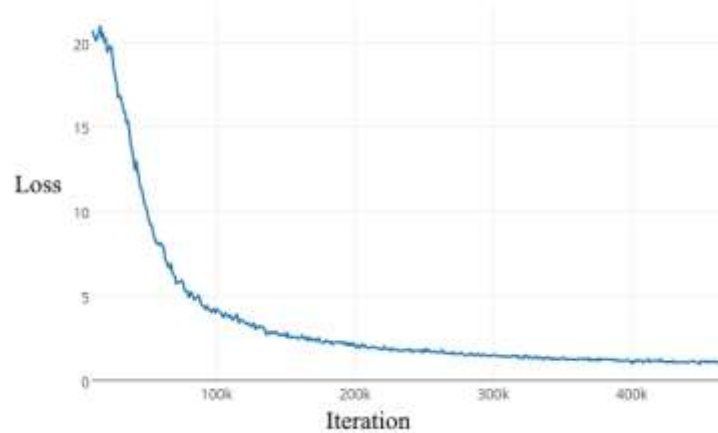


Рисунок 3.4 – Графік прогресу навчання спроектованої моделі штучної нейронної мережі

Тестування та аналіз програмного забезпечення тонічного аналізу музичних творів

Додаток, що реалізує інформаційну технологію тонічного аналізу музичних творів було розроблено як Web-додаток, отже він з себе представляє сервер до якого користувач надсилає надсилає запит на отримання сторінки. Перевіримо роботу серверу спробувавши завантажити сторінку. Так як сайт розміщено на локальному комп'ютері, то URL адреса до сайту буде

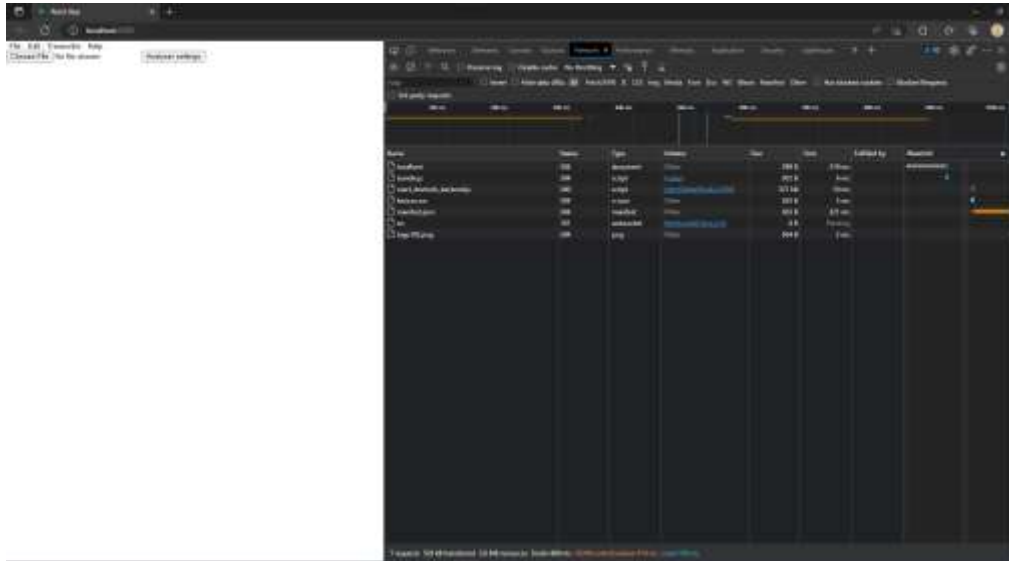


Рисунок 3.5 – Результат завантаження сторінки

Як бачимо з інструментів розробника, клієнт успішно отримав у відповідь сторінку з самим додатком. Також слід зазначити, що сторінку завантажено достатньо швидко – 880мс.

Далі перевіримо відкриття файлу. Для цього необхідно натиснути кнопку

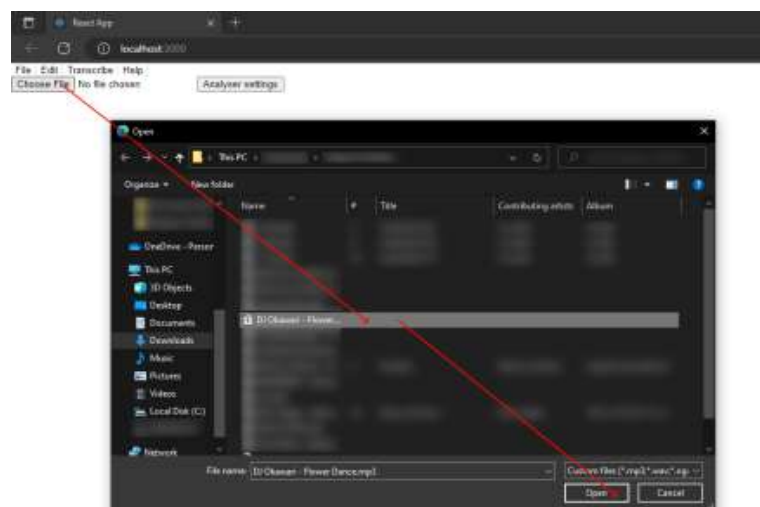


Рисунок 3.6 – Відкриття файлу в розробленому додатку

Після цього буде відкрите головне вікно додатку. Наступним кроком перевіримо генерацію візуалізації. Для цього необхідно натиснути кнопку Start. Після цього на екрані почне з'являтися візуалізація ліва у право (рисунок 3.7).

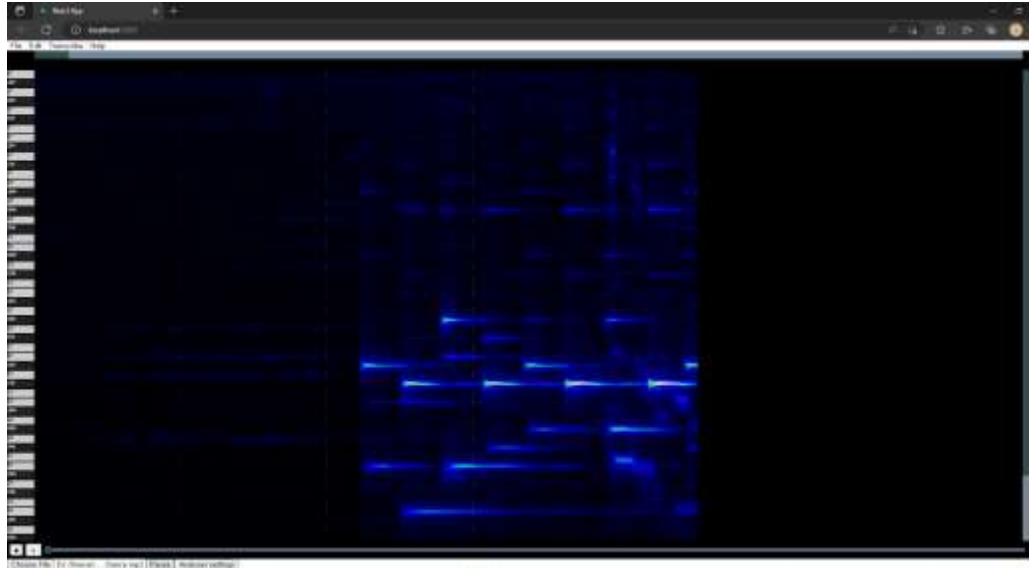


Рисунок 3.7 – Генерація візуалізації з вхідного аудіосигналу

Слід зауважити, що генерація візуалізації відбувається асинхронно з відтворенням аудіо, таким чином, під час генерації візуалізації користувач незалежно може відтворювати аудіофайл за допомогою центрального повзунка та кнопок керування відтворенням.

Розроблений додаток підтримує зміну налаштувань візуалізації, але при цьому увесь прогрес по генерації скидається і починається спочатку. В цих налаштуваннях можна задати якість, густину та діапазон. Змінимо ці параметри натиснувши кнопку Analyzer settings (рисунок 3.8 та 3.9).



Рисунок 3.8 – Вікно зміни налаштувань візуалізації

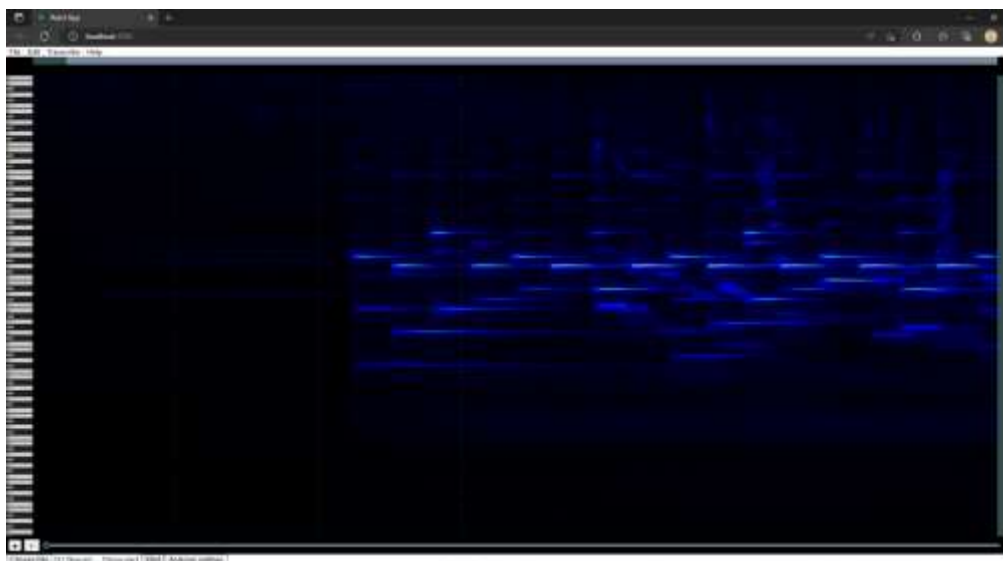


Рисунок 3.9 – Візуалізація із зміненими параметрами

Наступним кроком перевіримо функцію автоматичного транскрибування. Для цього необхідно натиснути кнопку Transcribe і у вікні яке, з'явилося знову натиснути Transcribe. Після цього на екрані швидко почнуть з'являтися ноти (рисунок 3.10).

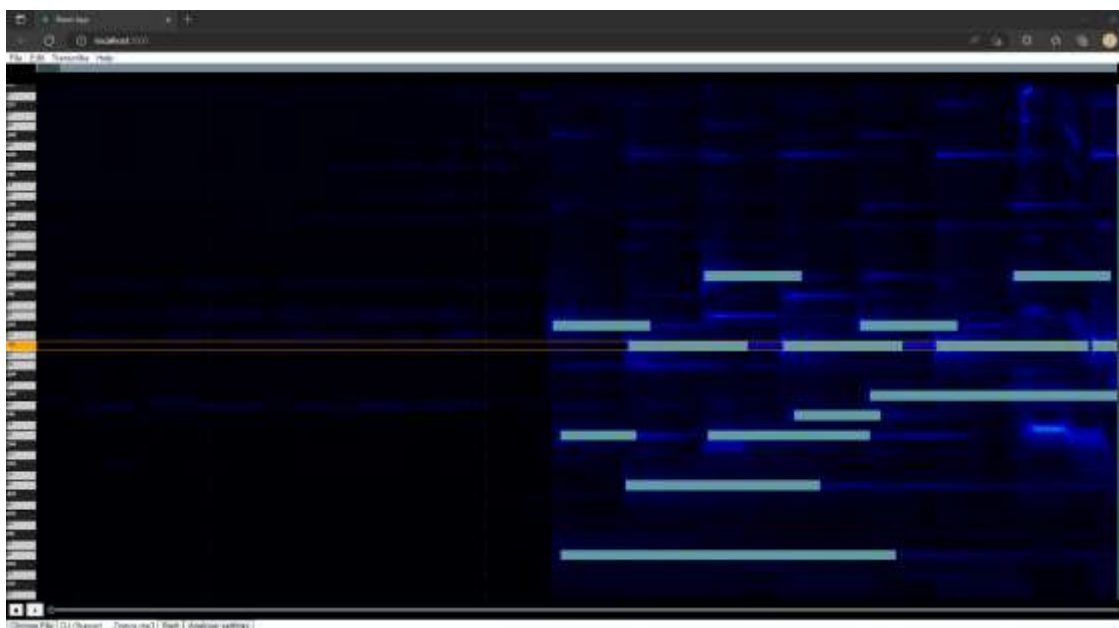


Рисунок 3.10 – Результат роботи автоматичного транскрибування

В результаті розробки додатку, що реалізує інформаційну технологію тонічного аналізу було досягнуто поставленої мети за рахунок реалізації функцій, що відсутні в системах-аналогах, а саме:

- Кросплатформенність
- Налаштування аналізу вхідних аудіоданих
- Відображення візуалізації
- Налаштування відображення візуалізації
- Автоматичне транскрибування
- Ручне коригування
- Керування відтворенням оригіналу

Порівняння наявності цих функцій з системами-аналогами наведено в таблиці 3.1

Таблиця 3.1 — Таблиця порівняння наявності функцій розробленого додатку та систем-аналогів

Функція	Розроблений продукт		Notation composer	Mei
Кросплатформенність				
Налаштування аналізу вхідних аудіоданих				
Відображення візуалізації				
Налаштування відображення візуалізації				
Автоматичне транскрибування				
Ручне коригування				
Керування відтворенням оригіналу				

З наведеної таблиці можна побачити, що розроблений додаток, який реалізує інформаційну технологію тонічного аналізу, має дві функції, які немає жодна інша система-аналог, а також комбінує в собі функції систем аналогів.

Висновок до розділу 3

В даному розділі магістерської роботи було обрано мову програмування для реалізації інформаційної технології тонічного аналізу музичних творів. Було визначено підхід до розробки додатку для максимальної ефективності. Розроблено графічний інтерфейс додатку. Сформовано навчальну вибірку для навчання штучної нейронної мережі з вчителем. Створено модель штучної нейронної мережі та було проведено її навчання виведення. Проведено тестування розробленого додатку та з'ясовано, що усі заплановані функції працюють у повній мірі. Було розширено функціонал програмного забезпечення тонічного аналізу музичних творів. Розроблений програмний продукт має дві нові функції які не мають системи аналогії.

ЕКОНОМІЧНА ЧАСТИНА

4.1 Комерційний та технологічний аудит науково-технічної розробки

Метою даного розділу є проведення технологічного аудиту, в даному випадку нового програмного продукту тонічного аналізатора музичних творів. Особливістю програми є те, що вона є браузерним додатком, що покликаний спростити процес транскрипції музичний творів за рахунок генерації візуалізації аудіо, що гнучко налаштовується, а також за допомогою інструментів автоматичної транскрипції.

Аналогом може бути AnthemScore 4, ціна 1600 грн; Melody Scanner, ціна 1720 грн.

Для проведення комерційного та технологічного аудиту залучають не менше 3-х незалежних експертів. Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням п'ятибальної системи оцінювання за 12-ма критеріями, у відповідності із табл. 4.1.

Таблиця 4.1 – Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Бали (за 5-ти бальною шкалою)					
Кри-терій					
Технічна здійсненність концепції					
	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах

Продовження табл. 4.1

Ринкові переваги					
	Багато аналогів на малому ринку	Ринкові п Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практик на здійсненність					
	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві

Продовження табл. 4.1

	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Усі дані по кожному параметру занесено в таблиці 4.2

Таблиця 4.2 – Результати оцінювання комерційного потенціалу розробки

Критерії оцінювання	ПІБ експертів		
	Експерт 1	Експерт 2	Експерт 3
	Бали		
Технічна здійсненність концепції			
Наявність аналогів на ринку			
Цінова політика			
Технічні та споживчі властивості виробу			
Експлуатаційні витрати			
Ринок збуту			
Конкурентоспроможність			
Фахівці з технічної і комерційної реалізації			
Фінансування			
Матеріально-технічна база			
Термін реалізації ідеї			
Супровідна документація			
Сума			
Середньоарифметична сума балів	/		

За даними таблиці 4.2 можна зробити висновок щодо рівня комерційного потенціалу даної розробки. Для цього доцільно скористатись рекомендаціями, наведеними в таблиці 4.3.

Таблиця 4.3 - Рівні комерційного потенціалу розробки

Середньоарифметична сума балів СБ , розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
	Низький
	Нижче середнього
	Середній
	Вище середнього
	Високий

Як видно з таблиці, рівень комерційного потенціалу розроблюваного нового програмного продукту є високим, що досягається за рахунок того, що програмний продукт відрізняється від існуючих тим, що дана технологія є браузерним додатком, що покликаний спростити процес транскрипції музичний творів за рахунок генерації візуалізації аудіо, що гнучко налаштовується, а також за допомогою інструментів автоматичної транскрипції.

4.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи

4.2.1 Основна заробітна плата розробників, яка розраховується за формулою:

$$Z = \frac{M \cdot T_p \cdot t}{P}$$

де M – місячний посадовий оклад конкретного розробника (дослідника), грн.;

T_p – число робочих днів в місяці, 22 днів;

t – число днів роботи розробника (дослідника).

Z
o
M
T
P
.

Результати розрахунків зведемо до таблиці 4.1.

Таблиця 4.1 – Основна заробітна плата розробників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник проекту				
Інженер				
Всього				

Так як в даному випадку розробляється програмний продукт, то розробник виступає одночасно і основним робітником, і тестувальником розроблюваного програмного продукту.

4.2.2 Додаткова заробітна плата розробників, які приймали участь в розробці обладнання.

Додаткова заробітна плата прийнято розраховувати як 13 % від основної заробітної плати розробників та робітників:

$$Z_d = (145090,91 \cdot 13 \% / 100 \%) = 18861,82 \text{ (грн.)}$$

4.2.3 Нарахування на заробітну плату розробників.

Згідно діючого законодавства нарахування на заробітну плату складають 22 % від суми основної та додаткової заробітної плати.

3

д

=

Н

з

=

$$H_3 = (145090,91 + 18861,82) \cdot 22 \% / 100 \% = 36069,60 \text{ (грн.)}$$

4.2.4. Оскільки для розроблювального пристрою не потрібно витратити матеріали та комплектуючі, то витрати на матеріали і комплектуючі дорівнюють нулю.

4.2.5 Амортизація обладнання, яке використовувалось для проведення розробки.

Амортизація обладнання, що використовувалось для розробки в спрощеному вигляді амортизація обладнання, що використовувалась для розробки розраховується за формулою:

=

де Ц – балансова вартість обладнання, грн.;

T – термін корисного використання обладнання згідно податкового законодавства, років

$t_{\text{вик}}$ – термін використання під час розробки, місяців

Розрахуємо, для прикладу, амортизаційні витрати на комп'ютер балансова вартість якого становить 26000 грн., термін його корисного використання згідно податкового законодавства – 2 роки, а термін його фактичного використання – 1,73 міс.

$$A_{\text{обл}} = \frac{26000}{2} \times \frac{1,73}{12} = 1871,212 \text{ грн.}$$

Аналогічно визначаємо амортизаційні витрати на інше обладнання та приміщення. Розрахунки заносимо до таблиці 4.2. Для розрахунку амортизації нематеріальних ресурсів використовується формула:

Але, так як вартість ліцензійної ОС та спеціалізованих ліцензійних

нематеріальних ресурсів менше 20000 грн, то даний нематеріальний актив не амортизується, а його вартість включається у вартість розробки повністю, $V_{нем.ак.} = 9000$ грн.

Таблиця 4.2 – Амортизаційні відрахування матеріальних і нематеріальних ресурсів для розробників

Найменування обладнання	Балансова вартість, грн.	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн.
Комп'ютер та комп'ютерна периферія				
Офісне обладнання				
Приміщення				
Всього				

2.6 Тарифи на електроенергію для побутових споживачів (промислових підприємств) відрізняються від тарифів на електроенергію для населення. При цьому тарифи на розподіл електроенергії у різних постачальників (енергорозподільних компаній), будуть різними. Крім того, розмір тарифу залежить від класу напруги (1-й або 2-й клас). Тарифи на розподіл електроенергії для всіх енергорозподільних компаній встановлює Національна комісія з регулювання енергетики і комунальних послуг (НКРЕКП). Витрати на силову електроенергію розраховуються за формулою:

$$= \dots$$

де B – вартість 1 кВт-години електроенергії для 1 класу підприємства, $B = 6,2$ грн./кВт;

P – встановлена потужність обладнання, кВт. $P = 0,5$ кВт;

Φ – фактична кількість годин роботи обладнання, годин.

$K_{\text{п}}$ – коефіцієнт використання потужності, $K_{\text{п}} = 0,9$.

$$V_e = 0,9 \cdot 0,5 \cdot 8 \cdot 38 \cdot 6,2 = 848,16 \text{ (грн.)}$$

2.7 Інші витрати та загальновиробничі витрати.

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками. Витрати за статтею «Інші витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників:

$$I_e = 145090,91 \cdot 105\% / 100\% = 152345,5 \text{ (грн.)}$$

До статті «Накладні (загальновиробничі) витрати» належать: витрати пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін. Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників:

$$H_{\text{нзв}}$$

$$\text{де } H_{\text{нзв}} \text{ – норма нарахування за статтею «Накладні (загальновиробничі) витрати».$$

$$H_{нзв} = 145090,91 * 130 \% / 100 \% = 188618 \text{ (грн.)}$$

2.9 Витрати на проведення науково-дослідної роботи.

Сума всіх попередніх статей витрат дає загальні витрати на проведення науково-дослідної роботи:

$$B_{заг} = 145090,91 + 18861,82 + 36069,60 + 9068,18 + 9000 + 848,16 + 152345,5 + \text{грн.}$$

Розрахунок загальних витрат на науково-дослідну (науково-технічну) роботу та оформлення її результатів.

Загальні витрати на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються ZB , визначається за формулою:

$$ZB = \frac{B_{заг}}{\eta} \text{ (грн)}, \quad (4.9)$$

де η – коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи.

Так, якщо науково-технічна розробка знаходиться на стадії: науково-дослідних робіт, то $\eta=0,1$; технічного проектування, то $\eta=0,2$; розробки конструкторської документації, то $\eta=0,3$; розробки технологій, то $\eta=0,4$; розробки дослідного зразка, то $\eta=0,5$; розробки промислового зразка, то $\eta=0,7$; впровадження, то $\eta=0,9$. Оберемо $\eta = 0,5$, так як розробка, на даний момент, знаходиться на стадії дослідного зразка:

$$ZB = 559902,31 / 0,5 = 1119805 \text{ грн.}$$

Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

В ринкових умовах узагальнювальним позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів цієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку. Саме зростання чистого прибутку забезпечить потенційному інвестору надходження додаткових коштів, дозволить покращити фінансові результати його діяльності, підвищить конкурентоспроможність та може позитивно вплинути на ухвалення рішення щодо комерціалізації цієї розробки.

Для того, щоб розрахувати можливе зростання чистого прибутку у потенційного інвестора від можливого впровадження науково-технічної розробки необхідно:

а) вказати, з якого часу можуть бути впроваджені результати науково-технічної розробки;

б) зазначити, протягом скількох років після впровадження цієї науково-технічної розробки очікуються основні позитивні результати для потенційного інвестора (наприклад, протягом 3-х років після її впровадження);

в) кількісно оцінити величину існуючого та майбутнього попиту на цю або аналогічні чи подібні науково-технічні розробки та назвати основних суб'єктів (зацікавлених осіб) цього попиту;

г) визначити ціну реалізації на ринку науково-технічних розробок з аналогічними чи подібними функціями.

При розрахунку економічної ефективності потрібно обов'язково враховувати зміну вартості грошей у часі, оскільки від вкладення інвестицій до отримання прибутку минає чимало часу. При оцінюванні ефективності інноваційних проектів передбачається розрахунок таких важливих показників:

- абсолютного економічного ефекту (чистого дисконтованого доходу);
- внутрішньої економічної дохідності (внутрішньої норми дохідності);
- терміну окупності (дисконтованого терміну окупності).

Аналізуючи напрямки проведення науково-технічних розробок, розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором можна об'єднати, враховуючи визначені ситуації з відповідними умовами.

4.3.1 Розробка чи суттєве вдосконалення програмного засобу (програмного забезпечення, програмного продукту) для використання масовим споживачем.

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

$$\Delta\Pi_i = (\pm\Delta C_0 \cdot N + C_0 \cdot \Delta N) \cdot i \cdot \lambda \cdot \rho \cdot (1 - \vartheta^{100}), \quad (4.10)$$

де $\pm\Delta C_0$ – зміна вартості програмного продукту (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу;

– кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки;

C_0 – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки, $C_0 = C_0 \pm \Delta C_0$;

C_0 – вартість програмного продукту у році до впровадження результатів розробки;

ΔN – збільшення кількості споживачів продукту, в аналізовані періоди часу, від покращення його певних характеристик;

λ – коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт $\lambda = 0,8333$.

ρ – коефіцієнт, який враховує рентабельність продукту;

ϑ – ставка податку на прибуток, у 2022 році $\vartheta = 18\%$.

Припустимо, що при прогнозованій ціні 800 грн. за одиницю виробу, термін збільшення прибутку складе 3 роки. Після завершення розробки і її вдосконалення, можна буде підняти її ціну на 50 грн. Кількість одиниць реалізованої продукції також збільшиться: протягом першого року – на 15000 шт., протягом другого року – на 20000 шт., протягом третього року на 25000 шт. До моменту впровадження результатів наукової розробки реалізації продукту не було:

$$\Delta\Pi_1 = (0*50 + (800 + 50) * 15000) * 0,8333 * 0,4 * (1 - 0,18) = 3279999,869 \text{ грн.}$$

$$\Delta\Pi_2 = (0*50 + (800 + 50) * (15000 + 20000)) * 0,8333 * 0,4 * (1 - 0,18) = 8131666,341 \text{ грн.}$$

$$\Delta\Pi_3 = (0*50 + (800 + 50) * (15000 + 20000 + 25000)) * 0,8333 * 0,4 * (1 - 0,18) = 13939999,442 \text{ грн.}$$

Отже, комерційний ефект від реалізації результатів розробки за три роки складе 25351665,65 грн.

Розрахунок ефективності вкладених інвестицій та періоду їх окупності.

Розраховуємо приведену вартість збільшення всіх чистих прибутків Π_i , що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$T \Delta\Pi_i (1 + \tau)^t, \quad (4.11)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої науково-дослідної (науково-технічної) роботи, грн;

T – період часу, протягом якою виявляються результати впровадженої науково-дослідної (науково-технічної) роботи, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований

p

i

0,05...0,15;

t – період часу (в роках).

Збільшення прибутку ми отримаємо починаючи з першого року:

ПП=

грн.

Далі розраховують величину початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

P

V

де $k_{инв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{инв}=2...5$, але може бути і більшим;

ZB – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

P

V

грн. Тоді абсолютний економічний ефект $E_{абс}$ або чистий приведений дохід впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = ПП - PV, \quad (4.13)$$

$$E_{абс} = 20175531,37 - 2239609,22 = 17935922,15 \text{ грн.}$$

Оскільки $E_{abc} > 0$ то вкладання коштів на виконання та впровадження результатів даної науково-дослідної (науково-технічної) роботи може бути доцільним.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність або показник внутрішньої норми дохідності (*IRR, Internal Rate of Return*) вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій E_e . Для цього використаємо формулу:

$$E_e = \sqrt[T_{жс}]{1 + \frac{E_{abc}}{PV}} - 1,$$

$T_{жс}$ – життєвий цикл наукової розробки, роки.

$$E_e = 3 \left(1 + 17935922,15/2239609,22 \right)^{1/3} - 1 = 1,081$$

Визначимо мінімальну ставку дисконтування, яка у загальному вигляді

визначається за формулою:

$$\tau = d + f,$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2022 році в Україні $d = (0,09...0,14)$;

– показник, що характеризує ризикованість вкладень; зазвичай, величина $f =$

$$\tau_{\min} = 0,14 + 0,05 = 0,19$$

Так як $E_v > \tau_{\min}$, то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій за формулою:

$$T_{ок} = \frac{1}{E_{\theta}}$$

$$T_{ок} = 1 / 1,081 = 0,93 \text{ р.}$$

Оскільки $T_{ок} < 3$ -х років, а саме термін окупності рівний 0,93 роки, то фінансування даної наукової розробки є доцільним.

4.4 Висновок до розділу 4

У розділі було виконано розрахунок витрат на розробку нового програмного продукту, сума яких складає 1119805 гривень. Було спрогнозовано орієнтовану величину витрат по кожній з статей витрат. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення, розраховано період окупності витрат для інвестора та економічний ефект при використанні даної розробки. В результаті аналізу розрахунків можна зробити висновок, що розроблений програмний продукт за ціною дешевший за аналог і є висококонкурентоспроможним. Період окупності складе близько 0,93 роки.

ВИСНОВКИ

У першому розділі було проведено аналіз предметної області “Транскрибування музичних творів”. Розглянуто проблеми розробки сучасного програмного додатку для спрощення процесу транскрибування музичних творів, а також розглянуто існуючі (на момент написання МКР) рішення задачі транскрипції музичних творів та виконано їх порівняльний аналіз. Знайдено невелику кількість рішень задачі. Вони не вирішують поставлену задачу в повній мірі та мають відносно велику ціну. Було сформульовано основні вимоги до роботи, в яких відображається основний функціонал програмного додатку.

У другому розділі магістерської роботи було визначено спосіб виведення оброблених даних користувачу, розроблено загальну структуру додатку. Розглянуто різні парадигми штучних нейронних мереж в області транскрибування музичних творів. Для розробки обрано багат шаровий перцептрон. Було сформульовано основні вимоги до інтерфейсу користувача, описано основні компоненти та принцип їх роботи. Наведено UML діаграму прецедентів роботи додатку. На діаграмі зображено можливі способи взаємодії користувача з інтерфейсом програми.

В третьому розділі магістерської роботи було обрано мову програмування для реалізації інформаційної технології тонічного аналізу музичних творів. Було визначено підхід до розробки додатку для максимальної ефективності. Розроблено графічний інтерфейс додатку. Сформовано навчальну вибірку для навчання штучної нейронної мережі з вчителем. Створено модель штучної нейронної мережі та було проведено її навчання. Проведено тестування розробленого додатку та з’ясовано, що усі заплановані функції працюють у повній мірі.

розрахунок витрат на розробку нового програмного продукту, сума яких

с

к

л

а

д

а

є

1

1

1

9

8

0

5

г

р

и

в

е

н

ь

.

Б

У

л

о

с

п

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гульчак Р.С., Барабан С.В. «Розробка тонічного аналізатора музичних творів» в матеріали конференції «Науково-технічна конференція підрозділів Вінницького національного технічного університету», Вінниця, 2010.
2. Девід Гергард «Computer music analysis, Simon Fraser Universit 1997. 2011.
3. Бородин Б.Б. История фортепианной транскрипции: эволюция направлений, стилей и методов в контексте художественной культуры: монография. 55с. [Електронний ресурс] – Режим доступу: [https://www.researchgate.net/publication/312211114](#)
5. Emmanouil Benetos, Simon Dixon, Dimitrios Giannoulis, Holger Krumholz.
6. A.
7. N.
8. M.
9. Методичні вказівки до виконання студентами-магістрантами економічної частини магістерських кваліфікаційних робіт / Уклад. В. О. Козловський – Вінниця: ВНТУ, 2012. – 22 с.
10. Річард Лайонс – Цифрова обробка сигналів 2006р. 656с.
Finding Using the Game of Checkers. IBM Journal of Research and Development 1959p. 210–229 с.
11. [Електронний ресурс] – Режим доступу: [https://www.researchgate.net/publication/312211114](#)
12. [Електронний ресурс] – Режим доступу: [https://www.researchgate.net/publication/312211114](#)
13. MDN – Javascript – Programming language [Електронний ресурс]
14. Багатошаровий перцептрон, принципи функціонування і навчання [Електронний ресурс] – Режим доступу: <https://melodyscanner.com>.

[Електронний ресурс] Режим доступу: <https://www.typescriptlang.org/>

16. W

e 17. React – A JavaScript library for building user interfaces [Електронний ресурс] Режим доступу: <https://reactjs.org/>

A 18. React – Introducing Hooks [Електронний ресурс] Режим доступу: <https://reactjs.org/docs/hooks-intro.html>

s 19. T

e 20. Прогнозування витрат на виконання науково-дослідної, дослідно-конструкторської та конструкторсько-технологічної роботи [Електронний ресурс]. URL: <https://web.posibnyky.vntu.edu.ua>

б 21. Ефективність наукових досліджень [Електронний ресурс]. URL: <https://studfile.net/preview/6872280/page:25/>

Електронний ресурс] Режим доступу: <https://webassembly.org/>

Г

і

ч

н

и

й

а

у

д

и

т

[

Е

л

е

к

ДОДАТКИ

ДОДАТОК А (ОБОВ'ЯЗКОВИЙ) РЕЗУЛЬТАТ ПЕРЕВІРКИ НА ПЛАГІАТ В ОНЛАЙН-СИСТЕМІ



Ім'я користувача:
Озеранський В.С. КН

ID перевірки:
1013334269

Дата перевірки:
20.12.2022 10:50:44 EET

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
20.12.2022 10:53:48 EET

ID користувача:
62038

Назва документа: 122МКР-ГульчакРС2022

Кількість сторінок: 37 Кількість слів: 5374 Кількість символів: 41607 Розмір файлу: 678.60 KB ID файлу: 1013094037

5.23% Схожість

Найбільша схожість: 5.23% з джерелом з Бібліотеки (ID файлу: 1008352932)

Не знайдено джерел з Інтернету

5.23% Джерела з Бібліотеки

1

Сторінка 39

0% Цитат

Не знайдено жодних цитат

Не знайдено жодних посилань

4.91% Вилучень

Деякі джерела вилучено автоматично (фільтри вилучення: кількість знайдених слів є меншою за 8 слів та 5%)

2.36% Вилучення з Інтернету

25

Сторінка 40

4.15% Вилученого тексту з Бібліотеки

160

Сторінка 40

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

9

ДОДАТОК Б (ОБОВ'ЯЗКОВИЙ)

ЛІСТИНГ ПРОГРАМИ

Файл analyzingPoint.ts

```
export class AnalyzingPoint {
  private bufferX: number[] = [];
  private bufferY: number[] = [];
  private length: number;
  private position: number = 0;
  private cyclePos: number = 0;
  private cycleStep: number = 0;

  constructor(
    public frequency: number,
    public sampleRate: number,
    cycles: number
  ){

    this.length = Math.floor(cycles * sampleRate / frequency);
    this.cycleStep = 2 * Math.PI * frequency / sampleRate;

    for(let i = 0; i < this.length; i++){
      this.bufferX.push(0.5 * Math.cos(i * this.cycleStep));
      this.bufferY.push(0.5 * Math.sin(i * this.cycleStep));
    }
  }

  public next(value: number){
    this.bufferX[this.position] = value * Math.cos(this.cyclePos);
    this.bufferY[this.position] = value * Math.sin(this.cyclePos);

    this.position = (this.position + 1) % this.length;
    this.cyclePos = (this.cyclePos + this.cycleStep) % (Math.PI * 2);
  }

  public calculateValue(): number {
    let sumX = 0;
    let sumY = 0;
    for(let i = 0; i < this.length; i++){
      sumX += this.bufferX[i];
      sumY += this.bufferY[i];
    }
    sumX /= this.length;
    sumY /= this.length;
    const res = Math.sqrt(sumX * sumX + sumY * sumY);
    return res;
  }
}
```

Файл analyzer.ts

```
import { Note } from '../helpers/noteHelper';
import { AnalyzedData, DataUpdate } from "../../models/data/analyzedData";
import { Subject } from "../common/observable";
import { generateNotes, getNoteFrequency } from "../helpers/noteHelper";
import { AnalyzingPoint } from "../analyzingPoint";

export interface AnalyzerPointConfig {
  frequency: number;
}

export interface AnalyzerOptions {
  tickLength: number;
  notes: {
    start: Note,
    end: Note,
    tuning: number,
    density: number
  };
  cyclesCount: number;
}

export const defaultAnalyzerSettings : AnalyzerOptions = {
  tickLength: 256,
  cyclesCount: 64,
  notes: {
    start: 'C3',
    end: 'B7',
    tuning: 440,
    density: 5
  }
}

export class Analyzer extends Subject<DataUpdate> implements AnalyzedData{

  public ticksAnalyzed: number = 0;
  public ticks: number[][] = [];
  public tickRate: number;
  public length: number;
  public pointsNumber: number;

  private tickLength: number;
  private points: AnalyzingPoint[];

  private position: number = 0;

  constructor(public settings: AnalyzerOptions, private audioBuffer:
AudioBuffer){
    super();
    this.tickLength = settings.tickLength;
    this.length = audioBuffer.length / settings.tickLength;
    this.tickRate = audioBuffer.sampleRate / settings.tickLength;
  }
}
```



```

        const notes = generateNotes(settings.notes).reverse();

        this.pointsNumber = notes.length;

        this.points = notes.map(note => new
AnalyzingPoint(getNoteFrequency(note, settings.notes.tuning), audioBuffer.sampleRate,
settings.cyclesCount));
    }

    public analyzeNext(count: number): boolean {
        const data = this.audioBuffer.getChannelData(0);

        const startTick = this.position;
        const endTick = this.position + count;

        for(let t = this.position; t < endTick; t++){
            for(let i = 0; i < this.tickLength; i++){
                const val = data[i + this.tickLength * t] * 0.5 + 0.5;
                this.points.forEach(p => p.next(val));
            }
            this.ticks[t] = this.points.map(p => p.calculateValue());
        }

        this.position = endTick;

        this.emit({from: startTick, to: endTick});

        return this.position < this.length;
    }
}

```

Файл renderHelper.ts

```

import { AnalyzedData } from "../../models/data/analyzedData";
import { ColorFunction, colorFunctions } from "../common/colorFunction";

export function renderBlock(data: AnalyzedData, startTick: number, length:
number, colorFunction : ColorFunction = colorFunctions.gradientBGRW){
    let imageData : ImageData;
    let imageIndex = 0;

    imageData = new ImageData(length, data.pointsNumber);
    for(let p = 0; p < data.pointsNumber; p++){
        for(let t = 0; t < length; t++){
            let value = Math.pow((data.ticks[startTick + t]?.[p] ?? 0)*10, 0.7);
            if (value > 1){
                value = 1;
            }

            const color = colorFunction(value);
            imageData.data[imageIndex] = (color & 0xFF0000) >> 16;
            imageData.data[imageIndex + 1] = (color & 0xFF00) >> 8;
            imageData.data[imageIndex + 2] = color & 0xFF;
            imageData.data[imageIndex + 3] = 0xFF;
            imageIndex += 4;
        }
    }
}

```

```

    }
  }

  const canvas = document.createElement('canvas');
  canvas.width = imageData.width;
  canvas.height = imageData.height;
  const ctx = canvas.getContext('2d');
  if (!ctx){
    throw new Error('Can\'t get 2d context form canvas');
  }
  ctx.putImageData(imageData, 0, 0);
  return canvas.toDataURL();
}

```

Файл noteHelper.ts

```

export type Note = `${
  'A' | 'B' | 'C' | 'D' | 'E' | 'F' | 'G' | 'H' |
  'a' | 'b' | 'c' | 'd' | 'e' | 'f' | 'g' | 'h'}${
  '#' | '' | 'b'}${
  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10
}`;

export const maxFrequency = 8000;
export const minFrequency = 16;

export const minTuning = 440 * Math.pow(2, -0.5 / 12);
export const maxTuning = 440 * Math.pow(2, 0.5 / 12);

const noteIndexes: {[key: string]: number} = {
  'c': 0,
  'd': 2,
  'e': 4,
  'f': 5,
  'g': 7,
  'a': 9,
  'b': 11,
  'h': 11,
} as const;

const nameMap: {[key: number]: string} = {
  0: 'C',
  2: 'D',
  4: 'E',
  5: 'F',
  7: 'G',
  9: 'A',
  11: 'B'
} as const;

export function validateTuning(tuning: number){
  if (tuning > maxTuning || tuning < minTuning){
    throw new Error(`Tuning "${tuning}Hz" is not allowed. The value must be between
"${minTuning}Hz" and "${maxTuning}Hz"`);
  }
}

```

```

export function getHighestNote(tuning: number = 440){
  return Math.floor(Math.log2(maxFrequency / tuning) * 12 + 9);
}

export function getLowestNote(tuning: number = 440){
  return Math.ceil(Math.log2(minFrequency / tuning) * 12 + 9);
}

export function getNoteFrequency(input: Note | number, tuning: number = 440){
  const noteIndex = typeof input === 'string' ? parseNote(input) : input;

  return tuning * Math.pow(2, (noteIndex - 9) / 12);
}

export function parseNote(note: Note){
  const res = /^[A-G](#[b]?)\d$/i.exec(note);
  if (!res){
    throw new Error(`Unknown note ${note}`);
  }
  const value = noteIndexes[res[1].toLowerCase()] + (res[2] === '#' ? 1 : (res[2]
=== 'b' ? -1 : 0)) + (Number.parseInt(res[3]) - 4) * 12;

  return value;
}

export function stringifyNote(noteIndex: number) : Note{
  const relativeIndex = (Math.round(noteIndex) % 12 + 12) % 12;
  const baseName = nameMap[relativeIndex] ?? (nameMap[relativeIndex - 1] + '#');
  return `${baseName}${Math.floor(Math.round(noteIndex) / 12) + 4}` as Note;
}

export interface NoteInfo{
  index: number,
  name: Note,
  baseNoteName: string,
  octave: number,
  isBlack: boolean,
  tuning: number
}

export interface GenerateNotesRangeOptions{
  start?: number | Note;
  end?: number | Note;
  density?: number;
  tuning?: number;
}

export function generateNotes(options?: GenerateNotesRangeOptions){
  const from = options?.start ?? getLowestNote(options?.tuning ?? 440);
  const to = options?.end ?? getHighestNote(options?.tuning ?? 440);
  const density = options?.density ?? 1;

  const fromIndex = typeof(from) === 'string' ? parseNote(from) : from;
  const toIndex = typeof(to) === 'string' ? parseNote(to) : to;

  const result: number[] = [];
  for(let i = fromIndex; i <= toIndex; i++){
    for(let j = 0; j < density; j++){
      result.push(
        Math.round(

```

```

        (i - 0.5 + 0.5 / density + j / density)
        * 1000
    ) / 1000
    );
    }
}

return result;
}

export function generateNotesFull(options?: GenerateNotesRangeOptions){

    const tuning = options?.tuning ?? 440;
    const from = options?.start ?? getLowestNote(tuning);
    const to = options?.end ?? getHighestNote(tuning);
    const density = options?.density ?? 1;

    const fromIndex = typeof(from) === 'string' ? parseNote(from) : from;
    const toIndex = typeof(to) === 'string' ? parseNote(to) : to;

    const result: number[] = [];
    for(let i = fromIndex; i <= toIndex; i++){
        for(let j = 0; j < density; j++){
            result.push(
                Math.round(
                    (i - 0.5 + 0.5 / density + j / density)
                    * 1000
                ) / 1000
            );
        }
    }

    return result.map<NoteInfo>(note => {

        const relativeIndex = (Math.round(note) % 12 + 12) % 12;
        const baseName = nameMap[relativeIndex] ?? (nameMap[relativeIndex - 1] + '#');
        const octave = Math.floor(Math.round(note) / 12) + 4;

        return {
            index: note,
            name: `${baseName}${Math.floor(Math.round(note) / 12) + 4}` as Note,
            isBlack: !nameMap[relativeIndex],
            baseNoteName: baseName,
            octave,
            tuning
        }
    });
}

```

ІЛЮСТРАТИВНА ЧАСТИНА
ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ТОНІЧНОГО АНАЛІЗУ МУЗИЧНИХ
ТВОРІВ

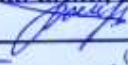
Виконав: студент 2-го курсу, групи
ІКН-21м спеціальності 122

«Комп'ютерні науки»

(шифр і назва напрямку підготовки, спеціальності)

 Гульчак Р.С.
(прізвище та ініціали)

Керівник: к.т.н., доцент каф. КН

 Барабан С.В.
(прізвище та ініціали)

« 15 » 12 2022р.

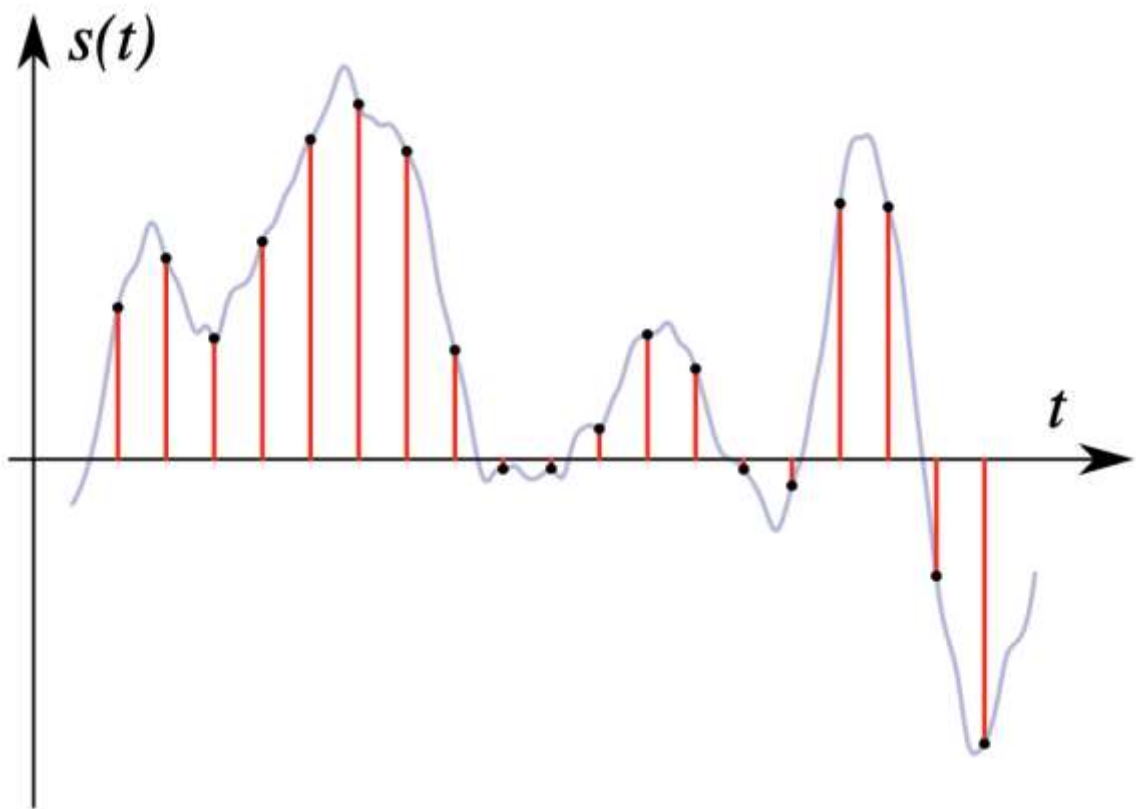


Рисунок В.1 – Графічне представлення ІКМ-сигналу

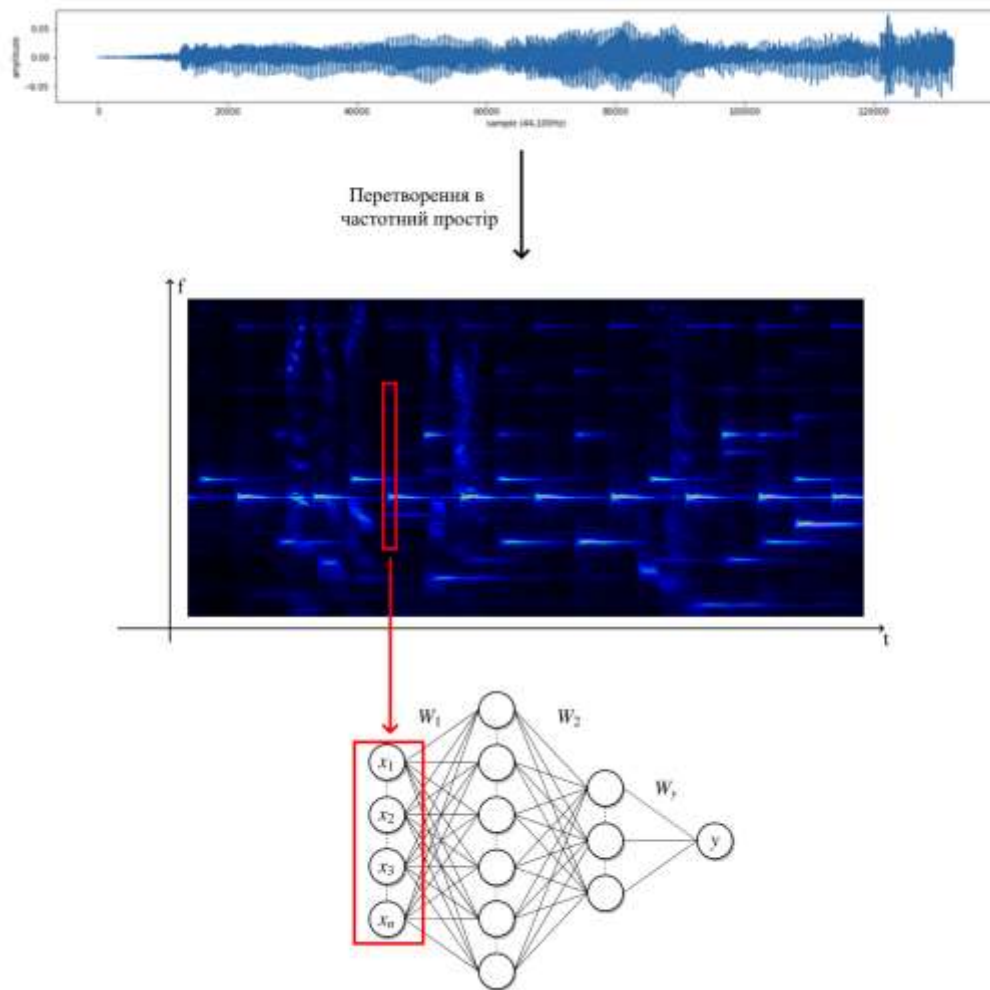


Рисунок В.2 – Схема перетворення дискретного аудіосигналу при представленні на вхід нейронної мережі

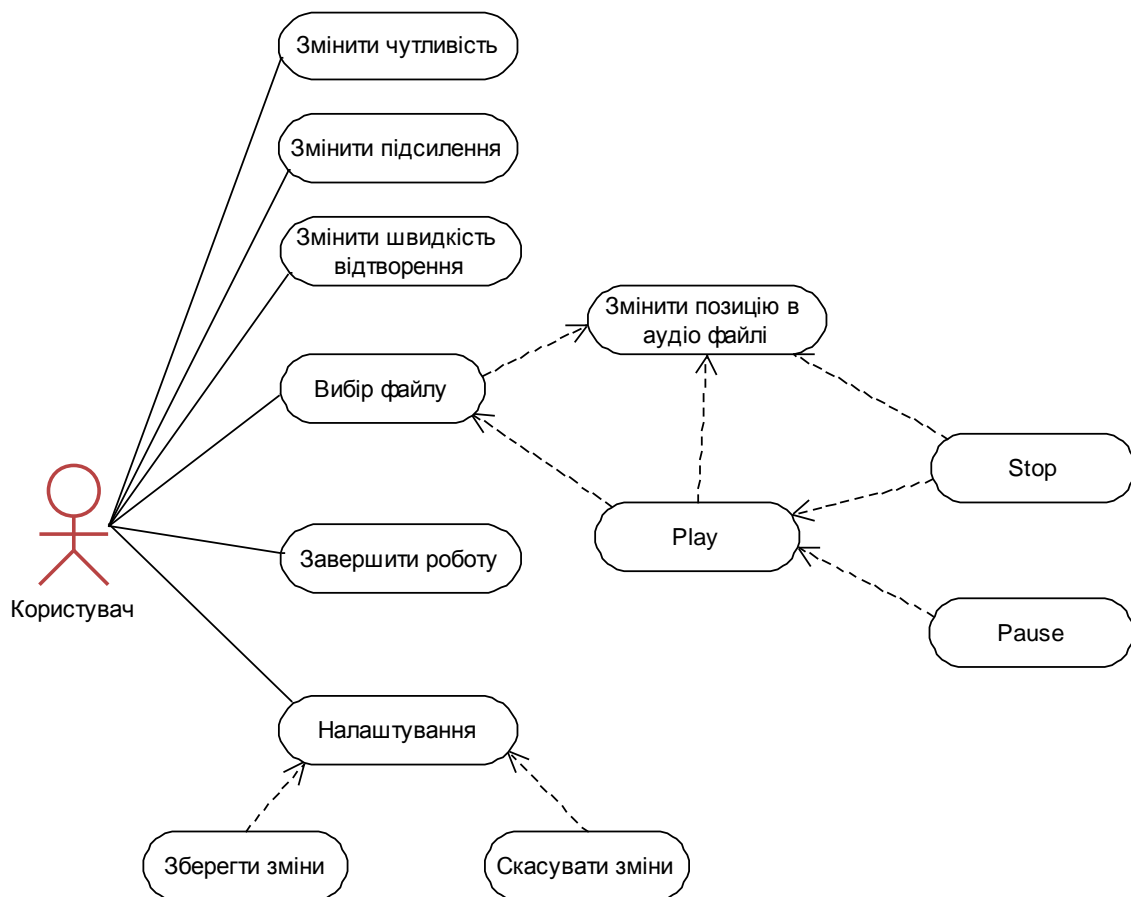


Рисунок В.3 – Діаграма прецедентів роботи додатку



Рисунок В.4 – Головне вікно додатку

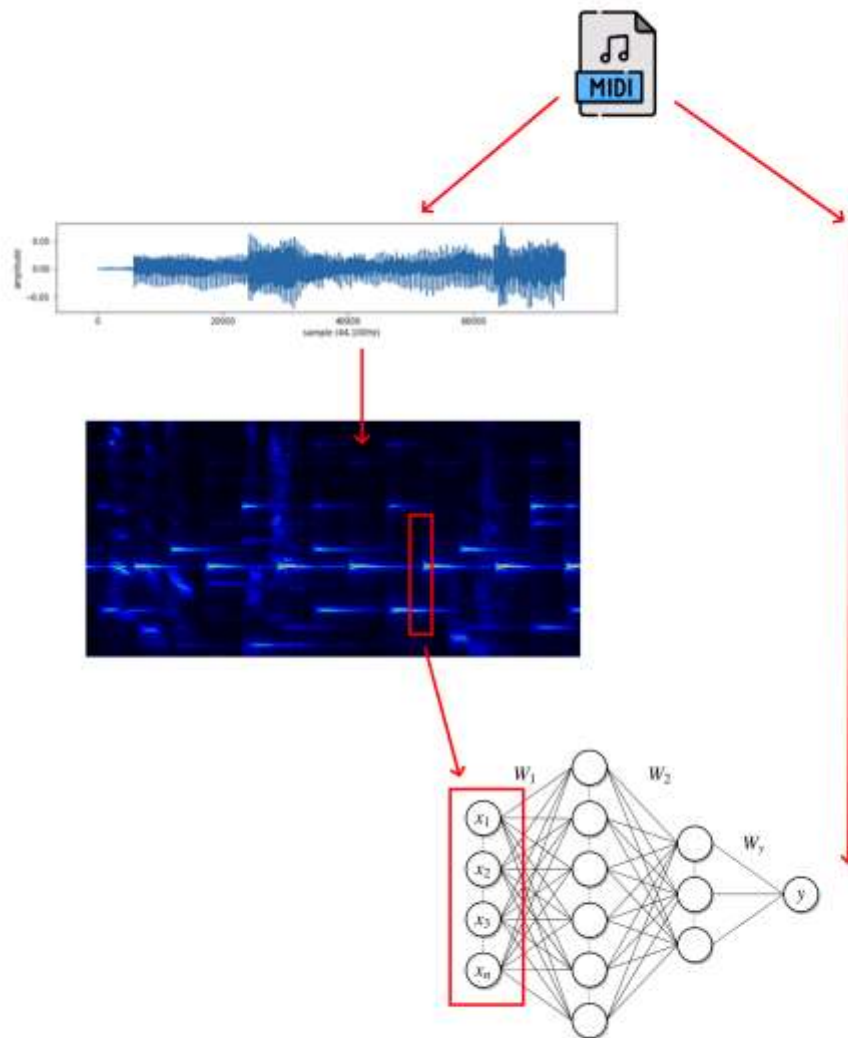


Рисунок В.5 – Графічне представлення способу генерування навчальної вибірки

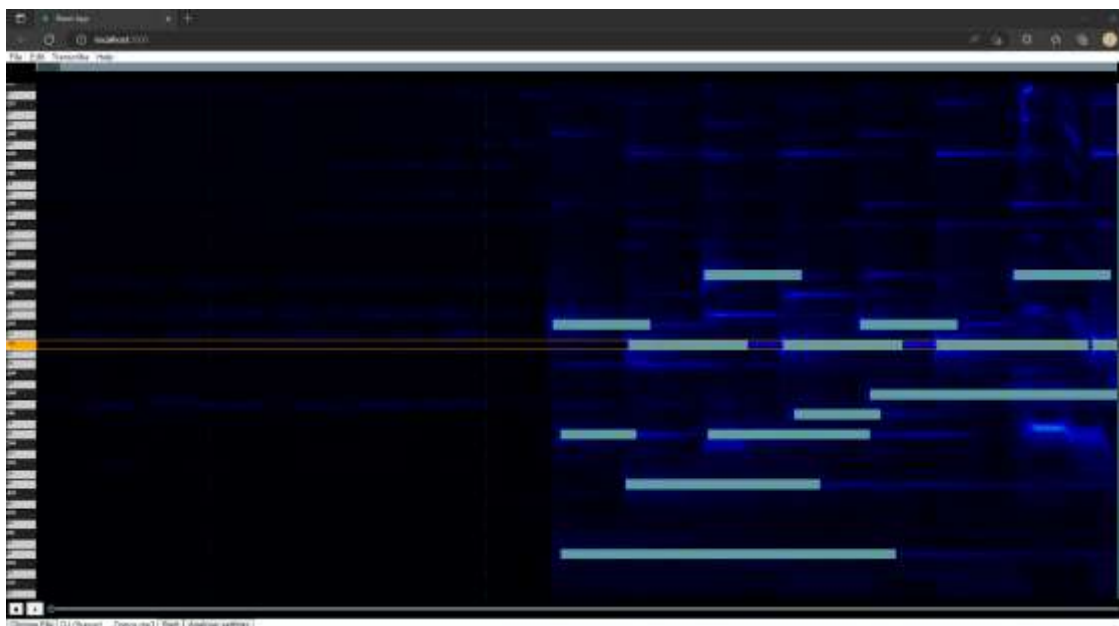


Рисунок В.6 – Результат роботи автоматичного транскрибування

ДОДАТОК Г (ДОВІДНИКОВИЙ) ІНСТРУКЦІЯ КОРИСТУВАЧА

Розроблений додаток представляє з веб-сайт. Для його запуску необхідно встановити node.js з офіційного сайту <https://nodejs.org/en/>. Після цього необхідно перезапустити систему. Наступним кроком потрібно інстальювати пакети, для цього необхідно відкрити термінал у коренній папці проекту (там де знаходиться файл packages.json) та ввести команду `npm install`. Встановлення пакетів займе декілька хвилин та після його завершення можна запускати проект. Для запуску необхідно ввести в терміналі `npm start` та зачекати доки буде запущено локальний сервер. Після запуску в будь-якому оновленому браузері необхідно перейти за посилання <http://localhost:4200> – якщо у вікні браузера відобразиться головне вікно додатку - все налаштовано вірно та проект запущено.

Для відкриття файлу необхідно натиснути кнопку Choose File, обрати аудіофайл серед файлів на комп'ютері та натиснути кнопку Open (рисунок Г.1).

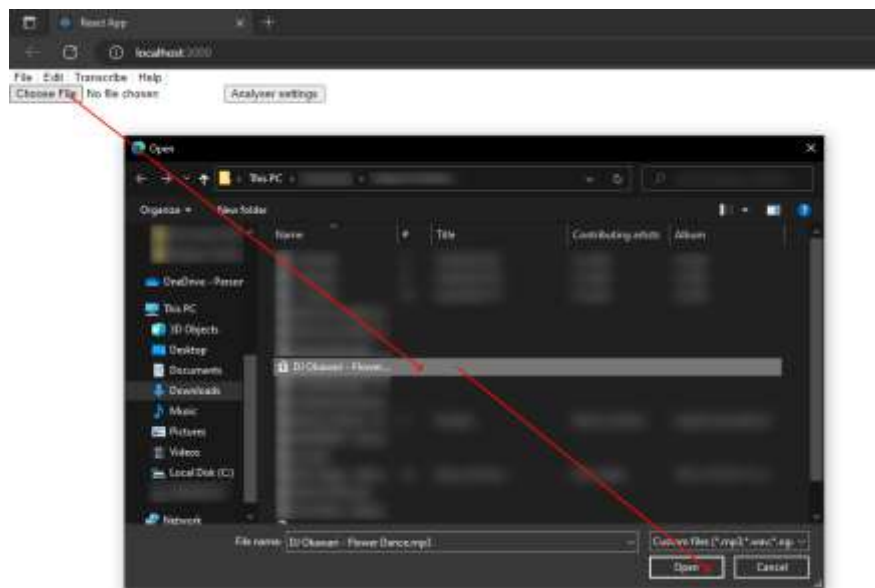


Рисунок Г.1 – Відкриття файлу в розробленому додатку

Edit – випадаюче меню з інструментами роботи з нотами:

— M

— D

— E
Eraser – інструмент для видалення нот;

— K
Wool – інструмент маніпулювання з існуючими нотами;

— T
Transcribe – відкриває вікно для запуску автоматичного транскрибування;

— I
Help – показує вікно з довідковою інформацією.

Також у нижній частині поруч з кнопкою керування генерацією знаходиться кнопка налаштування візуалізації. При натиску на неї відкривається вікно де можна задати діапазон та якість. Але слід зауважити, що функція автоматичного транскрибування активна лише коли задано налаштування візуалізації по замовчуванню.

ДОДАТОК Д
ДОВІДКА ПРО ВПРОВАДЖЕННЯ



МАЛЕ НАУКОВО-ВИРОБНИЧЕ ПІДПРИЄМСТВО "ТОВ "ІТІ"
Україно, 21021, м.Вінниця, вул. Келецька, 56

№ 375 від "15" 12 2022 р.

ДОВІДКА

Дана Гульчаку Роману Сергійовичу в тому, що результати, одержані ним в процесі виконання магістерської кваліфікаційної роботи, а саме алгоритми та програмні засоби тонічного аналізу музичних творів для автоматизованого їх транскрибування, планується використати в розробках ТОВ «ІТІ».

Зам. директора ТОВ «ІТІ»



Бодяк В.М.

ТОВ "ІТІ"