ПОЛТАВСЬКИЙ ДЕРЖАВНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЕКОНОМІКИ, УПРАВЛІННЯ, ПРАВА ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

Пояснювальна записка

до кваліфікаційної роботи на здобуття ступеня вищої освіти магістр

на тему: **«Технологія класифікації аудіоконтенту на основі нейронної мережі»**

Виконав: здобувач вищої освіти
за освітньо-професійною програмою
Інформаційні управляючі системи та технології спеціальності
126 Інформаційні системи та технології ступеня вищої освіти магістр
групи 126ІСТмд_22
Раскін А. М.
Керівник: Слюсар В. І.
Рецензент: Муравльов В. В.

Полтава – 2023 року

ВСТУП

Актуальність теми кваліфікаційної роботи підтверджується необхідністю класифікації аудіоконтенту на основі нейронних мереж. На даний час, для вирішення цього завдання досить перспективними є згорткові та рекурентні мережі, автоенкодери, трансформери та гібридні моделі. При цьому, вони використовувались, в основному, для інших завдань, наприклад, обробки зображень, природньої мови та ін. Отримані результати в цих галузях свідчать про доцільність їх застосування для обробки аудіоданих. Однак питання впливу архітектури нейронної мережі та підбору гіперпараметрів на продуктивність моделі глибокого навчання нейронних мереж класифікації аудіоданних потребує додаткових досліджень. Все це свідчить про актуальність теми роботи.

Зв'язок роботи з науковими програмами, темами. Робота відповідає дослідженням в рамках науково-дослідної роботи «Управління стратегією інноваційного розвитку підприємств в контексті підвищення їх конкурентоспроможності на аграрному ринку, сталого розвитку та забезпечення продовольчої безпеки держави» (2021 р.), що фінансувалась господарськими договорами із замовниками, Концепції розвитку штучного інтелекту в Україні (розпорядження Кабінету Міністрів України № 1787-р від 29.12.2021), тематиці досліджень Навчально-дослідної лабораторії інтелектуальних систем, комп'ютерних мереж та інтернет речей Кафедри інформаційних систем та технологій Полтавського державного аграрного університету.

Метою кваліфікаційної роботи є класифікація аудіоконтенту за допомогою нейронної мережі.

Завданнями кваліфікаційної роботи є:

– аналіз особливостей обробки аудіоданих за допомогою нейронних мереж;

– вибір архітектури нейронної мережі для класифікації аудіо;

– оцінка впливу архітектури та підбору гіперпараметрів на продуктивність нейронної мережі;

– обґрунтування рекомендацій щодо використання технології класифікації аудіоконтенту.

Об'єктом дослідження є процес класифікації аудіоконтенту за допомогою штучних нейронних мереж.

Предметом дослідження є залежність точності класифікації аудіоконтенту від архітектури нейронної мережі.

Методами дослідження є аналітичний, інформаційно-пошуковий, методи синтезу та навчання нейронних мереж класифікації зображень, робота з фреймворком Keras.

Інформаційна база кваліфікаційної роботи сформована з ресурсів, що містять інформацію про архітектури нейронних мереж, інструментарій для виконання глибокого машинного.

Елементи наукової новизни роботи полягають у створені моделі глибокого навчання нейронних мереж класифікації аудіоконтенту.

Практична значущість роботи полягає у розробці рекомендацій щодо використання моделі глибокого навчання нейронних мереж класифікації аудіоконтенту – можуть бути використані для подальших досліджень за даною тематикою та при проектуванні мобільних додатків.

Апробація результатів відбувалася в рамках V Міжнародної студентської конференції «Теоретичне та практичне застосування результатів сучасної науки» (жовтень 2023 р., м. Рівне), V Міжнародної студентської конференції «Цифровізація науки та сучасні тренди її розвитку» (листопад 2023 р., м. Житомир).

За результатами досліджень здійснено 2 публікації тез доповідей.

Структура кваліфікаційної роботи логічно пов'язана з завданнями досліджень і містить вступ, три розділи основної частини, висновки, список використаних джерел, додатки. Загальний обсяг пояснювальної записки кваліфікаційної роботи складає 69 сторінок формату А4. Вона містить 37 рисунків.

РОЗДІЛ 1
АНАЛІЗ ОСОБЛИВОСТЕЙ ОБРОБКИ АУДІОДАНИХ НА БАЗІ ШТУЧНОГО ІНТЕЛЕКТУ

1.1 Напрями використання обробки аудіо даних на основі нейронних мереж

В останні роки основним напрямом у сфері штучного інтелекту (AI) [1] є обробка природньої мови (NLP) [2], особливо з використанням нейронних мереж, що побудовані на архітектурі трансформерів [3]. Вони використовуються в системах голосових помічників, а голосові помічники міцно входять у наше життя. Тим не менш, важливою складовою успіху голосових помічників є те, що вони «голосові», тобто звернення до них здійснюється за допомогою голосу, що означає застосування аудіоданих.

Аудіодані в повсякденному житті можуть надходити в множині формах, таких як людська мова, музика, голоси тварин та інші природні звуки, а також звуки, створені людиною в результаті людської діяльності, наприклад автомобілів і механізмів. Враховуючи поширеність звуків у нашому житті та діапазон типів звуків, не дивно, що існує величезна кількість сценаріїв використання, які вимагають від нас обробки та аналізу звуку. Тепер, коли глибоке навчання досягло повноліття, його можна застосовувати для вирішення низки напрямів використання.

1. Класифікація аудіо (рис. 1.1). Це один із найпоширеніших випадків використання, який передбачає взяття звуку та призначення його одному з кількох класів. Наприклад, завдання може полягати в тому, щоб визначити тип або джерело звуку, наприклад, це машина заводиться, це молоток, свисток або гавкіт собаки. Очевидно, що можливості застосування великі. Це може бути застосовано для виявлення несправності машин або обладнання на основі звуку, який вони видають, або в системі спостереження для виявлення зламів безпеки та ін.

2. Розподіл і сегментація звуку (рис. 1.2). Аудіорозділення передбачає виділення цікавого сигналу від суміші сигналів, щоб потім його можна було використовувати для подальшої обробки. Наприклад, ви можете відокремити голоси окремих людей від багатьох фонових шумів або звуки скрипки від решти музичного виконання.

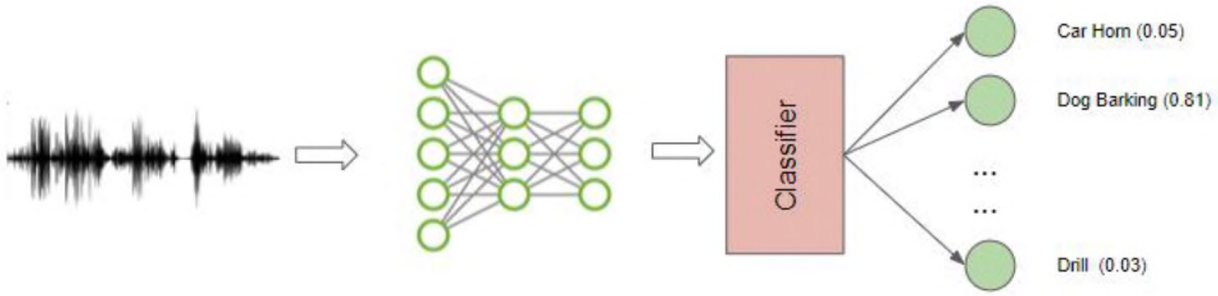

Рисунок 1.1 – Класифікація звичайних звуків [4]

Аудіосегментація використовується для виділення відповідних розділів аудіопотоку. Наприклад, його можна використовувати в діагностичних цілях для визначення різних звуків людського серця та виявлення аномалій.

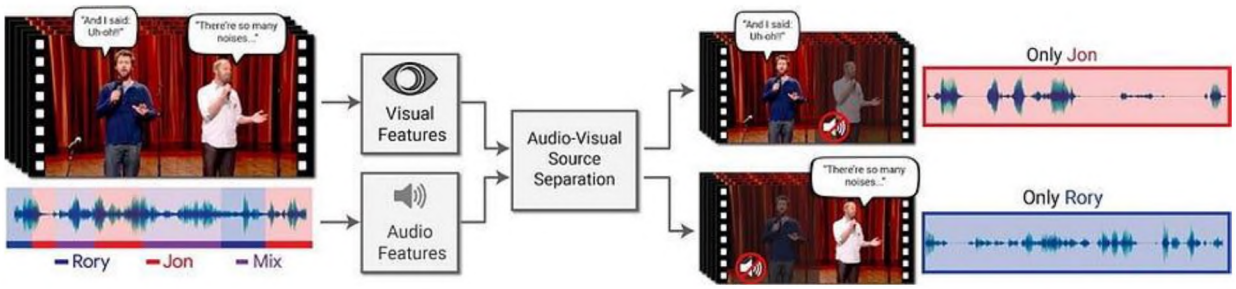

Рисунок 1.2 – Відокремлення окремих динаміків від відео [5]

3. Класифікація музичних жанрів і тегування (рис. 1.3). У зв'язку з популярністю сервісів потокового передавання музики, ще одна поширена програма – ідентифікація та класифікація музики на основі аудіо. Зміст музики аналізується, щоб визначити жанр, до якого вона належить. Це проблема класифікації з кількома ярликами, оскільки даний музичний твір може належати до кількох жанрів. напр. рок, поп, джаз, сальса,

інструментальна музика, а також інші аспекти, такі як «старі», «вокалістка», «щаслива», «музика для вечірок» тощо.

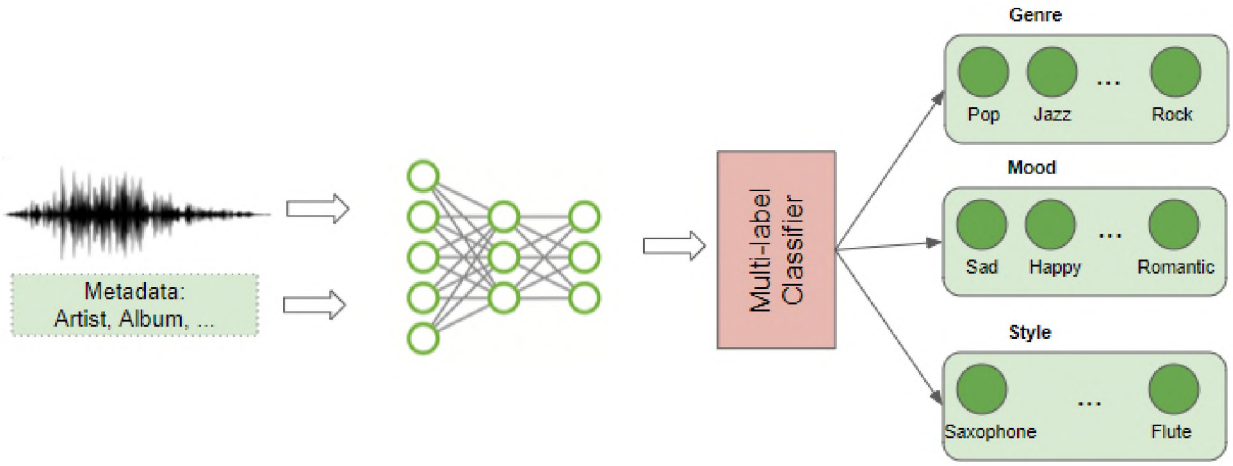

Рисунок 1.3 – Класифікація музичних жанрів і тегування [4]

Звичайно, окрім самого аудіофайлу, є метадані про музику, такі як виконавець, дата випуску, композитор, тексти пісень тощо, які можна інтегрувати у сервіси для додавання широкого набору тегів до музики. Це можна використовувати для індексування музичних колекцій відповідно до їхніх аудіофункцій, надання музичних рекомендацій на основі яку слухаєте.

4. Створення музики та транскрипція музики (рис. 1.4). Сьогодні ми бачили багато новин про використання глибокого навчання для програмного створення надзвичайно автентичних зображень обличчя та інших сцен, а також про можливість писати граматично правильні та розумні листи або новинні статті.

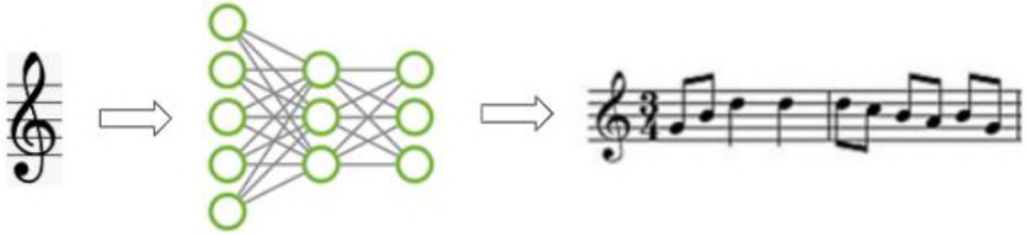

Рисунок 1.4 – Музичне покоління [4]

Подібним чином тепер можна створювати синтетичну музику, яка відповідає певному жанру, інструменту чи навіть певному стилю композитора. Певним чином транскрипція музики застосовує цю можливість навпаки. Щоб створити музичну ноту, яка містить музичні ноти, які присутні в музиці, потрібна певна акустика та її анотації.

5. Розпізнавання голосу (рис. 1.5). Технічно це також проблема класифікації, але стосується розпізнавання вимовлених звуків. Його можна застосовувати для ідентифікації статі мовця або його імені.

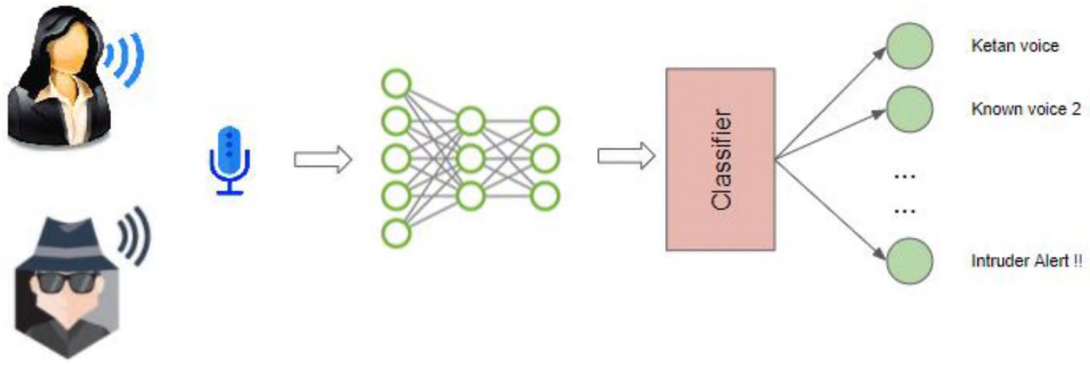

Рисунок 1.5 – Розпізнавання голосу для виявлення порушників [4]

Можна виявити людські емоції та визначити настрій людини за тоном її голосу, наприклад, людина щаслива, сумна, сердита чи напружена. Можна застосувати це до голосів тварин, щоб визначити тип тварини, яка видає звук, або потенційно визначити, чи це звук ніжного ласкавого муркотіння, погрозливого гавкоту чи переляканого лементу.

6. Перетворення мови в текст і перетворення тексту у мову (рис. 1.6). Маючи справу з людським мовленням, ми можемо піти далі і не просто розпізнати мовця, але й зрозуміти, що він говорить.

Рисунок 1.6 – Перетворення мови в текст [4]

Це передбачає виділення слів із аудіо на мові, якою він вимовлений, і транскрибування їх у текстові речення. Це одна з найскладніших програм, оскільки вона має справу не лише з аналізом аудіо, але й з НЛП і вимагає розвитку деяких базових мовних навичок для розшифровки окремих слів із вимовлених звуків. І навпаки, за допомогою синтезу мовлення можна піти в іншому напрямку, взяти письмовий текст і створити з нього мову, використовуючи, наприклад, штучний голос для розмовних агентів. Здатність розуміти людську мову, очевидно, дає змогу використовувати велику кількість корисних застосувань як у нашому бізнесі, так і в побуті. Відомими прикладами, які отримали широке поширення, є такі віртуальні помічники, як Alexa, Siri, Cortana та Google Home, що є зручними для споживача продуктами, створеними навколо цієї можливості.

1.2 Способи представлення аудіо у машинному навчанні

Часто робота з аудіосигналом здійснюється за допомогою аналізу як звуку, так і зображення спектрограми. Більшість звуків, які ми зустрічаємо, можуть не відповідати таким простим і регулярним періодичним моделям. Але сигнали різних частот можна додавати разом, щоб створити складені сигнали з більш складними повторюваними шаблонами. Усі звуки, які чуємо, включно з нашим людським голосом, складаються з таких хвиль. Людське вухо здатне розрізняти різні звуки на основі «якості» звуку, яка також відома як тембр. Можна сказати, що аудіо є фізичним поданням звуку, частота якого знаходиться в діапазоні від 20 Гц до 20 кГц (рис. 1.7). Ці звуки доступні в багатьох форматах, що дозволяє їх аналізувати комп'ютеру, наприклад mp3, wma, wav та ін. До роботи з аудіосигналом його потрібно оцифрувати, тобто перетворити звукову хвилю на ряд чисел (рис. 1.8). Це робиться шляхом виміру амплітуди звуку через фіксовані проміжки часу. Кожен такий вимір називається вибіркою, а частота вибірки – кількістю вибірок за секунду.

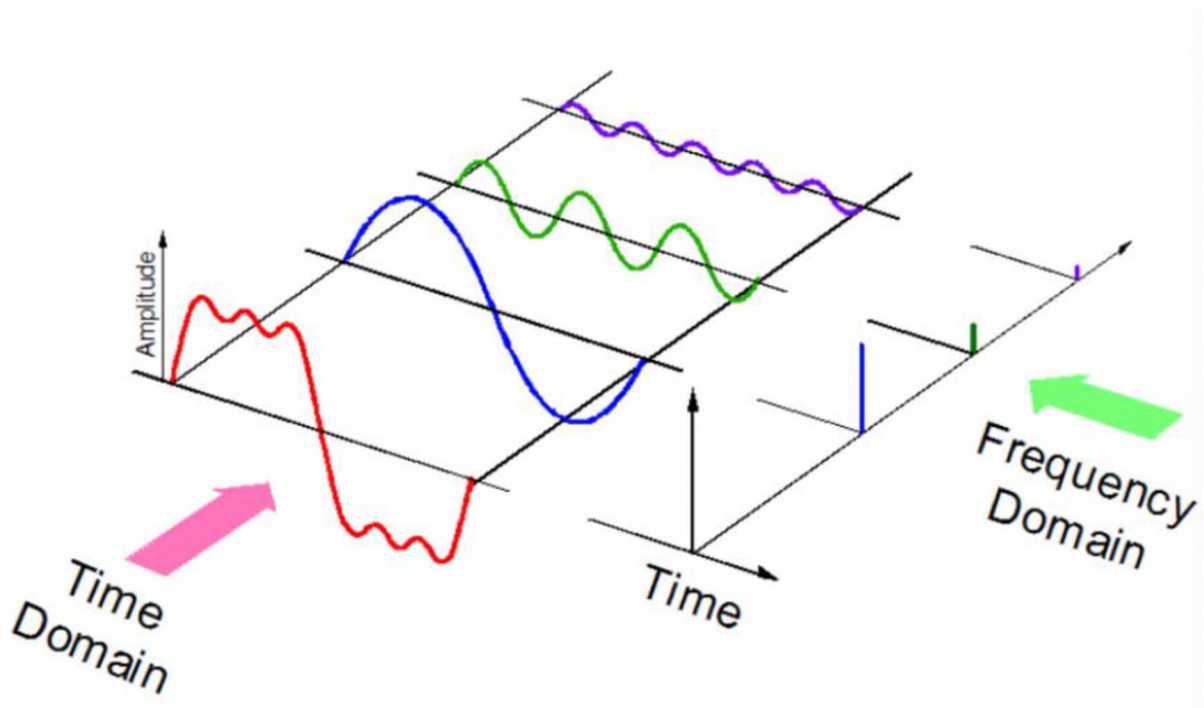

Рисунок 1.7. – Фізичне подання звуку

Наприклад, звичайна частота дискретизації становить близько 44100 вибірок за секунду. Це означає, що 10-секундний музичний кліп міститиме 441000 семплів. Завдяки дискретизації звуку з аудіозаписів можна отримувати досить багато різних характеристик, які допомагають у аналізі аудіо.

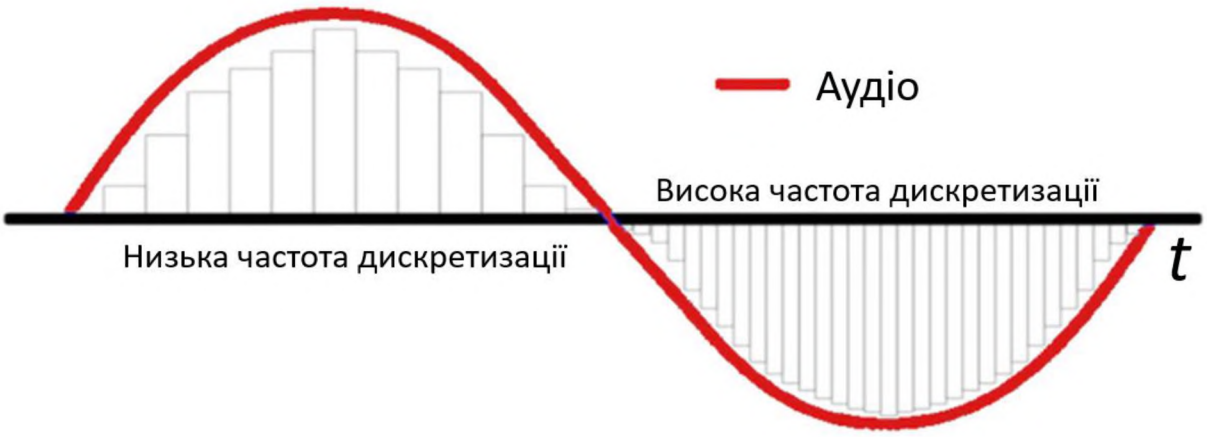

Рисунок 1.8 – Оцифровка сигналу

Сигнал аудіо можна представити як часовий ряд, де по осі Y буде амплітуда сигналу (рис. 1.9). Це дає нам уявлення про те, наскільки гучним чи

тихим є кліп у будь-який момент часу, але дає нам дуже мало інформації про те, які частоти присутні. Друге зображення є спектрограмою і відображає сигнал у частотній області. Спектрограми створюються за допомогою перетворень Фур'є [6] для розкладання будь-якого сигналу на його складові частоти.

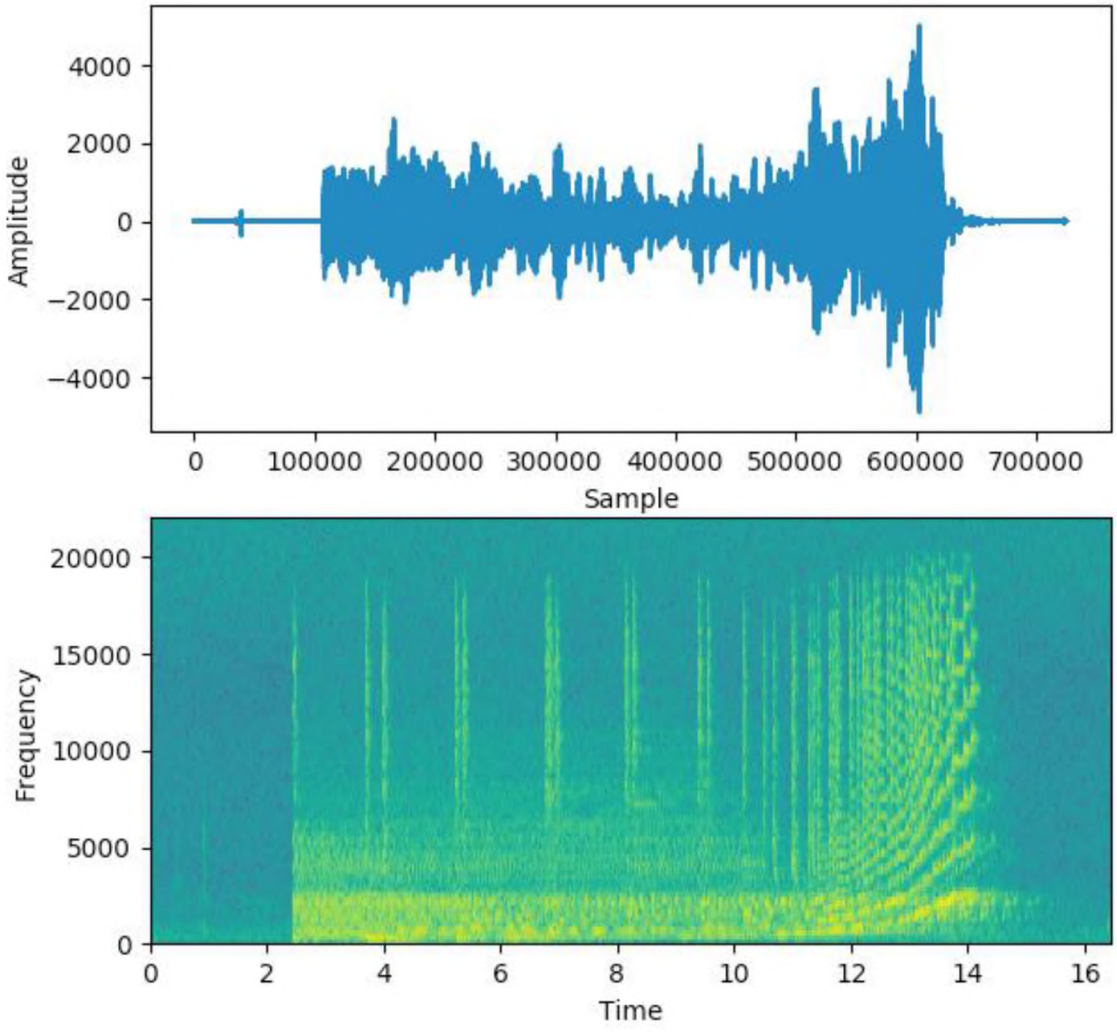

Рисунок 1.9 – Звуковий сигнал і його спектрограма

Людина може почути звук зосереджено у вузькому діапазоні частот і амплітуд. Те, як ми чуємо частоти звуку, називається «висотою». Це суб'єктивне враження від частоти. Люди не сприймають частоти лінійно. Вони більш чутливі до різниці між нижчими частотами, ніж вищими. Незважаючи на те, що в усіх випадках фактична різниця частот між кожною парою однакова на 100 Гц, пара на 100 Гц і 200 Гц звучатиме далі одна від

одної, ніж пара на 1000 і 1100 Гц. І навряд чи середньо-статистична людина зможе відрізнити пару 10000 і 10100 Гц. Однак це може здатися менш дивним, якщо усвідомимо, що частота 200 Гц насправді вдвічі перевищує частоту 100 Гц, тоді як частота 10100 Гц лише на 1 % вища за частоту 10000 Гц. Таким чином, люди сприймають частоти у логарифмічній шкалі, а не в лінійній. Щоб врахувати це застосовують шкалу Мела (Mel). Це шкала висот, за якою слухачі вважають, що кожна одиниця однакова за висотою від наступної (рис. 1.10).

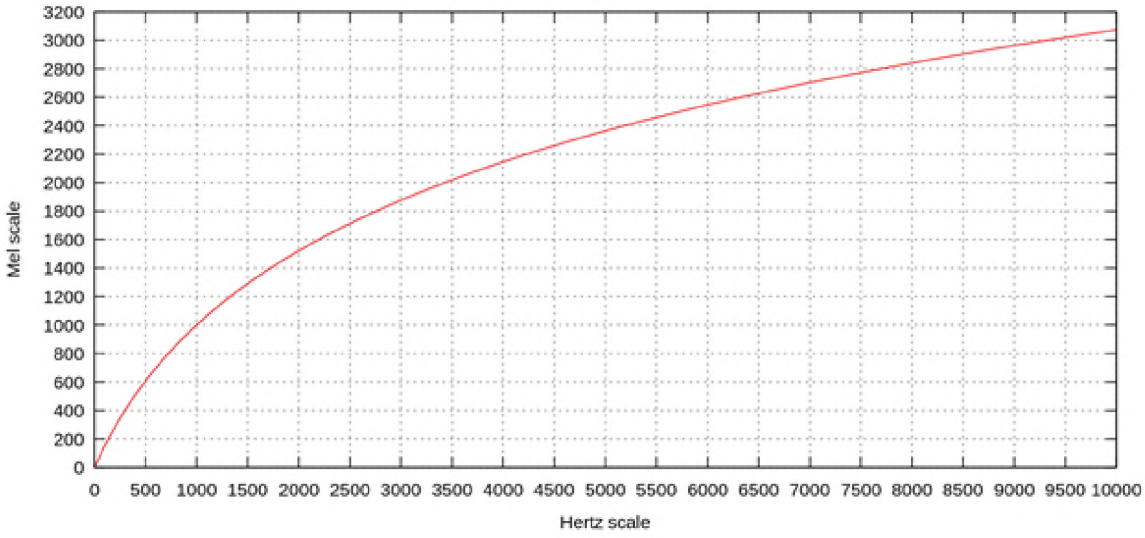

Рисунок 1.10 – Шкала Мела [7]

Сприйняття людиною амплітуди звуку це його гучність. І подібно до частоти, чуємо гучність логарифмічно, а не лінійно. Це враховуємо за допомогою шкали децибел (дБ). Таким чином, спектрограма Мела (рис. 1.11) вносить дві важливі зміни порівняно зі звичайною спектрограмою, яка відображає частоту та час: використовує шкалу Mel замість частоти на осі Y; використовує шкалу дБ замість амплітуди для позначення кольорів. Спектрограма – це стислий «знімок» аудіохвилі, і оскільки це зображення, воно добре підходить для введення в архітектури згорткових нейронних мереж (CNN), що розроблені для обробки зображень. Спектрограми генеруються із звукових сигналів за допомогою перетворень Фур'є. Воно розкладає сигнал на його складові частоти та відображає амплітуду кожної

частоти, присутньої в сигналі. Спектрограма розбиває тривалість звукового сигналу на менші сегменти часу, а потім застосовує перетворення Фур'є до кожного сегмента, щоб визначити частоти, що містяться в цьому сегменті. Потім він об'єднує перетворення Фур'є для всіх цих сегментів в єдиний графік. Він відображає частоту (вісь Y) і час (вісь X) і використовує різні кольори для позначення амплітуди кожної частоти. Чим яскравіший колір, тим вища енергія сигналу.

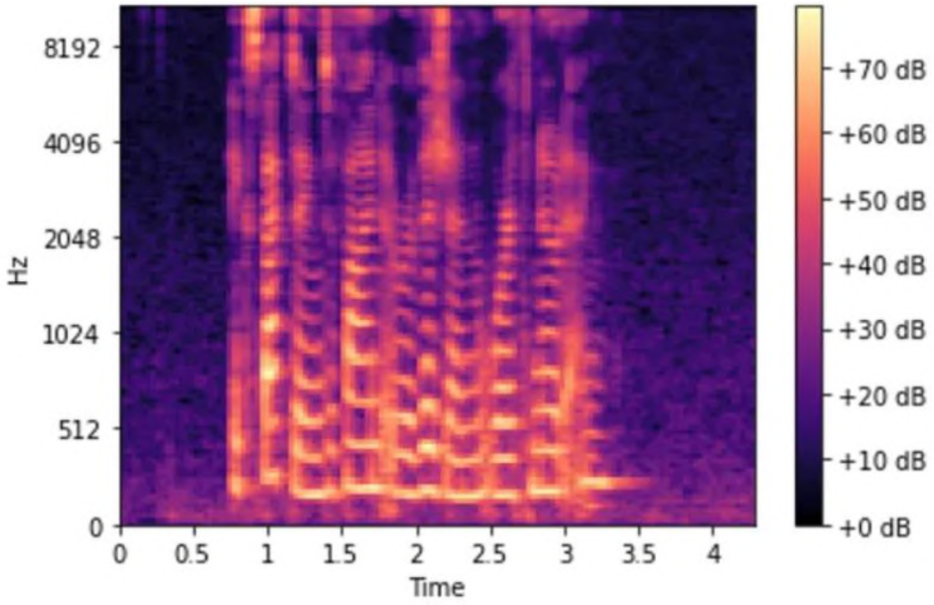

Рисунок 1.11 – Приклад MFCC

Таким чином, Мел-кепстральни коефіцієнти (Mel-Frequency Cepstral Coefficients, MFCC) дозволяють уявити аудіо у вигляді, найближчому до сприйняття звуку людиною.

1.3 Покращення функцій спектрограм і розширення даних

Основні проблеми обробки виникають через шум та спотворення на шляху від джерела сигналу до оцифрування з мікрофона. Можна для різних треків зіставляти оригінал із фрагментом, записаним у різних штучно

зашумлених умовах – і з багатьох прикладів знайти, які характеристики найкраще зберігаються. Виявляється, добре працюють піки спектрограми, виділені тим чи іншим способом – наприклад, як точки локального максимуму амплітуди. Висота піків не підходить (АЧХ мікрофона їх змінює), а ось їхнє місце розташування на сітці «частота-час» мало змінюється при зашумленні. Це спостереження, у тому чи іншому вигляді, використовується в багатьох відомих рішеннях, наприклад, Echoprint [8]. В середньому, на один трек виходить близько 300 тис. піків – такий обсяг даних набагато реальніше зіставляти з мільйонами треків у базі, ніж повну спектрограму запиту.

Mel Spectrograms добре працюють для більшості програм глибокого навчання аудіо. Для покращення якості обробки аудіо можна налаштовувати гіперпараметри спектрограми Мела. В роботі будемо використовувати назви параметрів, які використовуються у бібліотеці Librosa (інші бібліотеки матимуть еквівалентні параметри).

1. Смуги частот:

– fmin – мінімальна частота;

– fmax – максимальна частота для відображення;

– n_mels – кількість частотних діапазонів (тобто Мел-бінів – це висота спектрограми).

2. Розділи часу:

– n_fft – довжина вікна для кожного відрізка часу;

– hop_length – кількість зразків, на які потрібно пересувати вікно на кожному кроці (ширина спектрограми дорівнює загальній кількості зразків / довжині стрибка).

Моделі глибокого навчання рідко приймають це необроблене аудіо безпосередньо як вхідні дані. Тому, доцільно реалізовувати способи представлення аудіо як сукупності різних ознак. Поширеною технікою збільшення різноманітності вашого набору даних, особливо коли недостатньо даних, є штучне збільшення даних. Робимо це, дещо змінюючи наявні зразки даних. Наприклад, із зображеннями можемо злегка повертати зображення,

обрізати або масштабувати його, змінювати кольори або освітлення, або додавати шум до зображення. Оскільки семантика зображення суттєво не змінилася, така сама цільова мітка з оригінального зразка все ще застосовуватиметься до розширеного зразка. Наприклад, якщо зображення було позначено як «кішка», доповнене зображення також буде «кішкою». Але, з точки зору моделі, це виглядає як нова вибірка даних. Це допоможе моделі узагальнити ширший діапазон вхідних зображень. Так само, як і у випадку із зображеннями, існує кілька методів доповнення аудіоданих. Це збільшення можна виконати як на необробленому аудіо перед створенням спектрограми, так і на згенерованій спектрограмі. Збільшення спектрограми, зазвичай, дає кращі результати (рис. 1.12).

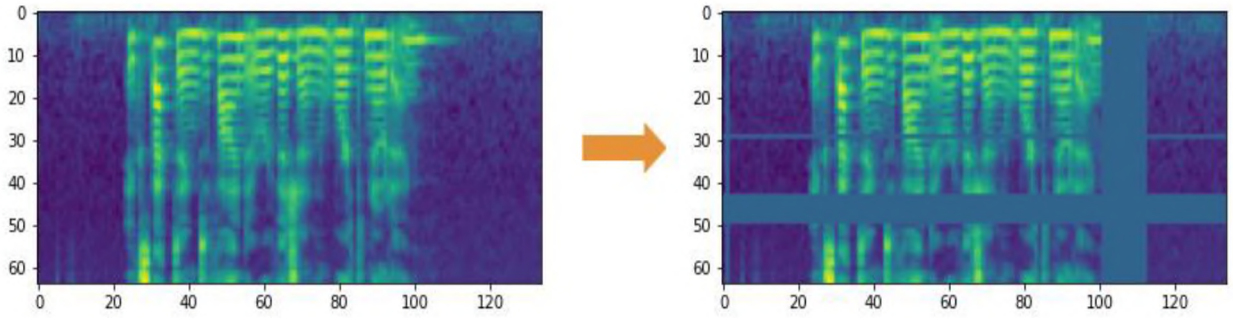

Рисунок 1.12 – Збільшення спектрограми

Нормальні перетворення, які використовували б для зображення, не застосовуються до спектрограм. Наприклад, горизонтальний переворот або незначний поворот суттєво змінить спектрограму та звук, який вона представляє. Замість цього ми використовуємо метод, відомий як SpecAugment де блокуємо ділянки спектрограми. Є два варіанти маски:

– частотна маска дозволяє довільно маскувати діапазон послідовних частот, додаючи горизонтальні смуги на спектрограмі;

– часова маска – подібна до частотних масок, за винятком того, що ми випадково блокуємо діапазони часу зі спектрограми за допомогою вертикальних смуг.

Доповнення необробленого аудіо. Воно має кілька варіантів.

1. Time Shift – зсув звуку вліво або вправо на випадкову величину (рис. 1.13):

– для таких звуків, як дорожній рух або морські хвилі, які не мають певного порядку, аудіо може обертатися;

– з іншого боку, для таких звуків, як людська мова, де порядок має значення, проміжки можна заповнити тишею.

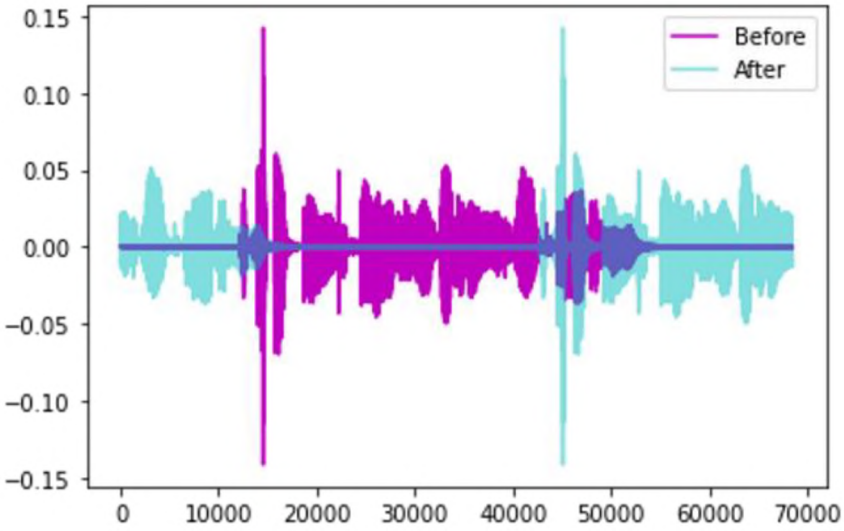

Рисунок 1.13 – Збільшення за допомогою Time Shift [4]

2. Pitch Shift – це випадкова зміна частоти частин звуку (рис. 1.14).

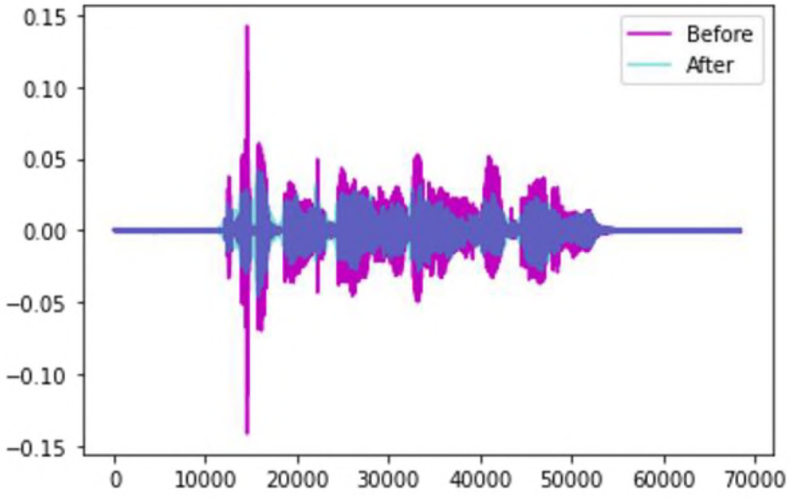

Рисунок 1.14 – Збільшення за допомогою Pitch Shift [4]

3. Time Stretch – довільне уповільнення або прискорення звуку (рис. 1.15).

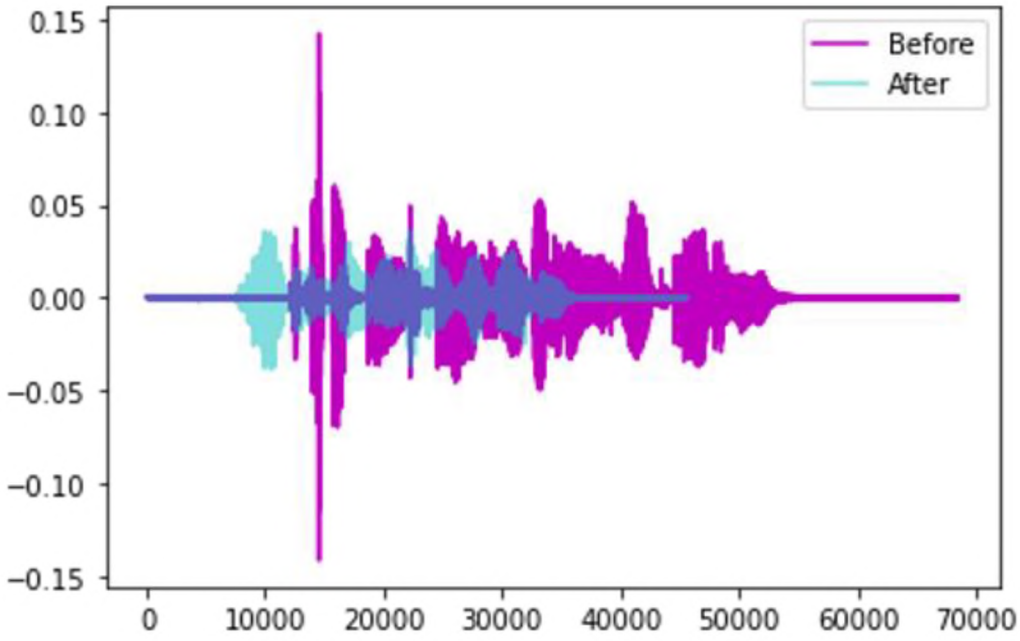

Рисунок 1.15 – Збільшення за допомогою Time Stretch [4]

4. Додати шум – додати до звуку випадкові значення (рис. 1.16).

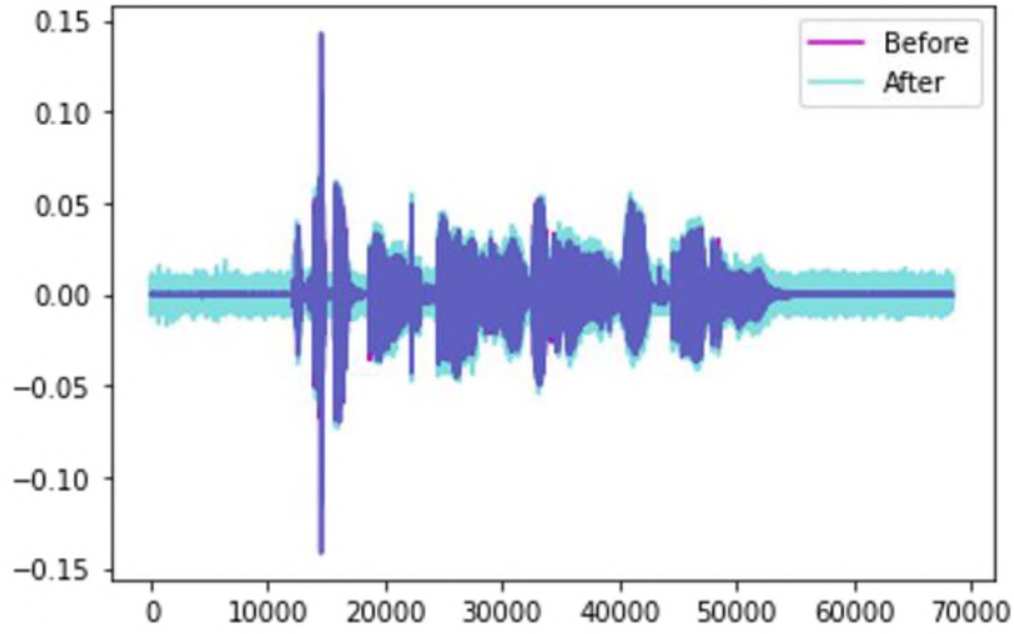

Рисунок 1.16 – Збільшення шляхом додавання шуму [4]

Подібно до того, як класифікація рукописних цифр за допомогою набору даних MNIST вважається проблемою типу «Hello World» для Computer Vision, можемо аналогічно розглядати завдання для глибокого навчання аудіо. Існує багато відповідних наборів даних для звуків різних типів. Ці набори даних містять велику кількість зразків аудіо, а також мітку класу для кожного зразка, яка визначає тип звуку на основі проблеми, яку намагаєтеся вирішити. Ці мітки класу часто можна отримати з деякої частини назви файлу аудіо зразка або з назви вкладеної папки, у якій знаходиться файл. Крім того, мітки класу вказуються в окремому файлі метаданих, зазвичай у форматі TXT, JSON або CSV. Що стосується більшості завдань глибокого навчання (рис. 1.17), треба виконати підготовку навчальних даних згідно схеми.

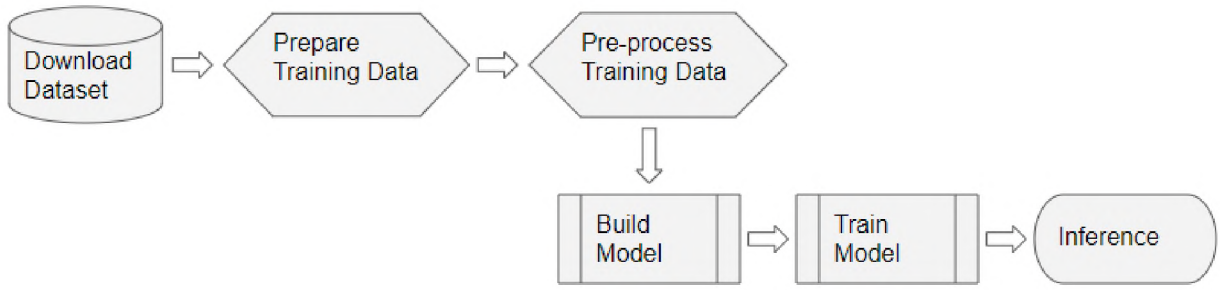

Рисунок 1.17 – Робочий процес глибокого навчання [4]

Навчальні дані для можуть бути досить простими:

– функції (X) – це шляхи аудіофайлів;

– цільові мітки (y) є іменами класів.

Оскільки набір даних має файл метаданих, який уже містить цю інформацію, можна використовувати її безпосередньо. Метадані містять інформацію про кожен аудіофайл (рис. 1.18).

Оскільки це файл CSV, можна використовувати бібліотеку Pandas для його зчитування. Можна підготувати дані про функції та мітки з метаданих. Це дає інформацію, необхідну для формування датасетів (рис. 1.19). Тобто, наявність файлу метаданих полегшила роботу.

	slice_file_name	fsID	start	end	salience	fold	classID	class
0	100032-3-0-0.wav	100032	0.0	0.317551	1	5	3	dog_bark
1	100263-2-0-117.wav	100263	58.5	62.500000	1	5	2	children_playing
2	100263-2-0-121.wav	100263	60.5	64.500000	1	5	2	children_playing
3	100263-2-0-126.wav	100263	63.0	67.000000	1	5	2	children_playing
4	100263-2-0-137.wav	100263	68.5	72.500000	1	5	2	children_playing

Рисунок 1.18 – Метадані

	relative_path	classID
0	/fold5/100032-3-0-0.wav	3
1	/fold5/100263-2-0-117.wav	2
2	/fold5/100263-2-0-121.wav	2
3	/fold5/100263-2-0-126.wav	2
4	/fold5/100263-2-0-137.wav	2

Рисунок 1.19 – Навчальні дані зі шляхами аудіофайлів
та ідентифікаторами класів

Багато наборів даних складаються лише з аудіофайлів, упорядкованих у структурі папок, з яких можна отримати мітки класів. Щоб підготувати навчальні дані в цьому форматі, виконуємо послідовність дій згідно (рис. 1.20).

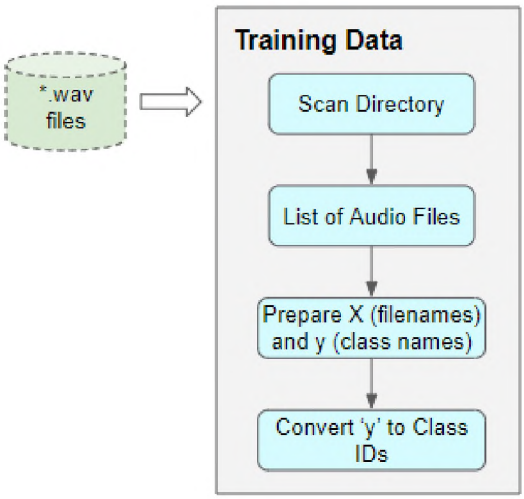

Рисунок 1.20 – Підготовка навчальних даних, коли метадані недоступні

Проскануємо каталог і підготуємо список усіх шляхів до аудіофайлів. Витягуємо мітку класу з імені кожного файлу або з імені батьківської вкладеної папки. Зіставляємо кожну назву класу з тексту на числовий ідентифікатор класу. З метаданими чи без них результат буде однаковим – функції, що складаються зі списку імен аудіофайлів і цільових міток, що складаються з ідентифікаторів класів.

Висновки до розділу 1

Моделі глибокого навчання працюють краще, ніж традиційні моделі класифікації звуку. Однак моделі глибокого навчання для аудіокласифікації здатні автоматично витягувати високовимірні характеристики зразків із великомасштабного набору даних без ручного вилучення ознак, якщо вхідні дані містять всю релевантну інформацію вихідних даних [9]. Моделі глибокого навчання здатні досягти вищих показників точності, ніж традиційні моделі. Це пов'язано з їх здатністю вивчати складні моделі та розпізнавати тонкі відмінності в аудіоданих. Більшість моделей глибокого навчання можуть навчатися швидше й точніше, ніж традиційні моделі, що робить їх ідеальними для класифікації та аналізу звуку в реальному часі.

Існують також деякі недоліки глибокого навчання. Моделі глибокого навчання потребують значних обчислювальних ресурсів, включаючи потужні графічні процесори та великий обсяг пам'яті, для навчання, що може бути дорогим і трудомістким. Крім того, навчання та оцінка систем аудіокласифікації з глибокими нейронними мережами (DNN) можлива лише з великою кількістю аудіоданих; без великого набору даних система може бути неуспішною [10]. Надмірна підгонка відбувається, коли модель була перенавчена на даних, які вона вже отримала, що призводить до низької продуктивності з новими невидимими даними [11]. З іншого боку, недостатня точність може виникнути, якщо модель не була достатньо

навчена. Моделі глибокого навчання значною мірою залежать від якості навчальних даних. Якщо дані шумні, упереджені та неповні, результати моделі можуть суттєво вплинути [12].

Використання моделей глибокого навчання вимагає обробки великих обсягів даних, що викликає занепокоєння щодо конфіденційності та безпеки даних. Якщо недобросовісні особи зловживають даними, це може призвести до серйозних наслідків, таких як крадіжка особистих даних, фінансові втрати та порушення приватного життя [13].

Тому, моделі глибокого навчання для класифікації аудіо набули широкого поширення через їх потенціал для вирішення складних завдань. Однак існують певні обмеження, такі як висока вартість обчислення, відсутність інтерпретації, надмірна підгонка, проблеми з конфіденційністю та безпекою даних, відсутність досвіду в області, залежність від якості даних і непередбачені набори даних.

РОЗДІЛ 2
ДОСЛІДЖЕННЯ АРХІТЕКТУР НЕЙРОННИХ МЕРЕЖ ДЛЯ ТЕХНОЛОГІЇ КЛАСИФІКАЦІЇ АУДІОКОНТЕНТУ

2.1 Класифікація архітектур нейронних мереж

Аудіокласифікація – це завдання аналізу аудіозаписів і присвоєння відповідних міток звуковому зразку [14]. Очевидно, що моделі глибокого навчання є більш надійними, коли навчаються з великою кількістю зразків для вивчення специфічних для завдання функцій, ніж традиційний метод машинного навчання, особливо для методів на основі трансформерів, без локального індуктивного зміщення в CNN [15]. Це особливо корисно під час роботи з аудіоданими, маркування яких може бути дорогим і трудомістким. Незважаючи на це, моделі глибокого навчання можуть адаптуватися до нових даних, не вимагаючи серйозних змін в архітектурі моделі. Загалом, глибоке навчання є потужним і гнучким підходом до класифікації аудіо, оскільки воно здатне перевершити традиційні підходи з точки зору точності, продуктивності, надійності, масштабованості та адаптивності. Застосування класифікації аудіо з використанням глибокого навчання стає все більш поширеним у різних областях, наприклад, комп'ютерний зір (CV), NLP, охорона здоров'я та промислова обробка сигналів. Згідно з [14], існує класифікація архітектур глибокого навчання для обробки аудіо (рис. 2.1).

Найбільш часто використовуваним критерієм оцінки для всіх моделей глибокого навчання класифікації аудіо є оцінка метрик оцінки. Метрики – це набір заходів, які використовуються для оцінки продуктивності моделей глибокого навчання, щоб визначити, наскільки добре модель може навчатися на даних навчання, і визначити сфери вдосконалення для оптимізації продуктивності моделі. Найпоширенішими з цих показників є точність, прецизійність, оцінка $F1$, крива ROC (робоча характеристика приймача) і оцінка AUC (площа під кривою ROC). Інші показники, такі як матриця

плутанини, чутливість і специфічність, також можуть бути використані для оцінки ефективності моделей глибокого навчання [16-18].

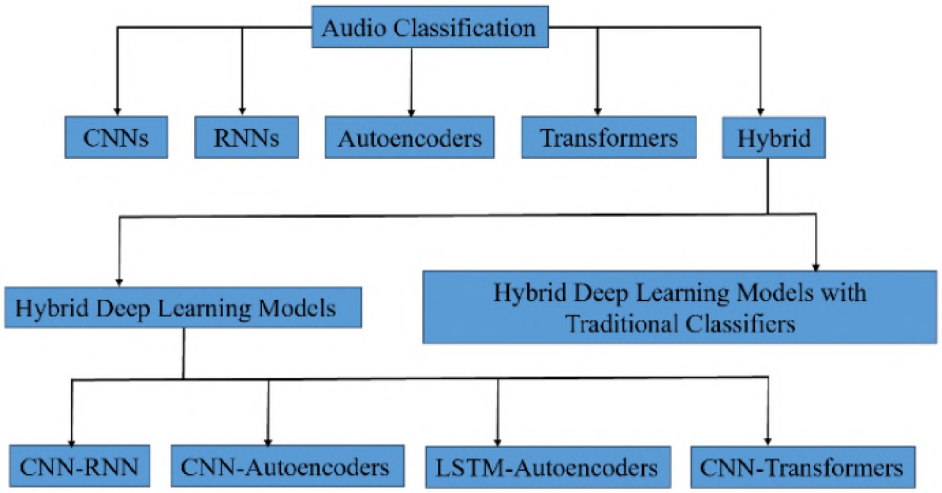

Рисунок 2.1 – Класифікація архітектур нейронних мереж

Точність – це відсоток правильно класифікованих алгоритмом точок даних порівняно з усіма точками даних, як показано в такому рівнянні:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} = \\ = \frac{TP + TN}{TP + TN + FP + FN}, \quad (2.1)$$

де TP, TN, FP і FN – позначають істинно позитивний, істинно негативний, хибно позитивний і хибно негативний, відповідно.

Точність прогнозування – це відношення правильно передбачених точок даних до загальної кількості прогнозованих точок даних і визначається за виразом:

$$\Pr ecision = \frac{TP + TN}{TP + TN + FP + FN}. \quad (2.2)$$

Прогнозування та чутливість вимірюють частку правильно класифікованих точок даних, що належать до певного класу, серед усіх точок даних, класифіковані як належність до цього конкретного класу в наборі даних і розраховані як:

$$\mathrm{Re}call\ /\ Sensitivity = \frac{TP}{TP\ +\ FN}. \tag{2.3}$$

Специфічність – це показник для вимірювання точності класифікатора в правильному визначенні всіх TN у наборі даних і може бути обчислений як:

$$Specificity = \frac{TN}{TN\ +\ FP}. \tag{2.4}$$

Оцінка F1 вимірює загальну продуктивність моделі, взявши середнє гармонійне значення точності та запам'ятовування, яке обчислюється як:

$$F1_{score} = \frac{2\cdot\ (\mathrm{Pr}ecision \cdot \mathrm{Re}call)}{\mathrm{Pr}ecision\ +\ \mathrm{Re}call} = \frac{TP}{TP\ +\ 0.5(FP\ +\ FN)}. \tag{2.5}$$

З іншого боку, площа під кривою (AUC) і крива робочих характеристик приймача (ROC) є графіками, які ілюструють показники TP і FP для даної моделі класифікації. Ці графіки використовуються для оцінки ефективності моделі щодо її здатності правильно класифікувати справжні позитивні та справжні негативні результати.

2.2 Архітектури на основі згортки

Згорткові нейронні мережі (CNN), зазвичай, використовуються для розпізнавання мовлення, класифікації аудіо, музичних рекомендацій, відокремлення аудіоджерел (тобто відокремлення різних аудіоджерел від одного запису) і багатьох інших областей застосування. CNN зробили революцію в галузі класифікації аудіо та створили широкий спектр застосувань. Вхідними сигналами до CNN є аудіо/мовні сигнали; однак методи на основі CNN, зазвичай, не використовують необроблені одновимірні (1D) сигнали як вхідні дані. На етапі попередньої обробки 1D-аудіо/мовні сигнали перетворюються з 1D-сигналу в 2D-сигнал. Потім 2-вимірне представлення аудіосигналу вводиться у модель CNN. Це перетворення 1D у 2D виконується для створення спектрограм. Проте FFT –

це оптимізоване застосування DFT, яке можна застосувати до дискретних сигналів у реальному часі та визначити як:

$$S(k)=\sum_{n=0}^{N-1}S(n)\exp(-j\frac{2\pi}{N}kn). \quad (2.6)$$

Спектр величин, $|S(k)|$ сигналу – це величина його частотного діапазону або номера частотної складової (k) при заданому номері вибірки (n) – це комплексне значення [82].

STFT – це вдосконалена форма перетворення Фур'є (FT), яка надає часові деталі сигналів як у часовій, так і в частотній областях. Використовуючи вікно в STFT, сигнал ділиться на сегменти часової області фіксованого розміру. Потім кожен сегмент піддається FT для виявлення різних особливостей сигналу. По суті, STFT використовує рівновіддалені, ідентичні та симетричні смугові фільтри в частотній області для аналізу сигналу. Математичне формулювання будь-якого сигналу $s(t)$ можна записати так:

$$S(f,t)=\int_{-T}^{T}s(\tau)w(\tau-t)\exp(-j2\pi f\tau)\,\partial\tau. \quad (2.7)$$

Щоб отримати таке представлення, сигнал $s(t)$ розбивається на сегменти за віконною функцією $w(t)$ у рівнянні (2.7). Довжина вікна має бути такою ж, як довжина сегментів сигналу, і передбачається, що сигнал виконує не змінюється (стаціонарно) всередині тривалості вікна. Спектрограма отримана за допомогою STFT шляхом взяття значення квадрата величини частотно-часового представлення [19, 20]:

$$Spectrogramm=|S(f,t)|^2 \quad (2.8)$$

Згідно п.1.2, Mel-спектрограму можна отримати за допомогою необробленого сигналу. На відміну від STFT, безперервне вейвлет-перетворення (CWT) не покладається на розміри вікна аналізу та зсув у часі для встановлення роздільної здатності за часом і частотою. Натомість CWT використовує фундаментальну форму хвилі, відому як «вейвлет», щоб

полегшити аналіз мовного сигналу. Цей метод передбачає згортання сигналу зі зсувом і стислі ітерації вейвлета, що досягається за допомогою часового зсуву. Підсумовуючи, необроблені 1D аудіосигнали або спектрограми можна використовувати як вхідні дані для моделі CNN.

CNN, як правило, складаються з кількох згорткових шарів, Rectified Linear Unit (ReLU), шарів об'єднання, повністю зв'язаних шарів (тобто щільних шарів) і шару Softmax. Згортковий рівень – це рівень у DNN, який застосовує фільтр операції згортки до вхідного сигналу для створення карт ознак, а потім передає отримані карти ознак на наступний рівень.

Для виділення значущих ознак із отриманих карт ознак у шарі згортки можна застосувати функцію активації ReLU. У ReLU від'ємні вхідні значення стають 0, а те саме вхідне значення залишається для невід'ємних чисел. Таким чином, ReLU підвищує нелінійність моделі, а також запобігає надмірній підгонці до навчених вхідних вибірок. ReLU є найпоширенішою функцією активації в CNN, хоча існують і інші функції активації, такі як сигмовидна активація, активація tanh та ін.

Рівень об'єднання – це рівень у CNN для зменшення розміру вхідних даних, зберігаючи важливу інформацію. Цього можна досягти шляхом об'єднання шарів, що зменшує розміри вхідної карти об'єктів. Це дозволяє моделі зосередитися на більш важливих функціях, таким чином підвищуючи точність. Найпоширенішими шарами об'єднання є максимальне та середнє об'єднання. Повністю підключений рівень у CNN використовується для з'єднання всіх нейронів попереднього рівня з усіма нейронами наступного шару. Повністю зв'язані шари зазвичай йдуть після згорткового шару. Повністю зв'язаний рівень – це рівень нейронної мережі з повністю зв'язаною конфігурацією, який використовується для обробки результатів згорткового шару після операції зведення.

Рівень softmax у CNN – це тип вихідного згорткового рівня, який використовує функцію активації softmax, яка є типом логістичної сигмоїдної функції, що використовується для перетворення довільних дійсних оцінок у

розподіл ймовірностей. У шарі softmax вихідний рівень створює розподіл ймовірностей за можливими класами для даного вхідного. Це дає змогу моделі передбачити, до якого класу належить певний вхідний сигнал на основі ймовірностей, які він створює. Це також хороший вибір для проблем класифікації з кількома класами, оскільки він створює ймовірності, сума яких дорівнює 1, що робить його ідеальним для призначення мітки класу вхідним даних. CNN дуже корисний для аналізу та класифікації зображень і дає багатообіцяючі результати для різноманітних програм аналізу мовлення, оскільки 1D аудіосигнали зазвичай перетворюються на 2D моделі (спектрограми) для аналізу за допомогою CNN.

На даний час, досить популярною моделлю є ConvNeXt [21]. Хоча вона розроблена для завдань зору, її було успішно адаптовано до класифікації аудіо на AudioSet [22] шляхом перетворення зразків аудіо на спектрограми log-mel і адаптації основи моделі ConvNeXt відповідно до вхідних аудіоданих. Це покращило сучасний рівень класифікації аудіо з використанням CNN, досягнувши кращої точності, ніж моделі типу PANN [23], маючи при цьому менше параметрів. Крім того, модель ConvNeXt-audio досягла позитивних результатів для завдань аудіосубтитрів і пошуку аудіо. Її особливістю є використання глибинної роздільної згортки (Depthwise Separable Convolution, DSC). Це особливий тип згорткової операції, який використовується для зменшення кількості обчислень і параметрів в нейронних мережах згортання, що робить їх більш ефективними в плані обчислень і пам'яті. При класичній згортці кожен фільтр обробляє весь вхідний об'єм даних (наприклад, усі кольорові канали зображення спектрограми). На відміну від цього, DSC розбиває цей процес на наступні дві частини.

1. DSC – на цьому кроці окремий фільтр застосовується до кожного каналу вхідних даних. Це означає, що замість обробки всіх каналів відразу для кожного каналу використовується свій фільтр.

2. Точкова згортка (Pointwise Convolution) – застосовується згортка

1×1, яка комбінує вихідні дані DSC з усіх каналів. Цей крок відповідає за створення нових ознак шляхом лінійної комбінації вихідних даних DSC.

Застосування DSC дозволяє значно зменшити кількість обчислень і параметрів, зберігаючи при цьому ефективність вилучення ознак. Це робить її популярною в мобільних та вбудованих системах, де ресурси обмежені, наприклад, MobileNet.

У [24] пропонується метод розширеної згортки з інтервалами навчання (Dilated Convolution with Learnable Spacings, DCLS). За допомогою простої заміни DSC на DCLS, яку можна зробити автоматично для всіх шарів моделі за допомогою певного коду, що наведений у Додатку А). DCLS емпірично довів свою ефективність для кількох завдань CV з використанням ImagNet1k [25]. Це призвело до створення моделі ConvNeXt-dcls [26] і моделі ConvFormer-dcls [27].

Системи аудіотегування донедавна в основному базувалися на згорткових нейронних мережах з адаптацією трансформаторів зору до обробки звуку. Моделі на основі PANN (наприклад, CNN), зокрема, містять блоки звичайних ядерних шарів згортки 3×3 [23]. У [28] моделі, подібні до PANN, були покращені з точки зору точності, розміру моделі та швидкості висновку шляхом додавання залишкових зв'язків та зміни розмірів ядра, кроку та заповнення за допомогою «параметра часового розміру, що зменшується». Інші ефективні архітектури CNN, такі як EfficientNet [29], також були протестовані на аудіотеги. У [30] використовувалися рівні DSC, що призвело до значного зниження складності моделі разом із підвищенням продуктивності. У [22] комбінація CNN і PANN також призвела до значного зменшення розміру моделі (приблизно на 60 %), водночас спостерігаючи приріст у продуктивності. У цьому останньому дослідженні ConvNeXt було адаптовано для виконання завдання аудіотегування в AudioSet. Він працював краще або нарівні з трансформерами AST [31] і PaSST-S [32].

2.3 Використання архітектури рекурентної мережі

RNN можуть приймати послідовність даних як вхідні дані та обробляти один елемент за раз і витягувати функції, тобто фіксувати часовий контекст даних. RNN можна використовувати для класифікації аудіосигналів, наприклад виявлення мови або музики в звукозаписі. У даному випадку, RNN навчені розпізнавати шаблони в аудіосигналі з часом. Послідовно обробляючи аудіокадри, RNN може навчитися розпізнавати характеристики різних звуків, дозволяючи йому точно класифікувати аудіо. RNN дозволяють комп'ютерам розуміти вміст аудіоданих завдяки їхній здатності вловлювати часовий контекст аудіосигналу. RNN можна використовувати для класифікації аудіосигналів шляхом виділення характеристик із сигналу, таких як частота, амплітуда та фаза. Крім того, RNN можна використовувати для класифікації музичних жанрів шляхом аналізу акустичних особливостей пісні, таких як ритм, висота та тембр. Хоча RNN успішні для послідовностей даних, таких як мовні сигнали, вони мають короткочасну пам'ять. У різних випадках нам можуть знадобитися короткострокові або довгострокові залежності. Для більш довгих послідовностей вхідних даних довготривала короткочасна пам'ять (LSTM), яка є типом RNN, використовується для запам'ятовування довготривалих залежностей у вхідних даних.

2.4 Архітектури автоенкодерів

Автоенкодер – використовує нейронні мережі для вивчення стислих даних (кодувань) немаркованих даних. Він складається з двох кроків (рис. 2.2): кодер спочатку перетворює вхідні дані в представлення нижчого розміру, а потім декодер відтворює вихідні дані із закодованого вхідного представлення. Автоенкодери використовують зворотне поширення, щоб дізнатися кодування вхідних даних, яке можна використовувати для

відтворення вхідного представлення з мінімальною втратою інформації. Це кодування називається «приховане представлення» або «прихований простір» даних. Автоенкодери ефективні для вивчення важливих функцій набору даних із скороченим набором функцій. Крім того, вони ефективні у виявленні викидів і аномалій у даних. Автоенкодери спочатку кодує аудіофункції у стиснене представлення, а потім використовує закодовані функції для навчання моделі класифікації. Потім цю модель можна використовувати для класифікації аудіосигналів на основі їх стислих представлень. Автоенкодери особливо корисні для завдань класифікації аудіо з великими вхідними даними, наприклад класифікація мови або класифікація музики. Вони також можуть бути використані для виявлення тонких змін у зразках аудіо, таких як зміни висоти чи темпу, які потім можуть бути використані для розрізнення різних класів зразків аудіо.

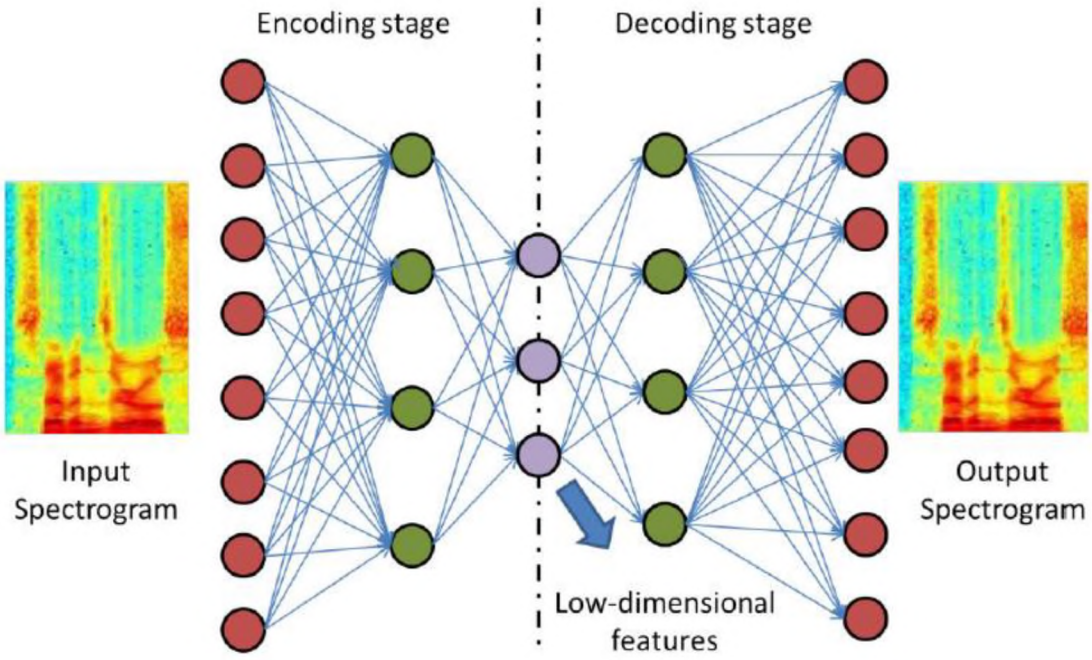

Рисунок 2.2– Архітектура автоенкодера [33]

На етапі попередньої обробки автоенкодери можна використовувати перед іншими моделями глибокого навчання, такими як CNN, для підвищення точності класифікації. Автоенкодери також можна

використовувати для зменшення складності вхідних даних, це допоможе зменшити кількість часу та ресурсів, необхідних для навчання та висновків.

2.5 Архітектури-трансформери

Трансформери є типом DNN з механізмом уваги (рис. 2.3). Останнім часом трансформери були застосовані до звукових сигналів за допомогою трансформерів спектрограм. Спочатку вони мали успіх Transformers [34] у сфері NLP і CV, вони були адаптовані до звукової модальності та отримали найсучасніші характеристики. Зокрема, нові методи на основі трансформерів були введені з використанням механізму уваги, щоб значно покращити класифікацію аудіо, дозволяючи моделі знати про глобальний контекст [35].

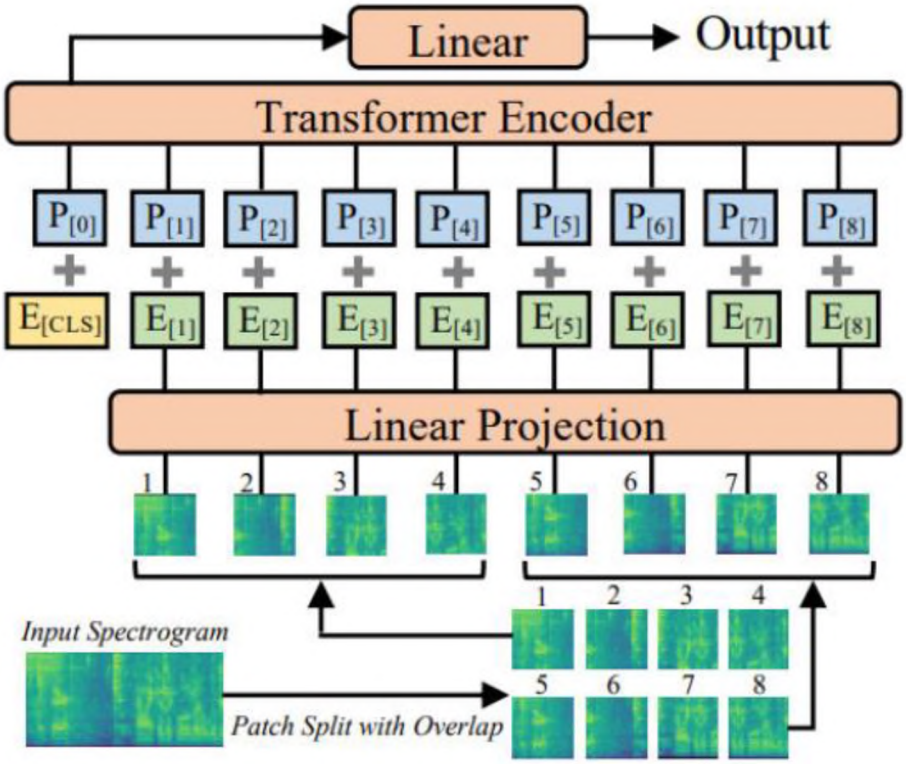

Рисунок 2.3 – Архітектура трансформера [36]

Методи на основі трансформерів, порівняно з CNN, можуть обробляти

дисперсію вхідної довжини, що є однією з переваг трансформерів. Цього можна досягти завдяки здатності багатовходового механізму уваги та працювати зі змінною довжиною вхідних послідовностей. Таким чином, трансформерні методи можуть швидко отримувати корисну інформацію глобального контексту, незалежно від тривалості аудіо. Трансформери потребують великої кількості навчальних даних. У випадках, коли дані обмежені, багато аудіоперетворювачів використовують моделі попереднього навчання та тонке налаштування. Patchout FaSt Spectrogram Transformer (PaSST) [37] і Audio Spectrogram Transformer (AST) [38] є двома провідними моделями для класифікація звуку. AST [38] є першою моделлю трансформера для аудіокласифікації та адаптує вагові коефіцієнти перед навчанням із мережі класифікації зображень трансформерів зору (ViTs) [39]. PaSST зменшує обчислення та складність пам'яті під час навчання трансформерів для звукової сфери.

Використання трансформерів для аудіокласифікації привернуло увагу завдяки їх багатообіцяючим результатам. Для ефективного використання трансформерів зазвичай враховуються такі кроки. Перші аудіосигнали перетворюються на візуальні спектрограми за допомогою методів виділення ознак (STFT, Mel-спектрограм, спектрограм log-Mel і MFCC). Спектрограми містять частотно-часову інформацію та служать вхідними даними для трансформера. Потім спектрограми часто ділять на невеликі сегменти фіксованої довжини, відомі як «патчі». Потім ці патчі розглядаються як окремі вхідні токени, подібні до слів у NLP. Тоді архітектура трансформера складається зі стеку шарів кодувальника, кожен з яких складається з механізмів самоконтролю та прямої нейронної мережі. Оригінальні моделі трансформера не мають згорткового шару, але містять шари прямого зв'язку. Потім застосовуються залишкове з'єднання та нормалізація шару. Механізм самоконтролю допомагає моделі вловлювати залежності між різними ділянками спектрограми, як і в оригінальній моделі трансформера. Глобальне середнє об'єднання та максимальне об'єднання – це загальні методи, які

використовуються для агрегування інформації в часовому вимірі після шарів кодувальника та забезпечення представлення звуку фіксованої довжини. Це зменшує часову розмірність вхідного звуку. Потім об'єднане представлення подається в один або кілька щільних шарів (повністю зв'язаних шарів). Вихідні дані цих шарів потім пропускаються через функцію активації softmax, яка створює ймовірності класів як вихідні дані. Нарешті, шляхом попереднього навчання на великих обсягах немаркованих аудіоданих за допомогою таких методів, як самоконтрольоване навчання або порівняльне навчання, перетворювачі для аудіокласифікації можуть отримати корисні звукові представлення. Потім ці представлення можна використовувати для точного налаштування моделі на конкретні завдання класифікації аудіо з використанням позначених даних.

У [35] представлено трансформер спектрограми, який є комбінацією різних стратегій виділення ознак для класифікації екологічних звуків. Вони вперше перетворили аудіосигнали в зображення спектрограми за допомогою ШПФ. Потім застосували різні блоки уваги, щоб покращити виділені характеристики з часової та частотної областей за допомогою трансформерного кодера. Вони протестували різні механізми уваги та отримали найкращі результати, використовуючи часово-частотний блок уваги на ESC-50 без попереднього навчання. У [40] досліджувано вплив підходу об'єднання функцій на рівні патча з використанням ViT для різних завдань класифікації звуку. Спочатку вони отримали спектрограму Mel аудіосигналу та розділили її на фрагменти, а потім ввели ці фрагменти в кодер ViT для об'єднання функцій. З об'єднаних патчів потім генерувалися нові патчі та вводилися в MLP для класифікації. Вони досліджували ефективність цього підходу в різних завданнях класифікації за допомогою ImageNet і AudioSet попередньо навчених вагових коефіцієнтів моделі.

У [16] проведено комплексний аналіз використання трансформерів проти різних базових моделей CNN для класифікації міського звуку. Вони досліджували продуктивність трансформера, базову лінію CNN, DenseNet,

Inception-V3 і варіанти попереднього навчання разом із доповненням даних. Як трансформер для порівняння використовувався трансформер спектрограми, запропоновано у [36]. Експерименти на UrbanSound8K, ESC-50 і ESC-10 продемонстрували, що модель трансформера, яка використовує передачу навчання від AudioSet, досягла найкращої точності.

У [41] розглянуто проблему ефективності проти великих параметрів. Багато моделей глибокого навчання (CNN) для аудіокласифікації вимагають великої кількості параметрів, що означає великий обсяг пам’яті для тестування. Однак багато мікроконтролерів не мають такої пам'яті для роботи. Щоб подолати цю проблему, вони запропонували крихітний перетворювач для аудіокласифікації за допомогою трансформера на основі BERT (розробленого для мовної моделі), який навчається на зображеннях Mel-спектрограми. Вони також досліджували різні методи доповнення даних. Запропонований крихітний трансформер містить близько 6000 параметрів і досягає точності 99,85 % для класифікації звуку навколишнього середовища на ESC-50. У [42] вдосконалено своє попереднє дослідження AST зі стратегією самоконтрольованого навчання. Хоча трансформери ефективні, вони вимагають великої кількості навчальних зразків для вивчення карт функцій. У невеликих областях, де навчальні вибірки обмежені, трансформери можуть бути не такими ефективними, як CNN. Тому автори намагалися пом'якшити цю проблему, інтегрувавши структуру самоконтрольованого навчання в AST. Вони запропонували генеративну модель, яка вивчає замасковані плями спектрограми. Експерименти показали, що запропонована модель підвищила точність немаркованих аудіоданих із наборів даних AudioSet і Librispeech.

У [43] розглянуто проблема ефективного навчання моделей трансформерів. Оскільки трансформери вимагають великої кількості зразків аудіосигналу для навчання, складність навчання зростає квадратично зі збільшенням вхідного сигналу. Щоб полегшити це, вони представили новий підхід до оптимізації та регулярізації трансформерів на аудіоспектрограмах,

використовуючи трансформер спектрограми з викривленнями для забезпечення ефективного навчання. Експерименти на Audioset показали, що їхній трансформер спектрограми patchout перевершує CNN як за продуктивністю, так і за швидкістю навчання.

У [44] запропонувано swin transformer, який містить метод попереднього навчання з самоконтролем для класифікації музичних жанрів. Трансформери Swin об'єднують менші фрагменти, заглиблюючись у архітектуру. Спочатку вони перетворили аудіосигнали на спектрограми та застосували доповнення даних, а потім застосували попереднє навчання під самоконтролем, використовуючи перехідне навчання та попереднє навчання. Вони вводять ці патчі в трансформер Swin. Експерименти на GTZAN показали, що поворотний трансформер підвищив точність.

Через потребу в даних трансформерів багато трансформерів для аудіокласифікації використовують попередньо підготовлені моделі з області зображень, такі як ImageNet. У [45] представлено вирішення даної проблеми та запропонував трансформер із самоконтролем під назвою ASiT, який зменшує залежність від попередньо навчених моделей із області зображення. Зокрема, загальні аудіорепрезентації отримані з локальним і глобальним контекстом шляхом застосування групового замаскованого моделювання та самоперегонки. Вони оцінили ASiT для класифікації аудіо/мовлення та досягли найсучасніших показників.

У [46] представлено Causal Audio Transformer (CAT), модель, яка використовує багатофункціональне виділення функцій Multi-Resolution Multi-Featured (MRMF) з акустичним блоком уваги для покращеного моделювання звуку. CAT також включає причинно-наслідковий модуль для зменшення надмірної підгонки, полегшення передачі знань і покращення інтерпретації. Експерименти показують, що CAT досяг кращих або порівнянних результатів порівняно з іншими моделями на ESC50, AudioSet і UrbanSound8K і може бути легко адаптований до інших моделей на основі трансформерів.

Частково успіх пояснюється глобальними сприйнятливими полями в Transformers для захоплення далекого контексту в аудіосигналах. Існуючі моделі аудіотрансформаторів успадковують структуру відомого Vision Transformer (ViT), головним чином тому, що одна з його найбільш широко використовуваних функцій, Mel-Spectrogram, має той самий формат, що й зображення. Замість того, щоб осі x і y переносити просторову інформацію в моделюванні зображення, вісь x спектрограми Mel позначає часову інформацію, тоді як вісь y несе дискретну частотну інформацію для аудіовходів.

Незважаючи на досягнення чудової продуктивності, у цих аудіотрансформерах все ще є не вирішені проблеми.

1. Зазвичай, застосовані акустичні представлення використовують різні частотно-часові перетворення та містять акустичну семантику різних масштабів і різної деталізації, що навряд чи може бути ефективно захоплені ViTs за допомогою звичайних механізмів самоуваги, патч-семплінгу та вбудовування.

2. Успішні проекти у завдання CV, таких як ResNET, MixUP, більш схильні до надмірної підгонки та менш узагальнені в акустичній області.

3. Вибір функцій і вивчення репрезентації мають вирішальне значення в CV, але на них часто не звертають уваги в акустичному моделюванні.

Зважаючи на вказані проблеми, можливо використання Causal Audio Transformer (CAT), який включає вилучення функцій Multi-Resolution Multi-Filter (MRMF), акустичну увагу та причинно-наслідковий модуль. Спектрограми, як стандартні вхідні дані в моделях аудіо, дискретизуються за допомогою перетворення Фур'є, що призводить до природного компромісу між часовою та частотною роздільною здатністю. CAT врівноважує компроміс, витягуючи всеохоплюючі патчі часових частот із різними роздільними здатностями та фільтрами, які пізніше поєднуються з позиційними 3D-вбудовуваннями. Потім пропонується акустична увага для

ефективного вилучення семантики з таких представлень, використовуючи патчі ознак як вхідні дані. Патчі від різних фільтрів рівномірно розподіляються між заголовками уваги, тоді як ми обчислюємо попарні уваги між патчами з різною роздільністю, але в межах однакових часових рамок, що дозволяє обмінюватися інформацією з різною деталізацією. Далі вводиться причинно-наслідковий модуль, щоб встановити необхідний і достатній зв'язок між вивченим представленням і прогнозованими мітками на основі контр-фактичних міркувань [47]. Він поширюється на контекст класифікації аудіо, де надається нижня межа ймовірності необхідності та достатності (PNS) з точки зору інтервенційного розподілу. Оскільки таку нижню межу можна оцінити лише на основі справжнього розподілу, пропонується причинно-наслідковий модуль, який вивчає відображення інтервенційного розподілу в набір даних спостереження (тобто той, який маємо), щоб пом'якшити надмірну підгонку, покращити інтерпретативність і забезпечити кращу передачу знань. CAT досягає продуктивності SOTA на ESC50, AudioSet і UrbenSound8K. Таким чином, реалізується CAT з MRMF (рис. 2.4), який має функції вилучення та акустичного моделювання. Причинно-наслідкові втрати з блоком реконструкції, який явно вимірює якість функції за допомогою PNS, зменшує надмірну підгонку та покращує передачу знань між різними наборами даних.

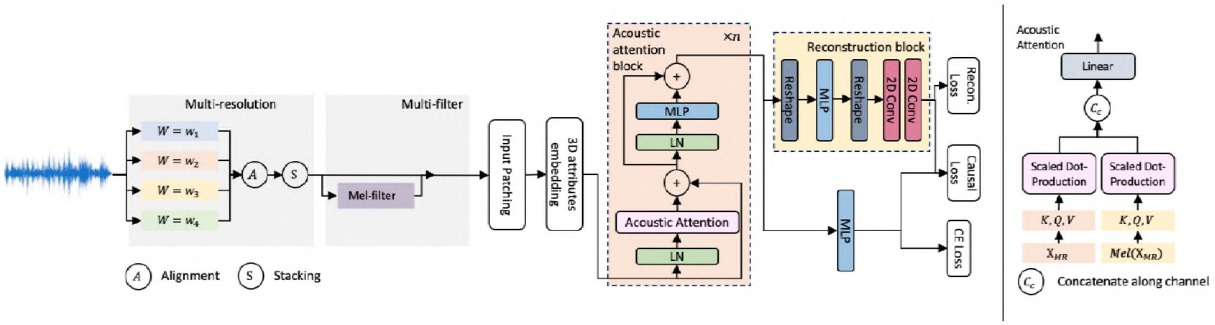

Рисунок 2.4 – Варіант реалізації CAT

Кілька наступних досліджень зосереджені на покращенні ефективності моделі: PaSST пропонує механізм виправлення, а HTS-AT використовує

ієрархічну основу. Щоб ще більше підвищити продуктивність, MBT представляє візуально-акустичне злиття, а PLSA представляє модельно-агностичну структуру. Однак мережеві структури в останніх дослідженнях значною мірою запозичені з ванільних трансформаторів, спочатку запропонованих для завдань NLP і CV, і вони більш схильні до надмірної підгонки та менш узагальнені для акустичних даних. Поняття причинності вперше введено в графічних імовірнісних моделях. Хоча причинно-наслідковий висновок є відносно новою концепцією в класифікації аудіо, він продемонстрував прогрес у машинному навчанні, яке можна інтерпретувати, і в навчанні представлення.

Моделювання причинно-наслідкових зв'язків серед генеруючих факторів значно сприяє вивченню репрезентативних ознак. У [48] використовують контрфактичну інформацію, яка допомагає передавати знання в різних областях. [49] доводить, що нижня межа вивченого представлення є необхідною та достатньою умовою для передбачення мітки. Однак така нижня межа знаходиться в умовах втручання, і тому її неможливо оцінити безпосередньо, не знаючи справжнього розподілу.

Таким чином, Causal Audio Transformer (CAT) витягує патчі функцій MRMF за допомогою 3D позиційного вбудовування. Потім патчі функцій надсилаються як вхідні дані для акустичного спостереження. Потім пропонується причинно-наслідковий модуль, щоб зменшити надмірну підгонку та покращити інтерпретацію та передачу знань.

У [50] запропоновано один із ранніх методів класифікації музичних жанрів на основі трансформерів. Спочатку автори перетворили аудіодані на Mel-спектрограми з логарифмічної амплітудою та застосували оригінальний багатовходовий трансформер уваги до GTZAN. Результати показали, що трансформери можна використовувати для класифікації музичних жанрів.

2-направлений трансформер із замаскованою прогностичною моделлю для класифікації музичних жанрів. Трансформер також використовує попередню обробку під назвою Pitch-to-Vector (Pitch2Vec), яка перетворює

аудіосигнали на векторні послідовності. Потім замаскований інтелектуальний кодер витягує 2-направлені представлення про музику за допомогою стратегії неконтрольованого навчання. Експерименти показали, що трансформер може досягти високої точності.

У [51] автори запропонували архітектуру на основі трансформера для обробки необроблених аудіосигналів без необхідності згорткових шарів для класифікації аудіо на наборі даних FSD50K. У порівнянні з AST, у цьому підході необроблені сигнали вводяться в трансформер для завдання класифікації. Вони досліджували продуктивність архітектури на основі трансформера, порівнюючи її з архітектурою CNN. Перевага їхньої архітектури полягає в тому, що не використовується попереднє навчання без нагляду, а також використовуються методи об'єднання з CNN та ідеї багатошвидкісної обробки сигналу з вейвлетів.

Ієрархічний семантичний аудіоперетворювач (HTS-AT) для вирішення проблеми великого обсягу пам'яті/тривалого часу навчання, а також потреби в попередньо навчених моделях із області зображення. HTS-AT використовує ієрархічну структуру, щоб зменшити кількість необхідної пам'яті та часу навчання. Він також використовує семантичний модуль для відображення в карти функцій класу для виявлення звукових подій. Оцінки AudioSet і ESC50 продемонстрували, що HTS-AT може досягти найсучаснішої продуктивності розпізнавання мовних команд.

У [52] досліджували продуктивність PaSST проти двох CNN у налаштуваннях навчання з нульовим ударом. У нульовому навчанні модель намагається передбачити невидимі класи за допомогою адаптованих представлень класів. Зокрема, автори досліджували продуктивність PaSST з двома моделями CNN (VGG і спеціальна модель CNN). Експерименти з 3-ма наборами даних, а саме AudioSet, ESC-50 і OpenMIC, показали, що PaSST перевершив аналоги CNN у нульовому навчанні в усіх завданнях.

У [53] об'єднав ієрархії функцій із різними масштабами в AST, що називається багатомасштабним AST (MAST) для класифікації звуку. Цей

MAST передбачає багаторазове патчування (розподіл спектрограм на патчі); у міру поглиблення мережі генеруються нові патчі та збільшуються розміри патчів. Таким чином, кількість патчів зменшується. Таким чином, виходить пірамідальна структура. Початкові рівні MAST обробляють високу часову роздільну здатність/низьке вбудовування, тоді як більш глибокі рівні отримують інформацію високого рівня. Експерименти показали, що MAST працює краще, ніж AST.

Інший MAST включає ієрархічну модель навчання в AST. Цей MAST використовує як 1D, так і 2D операції об'єднання, щоб зменшити розміри функцій і кількість токенів. Експерименти на різних наборах даних показали, що запропонований MAST ефективний для різних завдань класифікації звуку. Фреймворк на основі трансформера без згортки, який вивчає представлення з мультимодальних даних, таких як відео, аудіо та текст, у кінетичному середовищі. Цей фреймворк отримує дані з відео, аудіо та тексту та зливає їх у мультимодальні представлення за допомогою трансформера з немаркованих даних. Аудіодані вводяться як необроблені вейвлети, а аудіодані налаштовуються відповідно до AudioSet. Вони провели кілька експериментів, щоб перевірити ефективність своєї системи, виявивши, що вона ефективна.

2.6 Гібридні моделі

Гібридні моделі містять комбінацію різних архітектур глибокого навчання. Їх метою є поєднання сильних сторін різних моделей глибокого навчання. Наприклад, багато методів RNN спочатку використовують CNN для вилучення ознак із просторових вхідних даних, таких як зображення спектрограм, а потім вихідні дані CNN передаються в мережу RNN (LSTM). У той час як повторювані шари навчені ідентифікувати шаблони в часових вхідних даних, таких як аудіо або дані часових рядів. Приклад архітектури

CNN-LSTM наведено на рис. 2.4.

Подібним чином автокодери можна комбінувати з RNN (тобто LSTM) або CNN. Трансформери також можна комбінувати з CNN. Ця комбінація моделей забезпечує ефективну обробку як просторової, так і часової інформації. Ці гібридні моделі використовуються в різних додатках, наприклад, класифікація музичних жанрів, класифікація звуків навколишнього середовища, класифікація акустичних сцен і так далі. Також існують гібридні моделі глибокого навчання з традиційними класифікаторами для класифікації звуку.

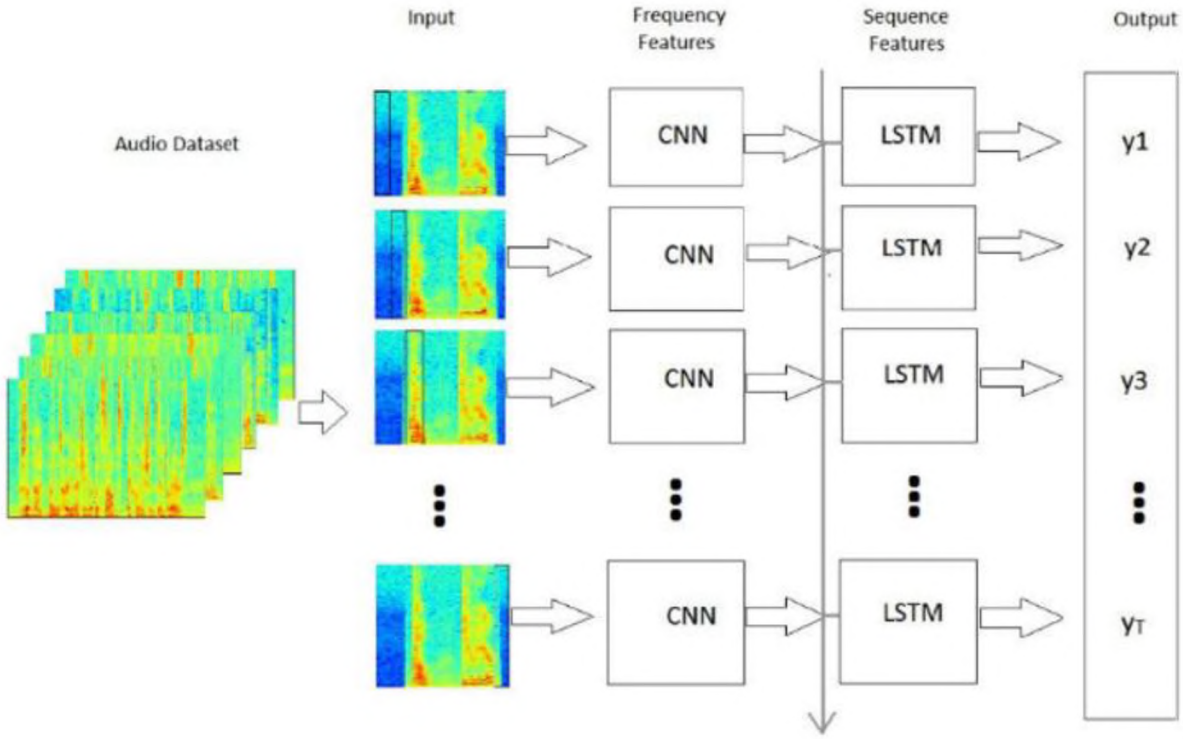

Рисунок 2.4 – Приклад архітектури CNN-LSTM [54]

Гібридні моделі з традиційними класифікаторами використовують моделі глибокого навчання (CNN і RNN) для вилучення ознак. Замість класифікації в наскрізній мережі витягнуті ознаки зводяться та вводяться в традиційні класифікатори.

Висновки до розділу 2

Моделі глибокого навчання дозволяють виявляти аномалії у звукових сигналах, таких як фоновий шум або інші небажані звуки. Потреба в автоматизованих системах класифікації звуків зростає, оскільки їхнє значення в повсякденному житті не можна недооцінювати. Такі системи використовуються в широкому діапазоні сфер, таких як відеоспостереження, голосова допомога, чат-боти, інтелектуальні пристрої безпеки та різноманітні реальні середовища, включаючи інженерні, промислові, побутові, міські, дорожні та природні.

В роботі досліджено архітектури нейронних мереж, такі як CNN, RNN, автоенкодери, трансформери та гібридні моделі. Для кожного типу архітектури також надається детальна інформація про компоненти архітектури. Розглянуті варіанти архітектур можуть застосовуватись для різноманітних завдань класифікації аудіо, наприклад, як мова, шум, музика, емоції, звуки навколишнього середовища, акустична сцена та ін. Значний успіх застосування трансформерів для завдань NLP спонукав до більш глибокого аналізу властивостей їх архітектур.

Охоплення різних моделей глибокого навчання для класифікації звуку, включно з гібридними моделями, робить його цінним ресурсом для тих, хто шукає розуміння різних архітектур.

В ході досліджень встановлено що при реалізації технології класифікації аудіоконтенту потрібно оцінити вплив архітектури та гіперпараметрів на продуктивність нейронної мережі. При цьому доцільно використовувати ознаки, що максимально можна виділити з аудіоданих.

РОЗДІЛ 3
РЕКОМЕНДАЦІЇ ЩОДО РЕАЛІЗАЦІЇ ТЕХНОЛОГІЇ КЛАСИФІКАЦІЇ АУДІОКОНТЕНТУ

3.1 Інструментарій для реалізації програмного коду

На даний час, в Python для роботи з аудіо можна використовувати кілька бібліотек:

– PyAudio – кросплатформова бібліотека для форматування та синтезу аудіо інформації;

– Speech Recognition – бібліотека, що є обгорткою над багатьма популярними сервісами/бібліотеками розпізнавання мовлення;

– Google Text To Speech (sTTS) – бібліотека для перетворення рядка на mp3 файл з мовою;

– LibROSA – яка застосовується для аналізу звукових сигналів і орієнтована на музику.

Надалі будемо, в основному, використовувати LibROSA. Розглянемо деякі інструменти цієї бібліотеки.

LibROSA – це Python-модуль для аналізу звукових сигналів, призначений для роботи з музикою. Він включає все необхідне для створення системи MIR (пошук музичної інформації) та детально задокументований разом із безліччю прикладів та посібників. Для роботи нейронної мережі з аудіо їй потрібні якісь ознаки аудіосигналу, щоб на основі цих ознак побудувати навчання. Згідно п.1.2, розглянемо найбільш використовувані та корисні ознаки.

Спектрограма показує залежність спектральної потужності сигналу від часу (рис. 3.1). Тобто показує інтенсивність частот у часі. Це дає уявлення часу осі x, частоти осі y, а відповідні амплітуди представляються кольором. Щоб завантажити аудіосигнал, використовуємо бібліотеку LibROSA, функцію load():

```
x, sr = librosa.load('audio_path')
# x - довжина звукової доріжки, # sr - частота дискретизації
```

Далі, щоб визначити спектр сигналу, використовуємо функцію librosa.stft():

```
X = librosa.stft(x) # обчислюємо спектр сигналу
```

Виводимо результат за допомогою бібліотеки matplotlib:

```
# змінюємо шкалу на децибели (для зручного відображення)
Xdb = librosa.amplitude_to_db(abs(X))
# виводимо спектрограму на екран
plt.figure(figsize=(14, 5)) #задаємо розмір
librosa.display.specshow(Xdb, sr=sr, x_axis='Time', y_
axis='Hz') #відобразимо спектрограми
plt.colorbar() #виведемо колірну шкалу
plt.show() #виводимо графік
```

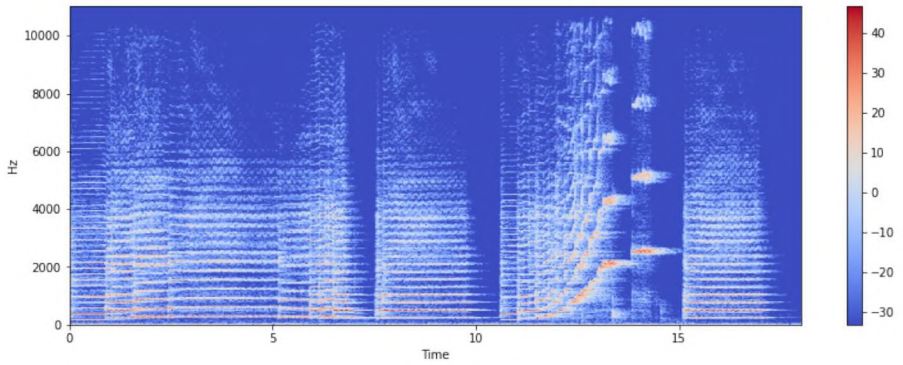

Рисунок 3.1 – Спектрограма

Частота перетину нуля (zero crossing rate) – рис. 3.2. Це частота зміни знаку сигналу, тобто частота, з якою сигнал змінюється з позитивного на негативний і назад.

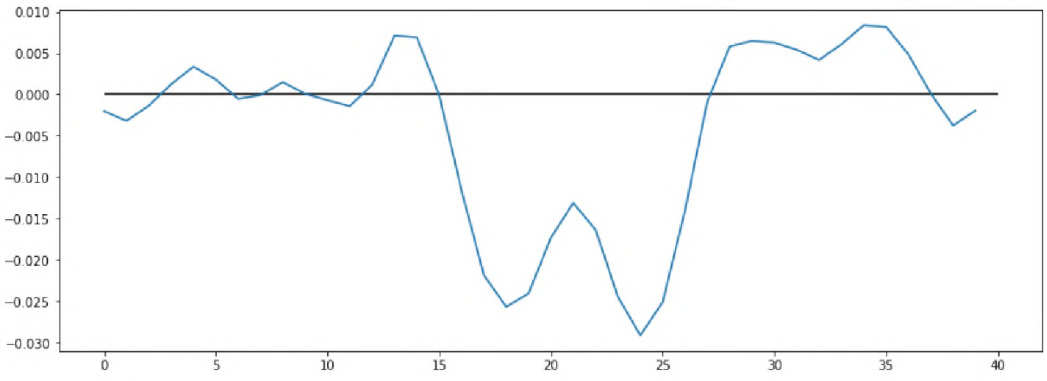

Рисунок 3.2 – Zero crossing rate

Наприклад, для металу та року цей параметр зазвичай вищий, ніж для інших жанрів, через велику кількість ударних. Для підрахунків перетину нуля використовуємо функцію zero_crossings():

```
# Розраховуємо перетину нуля
zero crossings = librosa.zero crossings(x[0:40], pad=False)
# Відображаємо результати
print(sum(zero crossings)) # Сумарна кількість перетинів
# Наявність перетину в кожній точці 8
```

Спектральний центроїд (рис. 3.3) вказує, де розташований центр мас звуку, і розраховується як середньозважене значення всіх частот. є гарним показником яскравості звуку, що широко використовується як автоматичний захід музичного тембру. У блюзових композиціях частоти розподілені рівномірно, у металі спектроїд лежить ближче до кінця спектра.

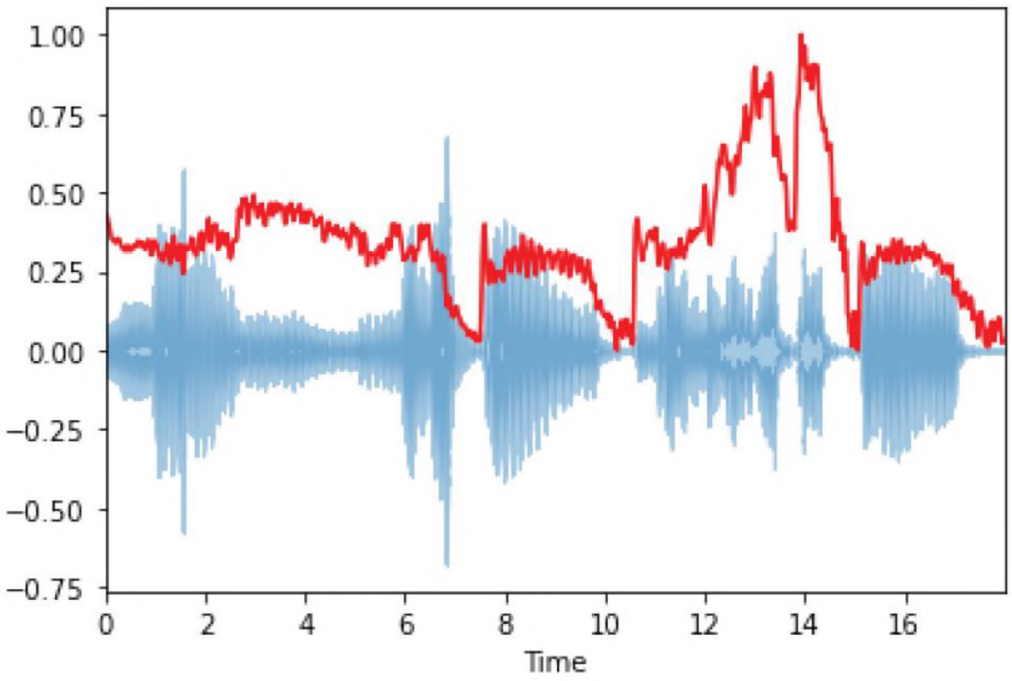

Рисунок 3.3 – Спектральний центроїд

Для обчислення спектрального центроїду використовуємо функцію librosa.feature.spectral_centroid():

```
import sklearn # Для нормування
# Обчислюємо спектральний центроїд
spectral centroids = librosa.feature.spectral centroid( sr=sr)[0]
```

```
# Обчислюємо час для візуалізації frames = range(len(spectral centroids)) t =
librosa.framestotime(frames)
# Нормалізуємо спектральний центроїд до відрізка 0-1
def normalize(x, axis=0):
return sklearn.preprocessing.minmaxscale(x, axis=axis)
# Будуємо графік сигналу та спектрального центроїду
librosa.display.waveplot(x, sr=sr, alpha=0.4)
# Побудуємо амплітуду сигналу
plt.plot(t, normalize(spectral centroids), color='r')
# Побудуємо графік нормалізованого спектрального центроїда plt.show()
# Виводимо графіки
```

Спектральний спад частоти (рис. 3.4) – це міра форми сигналу, що є частотою, нижче якої лежить певний відсоток від загальної спектральної енергії, наприклад, 85 %.

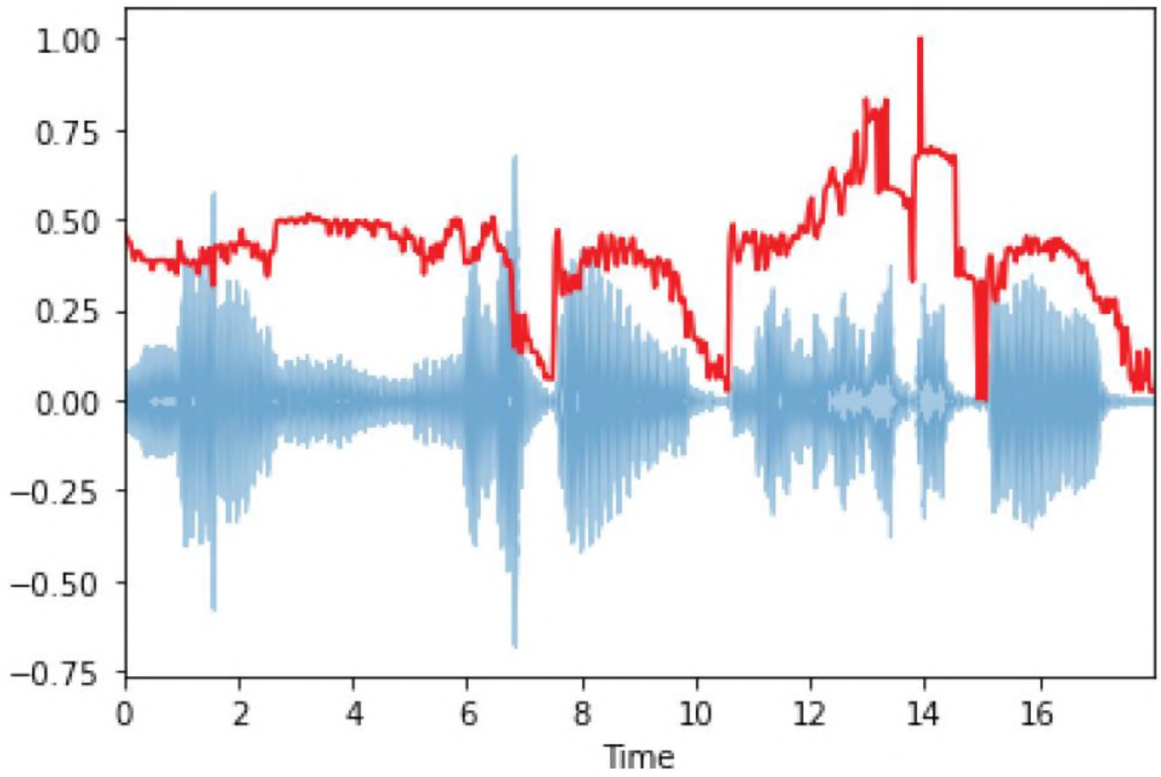

Рисунок 3.4 – Спектральний спад частоти

Для обчислення спектрального спаду частоти використовуємо функцію librosa.features.spectral_rolloff():

```
# Обчислюємо та відображаємо спектральний спад частоти
spectral_rolloff = librosa.feature.spectral_rolloff (x, sr = sr, roll percent = 0.85) [0]
librosa.display.waveplot (x, sr = sr, alpha = 0.4)
```

```
# Побудуємо амплітуду сигналу
plt .plot(t, normalize(spectral_rolloff), color='r')
# Побудуємо графік нормалізованого спектрального спаду частоти
plt.show()
# Виводимо графік
```

Параметром roll_percent=0.85 (за замовчуванням) вказуємо цей певний відсоток.

Mel-частотні спектральні коефіцієнти (MFCC) – є картинка з 20 рядків (20 фільтрів), де кожним фільтром (з відповідною частотою) буде пропущений аудіосигнал (рис. 3.5). Кожен рядок – це оброблений звуковий сигнал якимось частотним фільтром. MFCC робить більший внесок у нейронну мережу (щодо інших ознак).

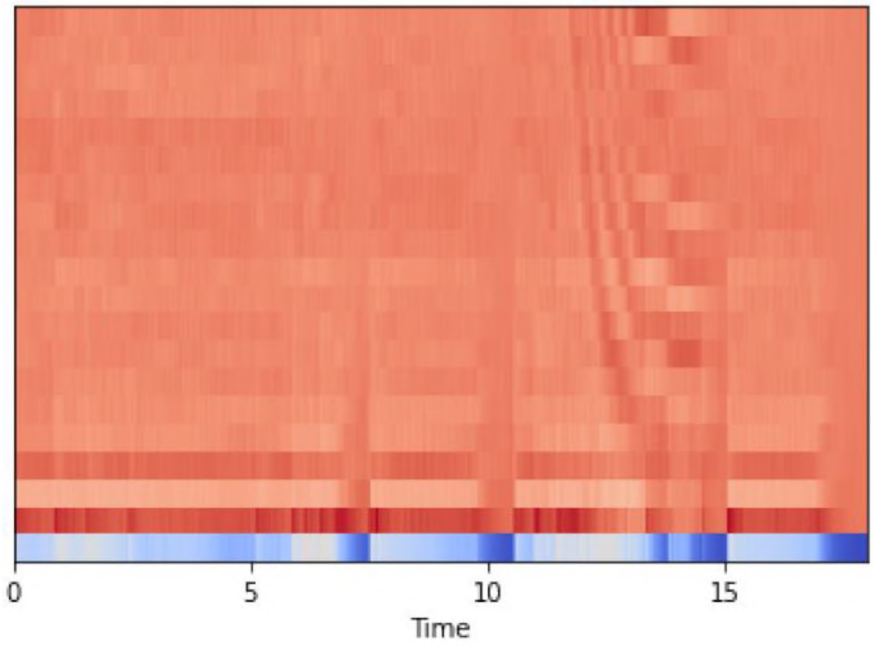

Рисунок 3.5 – MFCC

MFCC обчислюється за допомогою функції librosa.feature.mfcc():

```
# Обчислюємо та відображаємо MFCC
mfccs = librosa.feature.mfcc(x, sr=sr)
librosa.display.specshow(mfccs, sr=sr, x axis='time')
# Відобразимо спектрограми
plt.show() # Виведемо графік
```

Частота кольоровості (функція кольоровості) – це уявлення для музичного звуку, у якому весь спектр звуку проектується на 12 ділянок, які

мають 12 різних півтонів музичної октави (рис. 3.6). Для побудови частоти кольоровості використовуємо функцію librosa.feature. chroma_stft():

```
# Завантажуємо сигнал
x, sr = librosa.load(audio_path)
hop_length = 512
# Задаємо розмір відрізка сигналу,
# яким розраховується частоти кольоровості
# Розраховуємо та відображаємо частоту кольоровості
chromagram = librosa.feature.chroma stft(x, sr=sr, hop length=hop_length)
plt.figure(figsize=(10, 5))
# Задамо розмір графіка
librosa.display.specshow(chromagram, x axis = 'time', y axis='chroma',
hop_length=hop_length, cmap='coolwarm')
# Відобразимо спектрограми
plt.show() # Виведемо графік
```

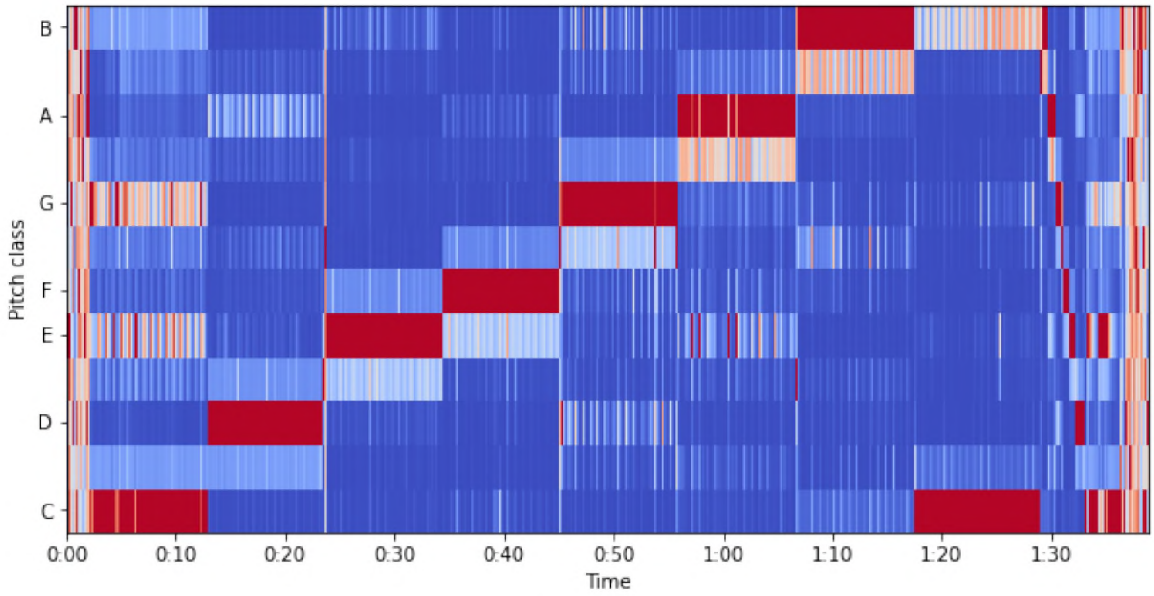

Рисунок 3.6 – Частота кольоровості

A, B, C, D, E, F, G – це ноти і між ними півтони. Тобто, представлено запис послідовного звукового сигналу 7 нот (приблизно по 10 секунд кожна). Комбінуючи перелічені ознаки, можна виконувати такі завдання, як розпізнавання мовлення, цифрове оброблення сигналів, а також класифікація, тегування та генерація музики.

Звичайно, це далеко не весь перелік значущих характеристик аудіосигналу і зазвичай кожен дослідник вибирає сам, які характеристики для вилучення з аудіофайлу він буде використовувати у своєму завданні.

Наприклад, можна також вибирати середні значення та стандартні відхилення Mel-коефіцієнтів:

```
mfcc_mean = np.mean(librosa.feature.mfcc(y=signal, sr=sr), axis=1)
mfcc_std = = np.std(librosa.feature.mfcc(y=signal, sr=sr), axis=1)
```

Середні значення та стандартні відхилення спектрального центроїду

```
cent_mean = np.mean(cent)
cent_std = np.std(cent)
```

Середні значення та стандартні відхилення спектрального спаду тощо.

```
roloff_mean = np.mean(roloff)
croloff_std = np.std(roloff)
```

Потім усі ці значення можна записати в датафрейм та працювати з ними

```
df = pd.DataFrame(audio_data)
df['labels'] = labels
```

Для подальшого аналізу необхідно виділити характеристики зі всіх аудіофайлів. Для цього, наприклад, ми можемо отримати середні значення крейдяні коефіцієнти для всіх наших аудіофайлів. Спочатку створюємо список audio_files з назвами файлів усіх композицій та відповідні їм мітки labels типу жанру:

```
audio_files = []
labels = []
labelind = -1
for label in os.listdir(dir):
        labelind +=1
        label_path = os.path.join(dir, label)
        for audio_file in os.listdir(label_path):
        audio_file_path = os.path.join(label_path, audio_file)
        audio_files.append(audio_file_path)
        labels.append(labelind)
```

Тепер створимо функцію, яка на вхід прийматиме аудіофайл, а потім отримуватиме середні значення коефіцієнтів для файлу.

```
def preprocess_audio(audio_file_path):
        audio, sr = librosa.load(audio_file_path)
        mfcc_mean = np.mean(librosa.feature.mfcc(y=audio, sr=sr),    axis=1)
        return abs(mfcc_mean)
```

Отримаємо список audio_data цифрових значень для всіх аудіофайлів:

```
audio_data =[]
for audio_file in audio_files:
    mfccs_mean = preprocess_audio(audio_file)
    audio_data.append(mfccs_mean)
```

Створимо масиви з характеристик аудіофайлів та їх міток

```
audio_data = np.array(audio_data)
labels = np.array(labels)
```

Таким чином, всі характеристики аудіофайлів можна об'єднати в один датафрейм і далі з ним працювати.

3.2 Оцінка продуктивності синтезованої моделі класифікації аудіо

В якості данних для навчання будемо використовувати датасет різножанрових записів бази даних GATZAN [55] додатку Marsyas – програмного середовища з відкритим вихідним кодом для обробки звуку з особливим упором на програми для пошуку музичної інформації. GTZAN складається із 1000 звукових доріжок кожні 30 секунд. Він містить 10 жанрів, кожен із яких представлений 100 треками. Всі доріжки є моно 16-бітовими аудіофайлами, частота дискретизації 22050 Гц у форматі .wav.

Для оцінки продуктивності моделі глибокого навчання класифікації 10 музичних жанрів використовувалось кілька архітектур мереж. Створений датасет складається з аудіозаписів 10 музичних жанрів: блюз, джаз, диско, класична, поп, кантрі, метал, рок, хіпхоп, реггі. При цьому, навчальна вибірка [56] містить по 80 записів на кожен із класів. Тривалість окремого запису становила 30 с. В свою чергу, перевірна вибірка містила по 15 записів на кожен з класів.

Для реалізації процесу навчання використовувався оптимізатор Adam (з параметрами $\beta 1 = 0.9$, $\beta 2 = 0.999$, $\varepsilon = 10^{-7}$ та відключеному параметрі *Amsgrad*), крок навчання 0,0001 і батч дорівнював 128. Вказані параметри оптимізатору Adam допомагають ефективно адаптувати швидкість навчання кожного параметра моделі, враховуючи як недавні, і довгострокові зміни градієнтів, наприклад:

$\beta 1$ – контролює експоненційне зважене середнє градієнтів. Він

відповідає за оцінку моменту першого порядку (середнє) градієнтів. Чим ближче значення до 1, тим більшим є вплив попередніх градієнтів на поточний момент. Зазвичай, значення β1 вибирається близько 0.9;

$\beta 2$ – контролює експоненційне зважене середнє квадратів градієнтів. Він використовується з метою оцінки моменту 2-го порядку (нецентрована дисперсія) градієнтів. Це допомагає стабілізувати швидкість навчання та зазвичай має значення близьке до 0.999;

ε – невелике число, яке додається для запобігання ділення на нуль під час обчислень, забезпечує чисельну стабільність у виразах, де дільник може наближатися до нуля. Зазвичай, вибирається дуже мале значення, таке як 1e-7 або 1e-8.

Перший варіант моделі, що досліджувався, наведений на рис. 3.7. Він показав продуктивність 37,3 % на 20-ій епосі.

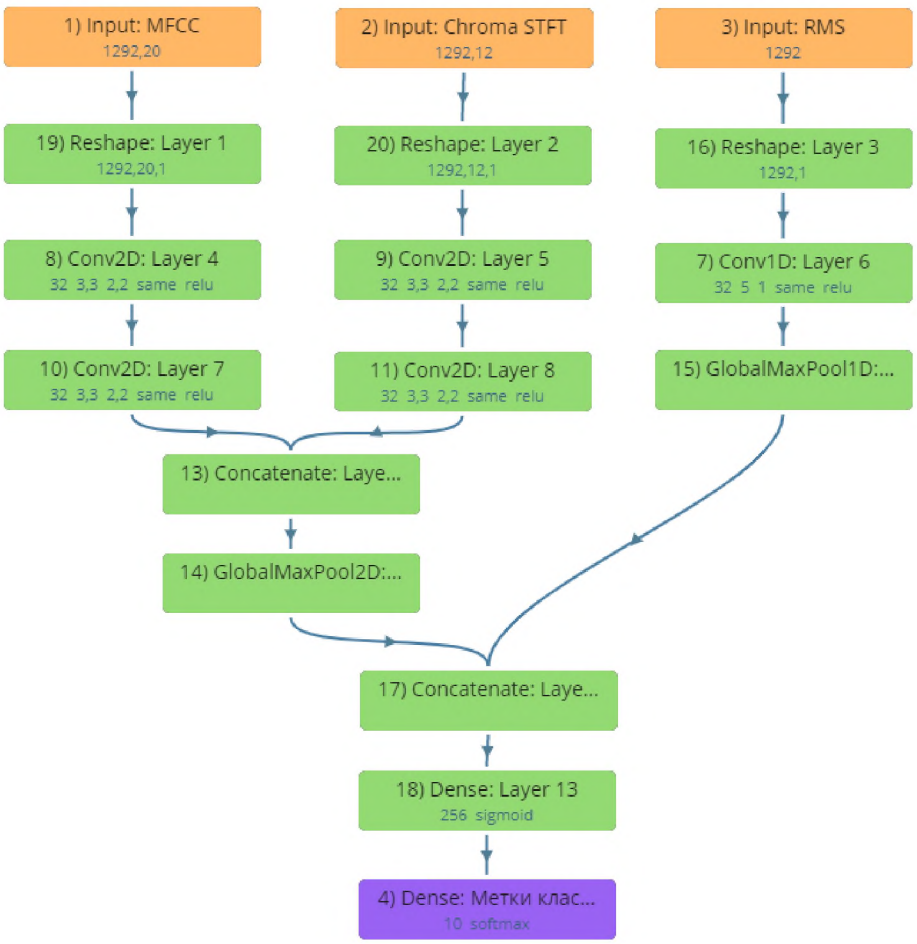

Рисунок 3.7 – Запропонована архітектура моделі класифікації аудіо

Згідно п. 1.3, на вхід моделі подаються Mel-коефіцієнти (масив MFCC розмірністю 1292×20), функція кольоровості (масив Croma STFT розмірністю 1292×12) і середньоквадратичне значення амплітуди звукових хвиль (масив RMS розмірністю 1292). При класифікації аудіо RMS (Root Mean Square) відноситься до методу вимірювання амплітуди звукового сигналу. RMS є статистичним мірою середньоквадратичного значення амплітуди звукових хвиль. У контексті аудіокласифікації RMS може використовуватися визначення загальної гучності або інтенсивності звукового сигналу. Важливо відзначити, що RMS не дає прямої інформації про зміст аудіо (наприклад, про тип звуку або музики), але може бути корисним для розділення звуків на категорії на основі їхньої гучності або енергетичного рівня. Наприклад, більш високі значення RMS можуть вказувати на більш гучні або енергійні звуки, тоді як нижчі RMS можуть відповідати тихим звукам або паузам. RMS часто використовується в поєднанні з іншими аудіоознаками, такими як спектральні характеристики, MFCC та ін., для створення більш складних і точних систем класифікації аудіо. Повнозв'язаний шар Dense (позначений № 18, див. рис. 3.7) має функцію активації Sigmoid, а на виході встановлений класифікатор на 10 класів (за кількістю жанрів).

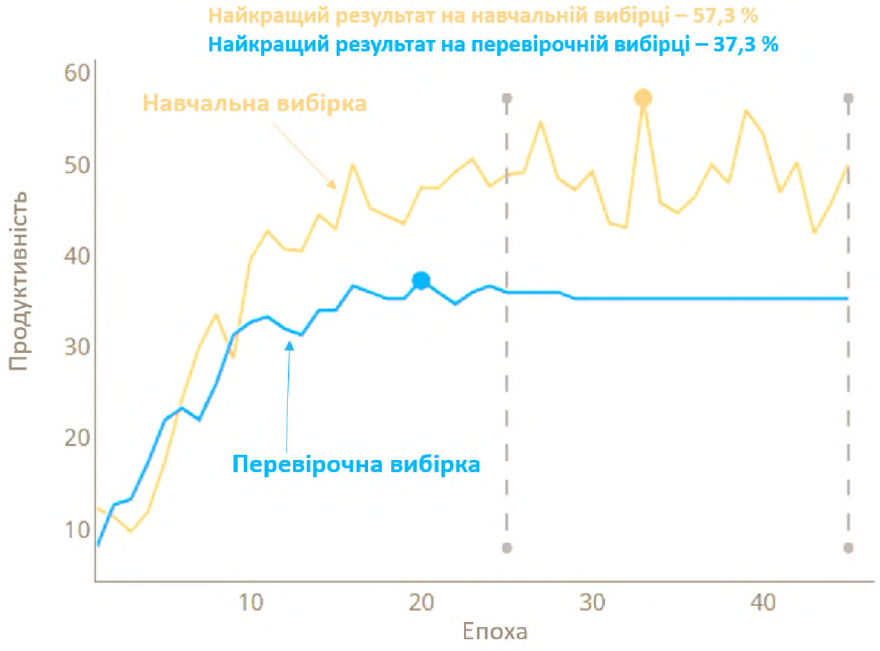

Рисунок 3.8 – Продуктивність першого варіанту моделі

Після налаштування параметрів останніх шарів зі зниженням розмірності до 200 та 80 та заміною Sigmoid на ReLU (рис. 3.9) вдалося підвищити точність до 62,7 % (рис. 3.10). При цьому, батч був збільшений з 8-ми до 32-ох.

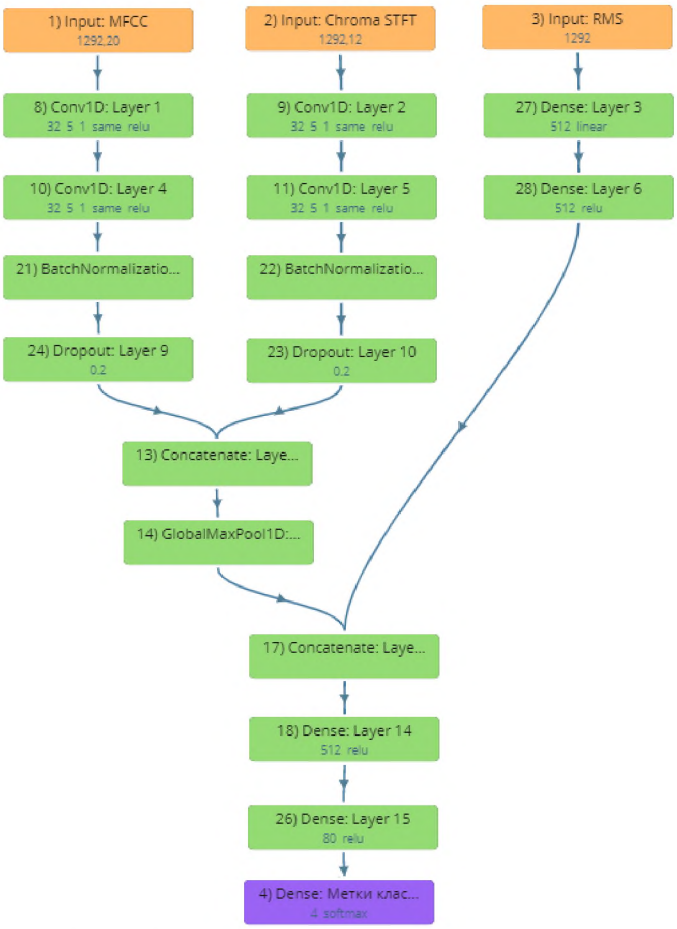

Рисунок 3.9 – 2-ий варіант моделі класифікації аудіо

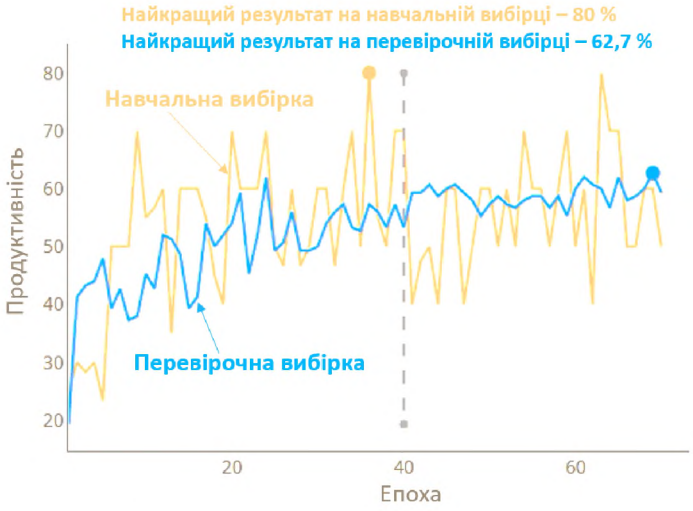

Рисунок 3.10 – Продуктивність 2-го варіанту

Таким чином, розмір батча помітно впливає на точність класифікації. На підтвердження цього було проведено дослідження для батчів 8, 16, 64. Надалі було виконано оптимізацію архітектури (рис. 3.11). Це дозволило отримати продуктивність 64 % (рис. 3.12).

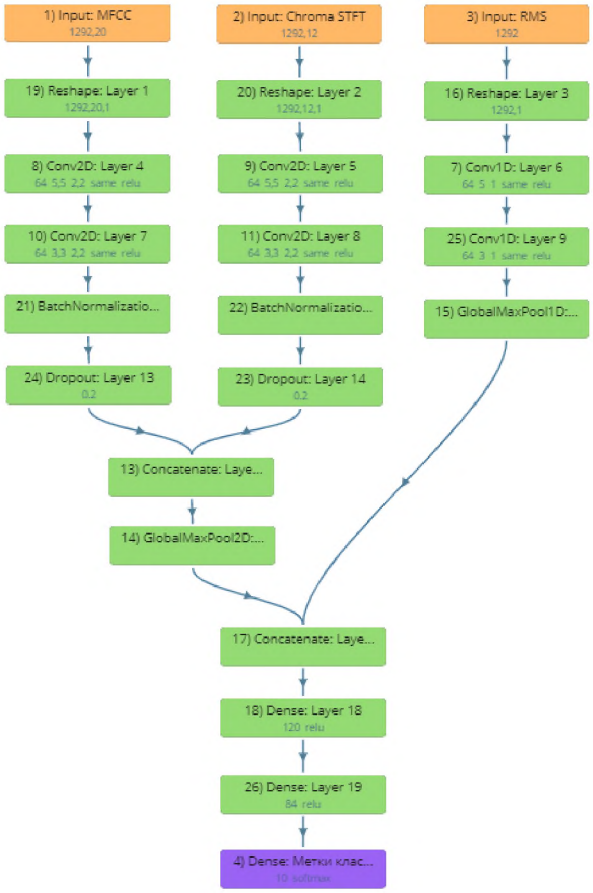

Рисунок 3.11 – Оптимізована архітектура моделі

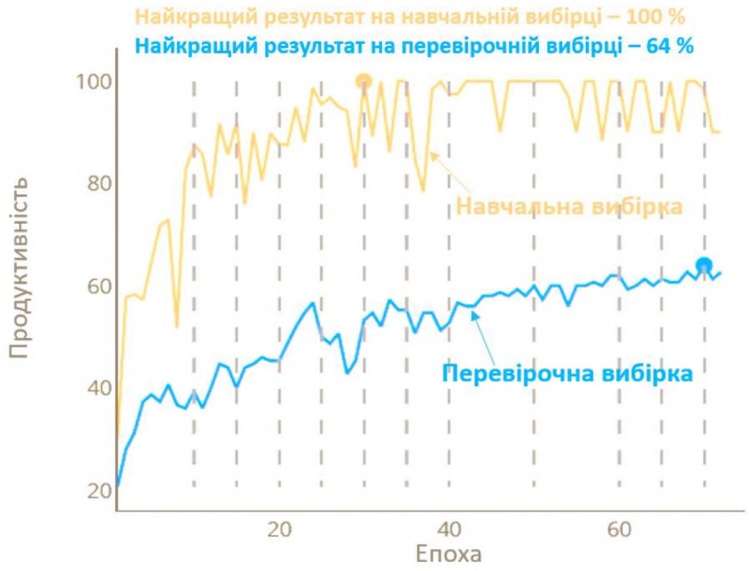

Рисунок 3.12 – Продуктивність оптимізованої архітектури моделі

Найменшу точність класифікації мав жанр «блюз» (13,3 %), а найкращу – класика (93,3 %) – рис. 3.13.

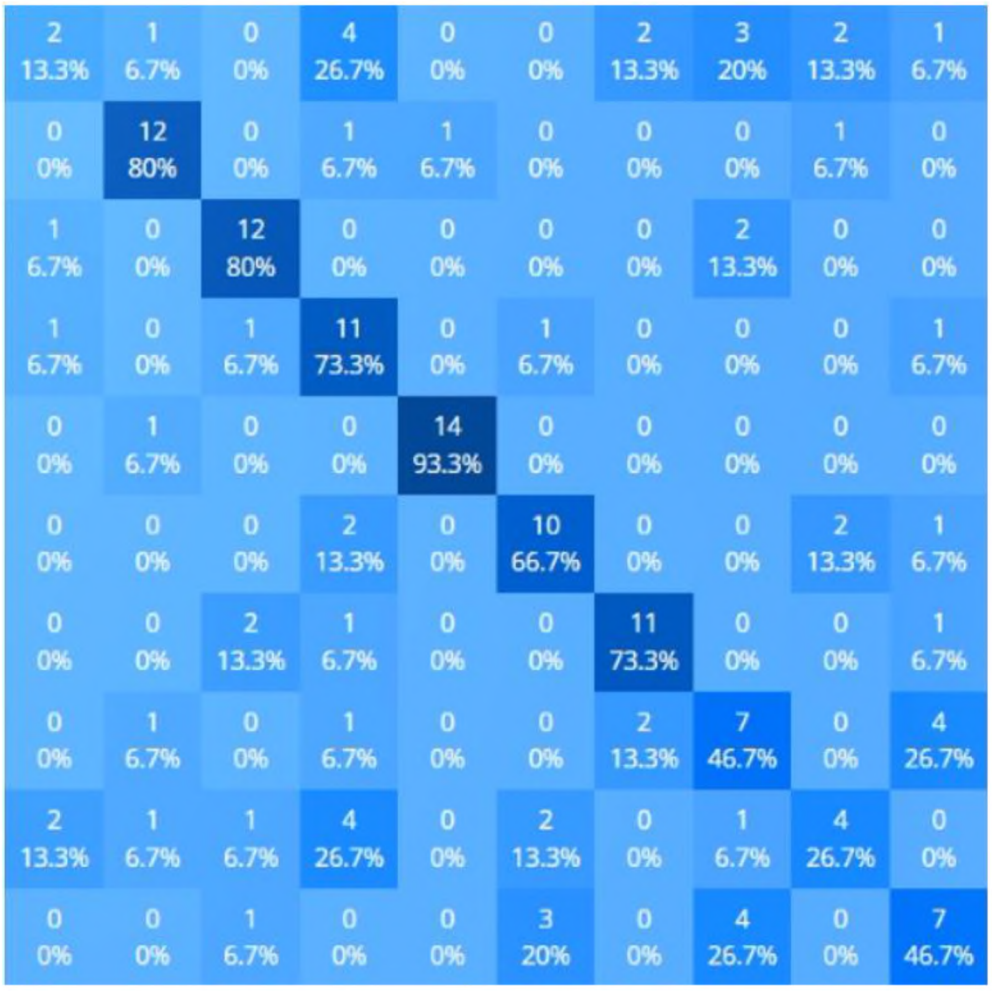

Рисунок 3.13 – Матриця помилок

У Додатку Б наведено фрагмент коду Python, що відповідає формуванню запропонованої моделі глибокого навчання класифікації аудіо на основі бібліотеки Keras.

3.3 Техніко-економічне обґрунтування прийнятих рішень

Запропонована в роботі аудіокласифікація на основі багатовходової нейронної мережі на основі згортки дозволяє отримати продуктивність 64 %. Дану модель можна використовувати не тільки для класифікації жанрів

музики та пошуку відповідного аудіо контенту, але і для класифікації певних подій під час технологічних виробничих процесів та ін. Підвищення ефективності можливо за рахунок використання гібридних моделей, в тому числі і на основі трансформерів. Згідно, п. 2.4, це дозволить підвищити продуктивність в середньому на 15-25 %. Однак, при цьому необхідно враховувати, що для трансформерів потрібен якісний і великий датасет. Ще одним з варіантів підвищення продуктивності може бути застосування попередньо навчених моделей. Однак, існуюча номенклатура CNN, в основному орієнтовано на обробку зображень реальних об'єктів, а не спектрограм MFCC та Chroma STFT.

Програмний код Python створений з використанням безкоштовних версій ресурсів, а також безкоштовних і повністю сформованих датасетів. Його довжина складає ≈ 6000 рядків. Згідно з існуючих досліджень, середня швидкість друку коду для програміста середнього рівня становить приблизно 25 рядків на одну годину. Відповідно, необхідний час на цей етап складає:

$$T = \frac{6000}{25} = 160 \text{ [год]}. \qquad (3.1)$$

Перерахувавши цей час у кількість робочих, отримаємо період на створення моделі – 1,5 місяця. Але крім створення самої моделі необхідно провести її навчання. Для цього можна використовувати хмарний сервіс Google Colab Pro+. На його оплату потрібно витрати за 2 місяця – 4000 грн. Після отримання повноцінної моделі глибокого навчання нейронної мережі класифікації аудіо проводиться етап її інтеграції до необхідних сервісів, наприклад, каналу у месенджері Telegram. З урахуванням процедури тестування, на дану операцію знадобиться 8-16 годин. Середня заробітна плата програміста-фрілансера Python в Україні залежить від низки чинників. Згідно [57], вона становить 45000 грн на місяць. Таким чином, мінімальні витрати на зарплату програміста складають 72000 грн. З урахуванням оплати використання Google Colab Pro+, загальні витрати складають 76000 грн.

Висновки до розділу 3

Модель глибокого навчання реалізується на мові Python. В якості базового інструменту виступає бібліотека LibROSA. Її використання дозволяє отримати низку ознак з вхідних аудіоданих.

В роботі запропоновано реалізувати технологію класифікації аудіоконтенту на основі моделі глибокого навчання нейронної мережі, на вхід якої подається кілька ознак, наприклад, MFCC, Chroma STFT, RMS.

Для забезпечення процесу навчання нейронної мережі використовується безкоштовний та тегований датасет GATZAN. За основу взята CNN. Під час оцінки продуктивності нейронної мережі оптимізувалась її архітектура та виконувався підбір гіперпараметрів.

Запропонований варіант модель глибокого навчання дозволив отримати продуктивність на рівні 64 % для перевірочної вибірки. З урахуванням невеликого обсягу наявного датасету, це підтверджує висунуті в роботі теоретичні положення щодо використання нейронних мереж для класифікації аудіоданих. Підвищення ефективності запропонованої моделі можливе за рахунок покращення якості датасету.

ВИСНОВКИ

У роботі проведено аналіз архітектур нейронних мереж CNN, RNN, автоенкодерів, трансформерів, а також гібридних моделей. Подано докладний опис компонентів кожного типу архітектури. Особливу увагу приділено трансформерам, через їх успіх в обробці NLP.

Зазначені архітектури можуть використовуватися для різних завдань класифікації аудіо, таких як розпізнавання мови, шумів, музики, емоцій, звуків довкілля, акустичних сцен та ін. Потенційно, запропоновані варіанти моделі глибокого навчання класифікації аудіо дозволяють визначати аномальні звуки.

Дослідження виявило, що при розробці технологій для класифікації аудіо важливо оцінити вплив архітектури мережі та її гіперпараметрів на ефективність. Використання характеристик, які можна вилучити з аудіоданих, є ефективним. На даний час, в Python для роботи з аудіо можна використовувати кілька бібліотек, наприклад, PyAudio, Speech Recognition, Google Text To Speech, LibROSA. Найбільш перспективною є бібліотека LibROSA, що дозволяє аналізувати різноманітні характеристики вхідних аудіоданих.

В роботі запропоновано технологія для класифікації аудіоконтенту що спирається на модель глибокого навчання. Ця модель приймає різні характеристики, такі як MFCC, Chroma STFT, RMS. При цьому, для обробки MFCC і Chroma STFT можна застосовувати нейронні мережі для обробки відео. Розглянутий підхід дозволяє використовувати для окремої характеристики визначену архітектуру нейронної мережі.

В процесі навчання використовується безкоштовний датасет GATZAN, В якості базової архітектури розглядається CNN. Ефективність моделі оцінювалася шляхом оптимізації архітектури та налаштування гіперпараметрів. Отримана модель показала продуктивність на рівні 64 % для тестової вибірки, що підтверджує теорії про ефективність нейронних мереж у

класифікації аудіоданих, висунуті у роботі. Якість моделі може бути покращена шляхом вдосконалення датасету.

Таким чином, результатами роботи є створена модель глибокого навчання нейронних мереж класифікації аудіоконтенту; рекомендації щодо використання технологія класифікації аудіоконтенту на основі нейронних мереж. Вони можуть бути використані для подальших досліджень за даною тематикою та при проектуванні мобільних додатків.