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 розпізнавання музичних патернів на основі спайкінгової нейронної мережі»

Виконав: студент 2-го курсу, групи 2КН-21м
спеціальності 122 «Комп'ютерні науки»
(шифр і назва напряму підготовки, спеціальності)
Чоботар Є.О.
(прізвище та ініціали)

Керівник: к.т.н., проф каф. КН
Месюра В.І.
(прізвище та ініціали)
«15» 12 2022 р.

Опонент: к.т.н., професор каф. АІІТ
Паламарчук Є.А.
(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 ІІ-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**

З А В Д А Н Н Я

НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Чоботарю Євгенію Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Інформаційна технологія розпізнавання музичних патернів на основі спайкінгової нейронної мережі

керівник роботи_ к.т.н., проф. кафедри КН Месюра В.І.
затверджені наказом вищого навчального закладу від "14" 09 2022 року № 203
2. Строк подання студентом роботи 18 листопада 2022 року
3. Вихідні дані до роботи:
вхідні дані – музичні патерни у форматі звукових файлів – .wav; навчальна вибірка - 2500 файлів, тестова вибірка - 1000 файлів; кількість патернів для розпізнавання 500, мова програмування – об'єктно-орієнтована.

4. Зміст текстової частини:
Вступ, аналіз предметної області розпізнавання музичних патернів, розробка інформаційної технології розпізнавання музичних патернів на основі спайкінгової нейронної мережі, програмна реалізація інформаційної технології нейромережевого розпізнавання музичних патернів, тестування та аналіз результатів роботи програми розпізнавання музичних патернів, економічна частина, висновки, список використаних джерел, додатки

5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень)
Алгоритм роботи програми розпізнавання музичних патернів, структура спайкінгової нейронної мережі, діаграма варіантів використання, UML діаграма класів програми розпізнавання музичних патернів, робочі вікна програми розпізнавання музичних патернів,

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	виконання прийняв
1-4	Месюра В.І., к.т.н., проф. каф. КН	14.09.22	14.09.22
5	Бурєннікова Н. В., д. е. н., проф. каф. ЕПВМ	19.09.22	19.09.22

7. Дата видачі завдання 14.09 2022 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз сучасного рівня інформаційних технологій розпізнавання музичних патернів. Постановка задач дослідження	14.09.22р. – 01.10.22р.	
2	Побудова моделей розпізнавання музичних патернів на основі нейронної мережі та функціонування нейронної мережі	02.10.22р. – 16.10.22р.	
3	Практичне застосування та оцінка ефективності розроблених моделей	17.10.22р. – 07.11.22р.	
4	Підготовка економічної частини	08.11.22р. – 11.11.22р.	
5	Апробація та/або впровадження результатів дослідження	13.11.22р. – 01.12.22р.	
6	Оформлення пояснювальної записки, графічного матеріалу та презентації	02.12.22р. – 14.12.22р.	

Студент (підпис) Чоботар Є.О.

Керівник роботи (підпис) Месюра В.І.

АНОТАЦІЯ

УДК 004.8

Чоботар Є. О. Інформаційна технологія розпізнавання музичних патернів на основі спайкінгової нейронної мережі. Магістерська кваліфікаційна робота зі спеціальності 122 – комп'ютерні науки, освітня програма - комп'ютерні науки. Вінниця: ВНТУ, 2022. 113 с.

На укр. мові. Бібліогр.: 24 назв; рис.: 21; табл. 15.

Дана магістерська кваліфікаційна робота присвячена розробці програмного забезпечення для розпізнавання музичних патернів на основі спайкінгової нейронної мережі. Були розглянуті та проаналізовані існуючі методи розпізнавання музичних патернів, як найбільш перспективний, було обрано нейромережевий метод. Було проаналізовано різні парадигми штучних нейронних мереж та обґрунтовано вибір для даної задачі спайкінгової нейронної мережі. Було розроблено структуру обраного типу мережі. Було спроектовано програму розпізнавання музичних патернів, написану мовою програмування Python у програмному середовищі PyCharm. Достовірність розпізнавання музичних патернів розробленої програми на 9,1% краще аналогу.

Графічна частина складається з 6 плакатів.

У економічному розділі визначено рівень комерційного потенціалу розробки, який становить 44 бала, що свідчить про комерційну важливість проведення даних досліджень. Термін окупності становить 0,99 р., що менше 3-х років і свідчить про комерційну привабливість науково-технічної розробки та може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Ключові слова: розпізнавання, музичний патерн, спайкінгова нейронна мережа.

ABSTRACT

Chobotar Ye. O. Information technology of recognition of musical patterns on the basis of spiking neural network. Master's thesis in the specialty 122 - computer science, educational program - computer science. Vinnytsia: VNTU, 2022. 113 p.

In Ukrainian language. Bibliogr .: 24 titles; fig.: 21; table 15.

This master's thesis is devoted to the development of software for recognizing music patterns based on the spiking neural network. The existing methods of music patterns were considered and analyzed, as the most promising, the neural network method was chosen. Different paradigms of artificial neural networks were analyzed and the choice of a spiking neural network for this task was substantiated. The architecture of the selected network type was developed. A music pattern recognition program written in the Python programming language in the PyCharm software environment has been designed. The reliability of recognition of music patterns of the developed program is 9,1% better than the analogue.

The graphic part consists of 5 posters.

In the economic section, the level of the commercial development potential is determined, which is 44 points, which indicates the commercial importance of conducting these studies. The payback period is 0.99 years, which is less than 3 years, which indicates the commercial attractiveness of the scientific and technical development and may encourage a potential investor to finance the implementation of this development and its introduction to the market.

Keywords: recognition, music pattern, spiking neural network.

ЗМІСТ

ВСТУП

Актуальність. При реалізації штучного інтелекту у ряді методів використовуються явні подання знань і ретельно спроектовані алгоритми перебору. Відмінний від цього підхід полягає в побудові інтелектуальних програм з використанням моделей, що імітують нейронні структури в людському мозку чи еволюцію різних альтернативних конфігурацій, як це робиться в генетичних алгоритмах і штучному житті.

Для того щоб вирішувати складні і погано формалізовані завдання і виник напрямок, який називається штучні нейронні мережі. Штучні нейронні мережі складаються з нейроноподобних елементів, з’єднаних між собою в мережу. Нейронні мережі знайшли застосування практично в усіх сферах обробки інформації.

Часто в програмах розпізнавання музичних патернів використовуються нейронні мережі на формальних нейронах, які пристосовані для розпізнавання не динамічних сигналів, а статичних векторних даних. Тому для їх використання доводиться перетворювати динамічні музичні сигнали у вектори чисел (застосовуючи виділення ознак з сигналів або їх розкладання в будь-який функціональний ряд), а ці вектори потім розпізнавати за допомогою нейронних мереж на формальних нейронах. Тому такі нейромережеві методи і засоби характеризуються низькою швидкодією і втратою корисної інформації через її перетворення в іншу форму.

Розпізнавання звуків необхідне для багатьох житєвих ситуацій, пошук людей за голосом, знаходження почутої музики, робота з голосовими асистентами, використання у військових цілях, тощо.

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.

Мета і завдання досліджень. Метою магістерської кваліфікаційної роботи є підвищення достовірності розпізнавання музичних патернів програмними засобами за рахунок використання спайкінгових нейронних мереж.

Для досягнення мети розробки необхідно виконати такі задачі:

- провести аналіз предметної області розпізнавання музичних патернів;
- розглянути існуючі методи розпізнавання музичних патернів та обрати й обґрунтувати вибір методу, який задовольняє мету даної магістерської кваліфікаційної роботи;
- розробити метод розпізнавання музичних патернів та структуру інформаційної технології розпізнавання музичних патернів на основі спайкінгової нейронної мережі;
- , розробити алгоритм роботи програмного засобу;
- виконати програмну реалізацію запропонованої інформаційної технології;
- провести тестування програмного продукту та виконати аналіз отриманих результатів.

Об’єкт дослідження – процес розпізнавання музичних патернів комп’ютерними засобами з використанням нейронних мереж.

Предмет дослідження – інформаційна технологія та програмні засоби розпізнавання музичних патернів комп’ютерними засобами з використанням нейронних мереж та достовірність їх роботи.

Методи дослідження. У роботі використані наступні методи наукових досліджень: системного аналізу, теорії штучних нейронних мереж для реалізації інформаційної технології, методи математичної статистики для

розробки процесу розв'язання задачі нейромережевого розпізнавання музичних патернів та обрахунків результатів експериментів із програмним засобом, об’єктно-орієнтованого програмування.

Наукова новизна одержаних результатів.

1. Набула подальшого розвитку інформаційна технологія нейромережевого розпізнавання музичних патернів, яка відрізняється використанням спайкінгової нейронної мережі, що дозволило підвищити достовірність програмних засобів розпізнавання музичних патернів.

Практичне значення одержаних результатів полягає в тому, що на основі проведених досліджень розроблено програмне забезпечення розпізнавання музичних патернів.

Запропонована інформаційна технологія сприяє підвищенню достовірності програмних засобів розпізнавання музичних патернів, зокрема:

- розроблено алгоритм роботи програмного забезпечення розпізнавання музичних патернів;
- розроблено програмні засоби для розпізнавання музичних патернів.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується коректністю постановки завдання, коректністю використання математичного апарату методів дослідження, експериментальними дослідженнями тестування програмної реалізації інформаційної технології розпізнавання музичних патернів. Адекватність розроблених математичних моделей підтверджується результатами експериментальних досліджень.

Особистий внесок здобувача. Усі результати, наведені у магістерській кваліфікаційній роботі, отримані самостійно. У працях, написаних у співавторстві, здобувачу належать: аналіз процесу розпізнавання музичних патернів та методів підвищення достовірності програмних засобів розпізнавання музичних патернів [1].

Апробація результатів роботи. Результати роботи були апробовані на конференції «L Науково-технічна конференція підрозділів Вінницького національного технічного університету (2021)», Вінниця, 2021.

Публікації. За результатами досліджень опубліковано одні тези доповіді на науково-технічній конференції[1].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ РОЗПІЗНАВАННЯ МУЗИЧНИХ ПАТЕРНІВ

1.1 Постановка задачі розпізнавання музичних патернів

Впродовж останніх років відбувається стрімкий розвиток технологій машинного навчання, які знайшли своє застосування в сферах обробки зображень та відео. Основні напрямки створення комп'ютерів з використанням штучного інтелекту пов'язані з вивченням візуальних даних. Сфера розпізнавання зображень та іншої не візуальної інформації дуже важлива. Можливість обробляти велику кількість даних у певному контексті принесе велику цінність. Одним із прикладів є автоматизований процес транскрипції тексту, який передбачає застосування аналізу старовинних документів, з огляду на складність та нерегулярність записів внаслідок ручних аспектів написання. Було б чудово отримати доступ до онлайн ресурсів старих документів у веб-бібліотеках.

Загальновідомо, що робота з розпізнавання музичних патернів переживає повільний процес розвитку порівняно з іншими напрямами. З іншого боку, майже весь розвиток даного сектору іде на розпізнавання візуальних даних. Дуже корисно застосовувати новітні технології в області розпізнавання обличь для визнання людей за допомогою камери. Глибинне навчання – це нове застосування машинного навчання для вивчення представлення даних. Наразі ця сфера переживає розквіт.

Дослідження щодо розпізнавання музичних патернів методами машинного навчання проводилися ще з середини 90-х років. Було створено різноманітні класифікатори [2, 3], в тому числі на базі нейронних мереж, методу опорних векторів, методу K-найближчих сусідів [4], карт самоорганізації та нейронних мереж RBF [5]. В даних роботах використовувалися різні набори музичних патернів, їх характеристик та

бібліотек музичних записів для навчальної вибірки. Деякі з робіт представляють аналіз великої кількості музичних патернів, проте класифікація виконується лише за музичними інструментами, якими створено ці патерни. В інших роботах представлена класифікація в межах одного сімейства музичних інструментів. Коректна класифікація музичних патернів залежить від наступних виокремлених процесів:

— вибору музичних інструментів для розпізнавання;

— вибору характеристик звуку;

— математичного представлення характеристик в наборі даних;

— зменшення розмірності набору даних;

— використання методів класифікації.

На початку експерименти проводились для малої кількості інструментів та характеристик. З часом удосконалювались методи та підходи до вирішення задачі і кількість інструментів у експериментах зростала.

З розвитком технологій, вдосконалювались і методи машинного навчання, і в останні роки ця галузь стала однією з передових у сфері розпізнавання музичних патернів. В останні роки запит на використання методів машинного навчання постійно зростає [6] і регулярно з’являються нові удосконалені інструменти для роботи з цими методами.

Методи вирішення задачі класифікації музичних патернів поділяють на два типи – класичні та сучасні. В середині 90-х використовувались класичні методи, проте останнім часом з розвитком техніки почали застосовуватись сучасні методи класифікації з різними комбінаціями, у тому числі й поєднанням сучасних та класичних методів.

Основною задачею дослідження є розпізнати музичний патерн який знаходиться в форматі .wav, в свою чергу розпізнати означає віднести цей патерн до одного із 500 класів, цими класами може бути як і сам інструмент який створює звук, так і крик людини чи тварини що використовувався в композиції, або музичний жанр. Повний перелік назв цих 500 класів музичних

патернів наведено у лістінгу прогами у додатку Б. Деякі приклади з них такі: чоловічий спів, жіночий спів, гітара, мандоліна, музика сальса, джаз, рок, піаніно, орган, саксофон, фламенко, щаслива музика (happy music), смішна музика (funny music), сумна музика (sad music), дощ, водоспад та інші.

1.2 Огляд відомих методів розпізнавання музичних патернів та обгрунтування вибору нейромережевого методу

Створення штучного інтелекту неможливе без аналізу звукових сигналів. Існує багато класичних методів розв’язання задач, пов’язаних із дослідженням аудіо, однак зі зростанням кількості інформації ці методи погіршують свої результати. В той же час сучасні методи, що базуються на використанні нейронних мереж, на великий обсягах даних покращують свої показники, оскільки етапами побудови моделі є підбір тренувальної вибірки та процес навчання.

Основною перешкодою переходу до використання машинного навчання для обробки аудіосигналів слугує складність цього процесу, для спеціалістів необхідно вивчати дві окремі галузі – обробки звуків та створення моделей штучного інтелекту. Отже, актуальною є задача переходу від класичних до сучасних методів розв’язання звукових задач. Для зменшення впливу даної проблеми пропонується створити загальне рішення для обробки аудіосигналів нейромережевими методами, яке допоможе спеціалістам швидше адаптуватись до нових технологій. Автоматичне розпізнавання джерела звуку відіграє важливу роль у розробці автоматичної індексації і додатків для пошуку баз даних. Ці програми мають потенціал у допомозі людям, здійснюючи пошук через величезну кількість цифрових аудіо-матеріалів, доступних сьогодні. Наприклад, було б дуже корисно мати змогу знайти звукові зразки, які «звучать схоже», як подано аудіо приклад. Аналіз музичного вмісту в цілому має багато

практичних застосувань, включаючи, наприклад, структуроване кодування, автоматичне коментування музичного сигналу та інструментів музикантів.

Автоматичне розпізнавання музичних інструментів є найважливішим завданням у вирішенні цих непростих проблем, а також може надавати корисну інформацію в інших областях розпізнавання джерел звуку. Однак аналізу музичного сигналу так і не вдалося досягти великого комерційного інтересу, як, наприклад, розпізнавання мовців та мови. Це тому, що теми навколо мовленнєвої обробки є більш комерційно застосовними, хоча обидві області вважаються дуже складними. Через побудову комп'ютерних систем, які «Слухають», ми також можемо отримати нові уявлення про людське сприйняття. Ця теза описує побудову та оцінку системи розпізнавання музичних патернів, яка здатна розпізнати поодинокі тони, які грають на інструментах.

Також відомі ще деякі методи як аналоговий і векторний. Суть аналогового методу є те, що на аналогові фільтри подається звуковий сигнал, який проходить через різні перетворювачі і на виході ми отримуємо сигнал, який порівнюється з відомини сигналами в базі даних. В свою чергу, векторний метод розбиває звуковий сигнал на вектори і далі ці вектори зрівнюються за різними методами, наприклад за методом К – найближчих сусідів. У таблиці 1.1 зображенно порівняння методів розпізнавання музичних патернів.

Таблиця 1.1 – Порівняння методів розпізнавання музичних патернів

	Точність розпізнавання	Складність реалізації	Швидкість роботи	Ресурсовитратність
Нейронні мережі	Висока	Середня	Висока	Висока
Аналоговий метод	Низька	Низька	Середня	Низька
Векторний метод	Висока	Висока	Низька	Висока

Згідно з вище вказаною таблицею ми можемо побичити, що метод розпізнавання музичних патернів за допомогою нейронних мереж є точним і швидким, не складним в реалізації, хоча і ресурсовитратним, в той час як інші методи не дають таких результатів.

1.3 Обгрунтування вибору аналогу до програми розпізнавання музичних патернів

В даній роботі представлено вирішення задачі розпізнавання музичних патернів [1]. Однак, результати за різних конфігурацій склали менше 62%. Для прикладу ми можемо взяти програму аналог Shazam [7] і схожу на неї програму Tunatic [8]. На рисунку 1.1 зображено приклад програми Shazam, а на рисунку 1.2 - приклад програми Tunatic.

Рисунок 1.1 – Приклад програми Shazam

Оскільки процес виділення джерела звучання з аудіо-запису є окремою задачею, для спрощення пропонується вирішити задачу створення програми для розпізнавання музичних патернів на основі спайкінгової нейронної мережі.

Рисунок 1.2 – Приклад програми Tunatic

До обробки аудіосигналів відносять задачі класифікації, музичного маркування, сегментації, виділення джерела звучання, відстеження ритму, підбір рекомендацій, виокремлення сигналів та транскрипції звукової доріжки. Для розв’язання подібних завдань нейромережевими методами необхідно вирішити декілька підзадач, а саме:

— вибір бібліотеки зразків для тренувальної вибірки;

— підбір найважливіших звукових характеристик;

— виокремлення обраних характеристик зі звукових сигналів;

— створення архітектури на тренування моделі нейронної мережі;

— перевірка отриманої моделі на тестовій вибірці.

Недоліком відомих програм є низька достовірність розпізнавання музичних патернів, звідки і випливає мета даної роботи – підвищення достовірності програмних засобів розпізнавання музичних патернів..

1.4 Висновок до розділу 1

У даному розділі було проведено аналіз предметної області розпізнавання музичних патернів, а саме – сформульовано постановку задачі, проведено огляд відомих методів розв'язання задачі розпізнавання музичних патернів. Найперспективнішим було обрано нейромережевий метод. Також було проведено аналіз програмних засобів розпізнавання музичних патернів та обгрунтовано вибір аналогу. Недоліком відомих програм є низька достовірність розпізнавання музичних патернів, звідки і випливає мета даної роботи – підвищення достовірності програмних засобів розпізнавання музичних патернів.

2 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ РОЗПІЗНАВАННЯ МУЗИЧНИХ ПАТЕРНІВ НА ОСНОВІ СПАЙКІНГОВОЇ НЕЙРОННОЇ МЕРЕЖІ

2.1 Обгрунтування вибору спайкінгової нейронної мережі для розпізнавання музичних патернів

На сьогодняшній день існує багато парадигм нейронних мереж, які можуть використовуватись для задачі розпізнавання музичних патернів. Серед них можна виділити такі нейронні мережі як: одношаровий персептрон, багатошаровий персептрон, мережа Хемінга, радільно-базисна нейронна мережа, спайкінгова мережа та ін. Розглянемо деякі з них, щоб обрати найбільш ефективну для задачі розпізнавання музичних патернів.

2.1.1 Багатошаровий перцептрон

Багатошарові мережі можна отримати каскадним з'єднанням одношарових мереж, де вихід одного шару є входом наступного шару, причому така мережа може привести до збільшення обчислювальної потужності лише в тому випадку, якщо активаційна функція між шарами буде нелінійною.

Багатошарові нейромережі здатні виконувати загальну класифікацію, відокремлюючи ті крапки, які утримуються в опуклих обмежених або необмежених областях.

Багатошаровий перцептрон – багатошарова нейромережа прямого поширення. Зазвичай складається з множини сенсорних нейронів, які утворюють вхідний шар, одного чи декількох прихованих шарів нейронів і одного вихідного шару нейронів. Вхідний сигнал поширюється мережею в прямому напрямк, від шару до шару [9].

Для навчання багатошарового перцептрону використовують алгоритм зворотного поширення похибки, який передбачає два проходи по всіх шарах

мережі: прямий і зворотний. Під час прямого проходу образ (вхідний вектор) подається на сенсорні вузли нейромережі, після чого поширюється від шару до шару. В результаті генерується набір вихідних сигналів, який є реакцією мережі на цей вхідний образ. Під час прямого проходу всі синаптичні ваги мережі фіксовані. Під час зворотного проходу всі синаптичні ваги налаштовуються відповідно до правила корекції помилок, а саме: фактичний вихід мережі віднімається від цільового виходу, в результаті чого формується сигнал помилки. Цей сигнал згодом поширюється мережею в напрямку, зворотному до напрямку синаптичних зв’язків. Синаптичні ваги налаштовуються з метою максимального наближення вихідного сигналу мережі до бажаного [9].

Багатошарові перцептрони мають три характерні ознаки:

1. Кожен нейрон мережі має нелінійну функцію активації. Важливо підкреслити, що така нелінійна функція є гладкою (тобто всюди диференційованою), на відміну від жорсткої порогової функції, використовуваної в перцептроні Розенблатта. Найпоширенішою формою функції, що задовольняє ці вимоги, є сигмоїдальна

$$y_i = \frac{1}{1+\exp(-v_j)},$$

де v_j – індуковане локальне поле (зважена сума всіх синаптичних входів плюс порогове значення) нейрона j;

y_j – вихід нейрона.

2. Мережа містить один чи декілька шарів прихованих нейронів. Ці нейрони, послідовно витягаючи найважливіші ознаки вхідного образу, дозволяють мережі навчатись розв'язувати складні задачі.

3. Мережа має високий ступень зв’язності, що реалізується завдяки синаптичним зв’язкам. Зміна рівня зв’язності вимагає зміни множини синаптичних зв’язків або їхніх вагових коефіцієнтів.

Комбінація цих властивостей поєднано із здатністю до навчання на власному досвіді забезпечує високу потужність багатошарового перцептрону. Однак, ті самі властивості є причиною неповноти знань про мережі такого типу. Це пояснюється тим, що, по-перше, розподілена форма нелінійності та висока зв’язність мережі значно ускладнюють теоретичний аналіз багатошарового перцептрону. По-друге, наявність прихованих нейронів робить процес навчання складнішим для візуалізації. Саме в процесі навчання необхідно визначити, які ознаки вхідного сигналу необхідно подавати прихованими нейронами. Отже, процес навчання ускладнюється, оскільки пошук необхідно виконувати у дуже широкій області можливих функцій, а вибір – серед альтернативних представлень вхідних образів [10].

Якщо розглянути просту двошарову мережу із двома нейронами в першому шарі, з'єднаними з єдиним нейроном у другому шарі, то кожен нейрон першого шару розбиває площину на дві напівплощини, утворюючи в просторі образів область V-форми, а нейрон другого шару реалізує різні функції при підходящому виборі ваг і порога. Аналогічно в другому шарі може бути використано три нейрони з подальшою розбивкою площини й створенням області трикутної форми. Включенням достатнього числа нейронів у вхідний шар може бути утворений опуклий багатокутник будь-якої бажаної форми. Крапки, не складові опуклої області, не можуть бути відділені про інших крапок площини двошаровою мережею.

На рис. 2.1. зображено структуру багатошарового перцептрона з двома прихованими шарами та одним вихідним шаром. Ця нейромережа є повнозв’язною – кожен нейрон у будь-якому шарі мережі зв’язаний з усіма нейронами попереднього шару. Сигнал передається нейромережею виключно в прямому напрямку від шару до шару.

У мережах цього типу існують два типи сигналів:

Функціональний сигнал – вхідний сигнал, що надходить у мережу та проходить вперед від одного нейрона до іншого всією мережею. Такий сигнал досягає кінця мережі у вигляді вихідного сигналу.

Сигнал похибки – бере початок на виході мережі і поширюється в зворотному напрямку від шару до шару. Отримав свою назву внаслідок того, що він обчислюється кожним нейроном мережі на основі функції похибки.

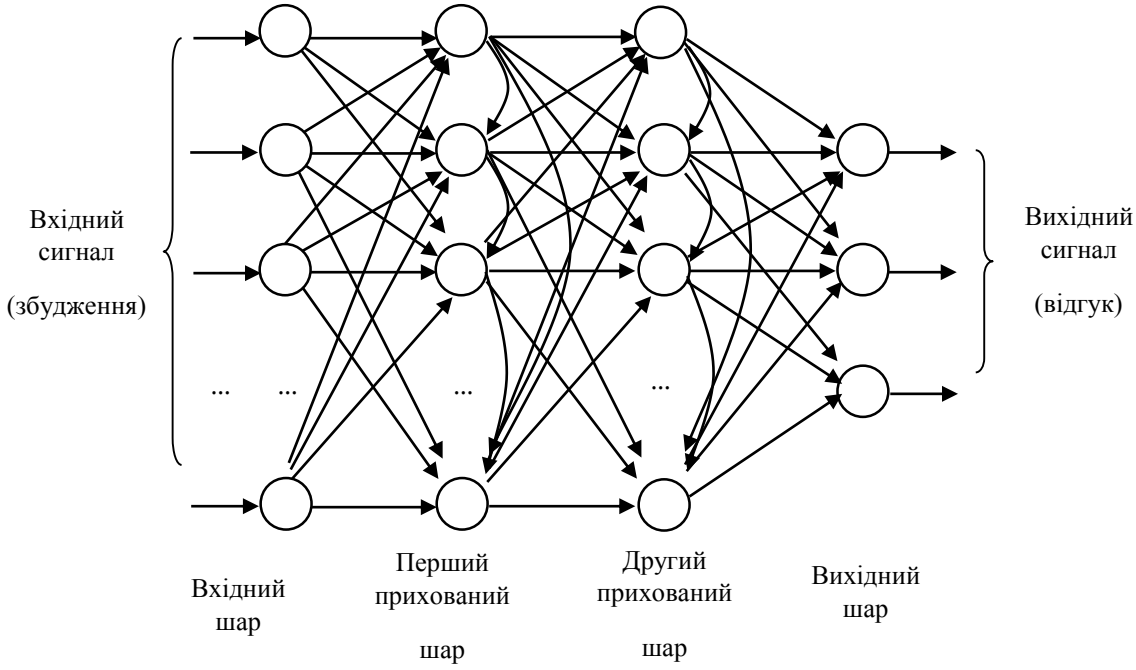


Рисунок 2.1 - Структура багатошарового перцептрона
з двома прихованими шарами.

Тришарова нейромережа є більш загальною. Її можливості, що класифікують, обмежені лише числом штучних нейронів і ваг. Обмеження на опуклість відсутні. Тепер нейрон третього шару приймає як вхід набір опуклих багатокутників, і їхня логічна комбінація може бути не опуклої форми. При додаванні нейронів і ваг число сторін багатокутника може необмежено зростати. Це дозволяє апроксимувати область будь-якої форми з будь-якою

точністю. У добавок не всі вихідні області другого шару повинні перетинатися. Отже, можливо поєднувати різні області, опуклі та неопуклі, видаючи на виході одиницю щораз, коли вхідний вектор належить однієї з них.

Будь-який прихований чи вихідний нейрон багатошарового перцептрона може виконувати обчислення двох типів:

1. Обчислення функціонального сигналу на виході нейрона, що реалізується у вигляді неперервної нелінійної функції від вхідного сигналу і синаптичних ваг, пов’язаних з цим вектором.

2. Обчислення оцінки вектора градієнта (тобто поверхні похибки за синаптичними вагами, пов’язаними зі входами цього нейрона), необхідного для зворотного проходу мережею.

З декількох причин дуже популярним є навчання багатошарового перцептрона за алгоритмом зворотного поширення похибки, а саме:

– локальність методу змін синаптичних ваг і порогів у багатошаровому перцептроні.

– ефективність методу обчислення всіх часткових похідних функцій вартості за вільними параметрами [11].

Однак, метод зворотного поширення похибки є ітераційним і вимагає щонайменше декількох проходів через мережу всієї навчальної вибірки, відповідно, застосування його для розв’язання задач інтелектуального аналізу даних є неможливим, оскільки через великий обсяг даних та їх багатовимірність процес триватиме занадто довго.

2.1.2 Нейронні мережі на основі радіальних базисних функцій

Побудову нейронної мережі, згідно з [11], можна звести до задачі апроксимації поверхні відгуку за деякими базовими точками в просторі великої вимірності. Тобто навчання еквівалентне побудові такої поверхні в багатовимірному просторі, яка б найточніше відповідала даним навчання, де критерій ”найкращої відповідності” обирають у деякому статистичному сенсі.

Базова архітектура RBF-мережі передбачає наявність трьох шарів нейронів, що виконують різні функції (рис. 2.2).

Вхідний шар складається із сенсорних елементів, які пов’язують мережу із зовнішнім середовищем. Другий шар є єдиним прихованим шаром, який переважно має значно більшу розмірність, ніж вхідний.

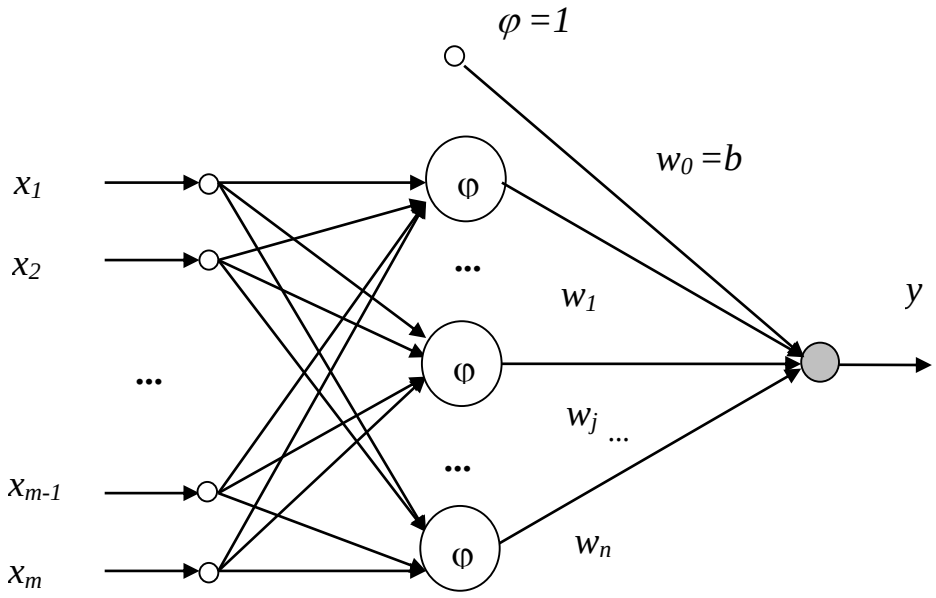
Кожен елемент прихованого шару використовує активаційну функцію Гауса (2.1).

$$h(x) = \exp(-\frac{\|x - c\|^2}{r^2}) \qquad (2.1)$$

де x – вхідний вектор,

c – центр функції Гаусса,

r – радіус функції Гаусса.



Вхідний шар | Прихований шар, що складається з n радіальних базисних функцій | Вихідний шар

Рисунок 2.2 - Мережа на основі радіальних базисних функцій

У центрі радіальної базисної функції (функції ядра) точка, визначена ваговим вектором, що пов’язаний із нейроном. Позиція та ширина функції ядра мають бути навчені на тренувальній вибірці прикладів. Як правило, кількість ядер є набагато меншою, ніж кількість навчальних прикладів.

Кожен вихідний елемент обчислює лінійну комбінацію цих радіальних базисних функцій за формулою (2.2).

$$f(x) = \sum_{j=1}^{m} w_j h_j(x) \tag{2.2}$$

З погляду апроксимації приховані елементи формують сукупність функцій, які створюють базисну систему для представлення вхідних прикладів у побудованому на ній просторі.

Така архітектура ґрунтується на висновках з теореми Ковера про роздільність образів, згідно з якими нелінійне перетворення складної задачі класифікації образів на простір більшої вимірності підвищує ймовірність лінійної роздільність образів. Також розмірність прихованого шару безпосередньо пов’язана із можливістю мережі апроксимувати гладке відображення ”вхід–вихід”. Чим вища розмірність прихованого шару, тим вища точність апроксимації [11].

Існують різні алгоритми навчання RBF-мереж. Основний алгоритм використовує двокрокову стратегію навчання або змішане навчання. Він оцінює позицію та ширину ядра з використанням алгоритму кластеризації „без вчителя”, а потім алгоритм мінімізації середньоквадратичної похибки „з вчителем” для визначення ваг між прихованим і вихідним шарами. Оскільки вихідні елементи є лінійними, застосовується неітераційний алгоритм. Після отримання цього початкового наближення використовується градієнтний спуск для уточнення параметрів мережі [10,11].

Такий змішаний алгоритм навчання RBF-мережі збігається набагато швидше, ніж алгоритм зворотного поширення похибки для навчання багатошарових перцептронів, однак RBF-мережа часто містить занадто велику кількість прихованих елементів, а отже, функціонування RBF-мережі в такому випадку є повільнішим за функціонування багатошарового перцептрона. Тому ефективність (похибка залежно від розміру мережі) RBF-мереж та багатошарового перцептрона залежить від розв'язуваної задачі [11].

Якщо порівнювати між собою багатошаровий перцептрон та мережу на основі радіальних базисних функцій, то можна зазначити, що багатошаровий перцептрон забезпечує глобальну апроксимацію нелінійного відображення, а RBF-мережа за допомогою локалізованих нелінійностей, що експоненційно зменшуються (функції Гауса) створює локальну апроксимацію нелінійного відображення.

2.1.3 Спайкінгові нейронні мережі

Спайкінгова нейромережа [12,13] приймає на входи серію імпульсів і видає імпульси на виході. У кожну мить кожен нейрон має деяке значення (аналог електричного потенціалу у біологічних нейронів) і, якщо це значення перевищує порогове, то нейрон посилає одиночний імпульс, після чого його власне значення падає до рівня нижче середнього значення (аналог процесу реабілітації у біологічних нейронів, так званий рефрактерний період) на 2-30 мс. При виведенні зі стану рівноваги потенціал нейрона починає плавно прагнути до середнього значення. Існує всього два параметри вагових зв'язків імпульсного нейрона - час затримки і величина ваги.

Також мережі Хемінга і радільно-базисна нейронна мережа призначенні для розпізнавання статичних даних, а на вхід необхідно подавати дані в форматі векторів або чисел, що потребує додаткової апаратної потужності для перетворення динамічного звукового сигналу в сигнал, який зможе оброблятися статичними нейромережами. В свою чергу, спайкінгова нейронна мережа може

обробляти динамічні дані, які подаються в форматі функціїї від часу, що нейтралізує необхідність перетворювати сигнал у вектор ознак і зменшує необхідну кількість обчислень для отримання результату.

Переваги спайкінгових нейронних мереж. Спайкінгові нейронні мережі мають ряд переваг над нейронними мережами попередніх поколінь [6]:

1) Спайкінгові нейронні мережі є динамічними, а значить відмінно підходять для роботи з динамічними процесами (розпізнавання мови, музики і динамічних зображень) [14];

2) Спайкінгові нейронні мережі багатозадачні, адже вхідні дані обробляються в нейронної мережі з зворотними зв'язками, а різні групи зчитують нейронів можуть бути навчені на рішення різних завдань;

3) Спайкінгові нейронні мережі здатні здійснювати розпізнавання з передбаченням (тобто не обов'язково володіти повною інформацією про об'єкт або знати результат процесу);

4) Спайкінгові нейронні мережі просто навчати, так як досить навчити тільки вихідні зчитувальні нейрони;

5) Спайкінгові нейронні мережі мають підвищену продуктивність обробки інформації і стійкість перед перешкодами, так як використовують часове кодування інформації;

6) Спайкінгові нейронні мережі вимагають меншого числа нейронів, так як кожен нейрон імпульсної нейронної мережі замінює два нейрона (збудливий і гальмуючий) класичної спайкінговій нейронні мережі;

7) Спайкінгові нейронні мережі мають високу швидкість роботи і великий потенціал розпаралелювання, так як для передачі імпульсу необхідно відправити 1 біт, а не безперервну величину, як в частотній спайкінговій нейронні мережі [15];

8) Спайкінгові нейронні мережі можуть донавчатися і перенавчатися в процесі роботи [15].

Недоліки спайкінгових нейронних мереж

1) Спайкінгові нейронні мережі недоцільно використовувати в системах з малим числом нейронів;

2) Не існує досконалого алгоритму навчання.

Таким чином, завдяки зазначеним перевагам спайкінгових нейронних мереж перед іншими нейронними мережами, оберемо їх для реалізації поставленної в даній роботі задачі розпізнавання музичних патернів.

2.2 Структура та порядок функціонування багатошарової спайкінгової нейронної мережі

В частотних спайкінгових нейронних мережах використовується сигнал, який приймає значення, залежне від частоти породження імпульсів певної групою нейронів (ваги нейронів, власне, і є формою подання цієї частоти). Проте, середня частота імпульсів в послідовності є досить поганим варіантом подання інформації, так як різні види стимуляції можуть приводити до однакової середньої частоті імпульсів [15]. Для позбавлення від цього недоліку в спайкінгових нейронних мережах використовуються наступні види представлення інформації [16]:

- фазовий (часовий) - інформація про сигнал задається точним (або в межах деякого вікна) становищем імпульсів у часі (щодо будь-якого загального опорного ритму головного мозку);

- синхронний (позиційний / просторовий / популяційний) - інформація про сигнал задається синхронної активністю різних груп нейронів, і, як наслідок, синхронною (або в межах деякого вікна) появою імпульсів на певних виходах мережі (наприклад, реагуючі на високі і низькі частоти слухові рецептори вуха равлика, знаходяться в різних зонах);

- час до появи першого імпульсу - інформація про сигнал задається часом затримки появи першого імпульсу на будь-якому виході;

- порядковий - інформація про сигнал задається порядком отримання імпульсів на виходах мережі;

- інтервальний (затримковий) - інформація про сигнал задається відстанню між імпульсами, які отримуються на виходах мережі;

- резонансний - інформація про сигнал задається щільною послідовністю імпульсів (чергою), що приводить до виникнення резонансу (одиночні імпульси загасають і не вносять ніякого внеску в передачу інформації).

Крім цього, існують види представлення інформації, що є змішаною формою кількох простих видів представлення інформації, наприклад: просторово-часовий - інформація задається не тільки певною послідовністю імпульсів у часі, але вони ще й повинні виходити від певної групи нейронів; для популяції - частотний - інформація задається підвищенням частоти породження імпульсів певної групою нейронів.

Архітектури спакінгової нейронної мережі можна розділити на наступні групи [15]:

1) Нейронна мережа прямого поширення - дані передаються строго в одному напрямку: від входів до виходів, зворотні зв'язки відсутні, а обробка може проходити по безлічі шарів;

2) Рекурентна нейронна мережа - окремі нейрони / популяції нейронів взаємодіють один з одним, тобто є зворотний зв'язок. Спайкінгові нейронні мережі такого виду володіють власною динамікою і високою обчислювальної здатністю;

3) Змішана нейронна мережа - всередині спайкінгової нейронної мережі деякі популяції нейронів належать до виду нейронних мереж прямого поширення , а деякі - до рекурентних. Взаємодія між популяціями може бути як односпрямованою, так і взаємною.

4) Синхронне порушення ланцюга - являє собою багатошарову мережу, в якій імпульсна активність може поширюватися у вигляді синхронної хвилі передачі пачок імпульсів від однієї популяції до подальшої;

5) Резервуарні обчислення - резервуарна спайкінгова нейронна мережа складається з резервуара, виконаного по рекурентному типу, і вихідних нейронів.

Для роботи було спроектовано трьохшарову нейронну мережу, в першому шарі якої знаходиться 8 вхідних нейронів, які приймають сигнал енергії певної смуги частот з фільтра і перетворюють функціональну залежність цього сигналу від часу у послідовність імпульсів (спайків). В прихованому шарі знаходиться 1500 нейронів (інтернейрони), які формують хвилі імпульсів, що циркулюють у інтернейронах і формують так званий динамічний стан мережі, який однозначно пов`язаний із вхідним музичним патерном. На третьому шарі ми маємо 500 вихідних нейронів, які будучи натренованими, видають результат роботи мережі – номер класу розпізнаного музичного патерну.

Схематичне зображення структури нейронної мережі показано на рисунку 2.3

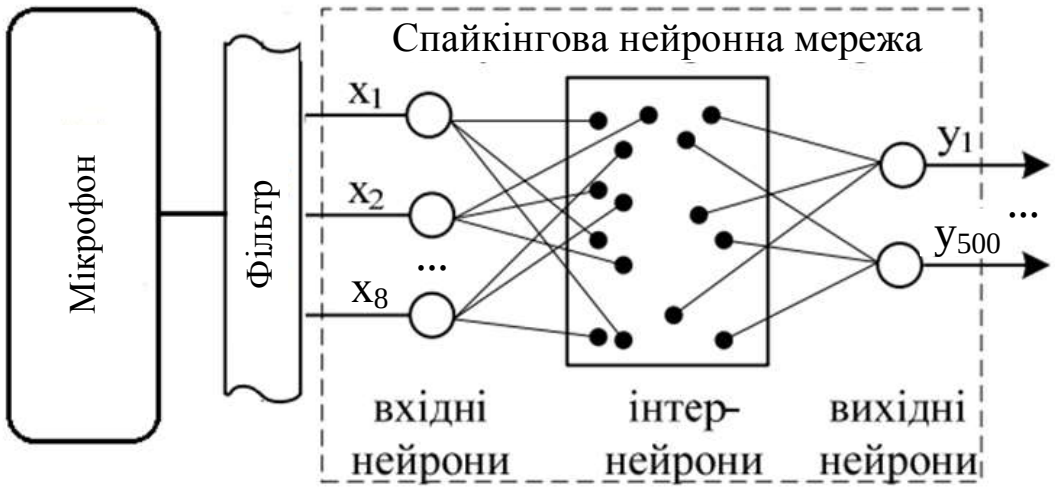


Рисунок 2.3 – Схематичне зображення структури нейронної мережі

Ілюстрація роботи спайкінгової нейронної мережі із сигналами після фільтрів представлена на рисунку 2.4

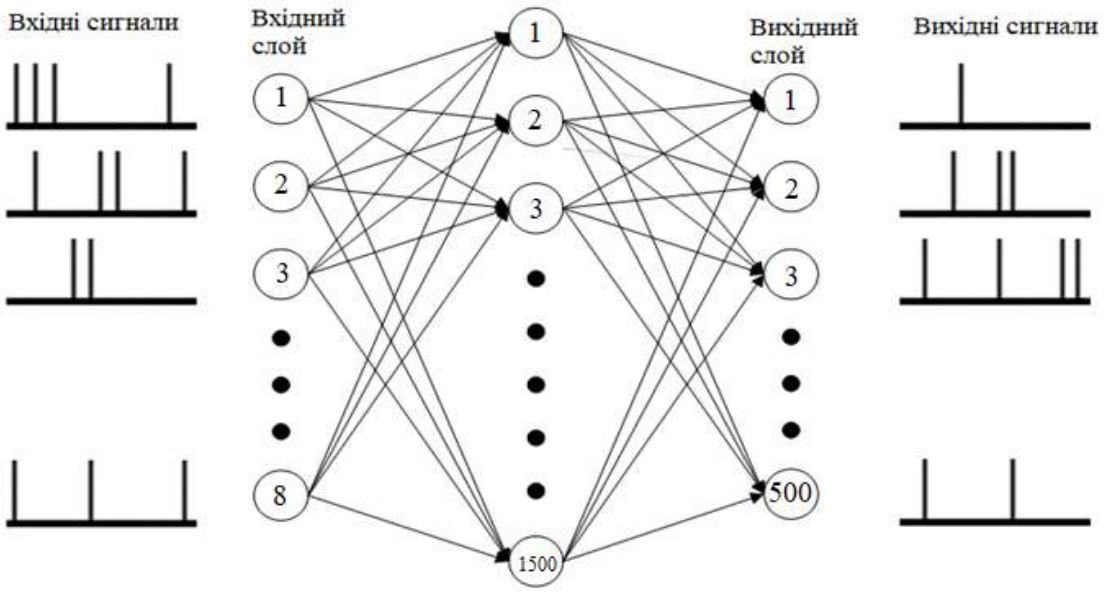


Рисунок 2.4 – Ілюстрація роботи спайкінгової нейронної мережі

Пояснення методів подання інформації в спайкінгових нейронних мережах подано на наступних рисунках: рисунок 2.5 – фазовий метод, рисунок 2.6 – синхронний метод, рисунок 2.7 – метод «час до першого імпульсу», рисунок 2.8 – порядковий метод, рисунок 2.9 – інтервальний метод.

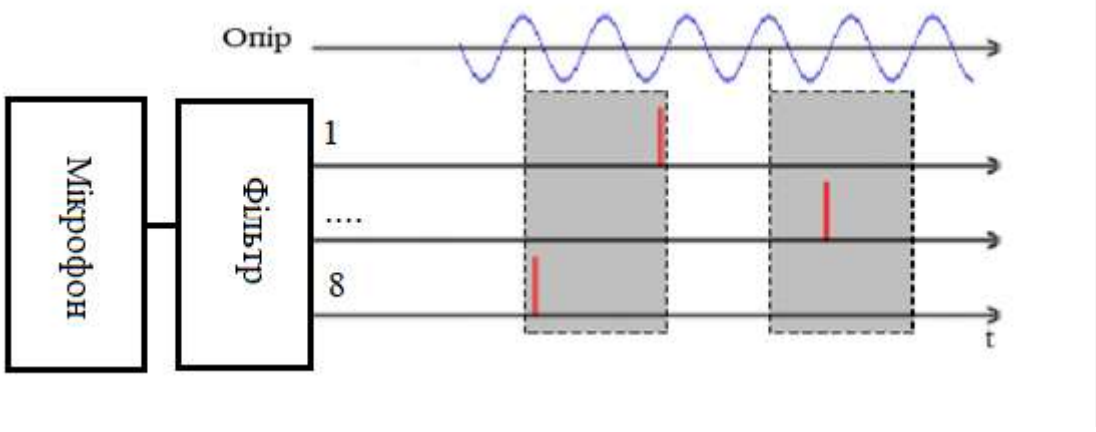


Рисунок 2.5 – Фазовий метод подання інформації в спайкінгових нейронних мережах

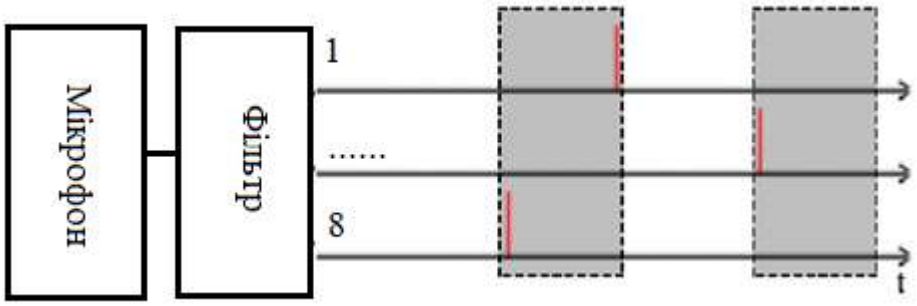


Рисунок 2.6 – Синхронний метод подання інформації в спайкінгових нейронних мережах

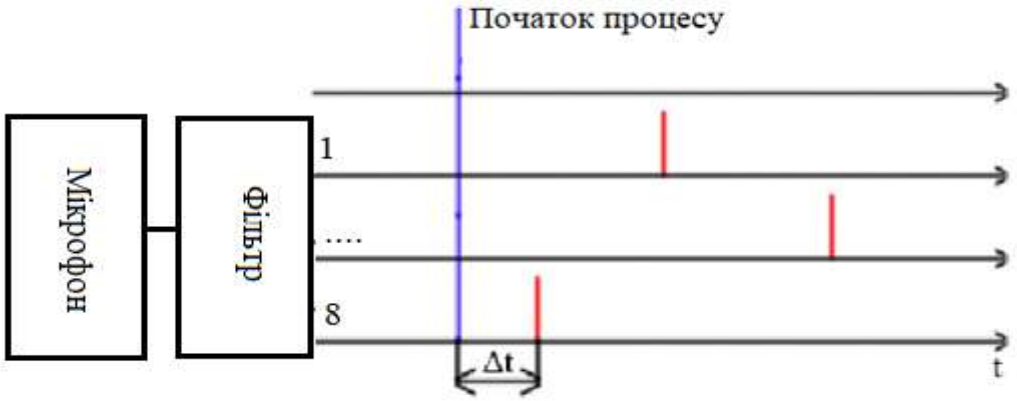


Рисунок 2.7 – Метод «час до появи першого імпульсу»

Рисунок 2.8 – Порядковий метод подання інформації в спайкінгових нейронних мережах

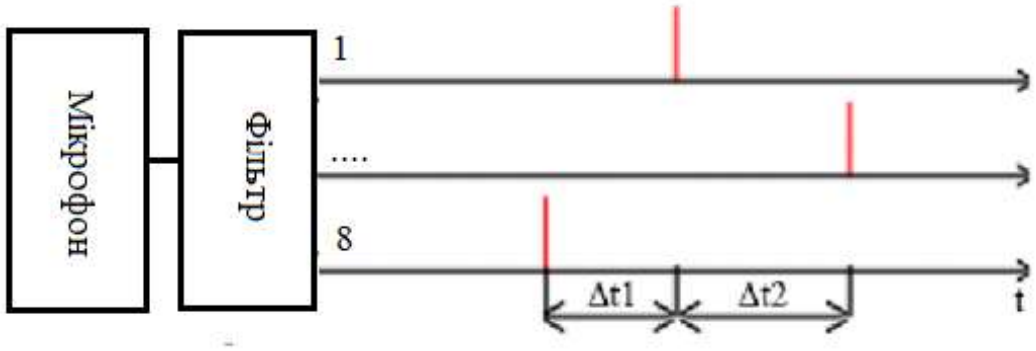


Рисунок 2.9 – Інтервальний метод подання інформації в спайкінгових нейронних мережах

Схематична модель одного з LIF нейронів, які використовуються при побудові мережі зображена на рисунку 2.10.

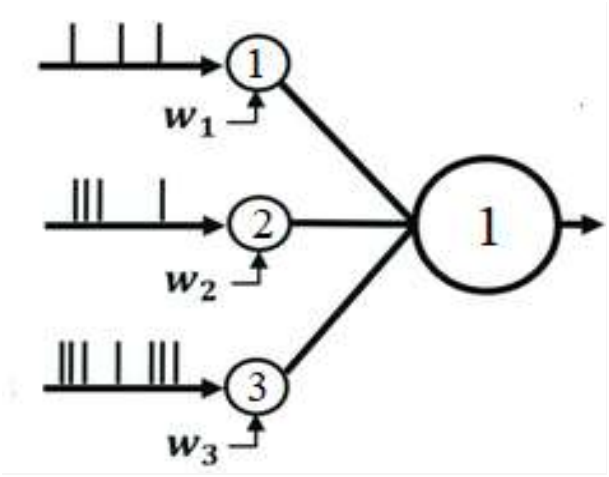


Рисунок 2.10 – Схематична модель LIF нейрона

Розрахунок значень ваг відбувається за формулою (2.3):

$$w = \sum_{i=1}^{n} w(k) \quad (2.3)$$

де: $w(k)$ - коефіцієнт швидкості навчання, w – ваги нейрона а n – коефіцієнт складності навчання.

2.3 Розробка методу розпізнавання музичних патернів на основі спайкінгової нейронної мережі

У роботі пропонується для розпізнавання музичних патернів використовувати спайкінгову нейронну мережу [15,16] - СНМ, тобто мережу, побудовану на спайкінгових нейронах. Для розв'язання цієї задачі не потрібно апаратно реалізовувати СНМ, а достатньо створити її комп'ютерну модель.

Передбачається, що система зможе розпізнавати музичні патерни як з мікрофона, так і з файлів формату .wav. Але для навчання нейромережі будуть використовуватись тільки файли звукового формату. Ці файли будуть братись із двох загальновідомих баз даних звукових файлів: AudioSet [17] та YouTube-8M [18].

Метод розпізнавання музичних патернів з використанням спайкінгової нейронної мережі можна сформулювати таким чином.

1. Створити (згенерувати) рекурентну спайкінгову нейронну мережу, складену з інтернейронів, у кількості N=1500 (взагалі рекомендується не менше 3m, де m – кількість еталонних музичних патернів (у даному випадку m=500), які мережа має «запам`ятати». З`єднання нейронів в мережі виконати за даними нейрофізіологічних досліджень [15]. Ваги зв`язків нейронів обрати невеликими випадковими.
2. Сформувати n вхідних нейронів по кількості спектральних смугових фільтрів звукового сигналу (у нашому випадку n=8). З`єднати кожен з них випадковим чином з не менш, ніж k нейронами мікромережі (n<k<N). Ваги зв`язків обрати випадковими.
3. Сформувати m вихідних нейронів по кількості музичних патернів, які необхідно розпізнавати. З`єднати кожен з них випадково з не менш, ніж *l* нейронами мікромережі (m<*l*<N). Ваги зв`язків обрати випадковими.
4. Застосувати алгоритм навчання з учителем на навчальній множині еталонних музичних патернів. При цьому підлаштовуються тільки ваги

зв`язків кожного з m вихідних нейронів. Як ідеальний цільовий вихід y(t) може виступати імпульсний сигнал з постійною (максимальною) частотою імпульсів ν_{max}, що дорівнює лабільності вихідного нейрона. Фактичний вихідний сигнал f(x(t)) буде являти собою послідовність імпульсів із довільними часовими проміжками між ними. Вхідний музичний фрагмент мережі тим ближче до еталонного музичного патерну, чим більше середня за період розпізнавання частота імпульсів вихідного нейрона, який відповідає даному еталонному музичному патерну.

5. Подати на вхід мережі досліджуваний музичний патерн з мікрофона або зі звукового файлу і зафіксувати який з m вихідних нейронів буде видавати максимальну кількість імпульсів. Саме цей нейрон і визначає еталонний музичний патерн, якому максимально відповідають вхідний звуковий фрагмент. Як оцінку міри схожості вхідного звукового фрагменту та еталонного музичного патерна можна використовувати відношення середньої за період розпізнавання частоти імпульсів вихідного нейрона до ν_{max} .

Як алгоритм навчання з учителем у п.4 може бути використаний, наприклад алгоритм SpikeProp [19].

2.4 Структура інформаційної технології розпізнавання музичних патернів на основі спайкінгової нейронної мережі

Структура інформаційної технології розпізнавання музичних патернів на основі спайкінгової нейронної мережі, детально описана у попередніх підпунктах, зображена на рис. 2.11.

Рисунок 2.11 - Структура інформаційної технології розпізнавання музичних патернів на основі спайкінгової нейронної мережі

Із рис. 2.11 видно, що в основі інформаційної технології лежить класифікація аудыосигналів, яка виконується спайкінговою нейронною мережею. Для роботи інформаційної технології розпізнавання музичних патернів на основі спайкінгової нейронної мережі вхідною є інформація про музичні патерни або з мікрофона, або із аудіофайлів формату .wav. Аудіосигнал

для подачі на нейромережу спочатку подається на фільтри частотного спектру у 8 смугах. А потім динамічний сигнал, виділений фільтрами, подається на вхідні нейрони мрежі, які є перетворювачами струм-частота. На виході вхідних нейронів отримується послідовність імпульсів, моменти появи яких однозначно пов`язані із залежністю енергії сигналу у цій частотній смузі від часу. Таким чином, кожний музичний паттерн представляється 8-канальним потоком імпульсів, який і є його характерним просторово-часовим «відбитком», що потім розпізнається нейромережею.

Далі для обробки цих просторово-часових «відбитків» потрібно ініціалізувати створення спайкінгової нейронної мережі. Потім здійснюється процес навчання цієї спайкінгової нейронної мережі на основі навчальної вибірки, що формується з еталонних музичних патернів. Після того як нейронна мережа навчена, її можна використовувати для розпізнавання музичних патернів. Для цього на вхід фільтрів подається невідомий звуковий фрагмент з мікрофона або з аудіофайла, який далі перетворюється фільтрами і вхідними нейронами у 8-канальний потік імпульсів. Цей потік проходить по інтернейронах та натренованих вихідних нейронах спайкінгової мережі і викликає на різних вихідних нейронах різні по частоті і моментах появи імпульсні сигнали. Далі відбувається процес фіксації середніх частот всіх вихідних нейронів та виведення інформації про номер класу вхідного патерну. Клас вхідного музичного патерну визачається по номеру вихідного нейрона, який видав максимальну кількість імпульсів за період тривалості вхідного аудіофрагмента.

Таким чином, розроблена структура інформаційної технології розпізнавання музичних патернів на основі спайкінгової нейронної мережі може бути використана для подальшої розробки програмних засобів.

2.5 Висновок до розділу 2

У розділі було обґрунтовано вибір типу нейронної мережі для задачі розпізнавання музичних патернів – спайкінгова нейромережа. Була розроблена структура спайкінгової нейронної мережі, проаналізовано види кодування інформації, порядок функціонування багатошарової спайкінгової нейронної мережі та математична модель спайкінгового нейрона. Було розроблено метод розпізнавання музичних патернів та структуру інформаційної технології розпізнавання музичних патернів на основі спайкінгової нейронної мережі.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ РОЗПІЗНАВАННЯ МУЗИЧНИХ ПАТЕРНІВ

Машинне навчання – це область з величезним вибором інструментів, бібліотек та фреймворків. Однак велика кількість варіантів призводить до розгубленості. Вибір фреймворку в перспективі може стати або перевагою, або недоліком майбутнього проекту.

Фреймворк диктує правила побудови архітектури програмного забезпечення. Ще на початковому етапі розробки задається поведінка за замовчуванням, каркас, який необхідно розширювати та змінювати відповідно до вказаних вимог протягом усього процесу розробки, а зміна архітектури проекту дуже затратна процедура. З іншого боку використання певного фреймворку дозволяє легко супроводжувати, модернізувати та масштабувати проект, відносно просто реалізовувати будь-які бізнес процеси.

Рішення на фреймворках, як правило, працюють значно швидше, безпечніші і витримують більші навантаження, ніж самописні системи. Отже, вибір інструментів розробки є важливим етапом створення програмного забезпечення, і в цьому розділі розглядатимуться основні підходи та програмні засоби, їхні плюси та мінуси для розробки продуктів з використанням машинного навчання.

3.1 Обґрунтування вибору мови програмування

Для розробки даної системи використовується об’єктно орієнтована мова програмування Python [20]. Для кращого розуміння вибору даної мови пропонується опис найпопулярніших мов програмування і їх порівняння.

Python — інтерпретована об'єктно-орієнтована мова високого рівня зі строгою динамічною типізацією. Розроблена в 1990 році Гвідо ван Россумом. Структури даних високого рівня разом із динамічною семантикою та

динамічним зв'язуванням роблять її привабливою для швидкої розробки програм, а також як засіб поєднування наявних компонентів. Python підтримує модулі та пакети модулів, що сприяє модульності та повторному використанню коду. Інтерпретатор Python та стандартні бібліотеки доступні як у скомпільованій, так і у вихідній формі на всіх основних платформах. В мові програмування Python підтримується кілька парадигм програмування, зокрема: об'єктно-орієнтована, процедурна, функціональна та аспектно-орієнтована.

Серед основних її переваг можна назвати такі:

- чистий синтаксис (для виділення блоків слід використовувати відступи);
- переносність програм (що властиве більшості інтерпретованих мов);
- стандартний дистрибутив має велику кількість корисних модулів (включно з модулем для розробки графічного інтерфейсу);
- можливість використання Python в діалоговому режимі (дуже корисне для експериментування та розв'язання простих задач);
- стандартний дистрибутив має просте, але разом із тим досить потужне середовище розробки, яке зветься IDLE і яке написане на мові Python;
- зручний для розв'язання математичних проблем (має засоби роботи з комплексними числами, може оперувати з цілими числами довільної величини, у діалоговому режимі може використовуватися як потужний калькулятор);
- відкритий код (можливість редагувати його іншими користувачами).

Python має ефективні структури даних високого рівня та простий, але ефективний підхід до об'єктно-орієнтованого програмування. Елегантний синтаксис Python, динамічна обробка типів, а також те, що це інтерпретована мова, роблять її ідеальною для написання скриптів та швидкої розробки прикладних програм у багатьох галузях на більшості платформ.

Інтерпретатор мови Python і багата стандартна бібліотека (як вихідні тексти, так і бінарні дистрибутиви для всіх основних операційних систем)

можуть бути отримані з сайту Python www.python.org, і можуть вільно розповсюджуватися. Цей самий сайт має дистрибутиви та посилання на численні модулі, програми, утиліти та додаткову документацію

Інтерпретатор мови Python може бути розширений функціями та типами даних, розробленими на C чи C++ (або на іншій мові, яку можна викликати із C). Python також зручна як мова розширення для прикладних програм, що потребують подальшого налагодження.

У табл. 3.1 указано порівняння популярних мов програмування

Таблиця 3.1 – Порівняння популярних мов програмування

№	Парадигми / Мови	Функціональна	Логічна	Процедурна	Об'єктно-орієнтована
1	Ada	-	-	+	+
2	C	-	-	+	-
3	C++		-	+	+
4	C#		-	+	+
5	Java	-	-	+	+
6	Haskell	+		+	-
7	Common LISP	+		+	+
8	Pyphon	+	+	+	+
9	Smalltalk	+		+	+
10	Delphi		-	+	+
11	Prolog		+	-	-

В даній роботі використовувалися принципи ООП. «ООП» означає «Об'єктно-орієнтоване програмування». Це такий підхід до написання програм, який ґрунтується на об'єктах, а не на функціях і процедурах. Ця модель ставить в центр уваги об'єкти, а не дії, дані, а не логіку. Об'єкт - реалізація класу. Всі реалізації одного класу схожі один на одного, але можуть мати різні параметри і значення. Об'єкти можуть задіяти методи, специфічні для них.

Принципи ООП:

1. Абстракція даних: подробиці внутрішньої логіки приховані від кінцевого користувача.

2. Наслідування: успадкування уможливлює повторне використання коду - якщо якийсь клас вже має якусь логіку і функції, нам не потрібно переписувати все це заново для створення нового класу, ми можемо просто включити старий клас в новий, цілком.

3. Інкапсуляція: включення в клас об'єктів іншого класу, питання доступу до них, їх видимості.

4. Поліморфізм: це властивість одних і тих же об'єктів і методів приймати різні форми.

Наведемо декілька операторів ООП в Python.

- `Оператор if` (якщо). Альтернативний блок після нього `else` (або).
- Оператори цикла `while` (доки) і `for` (для). Всередині циклу можна примінити команди `break` і `continue` для переривання цикла і переходу до наступної ітерації.
- Оператор оголошення класа `class`.
- Оператор определения функции, метода или генератора `def`.

3.2 Обґрунтування вибору середовища розробки

PyCharm — інтегроване середовище розробки для мови програмування Python. Надає засоби для аналізу коду, графічний зневаджувач, інструмент для запуску юніт-тестів і підтримує веб-розробку на Django. PyCharm [21] розроблена російською компанією JetBrains на основі IntelliJ IDEA.

PyCharm працює під операційними системами Windows, Mac OS X і Linux.

Можливості програми:

- Статичний аналіз коду, підсвічування синтаксису і помилок.
- Навігація серед проектів і сирцевого коду: відображення файлової структури проекту, швидкий перехід між файлами, класами, методами і використаннями методів.
- Рефакторинг: перейменування, витяг методу, введення змінної, введення константи, підняття і опускання методу тощо.
- Інструменти для веб-розробки з використанням фреймворку Django
- Вбудований зневаджувач для Python
- Вбудовані інструменти для юніт-тестування
- Розробка з використанням Google App Engine
- Підтримка систем контролю версій: загальний користувацький інтерфейс для Mercurial, Git, Subversion, Perforce і CVS з підтримкою списків змін та злиття

Колекція інструментів PyCharm з «коробки» включає вбудований відладник і тестовий прогін; Профілювальник Python; вбудований термінал; інтеграція з основними VCS і вбудованими інструментами бази даних; можливості віддаленої розробки з віддаленими перекладачами; вбудований ssh-термінал; і інтеграція з Docker і Vagrant, що значно спрощує роботу з програмою і початковим налаштуванням проекту.

На додаток до Python, PyCharm забезпечує першокласну підтримку різних середовищ веб-розробки Python, визначених мов шаблонів, JavaScript, CoffeeScript, TypeScript, HTML / CSS, AngularJS, Node.js і багатьох інших.

PyCharm інтегрується з IPython Notebook, має інтерактивну консоль Python і підтримує Anaconda, а також кілька наукових пакетів, включаючи Matplotlib і NumPy.

Через вище вказані характеристики PyCharm є одним із провідних середовищ програмування на мові Python. Підтримка управління базами даних та підтримка веб-фреймворків допомагає в розробці веб-частини нейронного перенесення стилю. Підтримка математичних модулів та Anaconda – безкоштовний дистрибутив пакетів для машинного навчання, забезпечує комфортну розробки моделей для машинного навчання.

3.3 Розробка алгоритму функціонування програмного забезпечення розпізнавання музичних патернів

Описано математичну модель нейронної мережі з використанням алгоритму зворотнього поширення помилки перетворення повинні бути організовані у вигляді бібліотек.

Для гарантії якості та перевірки роботи потрібно виконати окремо тестування кожного блоку. Програма має два варіанти використання, діаграма яких зображена на рисунку 3.1.

Така архітектура модулів робить гнучким процес їхнього використання. Модулі можна використовувати як окреме прикладне програмне забезпечення, а також розроблені модулі можна підключати до інших програм для того, щоб виконувати над музичними патернами більш складні завдання.

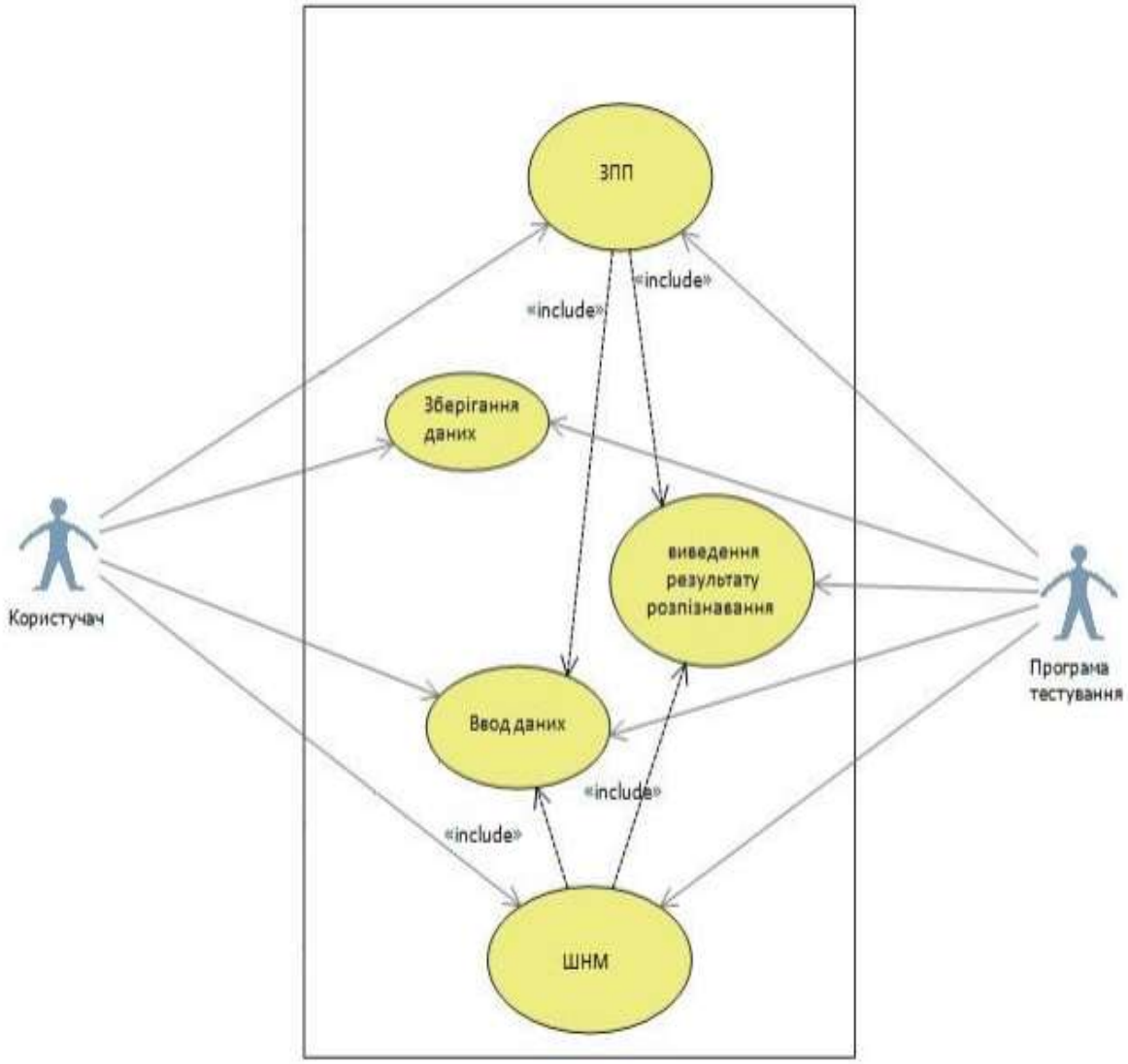


Рисунок 3.1 – Діаграма варіантів використання програми

Граф-схема алгоритму роботи програми розпізнавання музичних патернів на основі спайкінгової нейронної мережі зображенно на рисунку 3.2.

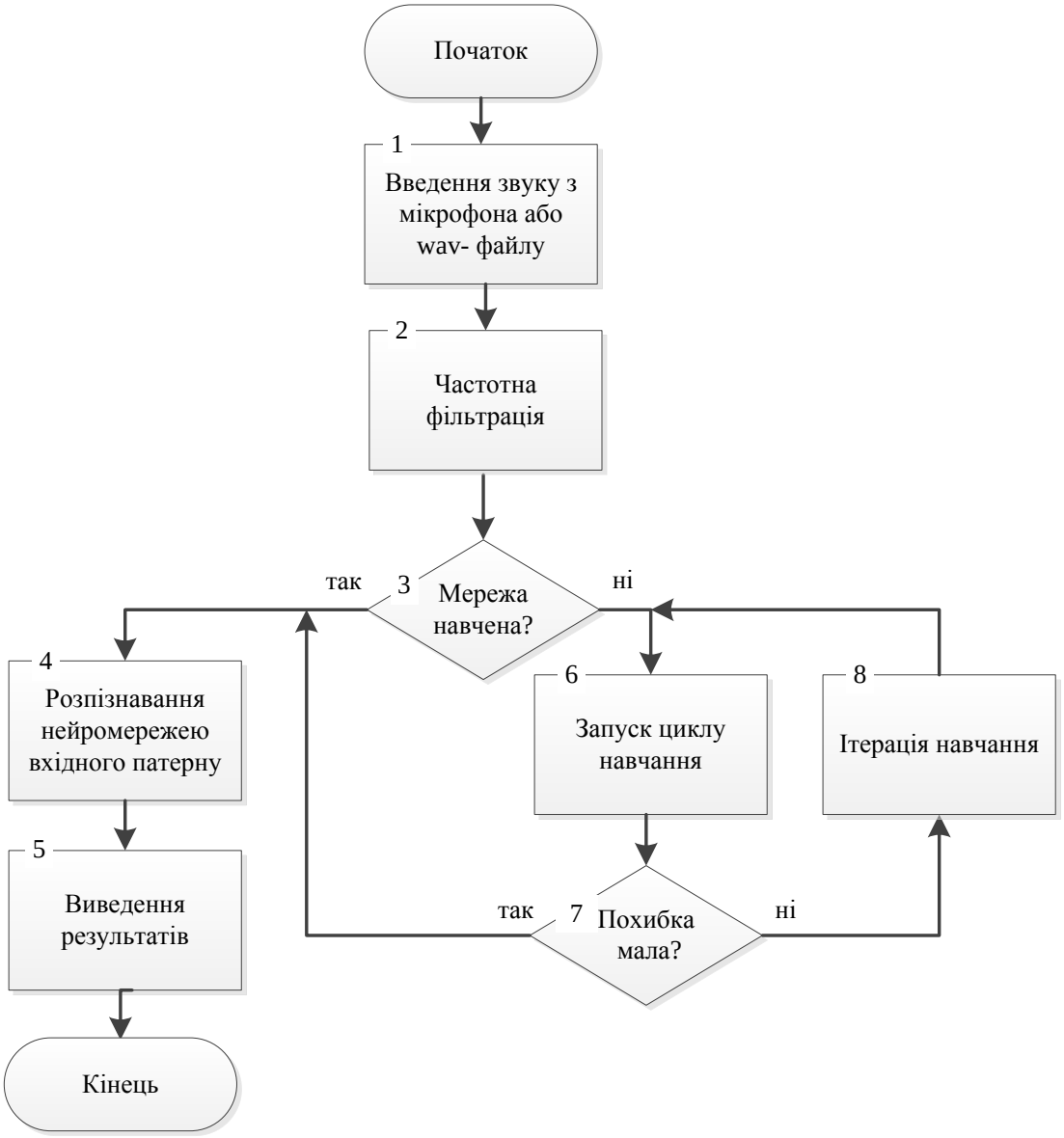


Рисунок 3.2 –Граф-схема алгоритму роботи програми розпізнавання музичних патернів на основі спайкінгової нейронної мережі

Цикл навчання нейронної мережі відбувається при використанні навчання з вчителем, цей метод необхідний для того щоб можна було класифікувати патерни які розпізнає мережа, оскільки ми маємо задати початкові значення і ввести в мережу значення класифікації музичних патернів, то інші типи навчання нейронної мережі нам не підходять.

3.4 Розробка UML-діаграми класів

Результатом проектування є програмне забезпечення розпізнавання музичних патернів на основі спайкінгових нейронної мереж. Для досягнення максимально точного розпізнавання математичну модель модифіковано. Спроектовано діаграму класів. Отримана структура системи дає змогу вирішити поставлену у технічному завданні задачу. На рисунку 3.3 зображенно діаграму класів програми розпізнавання музичних патернів

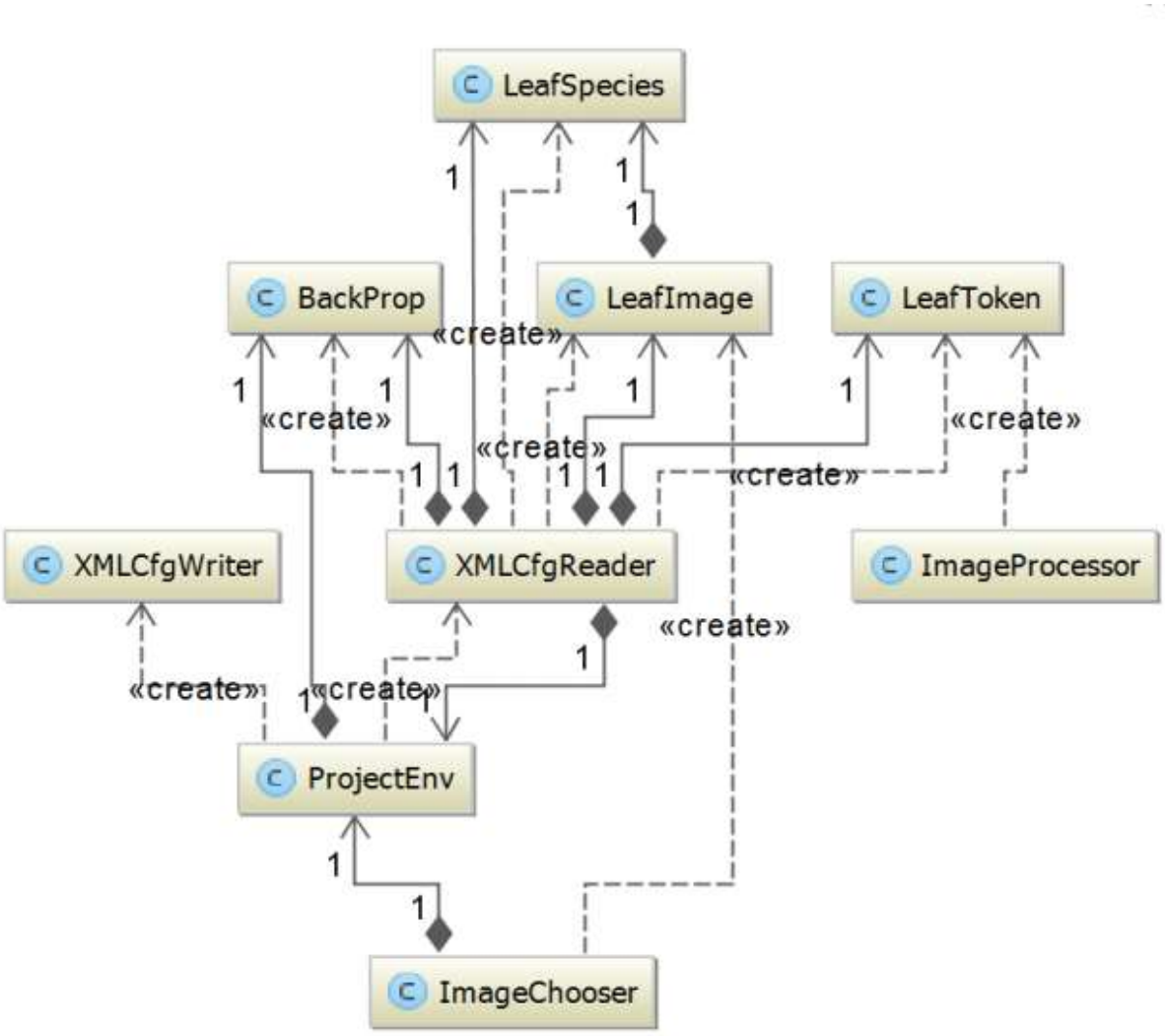


Рисунок 3.3 – Діаграма класів програми розпізнавання музичних патернів

3.5 Реалізація програми розпізнавання музичних патернів в середовищі PyCharm

За основу роботи була взята бібліотека TensorFlow. TensorFlow є системою машинного навчання Google Brain другого покоління, випущеною як відкрите програмне забезпечення 9 листопада 2015 року. В той час як еталонна реалізація працює на одиничних пристроях, TensorFlow може працювати на декількох центральних та графічних процесорах (включно з додатковими розширеннями CUDA для обчислень загального призначення на графічних процесорах). TensorFlow [22] доступна для 64-розрядних Linux, macOS, Windows, та для мобільних обчислювальних платформ, включно з Android та iOS.

Обчислення TensorFlow виражаються як станові графи потоків даних. Назва TensorFlow походить від операцій, що такі нейронні мережі виконують над багатовимірними масивами даних. Ці багатовимірні масиви називають «тензорами». В червні 2016 року Джефф Дін з Google заявив, що TensorFlow згадували 1 500 репозиторіїв на GitHub, лише 5 з яких були від Google.

Платформа спочатку розроблена командою Google Brain і використовуються в сервісах Google для розпізнавання мови, виділення облич на фотографіях, визначення схожості зображень, відсіювання спаму в Gmail, підбору новин у Google News і організації перекладу з урахуванням смислу. Розподілені системи машинного навчання можна створювати на типовому обладнанні, завдяки вбудованій підтримці в TensorFlow рознесення обчислень на кілька CPU або GPU.

Серед застосувань, для яких TensorFlow є основою, є програмне забезпечення автоматизованого опису зображень, таке як DeepDream. 26 жовтня 2015 року Google офіційно реалізувала RankBrain, який підтримує TensorFlow. RankBrain тепер обробляє суттєве число пошукових записів,

замінюючи та доповнюючи традиційні статичні алгоритми на основі результатів пошуку.

Іншими застосуванням є використання у складі програм FakeApp з метою безшовного поєднання фото- та відеозображень для створення підробних, але правдоподібних відео, відомих під назвою Deepfake.

TensorFlow надає бібліотеку готових алгоритмів чисельних обчислень, реалізованих через графи потоків даних (data flow graphs). Вузли в таких графах реалізують математичні операції або точки входу/виводу, в той час як ребра графа представляють багатовимірні масиви даних (тензори), які перетікають між вузлами. Вузли можуть бути закріплені за обчислювальними пристроями і виконуватися асинхронно, паралельно обробляючи разом все підходящі до них тензори, що дозволяє організувати одночасну роботу вузлів в нейронної мережі за аналогією з одночасною активацією нейронів в мозку.

Інтеграція TensorFlow з Python забезпечується не лише через pip, а й у дистрибутиві Anaconda.

Також однією із важливих частин програми є бібліотека Numpy. Numpy — розширення мови Python, що додає підтримку великих багатовимірних масивів і матриць, разом з великою бібліотекою високорівневих математичних функцій для операцій з цими масивами. Попередник Numpy, Numeric, був спочатку створений Jim Hugunin. Numpy — відкрите програмне забезпечення і має багато розробників.

Оскільки Python — інтерпретована мова, математичні алгоритми, часто працюють в ньому набагато повільніше ніж у компільованих мовах, таких як C або навіть Java. NumPy намагається вирішити цю проблему для великої кількості обчислювальних алгоритмів забезпечуючи підтримку багатовимірних масивів і безліч функцій і операторів для роботи з ними. Таким чином будь-який алгоритм, який може бути виражений в основному як послідовність операцій над масивами і матрицями, працює так само швидко, як еквівалентний код, написаний на C.

NumPy можна розглядати як гарну вільну альтернативу MATLAB, оскільки мова програмування MATLAB зовні нагадує NumPy: обидві вони інтерпретовані, і обидві дозволяють користувачам писати швидкі програми поки більшість операцій проводяться над масивами або матрицями, а не над скалярами. Перевага MATLAB у великій кількості доступних додаткових тулбоксів, включаючи такі як пакет Simulink. Основні пакети, що доповнюють NumPy, це: SciPy — бібліотека, що додає більше MATLAB-подібної функціональності; Matplotlib — пакет для створення графіки в стилі MATLAB. Внутрішньо як MATLAB, так і NumPy базується на бібліотеці LAPACK, призначеної для вирішення основних задач лінійної алгебри.

PyAudio – це модуль який дозволяє працювати з кросплатформенними бібліотекою PortAudio, програвати і записувати звуки.

Після того як файл був принятий за допомогою модуля PyAudio він обробляє його і передає їх нейронній мережі яка в свою чергу розпізнає його і після чого порівнює з вже відомими їй комбінаціями даних за допомогою модуля TensorFlow, якщо даний файл не вдалось розпізнати то мережа починає навчатись на ньому і додає його до списку вже відомих файлів.

За отримання початкових файлів відповідає код:

```
import argparse
import numpy as np
from scipy.io import wavfile

parser = argparse.ArgumentParser(description='Read file and process audio')
parser.add_argument('wav_file', type=str, help='File to read and process')

def process_file(wav_file):
    sr, data = wavfile.read(wav_file)
    if data.dtype != np.int16:
```

```
        raise TypeError('Bad sample type: %r' % data.dtype)

    # local import to reduce start-up time
    from audio.processor import WavProcessor, format_predictions

    with WavProcessor() as proc:
        predictions = proc.get_predictions(sr, data)

    print(format_predictions(predictions))

if __name__ == '__main__':
    args = parser.parse_args()
    process_file(**vars(args))
```

В даному коді викнується функція парсингу, parser і parser.add_argument. Функція parser відповідає за самий процес прийняття файлів з даними, тоді як функція parser.add_argument обмежує роботу і допускає до обробки лише файли з розширеням .wav.

Функція from audio.processor import WavProcessor, format_predictions забеспечує обробку файлів з розширеням .wav за допомогою іпморта спеціальної бібліотеки WavProcessor яка створює окремі процеси які і обробляють вхідні дані.

Далі дані, які були оброблені передаються до processor.py, далі дані оброблюються в коді за допомогою модуля TensorFlow. Оскільки початковий файл знаходиться в розширені .wav то тут також необхідно забезпечити роботу окремого процесу який буде обробляти файли з розширеням .wav, реалізація даного процесу виконується в коді:

```
__all__ = ['WavProcessor', 'format_predictions']
```

```
cwd = os.path.dirname(os.path.realpath(__file__))

class WavProcessor(object):
    _class_map = {}
    _vggish_sess = None
    _youtube_sess = None
```

Порівняння відбувається за допомогою побудови графіків функцій які звіряються і визначається значення введених даних, створення графіків відбувається за допомогою такого коду:

```
    def _init_vggish(self):
        graph = tf.Graph()
        with graph.as_default():
            sess = tf.Session()
            vggish.model.define_vggish_slim(training=False)
            vggish.model.load_vggish_slim_checkpoint(sess, params.VGGISH_MODEL)
        self._vggish_sess = sess
```

Навчання відбувається за допомогою функцій params і model. Функція params задає параметри по яким порівнюються значення, тоді як функція model визнаає моделі даних які будуть використовуватись:

```
    def _init_youtube(self):
        graph = tf.Graph()
        with graph.as_default():
            sess = tf.Session()
            youtube8m.model.load_model(sess,
params.YOUTUBE_CHECKPOINT_FILE)
```

```
        self._youtube_sess = sess

    def _init_class_map(self):
        with open(params.CLASS_LABELS_INDICES) as f:
            next(f)  # skip header
            reader = csv.reader(f)
            for row in reader:
                self._class_map[int(row[0])] = row[2]

    def get_predictions(self, sample_rate, data):
        samples = data / 32768.0  # Convert to [-1.0, +1.0]
        examples_batch = vggish.input.waveform_to_examples(samples, sample_rate)
        features = self._get_features(examples_batch)
        predictions = self._process_features(features)
        predictions = self._filter_predictions(predictions)
        return predictions

    def _filter_predictions(self, predictions):
        count = params.PREDICTIONS_COUNT_LIMIT
        hit = params.PREDICTIONS_HIT_LIMIT

        top_indices = np.argpartition(predictions[0], -count)[-count:]
        line = ((self._class_map[i], float(predictions[0][i])) for
                i in top_indices if predictions[0][i] > hit)
        return sorted(line, key=lambda p: -p[1])
```

За допомогою функцій shape і collection відбувається передача і накопичення даних по навчанню нейронної мережі, shape відповідає за передачу даних до списка відомих патернів, а collection за зберігання:

```
    def _process_features(self, features):
        sess = self._youtube_sess
        num_frames = np.minimum(features.shape[0], params.MAX_FRAMES)
        data = youtube8m.input.resize(features, 0, params.MAX_FRAMES)
        data = np.expand_dims(data, 0)
        num_frames = np.expand_dims(num_frames, 0)

        input_tensor = sess.graph.get_collection("input_batch_raw")[0]
        num_frames_tensor = sess.graph.get_collection("num_frames")[0]
        predictions_tensor = sess.graph.get_collection("predictions")[0]

        predictions_val, = sess.run(
            [predictions_tensor],
            feed_dict={
                input_tensor: data,
                num_frames_tensor: num_frames
            })

        return predictions_val
```

Використання модуля TensorFlow для створення так званих «тензорних» процесів які описуались раніше:

```
    def _get_features(self, examples_batch):
        sess = self._vggish_sess
        features_tensor = sess.graph.get_tensor_by_name(
            params.VGGISH_INPUT_TENSOR_NAME)
        embedding_tensor = sess.graph.get_tensor_by_name(
```

```
    params.VGGISH_OUTPUT_TENSOR_NAME)

[embedding_batch] = sess.run(
    [embedding_tensor],
    feed_dict={features_tensor: examples_batch}
)
postprocessed_batch = np.dot(
    self._pca_matrix, (embedding_batch.T - self._pca_means)
).T
return postprocessed_batch
```

3.5 Висновок до розділу 3

У розділі було обґрунтовано вибір мови програмування Python та середовища програмування PyCharm для задачі розпізнавання музичних патернів. Була описано використання бібліотек TensorFlow та NumPy для створення програмної реалізації спайкінгової нейронної мережі та її навчання, а також модуля PyAudio для прийняття та обробки звукових файлів і передачі їх на вхід нейронній мережі.

4 ТЕСТУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ РОБОТИ ПРОГРАМИ РОЗПІЗНАВАННЯ МУЗИЧНИХ ПАТЕРНІВ

4.1 Тестування програми розпізнавання музичних патернів на основі спайкінгової нейронної мережі

Проведемо тестування роботи програми розпізнавання музичних патернів. Навчання нейромережі відбувалось на аудіофрагментах із бази даних [17]. Обсяг навчальної вибірки - 2500 файлів. Для тестування програми було взято 1000 файлів, які не перетинаються з навчальними.

Для запуску програми необхідно ввести в командну консоль таку команду «parse_file.py Bb3.wav», де parse_file.py це файл, який приймає вхідні дані, а Bb3.wav це повний шлях до необхідного нам музичного файлу, шлях має бути відносним папки з проектом, тобто ми можемо вписати як Bb3.wav так і D:\pythondiplom\ Bb3.wav і значення виводу програми для нас залишиться незмінним. Коли програма запуститься вона почне перевірку, це можна буде побачити в консолі. На рисунку 4.1 зображенно приклад парсингу даних.

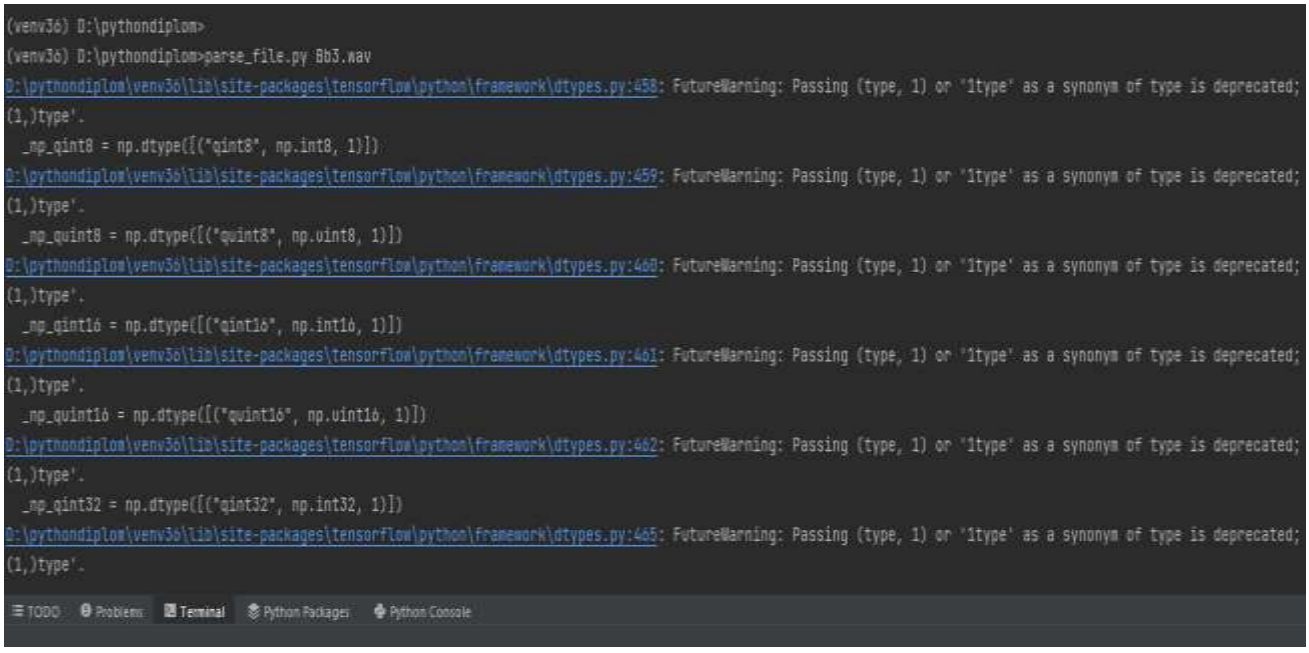


Рисунок 4.1 – Приклад парсингу даних

Після завершення парсингу даних виконується передача обробленого матеріалу в модуль TensorFlow, який в свою чергу звіряє дані з вже існуючими і предає інформацію. Приклад зображенно на рисунку 4.2.

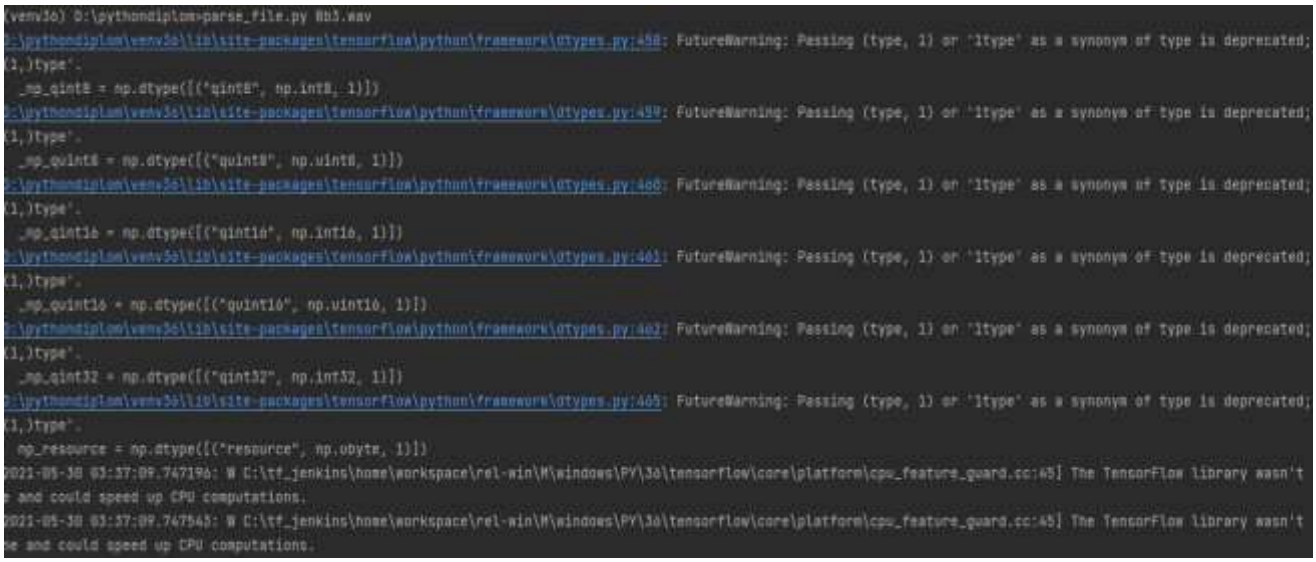


Рисунок 4.2 – Приклад розпізнавання модулем TensorFlow

Після цього ми отримуємо результат, який подається у вигляді переліку музичних патернів, на які був схожий вхідний аудіосигнал. Приклад отриманого результату зображенно на рисунку 4.3. Так, з останнього рядка тексту рис. 4.3 видно, що вхідний музичний патерн з подібністю 0,72 відповідає «щебетанню» (Chirp tone), з подібністю 0,25 відповідає «ксилофону» (Marimba, xylophone), з подібністю 0,22 відповідає «глокеншпілю» (Glockenspiel) і з подібністю 0,18 відповідає «гонгу» (Gong). Отже, виводиться назва класу, якому найбільше подібний вхідний аудіофрагмент і ще три класи по мірі спадання подібності.

Якщо нам необхідно дослідити фрагмент, який нейронна мережа не може розпізнати, то програма подасть запит на внесення даних про даний музичний патерн. Приклад такого запиту зображенно на рисунку 4.4.

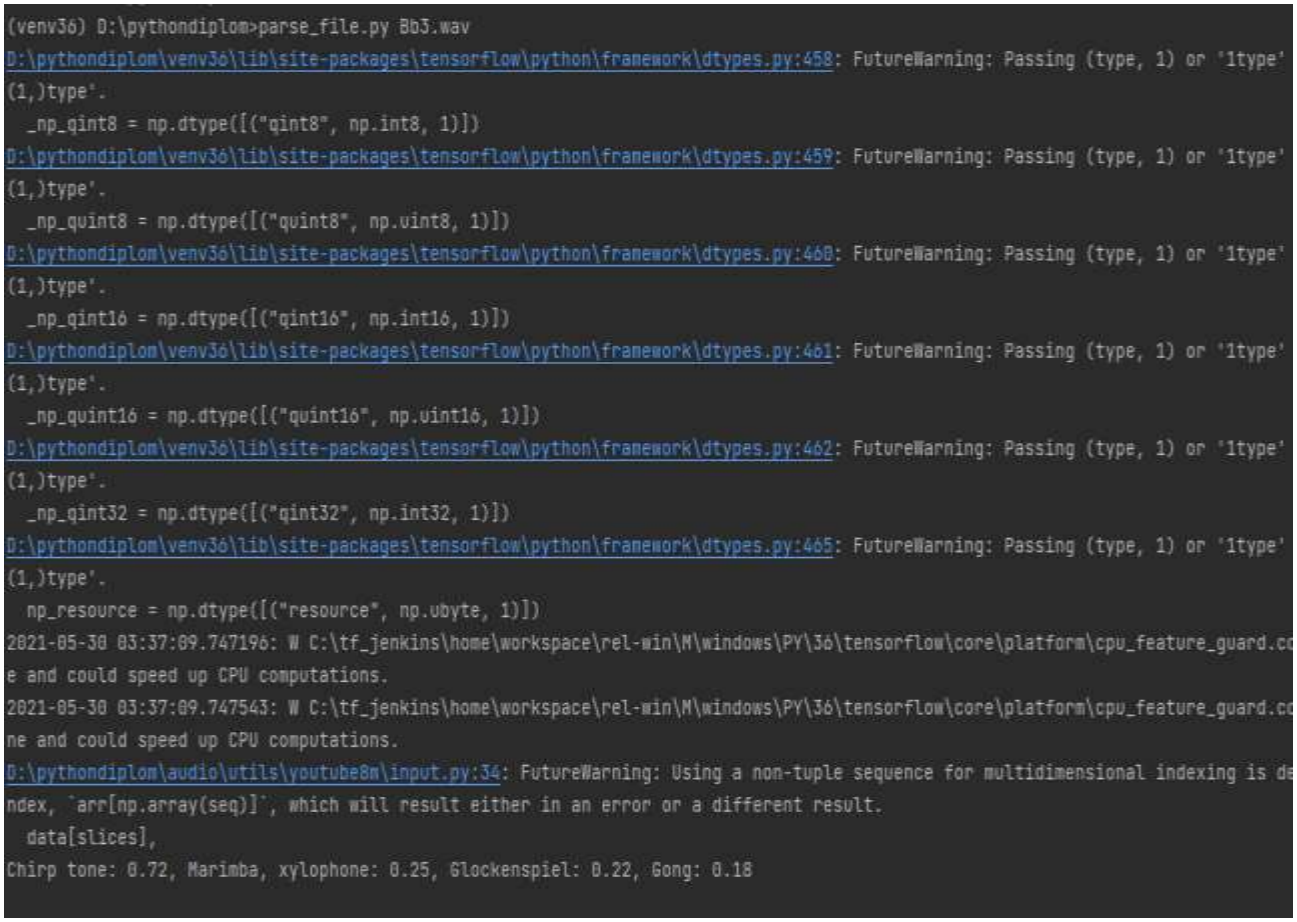


Рисунок 4.3 – Результат роботи програми

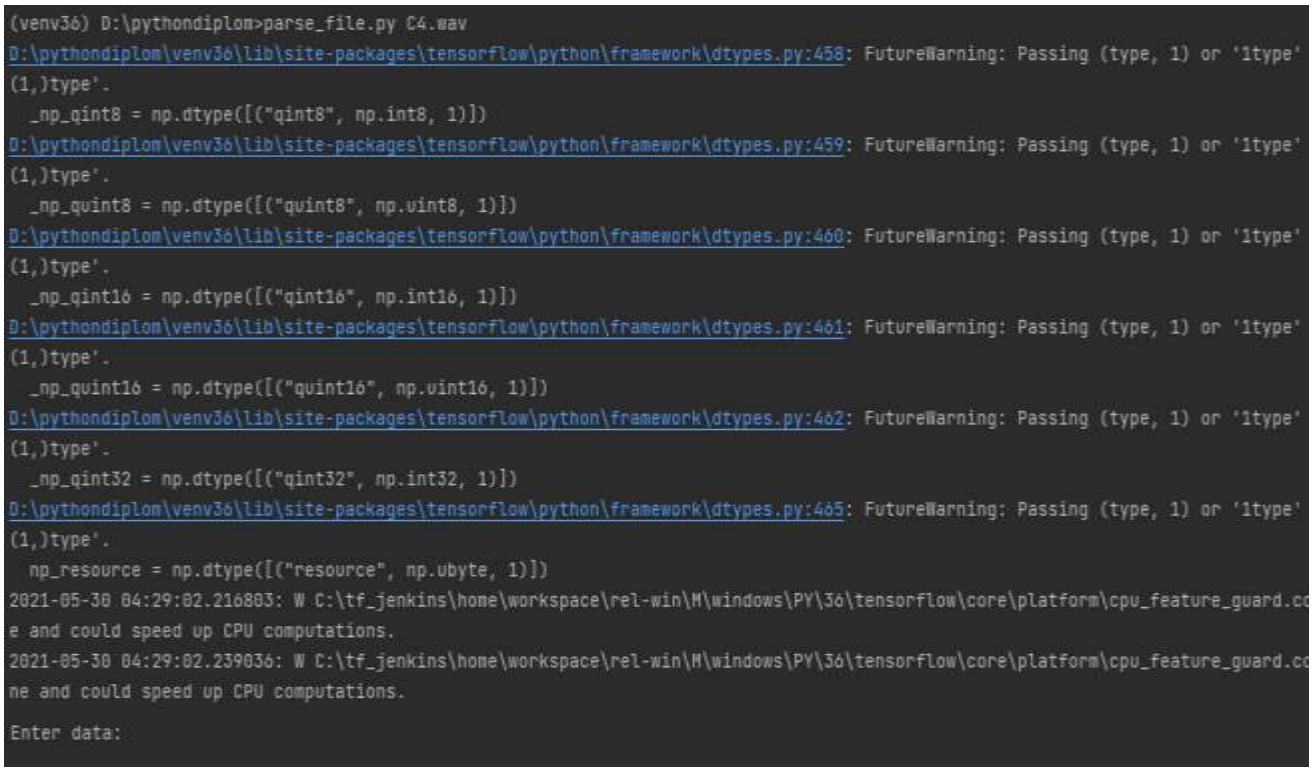


Рисунок 4.4 – Запит на введення даних для невідомого патерну

Введенні дані потряплять до списка з всіма відомими мережі значенями музичних патернів і будут використовуватися в подальшому, на рисунку 4.5 зображенно як після введення нових значень нейронній мережі вдалось розпізнати музичний патерн.

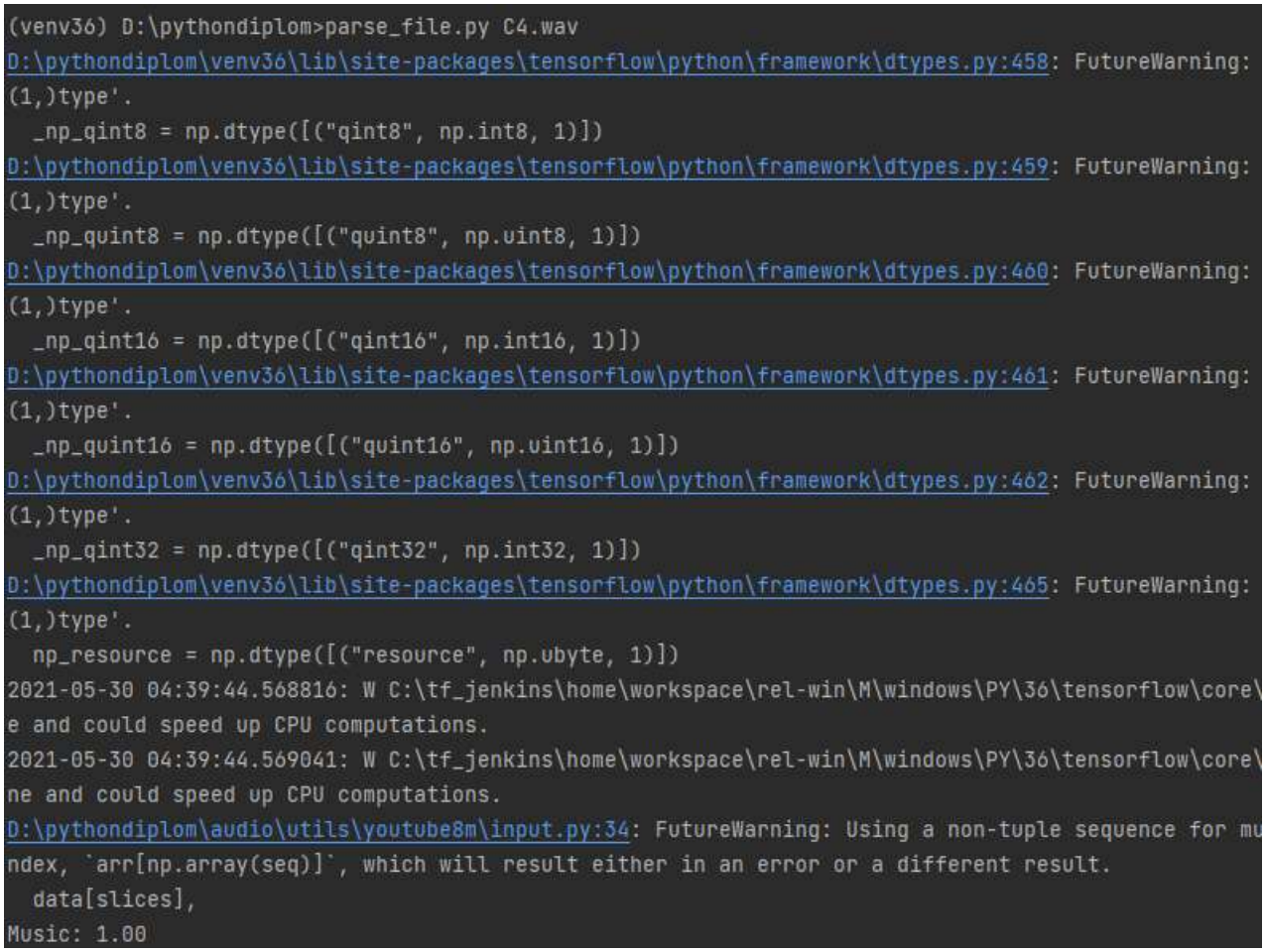


Рисунок 4.5 – Розпізнавання нового музичного патерну

Всі данній які зберігаються, таким чином потрапляють до сппеціального списку. Дані в цьому списку зберігаються в форматі index,mid,display_name, де index це порядковий номер, mid – значення побудованого графіка, а display_name - це назва даного патерну, яку повинен ввести користувач при

навчанні нейронної мережі. Дані зберігаються в форматі списку, що забезпечує спрощене користування ними і простоту в додавані нових музичних патернів.

4.2 Аналіз результатів роботи програми розпізнавання музичних патернів на основі спайкінгової нейронної мережі

Проведемо порівняння даного проекту з анатогом. В якості аналогу візьмемо програму Tunatic. Для тесту було відібрано 1000 музичних композицій і відправлено на розпізнавання в дану програму. Результатом стало правильно розпізнані 863 композиції, в той час як розроблена в ході роботи програма змогла розпізнати 954 композиції. Результати порівняння роботи обох програм зображені у таблиці 4.1.

Таблиця 4.1 – Результати порівняння роботи розробленої програми та аналога

Програма	Подано	Розпізнано	Достовірність розпізнавання, %
Програма, розроблема в ході роботи	1000 композицій	954 композиції	95.4%
Програма аналог Tunatic	1000 копмозицій	863 композиції	86.3%

Виходячи з табл. 4.1, ми можемо стрверджувати, що розроблена програма має достовірність розпізнавання 95,4 %, а програма –аналог – 86,3%, тобто достовірність розпізнавання порівняно з аналогом підвищено на 9,1% (95,4 – 86,3=9,1). Тобто мета роботи досягнута - достовірність розпізнавання музичних патернів підвищена на 9,1 %.

4.3 Висновок до розділу 4

У четвертому розділі в результаті тестування програми було доведено її працездатність та відповідність поставленому завданню. Тестування нейронної мережі проходило на вибірці з 1000 музичних композицій. Розроблена програма має достовірність розпізнавання музичних патернів 95,4 %, а програма –аналог – 86,3%, тобто достовірність розпізнавання порівняно з аналогом підвищено на 9,1% (95,4 – 86,3=9,1). Тобто мета роботи досягнута - достовірність розпізнавання музичних патернів підвищена на 9,1 %.

5 ЕКОНОМІЧНА ЧАСТИНА

Науково-технічна розробка має право на існування та впровадження, якщо вона відповідає вимогам часу, як в напрямку науково-технічного прогресу та і в плані економіки. Тому для науково-дослідної роботи необхідно оцінювати економічну ефективність результатів виконаної роботи.

Магістерська кваліфікаційна робота з розробки та дослідження «Інформаційна технологія розпізнавання музичних патернів на основі спайкінгової нейронної мережі» відноситься до науково-технічних робіт, які орієнтовані на виведення на ринок (або рішення про виведення науково-технічної розробки на ринок може бути прийнято у процесі проведення самої роботи), тобто коли відбувається так звана комерціалізація науково-технічної розробки. Цей напрямок є пріоритетним, оскільки результатами розробки можуть користуватися інші споживачі, отримуючи при цьому певний економічний ефект. Але для цього потрібно знайти потенційного інвестора, який би взявся за реалізацію цього проекту і переконати його в економічній доцільності такого кроку.

Для наведеного випадку нами мають бути виконані такі етапи робіт:

1) проведено комерційний аудит науково-технічної розробки, тобто встановлення її науково-технічного рівня та комерційного потенціалу;

2) розраховано витрати на здійснення науково-технічної розробки;

3) розрахована економічна ефективність науково-технічної розробки у випадку її впровадження і комерціалізації потенційним інвестором і проведено обґрунтування економічної доцільності комерціалізації потенційним інвестором.

5.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту дослідження за темою «Інформаційна технологія розпізнавання музичних патернів на основі

спайкінгової нейронної мережі» є оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної системи оцінювання за 12-ма критеріями, наведеними в табл. 5.1 [23].

Таблиця 5.1 – Рекомендовані критерії оцінювання науково-технічного рівня і комерційного потенціалу розробки та бальна оцінка

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено працездатність продукту в реальних умовах
Ринкові переваги (недоліки)					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	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
4	Технічні та споживчі властивості продукту значно гірші, ніж в	Технічні та споживчі влас-тивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в	Технічні та споживчі властивості продукту значно кращі, ніж в
5	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					
6	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
7	Активна конкуренція великих компаній на	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає

Продовження таблиці 5.1.

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних фахівців	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
9	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
10	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
11	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
12	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

Результати оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки потрібно звести до таблиці 5.2.

Таблиця 5.2 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

Критерії	Експерт (ПІБ, посада)		
	1	2	3
	Бали:		
1. Технічна здійсненність концепції	5	5	5
2. Ринкові переваги (наявність аналогів)	3	3	4
3. Ринкові переваги (ціна продукту)	4	4	4
4. Ринкові переваги (технічні властивості)	3	4	3
5. Ринкові переваги (експлуатаційні витрати)	2	2	2
6. Ринкові перспективи (розмір ринку)	3	2	3
7. Ринкові перспективи (конкуренція)	3	2	3
8. Практична здійсненність (наявність фахівців)	4	4	4
9. Практична здійсненність (наявність фінансів)	3	3	3
10. Практична здійсненність (необхідність нових матеріалів)	5	5	5
11. Практична здійсненність (термін реалізації)	5	4	4
12. Практична здійсненність (розробка документів)	5	4	5
Сума балів	45	42	45
Середньоарифметична сума балів $СБ_c$	44,0		

За результатами розрахунків, наведених в таблиці 5.2, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в табл. 5.3 [23].

Таблиця 5.3 – Науково-технічні рівні та комерційні потенціали розробки

Середньоарифметична сума балів СБ, розрахована на основі висновків експертів	Науково-технічний рівень та комерційний потенціал розробки
41…48	Високий
31…40	Вище середнього
21…30	Середній
11…20	Нижче середнього
0…10	Низький

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Інформаційна технологія розпізнавання музичних патернів на основі спайкінгової нейронної мережі» становить 44,0 бала, що, відповідно до таблиці

5.3, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки високий).

5.2 Оцінювання рівня новизни розробки

Виводячи на ринок новинку виробник вважає, що тієї новизни, якою наділена нова розробка є достатньо для того, щоб вона була сприйнята споживачем як нова. Але це не завжди так, в силу того, що споживач і виробник неоднозначно визначать її рівень новизни. Тому доцільним є визначення рівня новизни розробки отриманої в результаті досліджень за темою «Інформаційна технологія розпізнавання музичних патернів на основі спайкінгової нейронної мережі».

Саме визначення рівня і ступеня інтегральної новизни є найбільш актуальним, оскільки її рівень визначає ступінь однакового позитивного сприйняття новизни розробки як виробником, так і споживачем, а отже і ринком в цілому, а це, у свою чергу, є гарантією того, що новинка знайде своє місце на ринку, користуватиметься попитом у споживачів і забезпечить відшкодування витрат, зазнаних товаровиробником під час розроблення та виробництва технічної розробки [24].

Рівень новизни нової продукції розраховуємо експертним методом шляхом протиставлення нової продукції та її аналогів, що існують в даний час на ринку, за чинниками що визначають її значення, в системі «краще-гірше». Рівень новизни встановлюємо відносно рівня аналога (або продукту, що досить близький до аналога).

Для визначення *i*-го виду новизни, застосуємо чинники, які впливають на її рівень. Кожен чинник *i*-го виду новизни розраховуємо в балах. Більша кількість набраних балів свідчить про більший рівень новизни. Для оцінювання рівня новизни використаємо думки експертів, які встановлюють визначені бали відповідним чинникам. Бал відповідності проставляється в діапазоні від (-5 –

значно гірше аналога до +5 – значно краще аналога). Результати попереднього оцінювання зведемо до відповідного листа оцінювання (таблиця 5.4).

Таблиця 5.4 – Лист оцінювання рівня новизни експертами

Види та чинники		Бали та експерти		
		Експерт 1	Експерт 2	Експерт 3
1		*2*	*3*	*4*
Споживча новизна	Питома вага 0,24	Максимальний бал $Б_{i\ max}$		25
1. Зміна поведінкових звичок споживача		4	4	4
2. Ступінь задоволення потреб і запитів		5	5	5
3. Спосіб задоволення потреби		3	3	4
4. Формування нової потреби		4	4	4
5. Формування нового споживача		1	1	1
Середній бал експертів $Б_{i\ отр}$		17		
Товарна новизна	Питома вага 0,202	Максимальний бал $Б_{i\ max}$		30
1. Параметричні зміни показників продукції				
1.1. Якісні		4	4	4
1.2. Технічні		4	4	3
1.3. Економічні		3	3	3
1.4. Сервісні		4	4	4
2. Якість продукції по відношенню до конкурентів		3	3	3
3. Функціональні зміни		4	4	4
Середній бал експертів $Б_{i\ отр}$		22		
Виробнича новизна	Питома вага 0,042	Максимальний бал $Б_{i\ max}$		25
1. Рівень унікальності товару для підприємства		5	5	5
2. Рівень унікальності для галузі		3	4	3
3. Рівень унікальності товару для країни		1	1	1
4. Зміна виробничої системи		4	4	4
5. Відносно існуючого асортименту		2	2	2
Середній бал експертів $Б_{i\ отр}$		15		
Прогресивна новизна	Питома вага 0,2	Максимальний бал $Б_{i\ max}$		25
1. Зміна технології виготовлення		4	4	4
2. Рівень застосування нових компонентів і матеріалів		2	2	2
3. Зміна технологічного принципу дії виробу		1	2	1
4. Зміна конструктивного виконання		3	3	3
5. Рівень застосування інновацій		2	2	2
Середній бал експертів $Б_{i\ отр}$		12		

Продовження таблиці 5.4

Види та чинники		Бали та експерти		
		Експерт 1	Експерт 2	Експерт 3
1		*2*	*3*	*4*
Ринкова новизна	Питома вага 0,1	Максимальний бал $Б_{i\ MAX}$		20
1. Новий виріб на новому ринку		0	0	0
2. Новий виріб на відомому ринку		4	4	4
3. Модернізований виріб		2	2	2
4. Нова модель		3	3	3
Середній бал експертів $Б_{i\ отр}$		9		
Екологічна новизна	Питома вага 0,035	Максимальний бал $Б_{i\ MAX}$		20
1. Рівень екологічної чистоти технології виробництва		5	5	5
2. Рівень впровадження мало- та безвідходних технологій		5	5	5
3. Рівень екологічно небезпечних режимів експлуатації продукції		5	5	5
4. Рівень забруднення навколишнього середовища		5	5	5
Середній бал експертів $Б_{i\ отр}$		20		
Соціальна новизна	Питома вага 0,036	Максимальний бал $Б_{i\ MAX}$		20
1. Використання нового товару приводить до покращення стану здоров'я нації		0	0	0
2. Використання нового товару приводить до зростання доходів населення		0	0	0
3. Виробництво нового товару приводить до збільшення (зменшення) кількості робочих місць на підприємстві		4	5	4
4. Виробництво нового товару приводить до підвищення кваліфікації персоналу		3	3	3
Середній бал експертів $Б_{i\ отр}$		7		
Маркетингова новизна	Питома вага 0,145	Максимальний бал $Б_{i\ MAX}$		20
1. Нові методи маркетингових досліджень		0	0	0
2. Вживання нових стратегій сегментації ринку		3	3	3
3. Вибір нової маркетингової стратегії обхвату і розвитку цільового сегмента		2	3	2
4. Побудова нових каналів збуту		2	1	1
Середній бал експертів $Б_{i\ отр}$		7		

Значення *і*-го виду новизни розрахуємо за формулою [24]

$$I_i = \frac{Б_{i\,отр}}{Б_{i\,MAX}}, \qquad (5.1)$$

де $Б_{i\ отр}$ – отримана кількість балів за шкалою оцінок чинників, що визначають i-й вид новизни;

$Б_{i\ max}$ – максимальна кількість балів, що може бути отримана за i-м видом новизни.

Загальний рівень інтегральної новизни розраховуємо шляхом множення отриманого значення i-го виду новизни на її вагомість, причому вагомість i-го виду новизни визначаємо експертним методом, за формулою [24]:

$$N_{інт} = \sum_{i}^{n} W_i \cdot I_i \ , \tag{5.2}$$

де $N_{інт}$ – рівень інтегральної (сукупної) новизни;

W_i – вагомість (питома вага) i-го виду новизни;

n – загальна кількість видів новизни.

$$N_{інт} = (0{,}24 \cdot 17/25)+(0{,}202 \cdot 22/30)+(0{,}042 \cdot 15/25)+(0{,}2 \cdot 12/25)+(0{,}1 \cdot 9/20)+$$
$$(0{,}035 \cdot 20/20)+(0{,}036 \cdot 7/20)+(0{,}145 \cdot 7/20) = 0{,}578.$$

Отримане значення інтегрального рівня новизни зіставляємо зі шкалою, що наведена в табл. 5.5 [23].

Таблиця 5.5 – Рівні новизни нового товару та їхня характеристика

Рівні новизни товару	Значення інтегральної новизни	Характеристика товару	Вид нового товару
Найвища	1,00	Абсолютно новий товар	Новий товар, що наділений ознаками інноваційності (інноваційний товар)
Висока	0,8…0,99	Товар, який не має аналогів	
Значуща	0,6…0,79	Принципова зміна споживчих властивостей товару	
Достатня	0,4…0,59	Принципова технологічна модифікація товару	
Незначна	0,2…0,39	Кардинальна зміна параметрів	Новий товар
Помилкова	0,00…0,19	Малоістотна модифікація	

Згідно таблиці 5.5 розробка відповідає рівню при значенні інтегральної новизни 0,578 - достатня новизна; за характеристикою: принципова технологічна модифікація товару; вид розробки - новий товар, що наділений ознаками інноваційності (інноваційний товар).

5.3 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов'язані з проведенням науково-дослідної роботи на тему «Інформаційна технологія розпізнавання музичних патернів на основі спайкінгової нейронної мережі», під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

5.3.1 Витрати на оплату праці

До статті «Витрати на оплату праці» належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці.

Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників ($З_о$) розраховуємо у відповідності до посадових окладів працівників, за формулою [23]:

$$З_о = \sum_{i=1}^{k} \frac{M_{ni} \cdot t_i}{T_p}, \qquad (5.3)$$

де k – кількість посад дослідників залучених до процесу досліджень;

M_{ni} – місячний посадовий оклад конкретного дослідника, грн;

t_i – число днів роботи конкретного дослідника, дн.;

T_p – середнє число робочих днів в місяці, T_p=24 дні.

$$З_о = 16100{,}00 \cdot 72 / 24 = 48300{,}00 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці 5.6.

Таблиця 5.6 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Керівник дослідження	16100,00	670,83	72	48300,00
Науковий співробітник	14250,00	593,75	60	35625,00
Інженер-розробник програмного забезпечення	14100,00	587,50	60	35250,00
Технік	7050,00	293,75	20	5875,00
Всього				125050,00

Основна заробітна плата робітників

Витрати на основну заробітну плату робітників ($З_р$) за відповідними найменуваннями робіт НДР на тему «Інформаційна технологія розпізнавання музичних патернів на основі спайкінгової нейронної мережі» розраховуємо за формулою:

$$З_р = \sum_{i=1}^{n} C_i \cdot t_i , \qquad (5.4)$$

де C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

t_i – час роботи робітника при виконанні визначеної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду C_i можна визначити за формулою:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}} , \qquad (5.5)$$

де M_M – розмір прожиткового мінімуму працездатної особи, або мінімальної місячної заробітної плати (в залежності від діючого законодавства), приймемо M_M=6700,00 грн;

K_i – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду (табл. Б.2, додаток Б) [23];

K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

T_p – середнє число робочих днів в місяці, приблизно T_p = 24 дн;

$t_{зм}$ – тривалість зміни, год.

$C_1 = 6700{,}00 \cdot 1{,}50 \cdot 1{,}7 / (24 \cdot 8) = 88{,}98$ *грн.*
$З_{р1} = 88{,}98 \cdot 4{,}50 = 400{,}43$ *грн.*

Таблиця 5.7 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
Підготовка обладнання розпізнавання звукових сигналів	4,50	4	1,50	88,98	400,43
Підготовка комп'ютерного обладнання розробника програмного забезпечення	5,20	4	1,50	88,98	462,72
Монтаж приймачів звукових сигналів	3,00	5	1,70	100,85	302,55
Інсталяція програмного забезпечення моделювання звукових фрагментів	5,00	4	1,50	88,98	444,92
Компіляція програмних блоків розпізнавання звукових команд	16,00	3	1,35	80,09	1281,38
Налагодження програмних блоків	6,00	4	1,50	88,98	533,91
Формування бази даних	25,00	3	1,35	80,09	2002,15
Тренування нейронної мережі	8,00	3	1,35	80,09	640,69
Всього					6068,73

Додаткова заробітна плата дослідників та робітників

Додаткову заробітну плату розраховуємо як 10 … 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$З_{дод} = (З_о + З_р) \cdot \frac{Н_{дод}}{100\%}, \quad (5.6)$$

де $Н_{дод}$ – норма нарахування додаткової заробітної плати. Приймемо 10%.

$З_{дод}$ = (125050,00 + 6068,73) · 10 / 100% = 13111,87 грн.

5.3.2 Відрахування на соціальні заходи

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$З_н = (З_о + З_р + З_{дод}) \cdot \frac{Н_{зп}}{100\%} \quad (5.7)$$

де $Н_{зп}$ – норма нарахування на заробітну плату. Приймаємо 22%.

Зн = (125050,00 + 6068,73 + 13111,87) · 22 / 100% = 31730,73 грн.

5.3.3 Сировина та матеріали

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби і предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за темою «Інформаційна технологія розпізнавання музичних патернів на основі спайкінгової нейронної мережі».

Витрати на матеріали (*М*), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою:

$$M = \sum_{j=1}^{n} H_j \cdot Ц_j \cdot К_j - \sum_{j=1}^{n} B_j \cdot Ц_{вj}, \quad (5.8)$$

де H_j – норма витрат матеріалу j-го найменування, кг;

n – кількість видів матеріалів;

$Ц_j$ – вартість матеріалу j-го найменування, грн/кг;

$К_j$ – коефіцієнт транспортних витрат, ($К_j = 1,1 \ldots 1,15$);

B_j – маса відходів j-го найменування, кг;

$Ц_{вj}$ – вартість відходів j-го найменування, грн/кг.

$M_1 = 4,0 \cdot 265,00 \cdot 1,12 - 0 \cdot 0 = 1187,20$ *грн.*

Проведені розрахунки зведемо до таблиці 5.8.

Таблиця 5.8 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Величина відходів, кг	Ціна відходів, грн/кг	Вартість витраченого матеріалу, грн
Офісний папір CARBONIX Ultra	265,00	4,0	0	0	1187,20
Папір для записів CARBONIX Light A5	130,00	3,0	0	0	436,80
Органайзер офісний CARBONIX OFFICE	200,00	4,0	0	0	896,00
Канцелярське приладдя (набір офісного працівника)	260,00	4,0	0	0	1164,80
Картридж для принтера Canon LBP6000	1870,00	1,0	0	0	2094,40
Диск оптичний NewVybir CD-R	23,00	3,0	0	0	77,28
Flesh-пам'ять Kingston 32 GB	352,00	1,0	0	0	394,24
Тека для паперів CARBONIX BOX-ZX	135,00	3,0	0	0	453,60
Всього					6704,32

5.3.4 Розрахунок витрат на комплектуючі

Витрати на комплектуючі *($К_в$)*, які використовують при проведенні НДР на тему «Інформаційна технологія розпізнавання музичних патернів на основі спайкінгової нейронної мережі», розраховуємо, згідно з їхньою номенклатурою, за формулою:

$$К_в = \sum_{j=1}^{n} Н_j \cdot Ц_j \cdot К_j \qquad (5.9)$$

де $Н_j$ – кількість комплектуючих *j*-го виду, шт.;

$Ц_j$ – покупна ціна комплектуючих *j*-го виду, грн;

$К_j$ – коефіцієнт транспортних витрат, ($К_j$ *= 1,1 ... 1,15*).

$К_в = 1 \cdot 1850{,}00 \cdot 1{,}12 = 2072{,}00$ *грн.*

Проведені розрахунки зведемо до таблиці 5.9.

Таблиця 5.9 – Витрати на комплектуючі

Найменування комплектуючих	Кількість, шт.	Ціна за штуку, грн	Сума, грн
Зовнішня звукова карта Behringer U-PHORIA UMC404HD	1	1850,00	2072,00
Плата Arduino Uno	1	3120,00	3494,40
Кабель USB AM-BM для Arduino	1	415,00	464,80
Апаратний інтерфейс	8	123,00	1102,08
Декодер DTMF MT8870	1	94,50	105,84
Всього			7239,12

5.3.5 Спецустаткування для наукових (експериментальних) робіт

До статті «Спецустаткування для наукових (експериментальних) робіт» належать витрати на виготовлення та придбання спецустаткування необхідного для проведення досліджень, також витрати на їх проектування, виготовлення, транспортування, монтаж та встановлення. Витрати на виготовлення та придбання спецустаткування необхідного для проведення досліджень відсутні.

5.3.6 Програмне забезпечення для наукових (експериментальних) робіт

До статті «Програмне забезпечення для наукових (експериментальних) робіт» належать витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення, (програм, алгоритмів, баз даних) необхідних для проведення досліджень, також витрати на їх проектування, формування та встановлення.

Балансову вартість програмного забезпечення розраховуємо за формулою:

$$B_{прг} = \sum_{i=1}^{k} Ц_{iпрг} \cdot C_{прг.i} \cdot K_i \ , \qquad (5.10)$$

де $Ц_{iпрг}$ – ціна придбання одиниці програмного засобу даного виду, грн;

$C_{прг.i}$ – кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

K_i – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, ($K_i = 1,10…1,12$);

k – кількість найменувань програмних засобів.

$B_{прг} = 8410,00 \cdot 1 \cdot 1,12 = 9419,20$ *грн.*

Отримані результати зведемо до таблиці 5.10:

Таблиця 5.10 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
ОС Windows	1	8410,00	9419,20
Прикладний пакет Microsoft Office	1	7770,00	8702,40
Прикладний пакет обробки та аналізу звукових сигналів	1	7560,00	8467,20
Прикладний пакет мови програмування Python	1	4800,00	5376,00
TensorFlow	1	1320,00	1478,40
Бібліотека Numpy	1	3200,00	3584,00
Всього			37027,20

5.3.7 Амортизація обладнання, програмних засобів та приміщень

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, розраховуємо з використанням прямолінійного методу амортизації за формулою:

$$A_{обл} = \frac{Ц_б}{T_в} \cdot \frac{t_{вик}}{12}, \quad (5.11)$$

де $Ц_б$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{вик}$ – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$T_в$ – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$A_{обл} = (31250{,}00 \cdot 3) / (3 \cdot 12) = 2604{,}17$ *грн.*

Проведені розрахунки зведемо до таблиці 5.11.

Таблиця 5.11 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Персональний комп'ютер проектувальника EVEREST E5000	31250,00	3	3	2604,17
Портативний перенос-ний комп'ютер типу Acer Nitro 5 AN515-57-51H7 (NH.QEKEU.002) Black	35600,00	3	3	2966,67
Робоче місце інженера-програміста	9400,00	5	3	470,00
Пристрій виводу інф-мації	6800,00	5	3	340,00
Оргтехніка	9340,00	25	3	93,40
Приміщення лабораторії	326000,00	25	3	3260,00
Всього				9734,23

5.3.8 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію (Ве) розраховуємо за формулою:

$$B_e = \sum_{i=1}^{n} \frac{W_{yi} \cdot t_i \cdot Ц_e \cdot K_{вni}}{\eta_i}, \quad (5.12)$$

де W_{yi} – встановлена потужність обладнання на визначеному етапі розробки, кВт;

t_i – тривалість роботи обладнання на етапі дослідження, год;

$Ц_e$ – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), приймемо $Ц_e$ = 6,20 грн;

$K_{вni}$ – коефіцієнт, що враховує використання потужності, $K_{вni} < 1$;

η_i – коефіцієнт корисної дії обладнання, $\eta_i < 1$.

$B_e = 0,32 \cdot 480,0 \cdot 6,20 \cdot 0,95 / 0,97 = 952,32$ *грн.*

Проведені розрахунки зведемо до таблиці 5.12.

Таблиця 5.12 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Персональний комп'ютер проектувальника EVEREST E5000	0,32	480,0	952,32
Портативний переносний комп'ютер типу Acer Nitro 5 AN515-57-51H7 (NH.QEKEU.002) Black	0,05	400,0	124,00
Робоче місце інженера-програміста	0,10	400,0	248,00
Пристрій виводу інформації	0,50	9,0	27,90
Оргтехніка	0,50	5,0	15,50
Зовнішня звукова карта Behringer U-PHORIA UMC404HD	0,02	200,0	24,80
Всього			1392,52

5.3.9 Службові відрядження

До статті «Службові відрядження» дослідної роботи на тему «Інформаційна технологія розпізнавання музичних патернів на основі спайкінгової нейронної мережі» належать витрати на відрядження штатних працівників, працівників організацій, які працюють за договорами цивільно-правового характеру, аспірантів, зайнятих розробленням досліджень, відрядження, пов'язані з проведенням випробувань машин та приладів, а також витрати на відрядження на наукові з'їзди, конференції, наради, пов'язані з виконанням конкретних досліджень.

Витрати за статтею «Службові відрядження» розраховуємо як 20…25% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{св} = (З_о + З_р) \cdot \frac{Н_{св}}{100\%}, \quad (5.13)$$

де $Н_{св}$ – норма нарахування за статтею «Службові відрядження», приймемо $Н_{св}$ = 25%.

$B_{св}$ = (125050,00 + 6068,73) · 25 / 100% = 32779,68 грн.

5.3.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації

Витрати за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації» розраховуємо як 30…45% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{сп} = (З_о + З_р) \cdot \frac{Н_{сп}}{100\%}, \quad (5.14)$$

де $Н_{сп}$ – норма нарахування за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації», приймемо $Н_{сп}$= 40%.

$В_{сп}$ = (125050,00 + 6068,73) · 40 / 100% = 52447,49 грн.

5.3.11 Інші 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» розраховуємо як 50…100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_в = (З_о + З_р) \cdot \frac{Н_{ів}}{100\%}, \qquad (5.15)$$

де $Н_{ів}$ – норма нарахування за статтею «Інші витрати», приймемо $Н_{ів}$ = 55%.

$І_в$ = (125050,00 + 6068,73) · 55 / 100% = 72115,30 грн.

5.3.12 Накладні (загальновиробничі) витрати

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» розраховуємо як 100…150% від суми основної заробітної плати дослідників та робітників за формулою:

$$В_{нзв} = (З_о + З_р) \cdot \frac{Н_{нзв}}{100\%}, \qquad (5.16)$$

де $Н_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати», приймемо $Н_{нзв}$ = 125%.

$В_{нзв}$ = *(125050,00 + 6068,73) · 125 / 100% = 163898,42 грн.*

Витрати на проведення науково-дослідної роботи на тему «Інформаційна технологія розпізнавання музичних патернів на основі спайкінгової нейронної мережі» розраховуємо як суму всіх попередніх статей витрат за формулою:

$$В_{заг} = З_о + З_р + З_{дод} + З_н + М + К_в + В_{спец} + В_{прг} + А_{обл} + В_е + В_{св} + В_{сп} + І_в + В_{нзв}. \quad (5.17)$$

$В_{заг}$= *125050,00 +6068,73 +13111,87 +31730,73372 +6704,32 +7239,12 +0,00 + 37027,20 + 9734,23 +1392,52 +32779,68 +52447,49 +72115,30 +163898,42 = 559299,63 грн.*

Загальні витрати *ЗВ* на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховується за формулою:

$$ЗВ = \frac{В_{заг}}{\eta}, \quad (5.18)$$

де η - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, приймемо η=0,9.

ЗВ = 559299,63 / 0,9 = 621444,04 грн.

5.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів

тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку.

Результати дослідження проведені за темою «Інформаційна технологія розпізнавання музичних патернів на основі спайкінгової нейронної мережі» передбачають комерціалізацію протягом 4-х років реалізації на ринку.

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

ΔN – збільшення кількості споживачів продукту, у періоди часу, що аналізуються, від покращення його певних характеристик;

Показник	1-й рік	2-й рік	3-й рік	4-й рік
Збільшення кількості споживачів, осіб	1000	1750	2000	1750

N – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки, приймемо 15200 осіб;

$Ц_б$ – вартість програмного продукту у році до впровадження результатів розробки, приймемо 6400,00 грн;

$\pm\Delta Ц_o$ – зміна вартості програмного продукту від впровадження результатів науково-технічної розробки, приймемо 60,00 грн.

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta П_i$ для кожного із 4-х років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою [23]:

$$\Delta П_i = (\pm\Delta Ц_o \cdot N + Ц_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot (1 - \frac{\vartheta}{100}), \qquad (5.19)$$

де λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2022 році ставка податку на додану вартість складає 20%, а коефіцієнт $\lambda = 0,8333$;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту). Приймемо $\rho = 45\%$;

ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2022 році $\vartheta = 18\%$;

Збільшення чистого прибутку 1-го року:

$\Delta П_1 = (60,00 \cdot 15200,00 + 6460,00 \cdot 1000) \cdot 0,83 \cdot 0,45 \cdot (1 - 0,18/100\%) = 2257822,44$ грн.

Збільшення чистого прибутку 2-го року:

$\Delta П_2 = (60,00 \cdot 15200,00 + 6460,00 \cdot 2750) \cdot 0,83 \cdot 0,45 \cdot (1 - 0,18/100\%) = 5720204,79$ грн.

Збільшення чистого прибутку 3-го року:

$\Delta П_3 = (60,00 \cdot 15200,00 + 6460,00 \cdot 4750) \cdot 0,83 \cdot 0,45 \cdot (1 - 0,18/100\%) = 9677213,19$ грн.

Збільшення чистого прибутку 4-го року:

$\Delta П_4 = (60,00 \cdot 15200,00 + 6460,00 \cdot 6500) \cdot 0,83 \cdot 0,45 \cdot (1 - 0,18/100\%) = 13139595,54$ грн.

Приведена вартість збільшення всіх чистих прибутків *ПП*, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{i=1}^{T} \frac{\Delta П_i}{(1+\tau)^t}, \qquad (5.20)$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

T – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,15$;

t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$ПП = 2257822,44/(1+0,15)^1+5720204,79/(1+0,15)^2+9677213,19/(1+0,15)^3+$

$+13139595,54/(1+0,15)^4=1963323,86+4325296,63+6362924,76+7512606,40=20164$

151,64 грн.

Величина початкових інвестицій PV, які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки:

$$PV = k_{інв} \cdot ЗВ, \tag{5.21}$$

де $k_{інв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо $k_{інв}$=2;

$ЗВ$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 621444,04 грн.

$PV = k_{інв} \cdot ЗВ$ = 2 · 621444,04 = 1242888,08 грн.

Абсолютний економічний ефект $E_{абс}$ для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = ПП - PV \tag{5.22}$$

де $ПП$ – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 20164151,64 грн;

PV – теперішня вартість початкових інвестицій, 1242888,08 грн.

$E_{абс} = ПП - PV$ = 20164151,64 - 1242888,08 = 18921263,57 грн.

Внутрішня економічна дохідність інвестицій $E_в$, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$E_в = \sqrt[Тж]{1 + \frac{E_{абс}}{PV}} - 1, \quad (5.23)$$

де $E_{абс}$ – абсолютний економічний ефект вкладених інвестицій, 18921263,57 грн;

PV – теперішня вартість початкових інвестицій, 1242888,08 грн;

$T_ж$ – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримування позитивних результатів від її впровадження, 4 роки.

$$E_в = \sqrt[Тж]{1 + \frac{E_{абс}}{PV}} - 1 = (1+18921263,57/1242888,08)^{1/4} = 1,01.$$

Мінімальна внутрішня економічна дохідність вкладених інвестицій $\tau_{мін}$:

$$\tau_{мін} = d + f, \quad (5.24)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2022 році в Україні d =0,12;

f – показник, що характеризує ризикованість вкладення інвестицій, приймемо 0,4.

$\tau_{мін}$ = 0,12+0,4 = 0,52 < 1,01 свідчить про те, що внутрішня економічна дохідність інвестицій $E_в$, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну роботу за темою «Інформаційна технологія розпізнавання музичних патернів на основі спайкінгової нейронної мережі» доцільно.

Період окупності інвестицій $T_{ок}$ які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_в}, \quad (5.25)$$

де $E_в$ – внутрішня економічна дохідність вкладених інвестицій.

$T_{ок}$ = 1 / 1,01 = 0,99 р.

$T_{ок} <$ 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

5.5 Висновок до розділу 5

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Інформаційна технологія розпізнавання музичних патернів на основі спайкінгової нейронної мережі» становить 44,0 бала, що свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки високий). Також термін окупності становить 0,99 р., що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок. Отже можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Інформаційна технологія розпізнавання музичних патернів на основі спайкінгової нейронної мережі».

ВИСНОВКИ

При виконанні магістерської кваліфікаційної роботи розв’язано задачу розробки інформаційної технології та програмного забезпечення розв'язання задачі розпізннання музичних патернів на основі спайкінгової нейронної мережі.

У першому розділі магістерської роботи було проведено аналіз предметної області розпізнавання музичних патернів, а саме – сформульовано постановку задачі, проведено огляд відомих методів розв'язання задачі розпізнавання музичних патернів. Найперспективнішим було обрано нейромережевий метод. Також було проведено аналіз програмних засобів розпізнавання музичних патернів та обгрунтовано вибір аналогу. Недоліком відомих програм є низька достовірність розпізнавання музичних патернів, звідки і випливає мета даної роботи – підвищення достовірності програмних засобів розпізнавання музичних патернів.

У другому розділі магістерської кваліфікаційної роботи було обґрунтовано вибір типу нейронної мережі для задачі розпізнавання музичних патернів – спайкінгова нейромережа. Була розроблена структура спайкінгової нейронної мережі 8-1500-500, проаналізовано види кодування інформації, порядок функціонування багатошарової спайкінгової нейронної мережі та математична модель спайкінгового нейрона. Було розроблено метод розпізнавання музичних патернів та структуру інформаційної технології розпізнавання музичних патернів на основі спайкінгової нейронної мережі.

У третьому розділі було обґрунтовано вибір мови програмування Python та середовища програмування PyCharm для задачі розпізнавання музичних патернів. Була описано використання бібліотек TensorFlow та NumPy для створення програмної реалізації спайкінгової нейронної мережі та її навчання, а також модуля PyAudio для прийняття та обробки звукових файлів і передачі їх на вхід нейронній мережі.

У четвертому розділі в результаті тестування програми було доведено її працездатність та відповідність поставленому завданню. Тестування нейронної мережі проходило на вибірці з 1000 музичних композицій. Розроблена програма має достовірність розпізнавання музичних патернів 95,4 %, а програма –аналог – 86,3%, тобто достовірність розпізнавання порівняно з аналогом підвищено на 9,1% (95,4 – 86,3=9,1). Тобто мета роботи досягнута - достовірність розпізнавання музичних патернів підвищена на 9,1 %.

У п`ятому розділі визначено, що згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Інформаційна технологія розпізнавання музичних патернів на основі спайкінгової нейронної мережі» становить 44,0 бала, що свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки високий). Також термін окупності становить 0,99 р., що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок. Отже можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Інформаційна технологія розпізнавання музичних патернів на основі спайкінгової нейронної мережі».

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Є. О. Чоботар, І. К. Денисов, О. К. Колесницький Програмний модуль розпізнавання музичних патернів на основі спайкінгової нейронної мережі, в Матеріали конференції «L Науково-технічна конференція підрозділів Вінницького національного технічного університету (2021)», Вінниця, 2021. С.549-550 [Електронний ресурс]. Режим доступу: https://conferences.vntu.edu.ua/index.php/allvntu/index/pages/view/zbirn2021 Дата звернення: Черв. 2021

2. Арсенюк І. Р. Перспективні підходи до розв'язання задачі автоматизованого транскрибування музичних композицій / І. Р. Арсенюк, В. Ю. Кучеровський // Матеріали X Міжнародної науково-практичної конференції "Інтернет-Освіта-Наука" (ІОН-2016). – Вінниця: ВНТУ, 2016. – С. 45 – 47.

3. Арсенюк І. Р. Застосування сучасних методів розпізнавання мови для вирішення задачі автоматизованого транскрибування музичних композицій / І. Р. Арсенюк, В. Ю. Кучеровський // Матеріали XLVI науково-технічної конференції підрозділів ВНТУ, Вінниця, 2017. http://ir.lib.vntu.edu.ua/bitstream/handle/123456789/17217/2232.pdf?sequence=3

4. Altman, Naomi S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician 46 (3): 175–185. doi:10.1080/00031305.1992.10475879

5. Любунь З. М. Основи теорії нейромереж: текст лекцій / З. М. Любунь. – Львів : Видавничий центр ЛНУ ім. Івана Франка, 2006.

6. Гудфеллоу Я. Глубокое обучение / пер. с анг. А. А. Слинкина. 2-е изд., испр. М.: ДМК Пресс, 2018. 652 с.: цв. ил.

7. Shazam [Електронний ресурс]. Режим доступу: https://www.shazam.com/ru /

8. Tunatic [Електронний ресурс]. Режим доступу: http://tunatic.ru/

9. Хайкин С. Нейронные сети: полный курс / С. Хайкин – Издание 2.: Пер. с англ. – М.: Издательский дом «Вильямс», 2006. - 1104 с.

10. Круг П. Г. Нейронные сети и нейрокомпьютеры: учебное пособие / П. Г. Круг. — М. : Издательство МЭИ, 2002. — 176 с.

11.Руденко О.В. Штучні нейронні мережі: Навчальний посібник / О.В.Руденко, Є.В.Бодянський. - Харків : ТОВ «Компанія СМІТ», 2006. — 404 с. - ISBN 966-8630-73-X.

12.V. P. Kozemiako ; O. K. Kolesnytskyj ; T. S. Lischenko ; W. Wojcik and A. Sulemenov " Optoelectronic spiking neural network ", Proc. SPIE 8698, Optical Fibers and Their Applications 2012, 86980M (January 11, 2013); doi:10.1117/12.2019340; http://dx.doi.org/10.1117/12.2019340

13.Neurocomputer architecture based on spiking neural network and its optoelectronic implementation / Oleh K. Kolesnytskyj; Vladislav V. Kutsman; Krzysztof Skorupski; Mukaddas Arshidinova, Proc. SPIE 11176, Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2019, 1117609 (6 November 2019); doi: 10.1117/12.2536607

14.Колесницкий О. К. Метод распознавания многомерных временных рядов при помощи импульсных нейронных сетей / О. К. Колесницкий, Самра Муавия Хассан Хамо // Інформаційні технології та комп‘ютерна інженерія, 2006, - №2(6), С. 86-93.

15. Maass W. Real-time computing without stable states: A new framework for neural computation based on perturbations / W. Maass, T. Natschläger, H. Markram // Neural Computation. — 2002. — Vol. 14(11). — P. 2531—2560.

16.O. K. Kolesnytskyj, I. V. Bokotsey, S. S. Yaremchuk Optoelectronic Implementation of Pulsed Neurons and Neural Networks Using Bispin-Devices // Optical Memory & Neural Networks (Information Optics), 2010, Vol.19, №2, pp.154-165.

17.AudioSet - A large-scale dataset of manually annotated audio events [Електронний ресурс]. Режим доступу: https://research.google.com/audioset/

18. Google\Youtube-8M [Електронний ресурс]. Режим доступу: https://github.com/google/youtube-8m

19. S. B. Shrestha and Q. Song, "Robustness to Training Disturbances in SpikeProp Learning," in IEEE Transactions on Neural Networks and Learning Systems, vol. 29, no. 7, pp. 3126-3139, July 2018, doi: 10.1109/TNNLS.2017.2713125

20.Python [Електронний ресурс]. – Режим доступу: https://www.python.org/.

21. PyCharm - IDE для профессиональной разработки на Python [Електронний ресурс]. Режим доступу: https://www.jetbrains.com/ru-ru/pycharm/

22.Tensorflow [Електронний ресурс]. – Режим доступу: //www.tensorflow.org/.

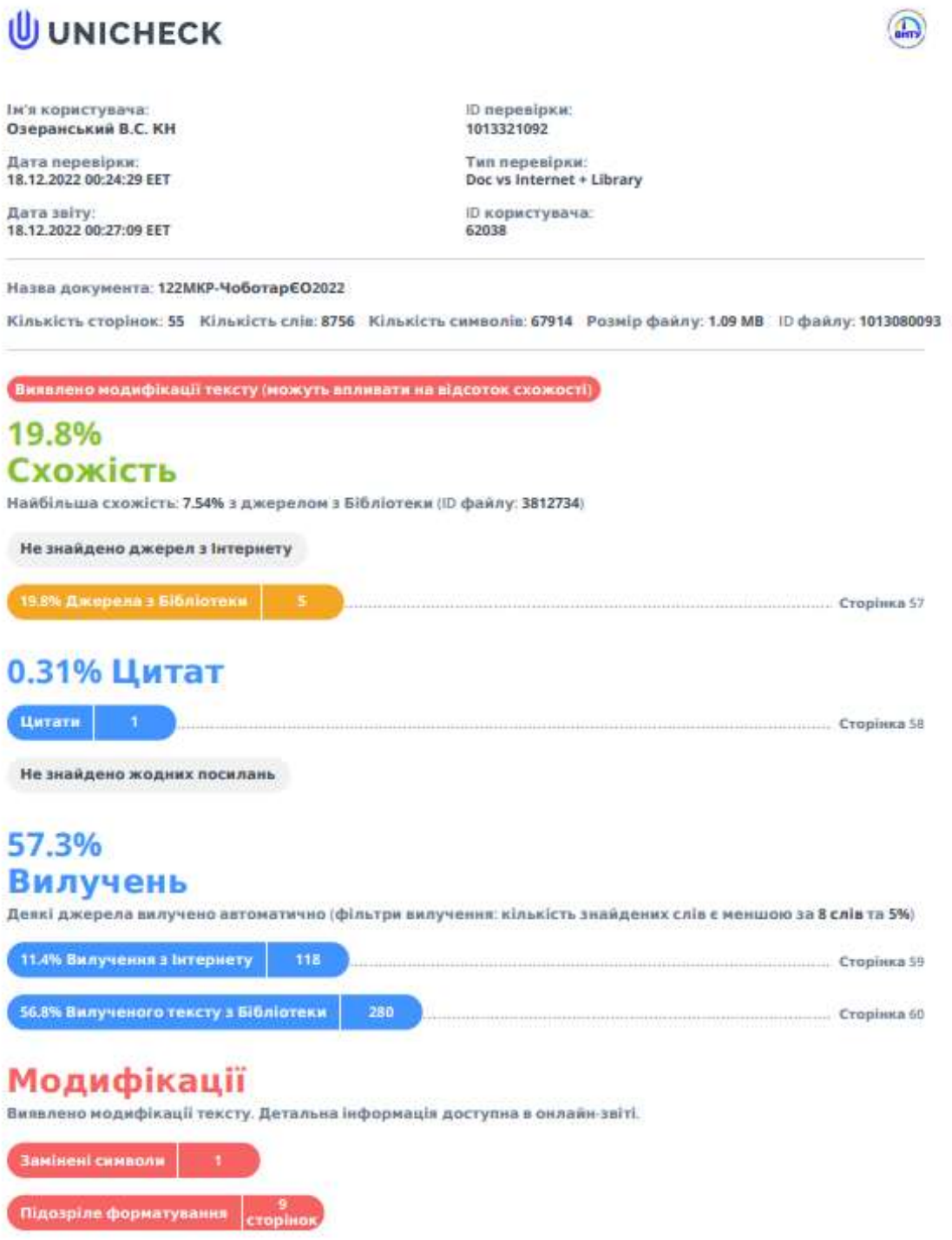
23.Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. – Вінниця : ВНТУ, 2021. – 42 с.

24.Кавецький В. В. Економічне обґрунтування інноваційних рішень: практикум / В. В. Кавецький, В. О. Козловський, І. В. Причепа – Вінниця : ВНТУ, 2016. – 113 с.

ДОДАТКИ

Додаток А (обов'язковий)

Результат перевірки на плагіат в онлайн-системі UNICHECK



UNICHECK

Ім'я користувача:
Озеранський В.С. КН

ID перевірки:
1013321092

Дата перевірки:
18.12.2022 00:24:29 EET

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
18.12.2022 00:27:09 EET

ID користувача:
62038

Назва документа: 122МКР-ЧоботарЄО2022

Кількість сторінок: 55 Кількість слів: 8756 Кількість символів: 67914 Розмір файлу: 1.09 MB ID файлу: 1013080093

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

19.8%
Схожість

Найбільша схожість: 7.54% з джерелом з Бібліотеки (ID файлу: 3812734)

Не знайдено джерел з Інтернету

19.8% Джерела з Бібліотеки 5 ..... Сторінка 57

0.31% Цитат

Цитати 1 ..... Сторінка 58

Не знайдено жодних посилань

57.3%
Вилучень

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

11.4% Вилучення з Інтернету 118 ..... Сторінка 59

56.8% Вилученого тексту з Бібліотеки 280 ..... Сторінка 60

Модифікації

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

Замінені символи 1

Підозріле форматування 9 сторінок

Додаток Б (обов'язковий)

Лістинг програми

parse_file.py

```
# Copyright (C) 2017 DataArt
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

import argparse
import numpy as np
from scipy.io import wavfile

parser = argparse.ArgumentParser(description='Read file and process audio')
parser.add_argument('wav_file', type=str, help='File to read and process')


def process_file(wav_file):
    sr, data = wavfile.read(wav_file)
    if data.dtype != np.int16:
        raise TypeError('Bad sample type: %r' % data.dtype)

    # local import to reduce start-up time
    from audio.processor import WavProcessor, format_predictions

    with WavProcessor() as proc:
        predictions = proc.get_predictions(sr, data)

    print(format_predictions(predictions))


if __name__ == '__main__':
    args = parser.parse_args()
    process_file(**vars(args))
```

class_labels_indices.csv

```
index,mid,display_name
0,/m/09x0r,"Speech"
1,/m/05zppz,"Male speech, man speaking"
2,/m/02zsn,"Female speech, woman speaking"
```

3,/m/0ytgt,"Child speech, kid speaking"
4,/m/01h8n0,"Conversation"
5,/m/02qldy,"Narration, monologue"
6,/m/0261r1,"Babbling"
7,/m/0brhx,"Speech synthesizer"
8,/m/07p6fty,"Shout"
9,/m/07q4ntr,"Bellow"
10,/m/07rwj3x,"Whoop"
11,/m/07sr1lc,"Yell"
12,/m/04gy_2,"Battle cry"
13,/t/dd00135,"Children shouting"
14,/m/03qc9zr,"Screaming"
15,/m/02rtxlg,"Whispering"
16,/m/01j3sz,"Laughter"
17,/t/dd00001,"Baby laughter"
18,/m/07r660_,"Giggle"
19,/m/07s04w4,"Snicker"
20,/m/07sq110,"Belly laugh"
21,/m/07rgt08,"Chuckle, chortle"
22,/m/0463cq4,"Crying, sobbing"
23,/t/dd00002,"Baby cry, infant cry"
24,/m/07qz6j3,"Whimper"
25,/m/07qw_06,"Wail, moan"
26,/m/07plz5l,"Sigh"
27,/m/015lz1,"Singing"
28,/m/0l14jd,"Choir"
29,/m/01swy6,"Yodeling"
30,/m/02bk07,"Chant"
31,/m/01c194,"Mantra"
32,/t/dd00003,"Male singing"
33,/t/dd00004,"Female singing"
34,/t/dd00005,"Child singing"
35,/t/dd00006,"Synthetic singing"
36,/m/06bxc,"Rapping"
37,/m/02fxyj,"Humming"
38,/m/07s2xch,"Groan"
39,/m/07r4k75,"Grunt"
40,/m/01w250,"Whistling"
41,/m/0lyf6,"Breathing"
42,/m/07mzm6,"Wheeze"
43,/m/01d3sd,"Snoring"
44,/m/07s0dtb,"Gasp"
45,/m/07pyy8b,"Pant"
46,/m/07q0yl5,"Snort"
47,/m/01b_21,"Cough"
48,/m/0dl9sf8,"Throat clearing"
49,/m/01hsr_,"Sneeze"
50,/m/07ppn3j,"Sniff"
51,/m/06h7j,"Run"
52,/m/07qv_x_,"Shuffle"
53,/m/07pbtc8,"Walk, footsteps"
54,/m/03cczk,"Chewing, mastication"

55,/m/07pdhp0,"Biting"
56,/m/0939n_,"Gargling"
57,/m/01g90h,"Stomach rumble"
58,/m/03q5_w,"Burping, eructation"
59,/m/02p3nc,"Hiccup"
60,/m/02_nn,"Fart"
61,/m/0k65p,"Hands"
62,/m/025_jnm,"Finger snapping"
63,/m/0l15bq,"Clapping"
64,/m/01jg02,"Heart sounds, heartbeat"
65,/m/01jg1z,"Heart murmur"
66,/m/053hz1,"Cheering"
67,/m/028ght,"Applause"
68,/m/07rkbfh,"Chatter"
69,/m/03qtwd,"Crowd"
70,/m/07qfr4h,"Hubbub, speech noise, speech babble"
71,/t/dd00013,"Children playing"
72,/m/0jbk,"Animal"
73,/m/068hy,"Domestic animals, pets"
74,/m/0bt9lr,"Dog"
75,/m/05tny_,"Bark"
76,/m/07r_k2n,"Yip"
77,/m/07qf0zm,"Howl"
78,/m/07rc7d9,"Bow-wow"
79,/m/0ghcn6,"Growling"
80,/t/dd00136,"Whimper (dog)"
81,/m/01yrx,"Cat"
82,/m/02yds9,"Purr"
83,/m/07qrkrw,"Meow"
84,/m/07rjwbb,"Hiss"
85,/m/07r81j2,"Caterwaul"
86,/m/0ch8v,"Livestock, farm animals, working animals"
87,/m/03k3r,"Horse"
88,/m/07rv9rh,"Clip-clop"
89,/m/07q5rw0,"Neigh, whinny"
90,/m/01xq0k1,"Cattle, bovinae"
91,/m/07rpkh9,"Moo"
92,/m/0239kh,"Cowbell"
93,/m/068zj,"Pig"
94,/t/dd00018,"Oink"
95,/m/03fwl,"Goat"
96,/m/07q0h5t,"Bleat"
97,/m/07bgp,"Sheep"
98,/m/025rv6n,"Fowl"
99,/m/09b5t,"Chicken, rooster"
100,/m/07st89h,"Cluck"
101,/m/07qn5dc,"Crowing, cock-a-doodle-doo"
102,/m/01rd7k,"Turkey"
103,/m/07svc2k,"Gobble"
104,/m/09ddx,"Duck"
105,/m/07qdb04,"Quack"
106,/m/0dbvp,"Goose"

107,/m/07qwf61,"Honk"
108,/m/01280g,"Wild animals"
109,/m/0cdnk,"Roaring cats (lions, tigers)"
110,/m/04cvmfc,"Roar"
111,/m/015p6,"Bird"
112,/m/020bb7,"Bird vocalization, bird call, bird song"
113,/m/07pggtn,"Chirp, tweet"
114,/m/07sx8x_,"Squawk"
115,/m/0h0rv,"Pigeon, dove"
116,/m/07r_25d,"Coo"
117,/m/04s8yn,"Crow"
118,/m/07r5c2p,"Caw"
119,/m/09d5_,"Owl"
120,/m/07r_80w,"Hoot"
121,/m/05_wcq,"Bird flight, flapping wings"
122,/m/01z5f,"Canidae, dogs, wolves"
123,/m/06hps,"Rodents, rats, mice"
124,/m/04rmv,"Mouse"
125,/m/07r4gkf,"Patter"
126,/m/03vt0,"Insect"
127,/m/09xqv,"Cricket"
128,/m/09f96,"Mosquito"
129,/m/0h2mp,"Fly, housefly"
130,/m/07pjwq1,"Buzz"
131,/m/01h3n,"Bee, wasp, etc."
132,/m/09ld4,"Frog"
133,/m/07st88b,"Croak"
134,/m/078jl,"Snake"
135,/m/07qn4z3,"Rattle"
136,/m/032n05,"Whale vocalization"
137,/m/04rlf,"Music"
138,/m/04szw,"Musical instrument"
139,/m/0fx80y,"Plucked string instrument"
140,/m/0342h,"Guitar"
141,/m/02sgy,"Electric guitar"
142,/m/018vs,"Bass guitar"
143,/m/042v_gx,"Acoustic guitar"
144,/m/06w87,"Steel guitar, slide guitar"
145,/m/01glhc,"Tapping (guitar technique)"
146,/m/07s0s5r,"Strum"
147,/m/018j2,"Banjo"
148,/m/0jtg0,"Sitar"
149,/m/04rzd,"Mandolin"
150,/m/01bns_,"Zither"
151,/m/07xzm,"Ukulele"
152,/m/05148p4,"Keyboard (musical)"
153,/m/05r5c,"Piano"
154,/m/01s0ps,"Electric piano"
155,/m/013y1f,"Organ"
156,/m/03xq_f,"Electronic organ"
157,/m/03gvt,"Hammond organ"
158,/m/0l14qv,"Synthesizer"

159,/m/01v1d8,"Sampler"
160,/m/03q5t,"Harpsichord"
161,/m/0l14md,"Percussion"
162,/m/02hnl,"Drum kit"
163,/m/0cfdd,"Drum machine"
164,/m/026t6,"Drum"
165,/m/06rvn,"Snare drum"
166,/m/03t3fj,"Rimshot"
167,/m/02k_mr,"Drum roll"
168,/m/0bm02,"Bass drum"
169,/m/011k_j,"Timpani"
170,/m/01p970,"Tabla"
171,/m/01qbl,"Cymbal"
172,/m/03qtq,"Hi-hat"
173,/m/01sm1g,"Wood block"
174,/m/07brj,"Tambourine"
175,/m/05r5wn,"Rattle (instrument)"
176,/m/0xzly,"Maraca"
177,/m/0mbct,"Gong"
178,/m/016622,"Tubular bells"
179,/m/0j45pbj,"Mallet percussion"
180,/m/0dwsp,"Marimba, xylophone"
181,/m/0dwtp,"Glockenspiel"
182,/m/0dwt5,"Vibraphone"
183,/m/0l156b,"Steelpan"
184,/m/05pd6,"Orchestra"
185,/m/01kcd,"Brass instrument"
186,/m/0319l,"French horn"
187,/m/07gql,"Trumpet"
188,/m/07c6l,"Trombone"
189,/m/0l14_3,"Bowed string instrument"
190,/m/02qmj0d,"String section"
191,/m/07y_7,"Violin, fiddle"
192,/m/0d8_n,"Pizzicato"
193,/m/01xqw,"Cello"
194,/m/02fsn,"Double bass"
195,/m/085jw,"Wind instrument, woodwind instrument"
196,/m/0l14j_,"Flute"
197,/m/06ncr,"Saxophone"
198,/m/01wy6,"Clarinet"
199,/m/03m5k,"Harp"
200,/m/0395lw,"Bell"
201,/m/03w41f,"Church bell"
202,/m/027m70_,"Jingle bell"
203,/m/0gy1t2s,"Bicycle bell"
204,/m/07n_g,"Tuning fork"
205,/m/0f8s22,"Chime"
206,/m/026fgl,"Wind chime"
207,/m/0150b9,"Change ringing (campanology)"
208,/m/03qjg,"Harmonica"
209,/m/0mkg,"Accordion"
210,/m/0192l,"Bagpipes"

211,/m/02bxd,"Didgeridoo"
212,/m/0l14l2,"Shofar"
213,/m/07kc_,"Theremin"
214,/m/0l14t7,"Singing bowl"
215,/m/01hgjl,"Scratching (performance technique)"
216,/m/064t9,"Pop music"
217,/m/0glt670,"Hip hop music"
218,/m/02cz_7,"Beatboxing"
219,/m/06by7,"Rock music"
220,/m/03lty,"Heavy metal"
221,/m/05r6t,"Punk rock"
222,/m/0dls3,"Grunge"
223,/m/0dl5d,"Progressive rock"
224,/m/07sbbz2,"Rock and roll"
225,/m/05w3f,"Psychedelic rock"
226,/m/06j6l,"Rhythm and blues"
227,/m/0gywn,"Soul music"
228,/m/06cqb,"Reggae"
229,/m/01lyv,"Country"
230,/m/015y_n,"Swing music"
231,/m/0gg8l,"Bluegrass"
232,/m/02x8m,"Funk"
233,/m/02w4v,"Folk music"
234,/m/06j64v,"Middle Eastern music"
235,/m/03_d0,"Jazz"
236,/m/026z9,"Disco"
237,/m/0ggq0m,"Classical music"
238,/m/05lls,"Opera"
239,/m/02lkt,"Electronic music"
240,/m/03mb9,"House music"
241,/m/07gxw,"Techno"
242,/m/07s72n,"Dubstep"
243,/m/0283d,"Drum and bass"
244,/m/0m0jc,"Electronica"
245,/m/08cyft,"Electronic dance music"
246,/m/0fd3y,"Ambient music"
247,/m/07lnk,"Trance music"
248,/m/0g293,"Music of Latin America"
249,/m/0ln16,"Salsa music"
250,/m/0326g,"Flamenco"
251,/m/0155w,"Blues"
252,/m/05fw6t,"Music for children"
253,/m/02v2lh,"New-age music"
254,/m/0y4f8,"Vocal music"
255,/m/0z9c,"A capella"
256,/m/0164x2,"Music of Africa"
257,/m/0145m,"Afrobeat"
258,/m/02mscn,"Christian music"
259,/m/016cjb,"Gospel music"
260,/m/028sqc,"Music of Asia"
261,/m/015vgc,"Carnatic music"
262,/m/0dq0md,"Music of Bollywood"

263,/m/06rqw,"Ska"
264,/m/02p0sh1,"Traditional music"
265,/m/05rwpb,"Independent music"
266,/m/074ft,"Song"
267,/m/025td0t,"Background music"
268,/m/02cjck,"Theme music"
269,/m/03r5q_,"Jingle (music)"
270,/m/0l14gg,"Soundtrack music"
271,/m/07pkxdp,"Lullaby"
272,/m/01z7dr,"Video game music"
273,/m/0140xf,"Christmas music"
274,/m/0ggx5q,"Dance music"
275,/m/04wptg,"Wedding music"
276,/t/dd00031,"Happy music"
277,/t/dd00032,"Funny music"
278,/t/dd00033,"Sad music"
279,/t/dd00034,"Tender music"
280,/t/dd00035,"Exciting music"
281,/t/dd00036,"Angry music"
282,/t/dd00037,"Scary music"
283,/m/03m9d0z,"Wind"
284,/m/09t49,"Rustling leaves"
285,/t/dd00092,"Wind noise (microphone)"
286,/m/0jb2l,"Thunderstorm"
287,/m/0ngt1,"Thunder"
288,/m/0838f,"Water"
289,/m/06mb1,"Rain"
290,/m/07r10fb,"Raindrop"
291,/t/dd00038,"Rain on surface"
292,/m/0j6m2,"Stream"
293,/m/0j2kx,"Waterfall"
294,/m/05kq4,"Ocean"
295,/m/034srq,"Waves, surf"
296,/m/06wzb,"Steam"
297,/m/07swgks,"Gurgling"
298,/m/02_41,"Fire"
299,/m/07pzfmf,"Crackle"
300,/m/07yv9,"Vehicle"
301,/m/019jd,"Boat, Water vehicle"
302,/m/0hsrw,"Sailboat, sailing ship"
303,/m/056ks2,"Rowboat, canoe, kayak"
304,/m/02rlv9,"Motorboat, speedboat"
305,/m/06q74,"Ship"
306,/m/012f08,"Motor vehicle (road)"
307,/m/0k4j,"Car"
308,/m/0912c9,"Vehicle horn, car horn, honking"
309,/m/07qv_d5,"Toot"
310,/m/02mfyn,"Car alarm"
311,/m/04gxbd,"Power windows, electric windows"
312,/m/07rknqz,"Skidding"
313,/m/0h9mv,"Tire squeal"
314,/t/dd00134,"Car passing by"

315,/m/0ltv,"Race car, auto racing"
316,/m/07r04,"Truck"
317,/m/0gvgw0,"Air brake"
318,/m/05x_td,"Air horn, truck horn"
319,/m/02rhddq,"Reversing beeps"
320,/m/03cl9h,"Ice cream truck, ice cream van"
321,/m/01bjv,"Bus"
322,/m/03j1ly,"Emergency vehicle"
323,/m/04qvtq,"Police car (siren)"
324,/m/012n7d,"Ambulance (siren)"
325,/m/012ndj,"Fire engine, fire truck (siren)"
326,/m/04_sv,"Motorcycle"
327,/m/0btp2,"Traffic noise, roadway noise"
328,/m/06d_3,"Rail transport"
329,/m/07jdr,"Train"
330,/m/04zmvq,"Train whistle"
331,/m/0284vy3,"Train horn"
332,/m/01g50p,"Railroad car, train wagon"
333,/t/dd00048,"Train wheels squealing"
334,/m/0195fx,"Subway, metro, underground"
335,/m/0k5j,"Aircraft"
336,/m/014yck,"Aircraft engine"
337,/m/04229,"Jet engine"
338,/m/02l6bg,"Propeller, airscrew"
339,/m/09ct_,"Helicopter"
340,/m/0cmf2,"Fixed-wing aircraft, airplane"
341,/m/0199g,"Bicycle"
342,/m/06_fw,"Skateboard"
343,/m/02mk9,"Engine"
344,/t/dd00065,"Light engine (high frequency)"
345,/m/08j51y,"Dental drill, dentist's drill"
346,/m/01yg9g,"Lawn mower"
347,/m/01j4z9,"Chainsaw"
348,/t/dd00066,"Medium engine (mid frequency)"
349,/t/dd00067,"Heavy engine (low frequency)"
350,/m/01h82_,"Engine knocking"
351,/t/dd00130,"Engine starting"
352,/m/07pb8fc,"Idling"
353,/m/07q2z82,"Accelerating, revving, vroom"
354,/m/02dgv,"Door"
355,/m/03wwcy,"Doorbell"
356,/m/07r67yg,"Ding-dong"
357,/m/02y_763,"Sliding door"
358,/m/07rjzl8,"Slam"
359,/m/07r4wb8,"Knock"
360,/m/07qcpgn,"Tap"
361,/m/07q6cd_,"Squeak"
362,/m/0642b4,"Cupboard open or close"
363,/m/0fqfqc,"Drawer open or close"
364,/m/04brg2,"Dishes, pots, and pans"
365,/m/023pjk,"Cutlery, silverware"
366,/m/07pn_8q,"Chopping (food)"

367,/m/0dxrf,"Frying (food)"
368,/m/0fx9l,"Microwave oven"
369,/m/02pjr4,"Blender"
370,/m/02jz0l,"Water tap, faucet"
371,/m/0130jx,"Sink (filling or washing)"
372,/m/03dnzn,"Bathtub (filling or washing)"
373,/m/03wvsk,"Hair dryer"
374,/m/01jt3m,"Toilet flush"
375,/m/012xff,"Toothbrush"
376,/m/04fgwm,"Electric toothbrush"
377,/m/0d31p,"Vacuum cleaner"
378,/m/01s0vc,"Zipper (clothing)"
379,/m/03v3yw,"Keys jangling"
380,/m/0242l,"Coin (dropping)"
381,/m/01lsmm,"Scissors"
382,/m/02g901,"Electric shaver, electric razor"
383,/m/05rj2,"Shuffling cards"
384,/m/0316dw,"Typing"
385,/m/0c2wf,"Typewriter"
386,/m/01m2v,"Computer keyboard"
387,/m/081rb,"Writing"
388,/m/07pp_mv,"Alarm"
389,/m/07cx4,"Telephone"
390,/m/07pp8cl,"Telephone bell ringing"
391,/m/01hnzm,"Ringtone"
392,/m/02c8p,"Telephone dialing, DTMF"
393,/m/015jpf,"Dial tone"
394,/m/01z47d,"Busy signal"
395,/m/046dlr,"Alarm clock"
396,/m/03kmc9,"Siren"
397,/m/0dgbq,"Civil defense siren"
398,/m/030rvx,"Buzzer"
399,/m/01y3hg,"Smoke detector, smoke alarm"
400,/m/0c3f7m,"Fire alarm"
401,/m/04fq5q,"Foghorn"
402,/m/0l156k,"Whistle"
403,/m/06hck5,"Steam whistle"
404,/t/dd00077,"Mechanisms"
405,/m/02bm9n,"Ratchet, pawl"
406,/m/01x3z,"Clock"
407,/m/07qjznt,"Tick"
408,/m/07qjznl,"Tick-tock"
409,/m/0l7xg,"Gears"
410,/m/05zc1,"Pulleys"
411,/m/0llzx,"Sewing machine"
412,/m/02x984l,"Mechanical fan"
413,/m/025wky1,"Air conditioning"
414,/m/024dl,"Cash register"
415,/m/01m4t,"Printer"
416,/m/0dv5r,"Camera"
417,/m/07bjf,"Single-lens reflex camera"
418,/m/07k1x,"Tools"

419,/m/03l9g,"Hammer"
420,/m/03p19w,"Jackhammer"
421,/m/01b82r,"Sawing"
422,/m/02p01q,"Filing (rasp)"
423,/m/023vsd,"Sanding"
424,/m/0_ksk,"Power tool"
425,/m/01d380,"Drill"
426,/m/014zdl,"Explosion"
427,/m/032s66,"Gunshot, gunfire"
428,/m/04zjc,"Machine gun"
429,/m/02z32qm,"Fusillade"
430,/m/0_1c,"Artillery fire"
431,/m/073cg4,"Cap gun"
432,/m/0g6b5,"Fireworks"
433,/g/122z_qxw,"Firecracker"
434,/m/07qsvvw,"Burst, pop"
435,/m/07pxg6y,"Eruption"
436,/m/07qqyl4,"Boom"
437,/m/083vt,"Wood"
438,/m/07pczhz,"Chop"
439,/m/07pl1bw,"Splinter"
440,/m/07qs1cx,"Crack"
441,/m/039jq,"Glass"
442,/m/07q7njn,"Chink, clink"
443,/m/07rn7sz,"Shatter"
444,/m/04k94,"Liquid"
445,/m/07rrlb6,"Splash, splatter"
446,/m/07p6mqd,"Slosh"
447,/m/07qlwh6,"Squish"
448,/m/07r5v4s,"Drip"
449,/m/07prgkl,"Pour"
450,/m/07pqc89,"Trickle, dribble"
451,/t/dd00088,"Gush"
452,/m/07p7b8y,"Fill (with liquid)"
453,/m/07qlf79,"Spray"
454,/m/07ptzwd,"Pump (liquid)"
455,/m/07ptfmf,"Stir"
456,/m/0dv3j,"Boiling"
457,/m/0790c,"Sonar"
458,/m/0dl83,"Arrow"
459,/m/07rqsjt,"Whoosh, swoosh, swish"
460,/m/07qnq_y,"Thump, thud"
461,/m/07rrh0c,"Thunk"
462,/m/0b_fwt,"Electronic tuner"
463,/m/02rr_,"Effects unit"
464,/m/07m2kt,"Chorus effect"
465,/m/018w8,"Basketball bounce"
466,/m/07pws3f,"Bang"
467,/m/07ryjzk,"Slap, smack"
468,/m/07rdhzs,"Whack, thwack"
469,/m/07pjjrj,"Smash, crash"
470,/m/07pc8lb,"Breaking"

471,/m/07pqn27,"Bouncing"
472,/m/07rbp7_,"Whip"
473,/m/07pyf11,"Flap"
474,/m/07qb_dv,"Scratch"
475,/m/07qv4k0,"Scrape"
476,/m/07pdjhy,"Rub"
477,/m/07s8j8t,"Roll"
478,/m/07plct2,"Crushing"
479,/t/dd00112,"Crumpling, crinkling"
480,/m/07qcx4z,"Tearing"
481,/m/02fs_r,"Beep, bleep"
482,/m/07qwdck,"Ping"
483,/m/07phxs1,"Ding"
484,/m/07rv4dm,"Clang"
485,/m/07s02z0,"Squeal"
486,/m/07qh7jl,"Creak"
487,/m/07qwyj0,"Rustle"
488,/m/07s34ls,"Whir"
489,/m/07qmpdm,"Clatter"
490,/m/07p9k1k,"Sizzle"
491,/m/07qc9xj,"Clicking"
492,/m/07rwm0c,"Clickety-clack"
493,/m/07phhsh,"Rumble"
494,/m/07qyrcz,"Plop"
495,/m/07qfgpx,"Jingle, tinkle"
496,/m/07rcgpl,"Hum"
497,/m/07p78v5,"Zing"
498,/t/dd00121,"Boing"
499,/m/07s12q4,"Crunch"
500,/m/028v0c,"Silence"
501,/m/01v_m0,"Sine wave"
502,/m/0b9m1,"Harmonic"
503,/m/0hdsk,"Chirp tone"
504,/m/0c1dj,"Sound effect"
505,/m/07pt_g0,"Pulse"
506,/t/dd00125,"Inside, small room"
507,/t/dd00126,"Inside, large room or hall"
508,/t/dd00127,"Inside, public space"
509,/t/dd00128,"Outside, urban or manmade"
510,/t/dd00129,"Outside, rural or natural"
511,/m/01b9nn,"Reverberation"
512,/m/01jnbd,"Echo"
513,/m/096m7z,"Noise"
514,/m/06_y0by,"Environmental noise"
515,/m/07rgkc5,"Static"
516,/m/06xkwv,"Mains hum"
517,/m/0g12c5,"Distortion"
518,/m/08p9q4,"Sidetone"
519,/m/07szfh9,"Cacophony"
520,/m/0chx_,"White noise"
521,/m/0cj0r,"Pink noise"
522,/m/07p_0gm,"Throbbing"

```
523,/m/01jwx6,"Vibration"
524,/m/07c52,"Television"
525,/m/06bz3,"Radio"
526,/m/07hvw1,"Field recording"
```

setup.py

```
import os
import re

from setuptools import setup

try:
    import pypandoc
    long_description = pypandoc.convert('README.md', 'rst')
except(IOError, ImportError):
    long_description = open('README.md').read()


def get_version(package):
    init_py = open(os.path.join(package, '__init__.py')).read()
    return re.search("__version__ = ['\"]([^'\"]+)['\"]", init_py).group(1)


def get_packages(package):
    packages = []
    for dirpath, dirnames, filenames in os.walk(package):
        if os.path.exists(os.path.join(dirpath, '__init__.py')):
            packages.append(dirpath)
    return packages


def get_package_data(package):
    filepaths = []
    for dirpath, dirnames, filenames in os.walk(package):
        if os.path.exists(os.path.join(dirpath, '__init__.py')):
            continue

        base = dirpath.replace(package + os.sep, '', 1)
        for filename in filenames:
            filepaths.append(os.path.join(base, filename))

    return {package: filepaths}


setup(name='devicehive_webconfig',
      version=get_version('devicehive_webconfig'),
      author='DataArt (http://dataart.com)',
      author_email='info@devicehive.com',
      url='https://devicehive.com',
      license='Apache License 2.0',
      description='DeviceHive Python web configurator',
```

```
long_description=long_description,
keywords='iot cloud m2m gateway embedded devicehive configurator web ui',
packages=get_packages('devicehive_webconfig'),
package_data=get_package_data('devicehive_webconfig'),
install_requires=['devicehive>=2.1.3', 'six>=1.11.0'],
python_requires=">=2.7, !=3.0.*, !=3.1.*, !=3.2.*, !=3.3.*, !=3.4.*",
classifiers=[
    'Development Status :: 4 - Beta',
    'Environment :: Console',
    'Environment :: Web Environment',
    'Intended Audience :: Developers',
    'Intended Audience :: Information Technology',
    'License :: OSI Approved :: Apache Software License',
    'Operating System :: MacOS',
    'Operating System :: Microsoft :: Windows',
    'Operating System :: POSIX',
    'Operating System :: Unix',
    'Programming Language :: Python',
    'Programming Language :: Python :: 2',
    'Programming Language :: Python :: 2.7',
    'Programming Language :: Python :: 3',
    'Programming Language :: Python :: 3.5',
    'Programming Language :: Python :: 3.6',
    'Programming Language :: Python :: Implementation :: CPython',
    'Programming Language :: Python :: Implementation :: PyPy',
    'Topic :: Home Automation',
    'Topic :: Internet',
    'Topic :: Software Development :: Embedded Systems',
])
```

Додаток В (обов'язковий)

Додаток В (обов'язковий) 105

ІЛЮСТРАТИВНА ЧАСТИНА

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ РОЗПІЗНАВАННЯ МУЗИЧНИХ ПАТЕРНІВ НА ОСНОВІ СПАЙКІНГОВОЇ НЕЙРОННОЇ МЕРЕЖІ

Виконав: студент 2-го курсу,
групи 2КН-21м
спеціальності 122 «Комп'ютерні науки»
(шифр і назва напряму підготовки, спеціальності)
Чоботар Є.О.
(прізвище та ініціали)

Керівник: к.т.н., проф. каф. КН
Месюра В.І.
(прізвище та ініціали)
«15» 12 2022 р.

Вінниця ВНТУ - 2022 рік

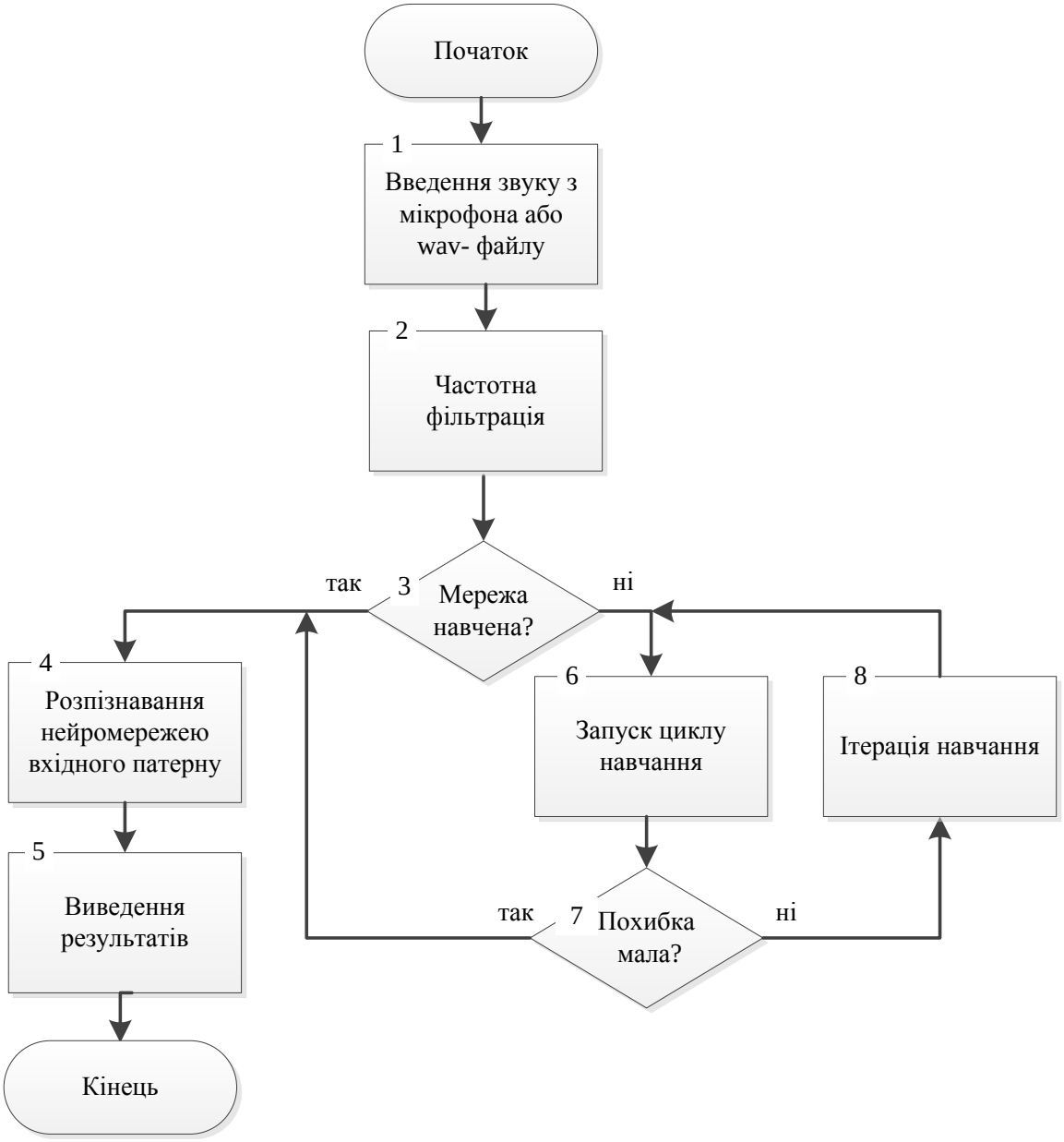


Рисунок В.1 – Граф-схема загального алгоритму роботи програми розпізнавання музичних патернів на основі спайкінгової нейронної мережі

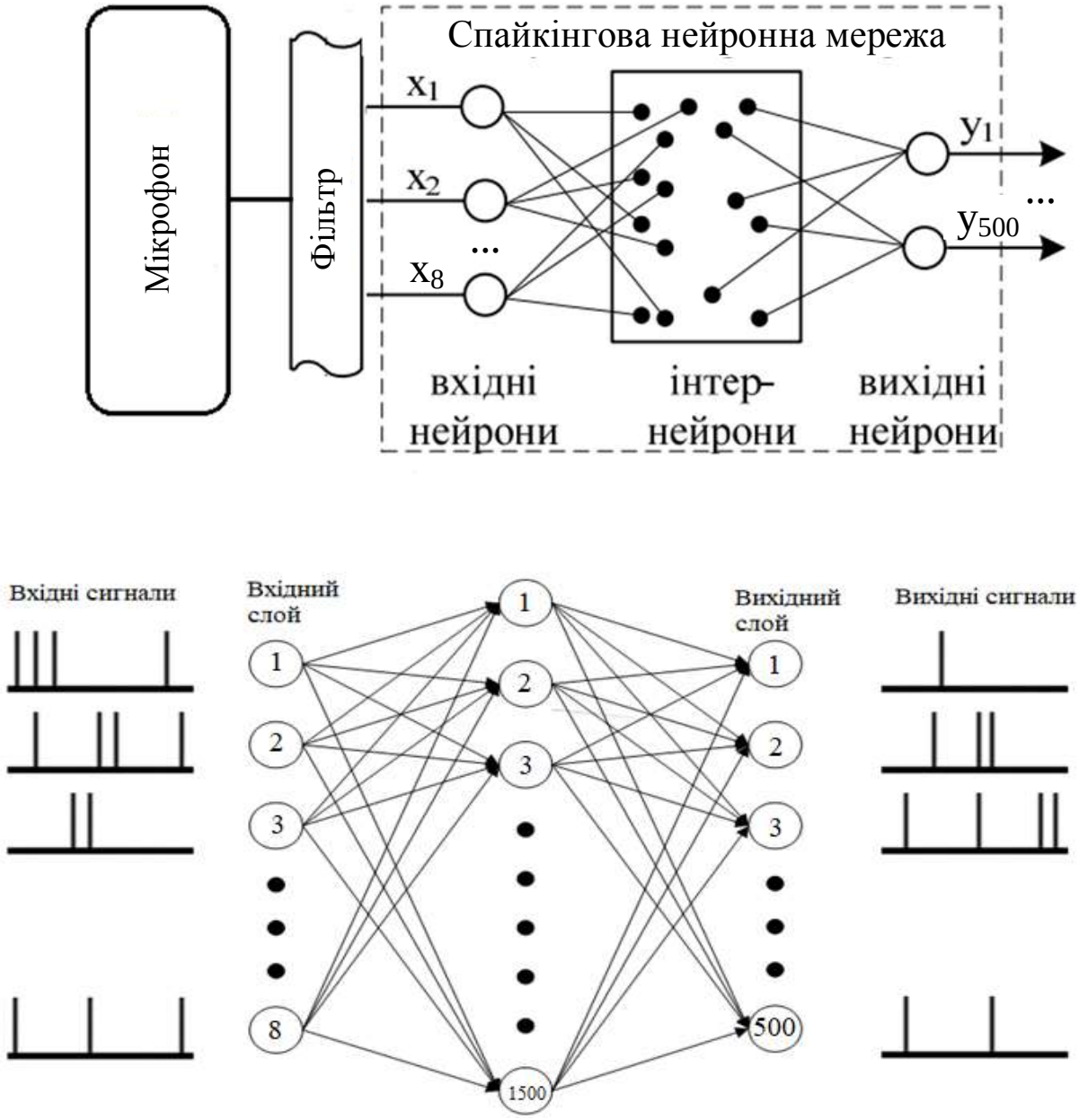


Рисунок В.2 – Структура спайкінгової нейронної мережі

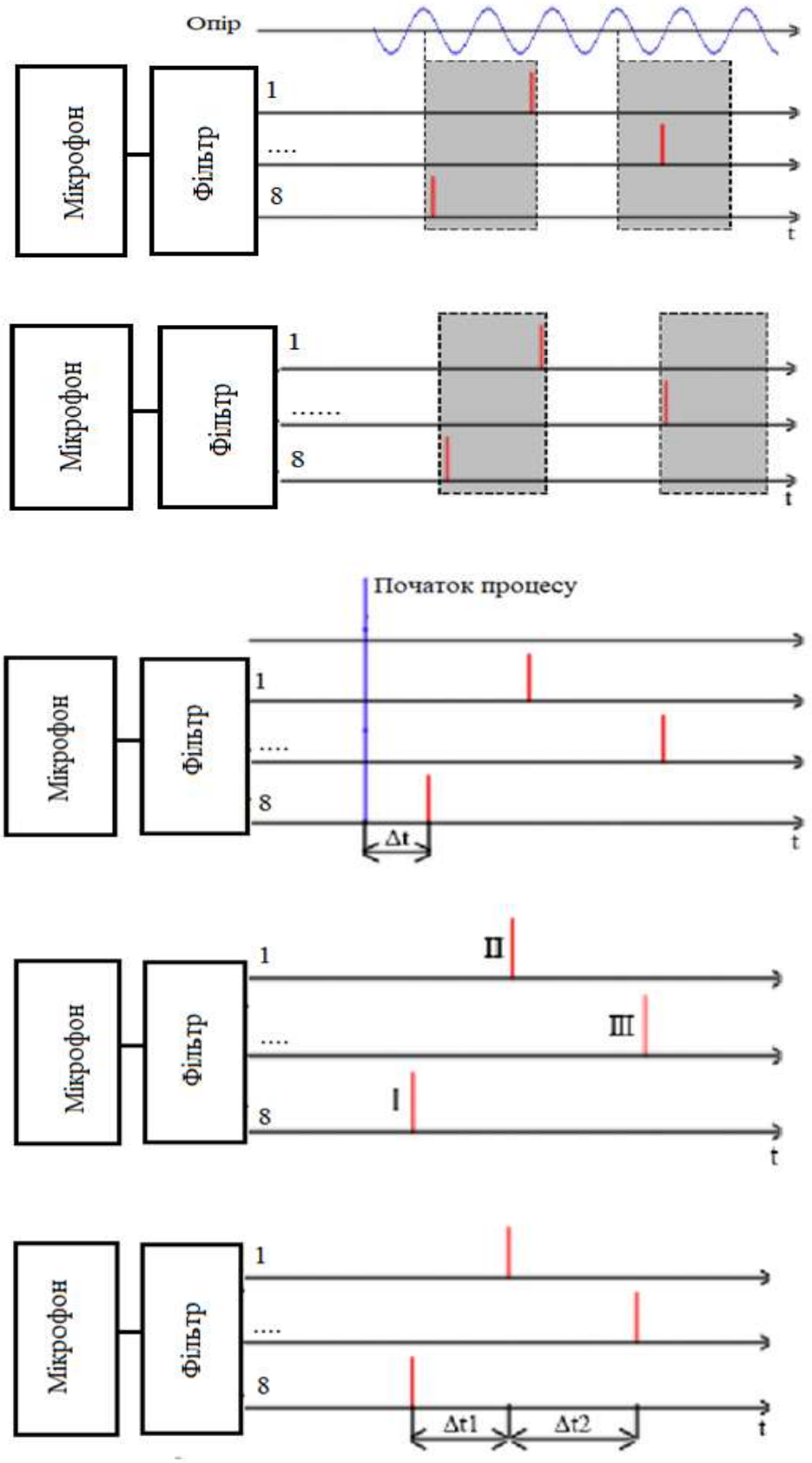


Рисунок В.3 – Методи подання інформації в спайкінгових нейронних мережах

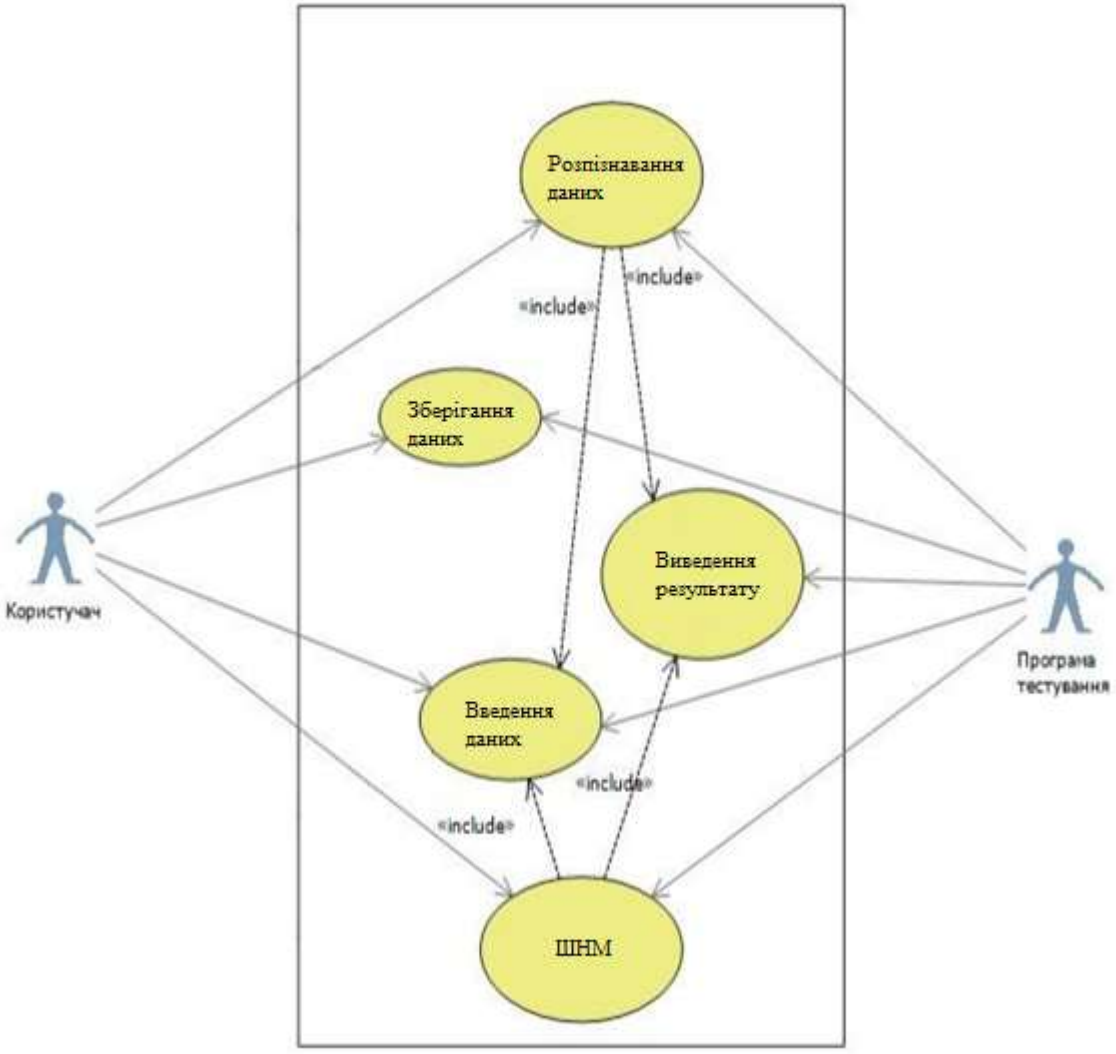


Рисунок В.4 – Діаграма варіантів використання програмного забезпечення розпізнавання музичних патернів

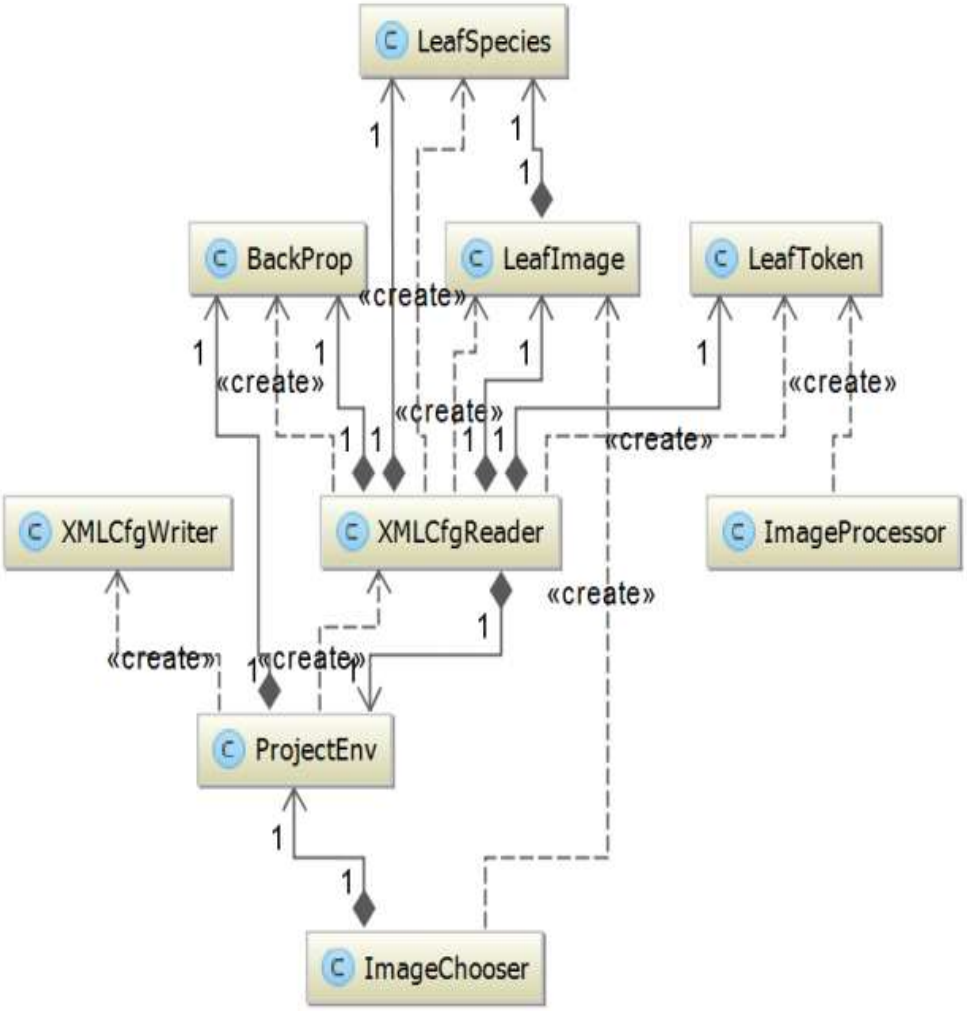


Рисунок В.5 – Діаграма класів програми розпізнавання музичних патернів

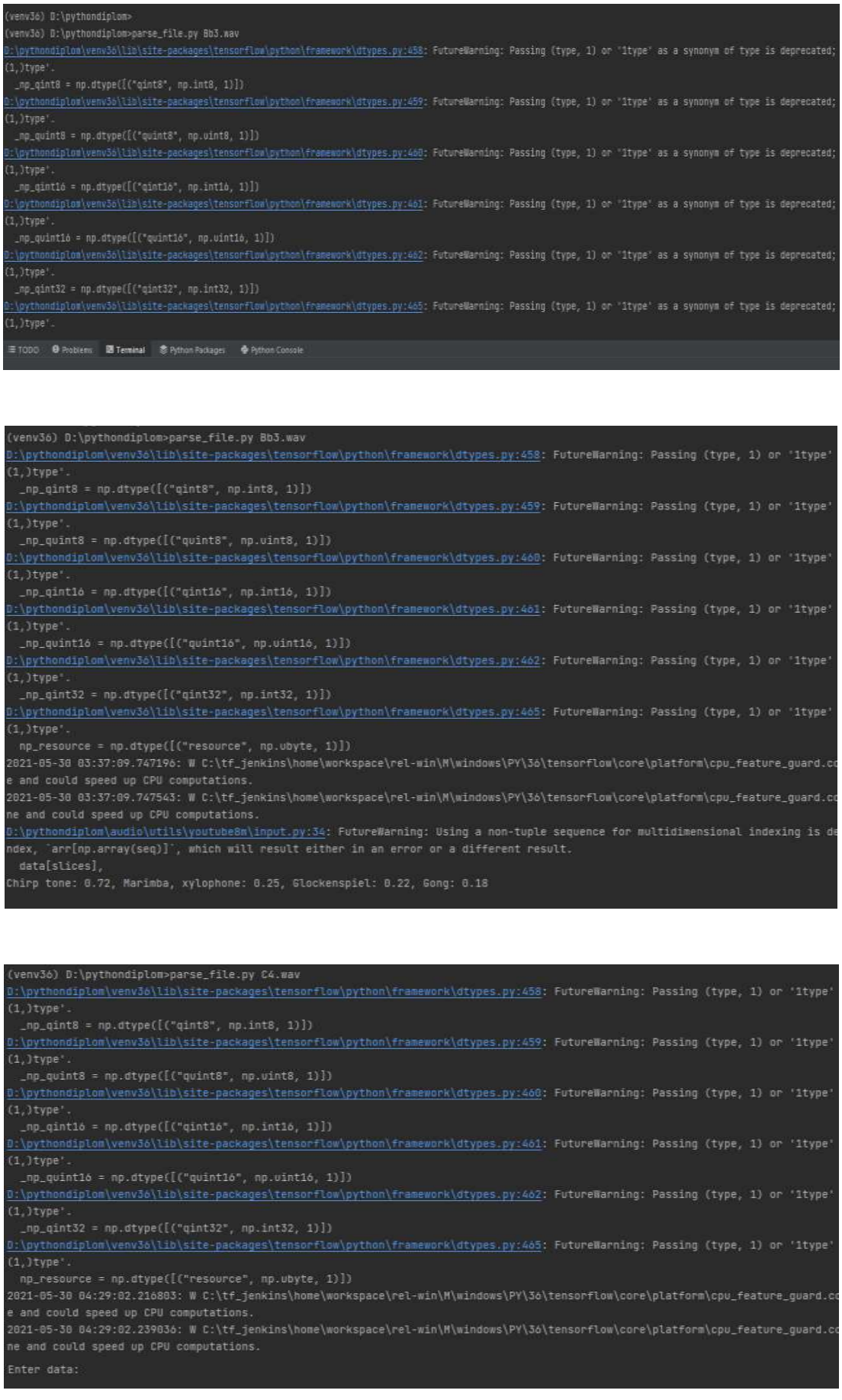


Рисунок В.6 – Результати роботи програми розпізнавання музичних патернів

Додаток Г (довідниковий)

Інструкція користувача

Для запуску програми необхідно встановити додаткове програмне забезпечення, а саме:

Numpy 1.13.3

Scipy 0.19.1

Resampy 0.2.0

PyAudio 0.2.11

Tensorflow 1.3.0

Six 1.11.0

Devicehive 2.1.1

Для цього можна виконати команду в терміналі:

-e git://github.com/devicehive/devicehive-python-webconfig.git@a792db1babcedb5baa68ec6ba6ebcf0041f20469#egg=devicehive_webconfig

або встановити це все вручну.

Далі для запуску роботи програми необхідно виконати наступну команду в консолі: «parse_file.py /path/to/the/file», де /path/to/the/file – це шлях до файла який необхідно розпізнати, при чому цей файл мусить мати розширеня .wav інакше система не буде працювати. Результат роботи виконання програми можна побачити на рисунку Г.1.

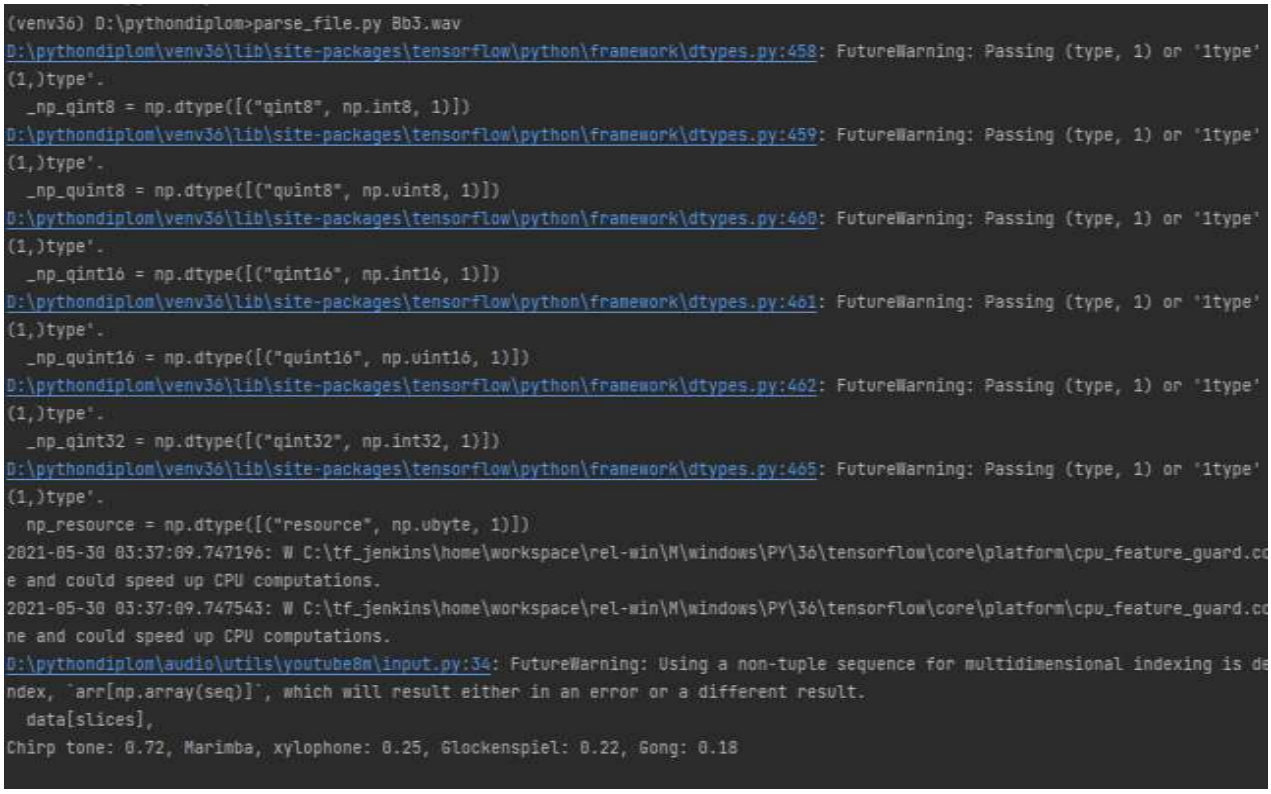


Рисунок Г.1 - Результат роботи виконання програми