НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

«На правах рукопису»
УДК 004.8

До захисту допущено:
Завідувач кафедри
_____ Ігор ПАРХОМЕЙ
«___»_____2020 р.

Магістерська дисертація

на здобуття ступеня магістра

за освітньо-професійною програмою «Інформаційне забезпечення робототехнічних систем»

зі спеціальності 126 «Інформаційні системи та технології»

на тему: «Інтелектуальна система класифікації типів аудіоконтенту»

Виконав:
студент ІІ курсу, групи ІК-91мп
Жіляєв Артем Михайлович _____

Керівник:
доцент, к.т.н, доц.,
Олійник Володимир Валентинович _____

Консультант з нормоконтролю:
доцент, к.т.н., доц.,
Пасько Віктор Петрович _____

Рецензент:
доцент каф. АСОІУ, к.т.н., доц.
Муха Ірина Павлівна _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____

Київ – 2020 року

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

Рівень вищої освіти – другий (магістерський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційне забезпечення робототехнічних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Ігор ПАРХОМЕЙ

«___»_____2020 р.

ЗАВДАННЯ
на магістерську дисертацію студенту

Жіляєв Артем Михайлович

1. Тема дисертації «Інтелектуальна система класифікації типів аудіоконтенту», науковий керівник дисертації Олійник Володимир Валентинович, к.т.н, старший викладач, затверджені наказом по університету від « 26 » жовтня 2020р. № 3132-с

2. Термін подання студентом дисертації 18.11.2020

3. Об’єкт дослідження – класифікація типів аудіоконтенту

4. Вихідні дані – Створення інтелектуальної системи класифікації типів аудіоконтенту для розв’язку задачі визначення типів радіоконтенту з аудіопотоку

5. Перелік завдань, які потрібно розробити – аналіз проблеми та існуючих рішень; класифікації аудіоконтенту, аналіз і реалізація алгоритму класифікації довгих аудіоданих на основі радіотипів; розробка та навчання моделей нейромереж; дослідження ефективності розроблених моделей, точності класифікації та надійності у використанні.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу – чотири інформаційні схеми та два плакати

7. Орієнтовний перелік публікацій – одна публікація

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

Перевірка на співпадіння	доцент Лісовиченко О.І.		
Нормоконтроль	доцент Пасько В.П.		

9. Дата видачі завдання 10.09.2020 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Аналіз предметної області	12.09.2020 р.	
2	Постановка задачі	15.09.2020 р.	
3	Аналіз інформаційного забезпечення	22.09.2020 р.	
4	Аналіз алгоритмічного забезпечення	26.09.2020 р.	
5	Розробка алгоритмічного забезпечення	15.10.2020 р.	
6	Розробка програмного забезпечення	10.11.2020 р.	
7	Маркетинговий аналіз стартап-проекту	12.11.2020 р.	
8	Висновки	16.11.2020 р.	
9	Попередній захист	16.11.2020 р.	
10	Нормоконтроль	10.12.2020 р.	
11	Перевірка на співпадіння	10.12.2020 р.	
12	Захист	22.12.2020 р.	

Студент Артем ЖІЛЯЄВ

Науковий керівник Володимир ОЛІЙНИК

АНОТАЦІЯ

У роботі проведено дослідження класифікації типів аудіоконтенту та розроблено інтелектуальну систему класифікації довгих аудіоданих, розглянуто типи радіоконтенту, процес вилучення аудіофіч, розглянуто особливості існуючих підходів до класифікації аудіоконтенту та попередня обробка звукового сигналу та методи покращення вилучених ознак.

Об’єктом дослідження є класифікація типів аудіоконтенту.

Було виконано всі кроки для переходу нейронних мереж від обробки даних до класифікації кінцевих ознак. Отримано результати тестів для двох різних розмічених наборів даних, створених за допомогою бібліотеки на мові python, було класифіковано їх із зміненими параметрами для роботи з довгим аудіоконтентом за допомогою інтерфейсу 8M, проведено аналіз отриманих результатів, їх точність та визначено необхідні зміни для покращення інтелектуальної системи.

Ключові слова: інтелектуальна система, класифікація довгого аудіоконтенту, нейронна мережа, розмічені набори даних.

Розмір пояснювальної записки – 87 аркушів, містить 35 ілюстрацій, 23 таблиці, 6 додатків.

ABSTRACT

In this work, the research conducted on the classification of types of audio content. The intellectual system of classification of long audio data was developed. Types of radio content, process of extraction of audio files was considered in this work. The features of existing approaches to classification of audio content and preliminary processing of an audio signal and methods of improvement of the removed features was considered in this explanatory note.

The object of research is the classification of audio content's types.

All steps were performed to transition neural networks from data processing to classification of final features. The results of tests for two different marked data sets created with the additional feature extractor library in python, they were classified with changed parameters for working with long audio content using the 8M interface, the results were analyzed, their accuracy and necessary changes to improve intelligent system.

Key words: intelligent system, classification of long audio content, neural network, marked data sets.

Explanatory note size - 87 pages, contains 35 illustrations, 23 tables, 6 applications.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ІС – Інтелекутальна система

CNN – Convolutional neural network

CAB-CNN – Classifier-Attention-Based CNN

RNN – Recurrent neural network

LSTM – Long short-term memory

FFC – Fast Fourier transform

DFT – Discrete Fourier transform

MFCCs – Mel-Frequency Cepstral Coefficients

ZCR – Zero-Crossing Rate

LPC – Linear Predictive Coding

GMM – Gaussian Mixture Model

SVM – Support Vector Machine

DCT – Discrete Cosine Transformation

ВСТУП

ІС для роботи з аудіоконтентом – це зростаюча область досліджень із численними реальними програмами. Незважаючи на те, що існує велика кількість досліджень у суміжних аудіополях, таких як голос, мова та музика, робота над класифікацією типів аудіоконтенту порівняно обмежена.

Так само, спостерігаючи останні досягнення в галузі класифікації зображень, де згорткові нейронні мережі використовуються для класифікації зображень з високою точністю та масштабом, виникає питання про застосовність цих методів в інших сферах, таких як класифікація довгих аудіоданих.

Останніми роками підходи до глибокого навчання набули значного інтересу як спосіб класифікації звуків, розпізнавання мови, генерації та стилізації музики, усі вони використовують схожий алгоритм для роботи з нейромоделями. У цій роботі я застосував різні типи підходів та нейромереж для класифікації довгих аудіоданих та емпірично протестував на нерозміченних даних роботу кожної створеної моделі. У випадку мовних та музичних даних було доведено, що вивчені моделі задовольняють умовам розпізнавання та класифікації. Окрім того, деякі представлені алгоритми класифікації аудіоконтенту отримані з маркованих аудіоданих, демонструють дуже хорошу продуктивність для багатьох завдань класифікації аудіо.

Сподіваюсь, що ця робота надихне на нові дослідження підходів глибокого навчання, що застосовуються до широкого кола завдань розпізнавання аудіо.

РОЗДІЛ 1. ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Загальний огляд об‘єкта дослідження

Класифікація типів аудіоконтенту – це процес обробки та аналізу довгих аудіозаписів. Цей процес належить до класифікації звуків, що лежать в основі багатьох сучасних технологій машинного навчання, включаючи віртуальних помічників, автоматичне розпізнавання мови та програми для перетворення тексту в мовлення. Ви також можете знайти його в інтелектуальному обслуговуванні, розумних системах безпеки та мультимедійній індексації та пошуку.

Такі проекти класифікації звуку, як згадані вище, починаються з оброблених анотованих звукових даних, як наприклад – MFCCs або filter banks. Відповідна ЕОМ потребує цих даних у векторному вигляді, щоб навчитися класифікувати аудіоконтент. Використовуючи ці дані, вони розвивають здатність розрізняти та класифікувати звуки для виконання конкретних завдань. Процес обробки часто включає виділення відповідних анотованих звукових даних на основі конкретних потреб задач за допомогою спеціальних служб класифікації аудіо.

Я представлю чотири широко використовувані типи класифікації та відповідні випадки використання для кожного:

- Класифікація акустичних даних: цей тип класифікації, також відомий як виявлення акустичних подій, визначає місце запису аудіосигналу. Це означає розмежування середовищ, таких як ресторани, школи, будинки, офіси, вулиці тощо. Одним із методів класифікації акустичних даних є створення та обслуговування звукових бібліотек для аудіо-мультимедіа. Він також відіграє роль у моніторингу екосистем. Одним із прикладів цього є оцінка чисельності риби в певній частині океану на основі їх акустичних даних.
- Класифікація звуків навколишнього середовища: як випливає з назви, це класифікація звуків, що зустрічаються в різних середовищах.

Наприклад, розпізнавання міських зразків звуку, таких як звукові сигнали автомобілів, дорожні роботи, сирени, людські голоси тощо. Це використовується в системах безпеки для виявлення таких звуків, як розбиття скла. Він також використовується для інтелектуального обслуговування шляхом виявлення невідповідностей звуку в заводських машинах. Він навіть використовується для розрізнення голосів тварин для спостереження та збереження дикої природи.

– Класифікація музики: процес класифікації на основі таких факторів, як жанр або відтворені інструменти музики. Ця класифікація відіграє ключову роль в організації аудіобібліотек за жанрами, вдосконаленні алгоритмів рекомендацій та виявленні тенденцій та уподобань слухачів за допомогою аналізу даних.

Класифікація мовлення: це класифікація записів рідної мови, заснована на розмовній мові, діалекті, семантиці чи інших мовних особливостях певних груп людей. Іншими словами, це класифікація людської мови з елементами національних та етнічних особливостей. Цей тип класифікації аудіо найчастіше зустрічається у чат-ботах та віртуальних помічниках, але також поширений у машинному перекладі та програмах перетворення тексту в мовлення.

В даній задачі класифікація буде стосуватися груп усіх типів аудіоданих, так як радіоконтент представляє собою складний вид аудіоконтенту і відповідно є предметом дослідження даного дипломного проекту.

1.2. Виділення предмету дослідження

Розпізнавання довгих аудіоданих на прикладі радіо має за основу комплекс усіх типів класифікації аудіо: від звуків до мовлення.

Розуміння того, як класифікувати складні аудіодані, є одним з найбільших викликів в галузі класифікації аудіо. Деякі роботи [1, 2] показали, що вивчення розрідженого подання звуків призводить до навчання фільтрів, які відповідають фільтрам нейронів як і в сприйнятті звуку у людей. У

відповідних роботах Grosse [3] запропонував ефективний алгоритм розрідженого кодування слухових сигналів та продемонстрував його корисність у завданнях класифікації аудіо. Проте як і в задачах Giusepry [4] з класифцікаії типів радіо новин метод найближчого сусіда та статистична HMM модель не може дати необхідну точність класифікації для більшості практичних задач класифікації аудіоконтенту. На заміну застарілим методам прийшло використання DNN, і як в обробці так і в класифікації аудіоконтенту.

Задача для якої будуть використовуватись створені моделі в цій роботі – це проблема моніторингу радіо в регіональних та державних установах, зокрема задача моніторингу рекламних та мовних квот на радіо.

Для розуміння предмету дослідження необхідно проінформувати про область та специфіку для подальшої роботи з впровадження. Зокрема для моніторингу радіоефіру та створення відповідних актів моніторингу використовуються держслужбовці Національної ради України з питань телебачення і радіомовлення в усіх областях країни. Через доволі рутинні дії наглядача та можливість місцевої корупції, Національна рада зацікавлена в автоматизації процесу моніторингу радіо. Також у створені відповідної інтелектуальної системи зацікавлені власники телерадіокомпаній для оцінки власного продукту.

Відповідно до Пенчук існують класифікації радіомовлення на території України [5]:

Класифікація радіомовлення за тематичною наповненістю:

1. Інформаційні передачі.
2. Політико-ідеологічні передачі.
3. Історико-культурологічні передачі.
4. Соціально-економічні передачі.
5. Освітньо-пізнавальні передачі.
6. Релігійні передачі.
7. Дитячі передачі.
8. Юнацькі передачі.

9. Молодіжні передачі.
10. Літературно-драматичні передачі.
11. Музично-розважальні передачі.
12. Рекламні передачі.

Класифікація програм радіомовлення за комунікативними характеристиками:

1. Монологічні радіопередачі.
2. Діалогічні радіопередачі.
3. Полілогічні радіопередачі.
4. Інтерактивні радіопередачі.

Класифікація програми радіомовлення за додатковим шумово-звуковим наповненням:

1. Музичні радіопередачі.
2. Немузичні радіопередачі.
3. Класифікація за типами мовлення.
4. Інформаційне мовлення.
5. Інформаційно-музичне мовлення.
6. Музично-інформаційне мовлення.
7. Музичне мовлення.

Для задачі моніторингу головну роль відіграє наповнення, відповідно з актами моніторингу 2019 [6], виділено основні типи які будуть класифікуватись і цій роботі:

1. Інформаційні передачі, з яких виділятимуться:
   1.1. За комунікативними характеристиками або в прикладному плані – кількістю голосів:
      - Монологічні радіопередачі (1 особа).
      - Діалогічні радіопередачі (2 особи).
      - Інтерактивні радіопередачі (більше).
   1.2. За критерієм мовлення:
      - Інформаційне мовлення (голос).

- Інформаційно-музичне мовлення (голос та музика).
- Музичне мовлення (превалює музика).

1.3.За мовою:

- іноземна.
- українська.

2. Музично-розважальні передачі.
3. Рекламні передачі.

В результатах дослідження і надалі представлені типи будуть мати наступний вигляд в дереві класифікацій типів радіоконтенту (додаток 1). Критерії оцінки ефективності створеної ІС заключаються в точності отриманих моделей на тестових данних.

1.3. Існуючі реалізації класифікації типів радіоконтенту

Серед небагатьох варіантів існуючих реалізацій класифікації довгих аудіоданих на основі радіоконтенту, було знайдено застарілу проте найбільш вдалу реалізацію за допомогою MPEG-7 професора Ignasi Esquerra та Giuseppe Dimattia в роботі “An Automatic Audio Classification System for Radio Newscast”. В Роботі описана реалізація класифікації за допомогою стандарту MPEG-7.

1.3.1. Відомості про MPEG-7 реалізацію для класифікації радіоконтенту

У вересні 2001 року група експертів із рухомих зображень (MPEG) визначила міжнародний стандарт, який називається Інтерфейс опису мультимедійного вмісту, або MPEG-7.

MPEG-7 – це перша повна робота з опису мультимедійних даних. MPEG-7 –це робота з визначення стандартного набору дескрипторів для різних типів мультимедійних даних та методів визначення інших дескрипторів, а також структур дескрипторів та їх взаємозв’язків. Незважаючи на те, що MPEG-7 не спрямований на певну область застосування, однією з

основних областей його застосування буде пошук та отримання мультимедійного вмісту.

Цей стандарт, також відомий як "Інтерфейс опису мультимедійного вмісту", забезпечує стандартизований набір технологій для опису мультимедійного вмісту. Стандарт розглядає широкий спектр мультимедійних додатків та вимог, надаючи систему метаданих для опису особливостей мультимедійного вмісту [8]. У цьому стандарті зазначено:

- Схеми опису (DS) описувати сутності або відносини, що стосуються мультимедійного вмісту. Схеми опису визначають структуру та семантику їх компонентів, які можуть бути схемами опису, дескрипторами або типами даних.
- Дескриптори (D) описувати особливості, атрибути або групи атрибутів мультимедійного вмісту.
- Типи даних є основними типами даних, що використовуються багато разів для схем опису та дескрипторів.
- Мова визначення опису (DDL) визначає схеми опису, дескриптори та типи даних, вказуючи їх синтаксис, і дозволяє їх розширення.
- Системні інструменти підтримка доставки описів, мультиплексування описів з мультимедійним вмістом, синхронізація, формат файлів тощо.

1.3.2. Схеми опису (DS) для опису особливостей типів радіоконтенту

Схеми опису (DS) визначають структуру та семантику взаємозв’язків між їх компонентами, які можуть бути як дескрипторами, так і схемами опису. DS надають стандартизований спосіб опису в XML важливих концепцій, пов'язаних з описом вмісту AV та управлінням вмістом, щоб полегшити пошук, індексацію, фільтрацію та доступ.

DS визначаються за допомогою мови визначення мови опису MPEG-7 (DDL), яка базується на мові схеми XML, і створюються як екземпляри як документи або потоки. Отримані описи можуть бути виражені у текстовій

формі (тобто зручний для читання XML для редагування, пошуку, фільтрації) або стислій двійковій формі (тобто для зберігання або передачі).

DS MPEG-7 призначені в першу чергу для опису регіонів, сегментів, об'єктів, подій, які є функціями AV високого рівня, а також деяких постійних метаданих, таких як заголовок, автор та дата створення. Поєднуючи D, DS та представляючи взаємозв'язки між ними, можна визначити складні описи. У MPEG-7 DS класифікуються відповідно до відповідних носіїв, таких як аудіо, візуальний домен або мультимедіа. Зазвичай мультимедійні DS описують вміст, що складається з комбінації аудіо, візуальних даних і, можливо, текстових даних, тоді як аудіо або візуальні DS посилаються конкретно на функції, унікальні для аудіо- чи візуального домену, відповідно [7]. DS можна згрупувати у 5 різних класів відповідно до їх функціональних можливостей:

- Опис вмісту: Представлення сприйнятливої інформації
- Управління вмістом: Інформація про медіа-функції, створення та використання AV-вмісту.
- Організація вмісту: Представлення аналізу та класифікації декількох вмістів AV.
- Навігація та доступ: Специфікація коротких викладів та варіацій змісту

Взаємодія з користувачем: Опис уподобань користувачів та історії використання, що стосуються споживання мультимедійних матеріалів.

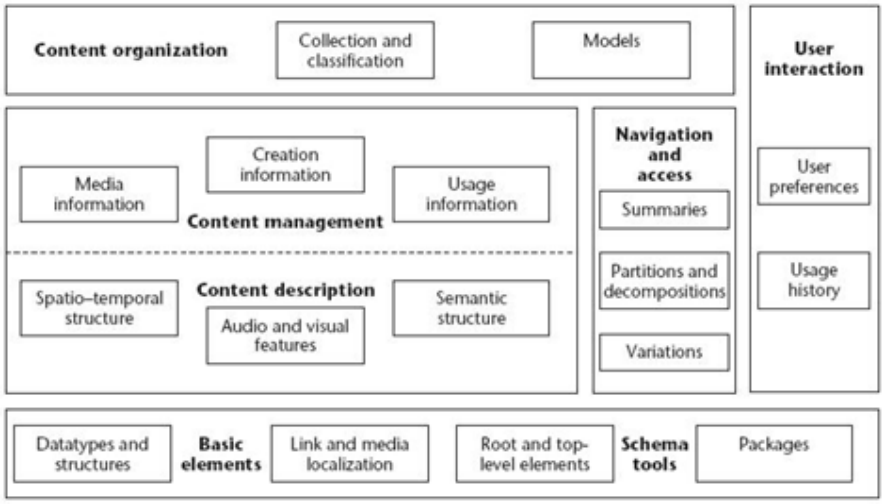


Рис. 1.1. Огляд мультимедійних DS-файлів MPEG-7

1.3.3. Мова визначення MPEG-7 (DDL) для опису схем

Для MPEG-7 однією з основних робіт було визначення мови (Language Definition Language) для опису DS. Під час процесу проектування мови визначення опису (DDL) розробники зацікавлені досягти наступних моментів за допомогою DDL:

- Запобігання витонченості, іншими словами, що відповідає основним особливостям, а також здатності до розширення.
- Покриття вимог MPEG-7 та використання MPEG-7 Ds, DS.
- Використання XML-структури та взаємодії зі схемою XML.

DDL - це мова, яка дозволяє створювати DS і MP DS MPEG-7. Схема DDL (файл DDL) визначає обмеження, які повинен поважати дійсний опис MPEG-7. це є закодований у XML. DDL використовує схему XML із міркувань взаємодії. Однак, оскільки описи мультимедійного вмісту вимагають специфічних функцій, які не визначені у схемі XML (наприклад, типи масивів та матриць), DDL додає ці функції до мови. Отже, MPEG-7 DDL приймає більшу частину специфікації схеми XML і додає MPEG-7-специфічні механізми на додачу. Деякі з важливих питань, які підтримуються DDL, - це розширена схема XML, структурні обмеження та типи даних DS, їх використання, DS в інших DS та визначення груп, такі як Attribute Group для простих визначень.

За допомогою DDL можна виражати структурні обмеження та обмеження типу даних. Структурні обмеження визначають правила, які повинен дотримуватись дійсний опис з точки зору включення елементів. Обмеження типу даних визначають тип та можливі значення даних в описі. DDL дозволяє визначати комплексні типи та прості типи. ComplexTypes визначають структурні обмеження, тоді як simpleTypes виражають обмеження типу даних. Більше того, DDL дозволяє повторно використовувати існуючі complexTypes або simpleTypes способом, подібним до успадкування в об'єктно-орієнтованому програмуванні [14].

1.3.4. Деталі стандарту MPEG-7

Цей стандарт підрозділяється на вісім частин:

Частина 1 - Системи: визначає інструменти для підготовки описів для ефективного транспортування та зберігання, стиснення описів та забезпечення синхронізації вмісту та описів.

Частина 2 - Опис Визначення мови: визначає мову для визначення стандартного набору інструментів опису (DS, Ds та типів даних) та для визначення нових інструментів опису.

Частина 3 - Візуальна: визначає інструменти опису, що стосуються візуального вмісту.

Частина 4 - Аудіо: визначає інструменти опису аудіовмісту [9].

Частина 5 - Схеми мультимедійного опису: визначає загальні засоби опису, що стосуються мультимедіа, включаючи аудіо та візуальний вміст [13].

Частина 6 - Довідкове програмне забезпечення: забезпечує програмну реалізацію стандарту.

Частина 7 - Відповідність: визначає керівні принципи та процедури для перевірки відповідності реалізацій стандарту.

Частина 8 - Видобування та використання: надає настанови та приклади вилучення та використання описів.

У цій роботі частини стандарту MPEG-7, які я використовував як посилання, є лише частинами 4, 5 та 6.

Зважаючи на ці частини є сенс відтворити шари ІС у вигляді наданому описаним стандартом.

MPEG-7 Audio забезпечує структури - разом із мультимедійними схемами опису, частиною стандарту – для опису аудіовмісту. Використовуючи ці структури, як набір низькорівневих дескрипторів для аудіофункцій Ignasi Esquerra, використав підхід який перекриває багато програмних рішень (наприклад, задач визначення спектральних, параметричних та тимчасових особливості сигналу), та інструменти опису високого рівня, які є більш специфічними для набору програм. Ці інструменти високого рівня

включають загальне розпізнавання звуку та індексацію інструментів опису, інструментальні інструменти опису тембру, інструменти опису звукового вмісту, схему опису звукового підпису та мелодійні інструменти опису для полегшення запиту-гудіння [14].

Мультимедійні схеми опису MPEG-7 (також звані MDS) включають набір інструментів опису (дескриптори та схеми опису), що стосуються як загальних, так і мультимедійних об’єктів.

Загальні сутності - це ознаки, які використовуються в аудіо- та візуальних описах, а отже, "загальні" для всіх засобів масової інформації. Це, наприклад, "вектор", "час", засоби текстового опису, керовані словникові запаси тощо.

MDS - це структури метаданих для опису та анотації мультимедійного вмісту і забезпечують спосіб описати у XML важливі поняття, пов’язані з мультимедійним вмістом. Завдання полягає у забезпеченні сумісного пошуку, індексації, фільтрації та доступу, забезпечуючи взаємодію між пристроями, які мають справу з описом мультимедійного вмісту.

1.3.5. Програмне забезпечення MPEG-7: eXperimentation Model

Програмне забезпечення eXperimentation Model (XM) - це симуляційна платформа для дескрипторів MPEG-7, схем опису (DS), схем кодування (CS) та мови визначення опису (DDL). Окрім нормативних компонентів, платформа моделювання потребує також деяких ненормативних компонентів, по суті для виконання певного процесуального коду, який повинен бути виконаний на структурах даних. Структури даних та процедурний код разом утворюють додатки [14].

Низькорівневі дескриптори звуку MPEG-7 мають загальне значення для опису звуку. Існує 17 часових і спектральних дескрипторів, які можуть бути використані в різних додатках. Ці дескриптори можуть бути автоматично вилучені з аудіо та відображати зміну властивостей аудіо з часом або частотою. На основі цих дескрипторів часто можливо проаналізувати схожість

між різними аудіофайлами. Таким чином, можна ідентифікувати ідентичний, схожий чи несхожий аудіовміст. Це також забезпечує основу для класифікації аудіовмісту [16]. Дескриптори звуку низького рівня можна приблизно розділити на такі групи:

1. Основні спектральні дескриптори: Це описи аудіовмісту в часовій області.
2. Основні спектральні дескриптори: Усі чотири основні спектральні звукові дескриптори отримані в результаті єдиного часово-частотного аналізу звукового сигналу. Вони описують звуковий спектр з точки зору його оболонки, центроїда, поширення та площини.
3. Дескриптор параметрів сигналу: Два дескриптори параметрів сигналу застосовуються лише до періодичних або квазіперіодичних сигналів. Вони описують основну частоту звукового сигналу, а також гармонійність сигналу.
4. Часові тимбральні дескриптори: Тимбральні часові дескриптори можуть бути використані для опису часових характеристик сегментів звуків. Вони особливо корисні для опису музичного тембру (характерна якість тону незалежно від висоти та гучності).
5. Спектральні тимбральні дескриптори Тимбральні спектральні дескриптори - це спектральні особливості в лінійному частотному просторі, особливо застосовні до сприйняття музичного тембру.
6. Тимбральні дескриптори не пристосовані для опису аудіо сегментів, витягнутих з радіовипусків, тому не будуть розглядатися далі.

Представлення спектральної бази: Два дескриптори спектральної основи представляють низьковимірні проекції високовимірного спектрального простору для полегшення компактності та розпізнавання. Ці дескриптори використовуються в основному з інструментами опису звукової класифікації та індексації, але можуть бути корисними і для інших типів програм.

У стандарті MPEG-7 є два способи опису низькорівневих функцій звуку:

– Функцію LLD можна виділити із звукових сегментів змінної довжини, щоб позначити регіони з різними акустичними властивостями. У цьому випадку зведений дескриптор, витягнутий із сегмента, зберігається як опис аудіосегменту MPEG-7. Аудіосегмент являє собою часовий інтервал аудіоматеріалу, який може коливатися від довільно коротких інтервалів до всієї аудіо частини медіа-документа.
– Функцію LLD можна регулярно витягувати із звукових кадрів. У цьому випадку отримані вибіркові значення зберігаються як опис MPEG-7 ScalableSeries.

Так само можна навести основні параметри та позначення, які будуть використані для опису вилучення дескрипторів на основі кадру.

У часовій області для вхідного звукового сигналу будуть використовуватися такі позначення:

– індекс часових рамок.
– інтервал часу між двома послідовними часовими рамками.
– ціле число вибірок часу, що відповідає:
– тривалість часових рамок
– позначає ціле число вибірок часу, що відповідає.
– загальній кількості часових рамок.

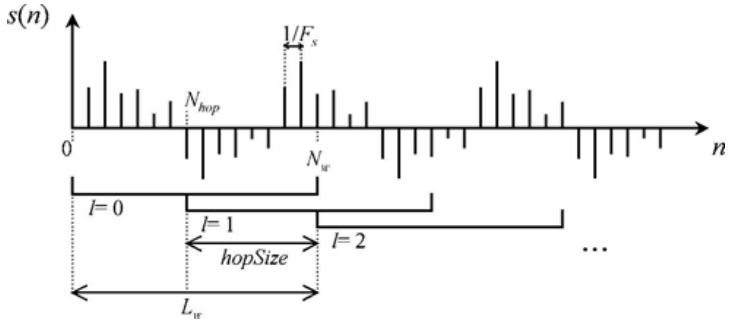


Рис. 1.2. Позначення дескрипторів на основі фреймі

1.4. Вимоги до розробки за результатами аналізу існуючих реалізацій

Робота зроблена в 2008 році Ignasi Esquerra та Giuseppe Dimattia є найбільш відповідною темі дипломної роботи і розкриває суть задачі з точки

зору класифікації радіоконтенту. Але алгоритмізація дескрипторних характеристик, зокрема, використання методу найближчого сусіда та моделі Маркова для знаходження і вилучення аудіофіч наразі є застарілою. Маючи великий спектр рішень класифікації звуків, мовлення, звучання інструментів і т. д. на основі DNN, в тому числі і в процесі вилучення фіч, було б неефективним використовувати застарілі підходи, адже нейромережі можуть дати більшу точність і гнучку реалізацію для будь-яких задач класифікації.

Висновки до розділу

У цьому розділі було проведено огляд об‘єкта та предмета дослідження, виявлено що запропоновані алгоритми класифікації є вузькоспеціалізованими, а реалізації – застарілими.

Як згадувалося, перша частина інтелектуальної системи стосується ідентифікації різних радіотипів. Чотири основні класи аудіо які можна виділити з них – це мова, музика, шум і тиша, але залежно від програми розглядаються більш конкретні класи, такі як голосне озвучування, мова з музикою та різні класи шуму.

Завдання сегментування або класифікації аудіо за різними класами було реалізовано [3] з використанням ряду різних схем. Слідом за роботою [4] існує декілька підходів для побудови моделі такої системи. Необхідно врахувати два аспекти - вибір ознак та класифікаційної моделі.

Більшість реалізацій працюють з окремими типами музики, голосу або мовлення, проте серед них в пункті 1.4 був обраний підхід щодо реалізації класифікації через стандарт MPEG-7 як відправна точка.

MFCCs з великим успіхом використовуються в системах розпізнавання мови, а згодом виявилися досить успішними і в завданнях класифікації звуку [2]. Інші ознаки (фічі) також були запропоновані на основі акустичних спостережень [5]. В пункті 1.4 запропоновано нові типи фіч на основі різних спостережень за характеристиками, що відокремлюють мовлення, музику та інші можливі класи аудіо. Характеристики, як правило, поділяються на основі

часового горизонту, який вони виділяють. Найпростіші запропоновані функції включають часову область та спектральні особливості. Характеристики часової області, як правило, представляють міру енергії або нульового значення перетину (ZCR).

Іншим аспектом, який слід врахувати, є класифікаційна схема, яку слід використовувати. Запропоновано ряд підходів до класифікації, які можна розділити на схеми, що базуються на правилах та моделях. Підходи, засновані на правилах, використовують деякі прості правила, відраховані від властивостей ознак (фіч). Оскільки ці методи залежать від порогових значень, вони не дуже стійкі до змінних умов, але можуть бути здійсненними для реалізації в режимі реального часу.

Підходи, засновані на моделях, включали класифікатори максимуму Posteriori (MAP), Модель суміші Гауса (GMM), K-найближчий сусід (K-NN) та лінійні персептрони, описаними Giuseppe Dimattia [1] є застарілими. Іншим підходом у цьому контексті є моделювання часової послідовності ознак або ймовірності перемикання між різними класами. Приховані моделі Маркова (HMM) враховують це, але не можуть дати необхідної точності. Опираючись на дослідження компанії Google [5] слід визнати, що найкращим варіантом рішення для класифікації довгих аудіоданих є CNN.

Зважаючи на великий об’єм підходів та рішень, було вирішено обрати декілька моделей та провести їх аналіз на практиці, так як предмет дослідження стосується декількох областей класифікацій звуків. Серед обраних підходів щодо моделей та мел-коефіцієнтів обрано – комплекс моделей існуючих для реалізації, а саме lstm, dbof, logistic та вилучення MFCC фіч.

Моделі та практична реалізація задачі класифікації аудіоконтенту на основі радіотипів представлені у наступному розділі. Відповідно зроблено порівняння отриманих моделей та їх характеристики та обґрунтовано найкращу реалізацію.

РОЗДІЛ 2. ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1. Перетворення аудіосигналу і класифікація нейромережею

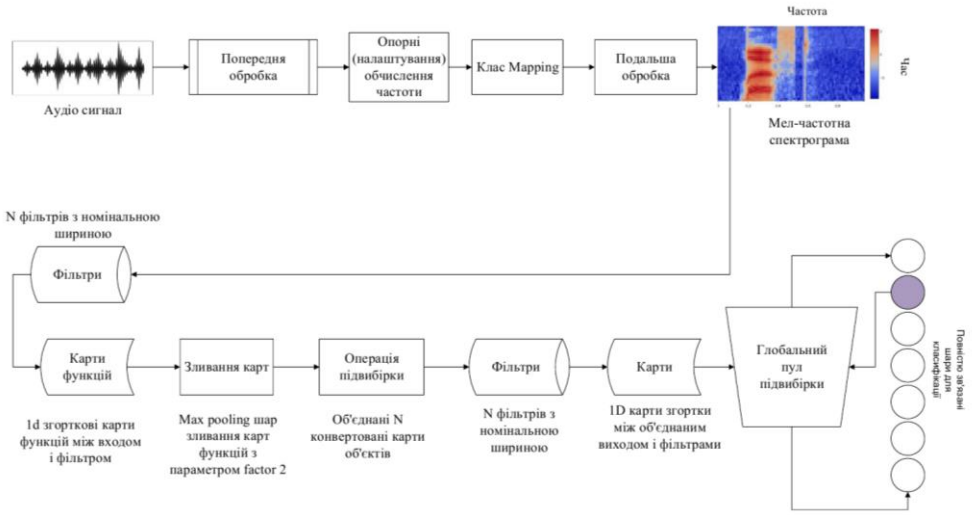


Рис. 2.1. Схема системи класифікації аудіо (додаток А)

В даній дипломній роботі у додатку А (тут – рис. 2.1.) описана загальна схема системи класифікації аудіо. Аудіосигнал перш ніж буде розроділений на відповідні класи радіотипів щодо його характеру проходить через відповідні перетворення та обчислення. Зазвичай для попередньої обробки використовується FFT, далі – опорні обчислення для виділення необхідних областей, клас mapping передає загальне представлення у векторному вигляді.

Для подальшої обробки зазвичай використовують дискретне перетворення Фур'є (детально postprocessing описано в розділі 3 пункт 3.4). Після процесінгу аудіо отримуємо мел-кепстральну спектрограму, для її характеристик виконується процес вилучення аудіофіч (надалі – фіч) за допомогою CNN, який детально описаний далі в пункті 2.1.1. Після вилучення фіч яке складається з відповідних кроків залежних від конкретної реалізації відбувається процес класифікації вилучених фіч, кількість яких на момент виходу з відповідної бібліотеки-екстрактора зазвичай дорівнює 13 на приблизно 1 мс в залежності від розміру вікна.

За процес класифікації отриманих даних відповідають схожі моделі, які також застосовуються до інших видів класифікації, як наприклад класифікації

зображень. Модель класифікації повертає лейбл або заголовок для сегменту поданого на початковий вхід аудіосигналу.

Для даної задачі серед існуючих практичних реалізацій використання нейромереж для класифікації аудіо слід виділити ті, область яких найбільш схожа по відношенню до класифікації довгого аудіоконтенту. Серед таких є робота групи дослідників Google [17], де один із дослідників Shawn Hershey описав роботу над архітектурою CNN для класифікації великих аудіо. В ній затверджено умови для класифікації саундтреків з датасету з 70 мільйонів відео. Зважаючи на те що CNN виявились дуже ефективними в класифікації зображень і демонструють таку саму перспективу щодо звуку. Вони використали різні архітектури CNN, щоб класифікувати саундтреки набору даних із навчальних відео (5,24 мільйона годин) за допомогою 30 871 міток рівня відео. Де розглянули повністю зв’язані глибокі нейронні мережі (DNN), AlexNet, VGG, Inception та ResNet. В ній дослідили різний розмір як навчального набору, так і лексики лейблів, з’ясувавши, що аналоги CNN, що використовуються в класифікації зображень, добре справляються з нашим завданням класифікації аудіо, а більша кількість тренувань та міток допомагають підвищити точність IC. Модель, яка використовує фічі з цих класифікаторів, працює набагато краще, ніж необроблені фічі для завдань класифікації Audio Set та Acoustic Event Detection (AED).

Датасет YouTube-100M було розділено на кадри, що не перекриваються, по 960 мс. Це дало приблизно 20 мільярдів прикладів із 70 мільйонів відео.

Кожен фрейм успадковує всі мітки батьківського аудіо. Кадри 960 мс розкладаються швидким перетворенням Фур'є із застосуванням 25 мс екранування спектрограми кожні 10 мс. Отримана спектрограма інтегрується в 64 роздільних MFCCs, і величина кожного трансформується після додавання невеликого зміщення, щоб уникнути числових проблем.

Це дає Mel-спектрограмні плями розміром 96 × 64 засічки, які утворюють вхідні дані для всіх класифікаторів.

Під час навчання передається 128 прикладів MFCCs шляхом випадкової вибірки з усіх засічок. Усі експерименти використовували TensorFlow і тренувались асинхронно на декількох графічних процесорах за допомогою оптимізатора Adam. Усі моделі використовували остаточний сигмоподібний шар, а не шар softmax, оскільки кожен приклад може мати декілька лейблів класифікації. Перехресна ентропія була функцією втрат. З огляду на великий розмір тренувального набору, в роботі не використовувалась функція відсіву, зменшення ваги чи інші загальні методи регуляризації. Для моделей, що навчаються на 7M або більше прикладів, не було помічено доказів перенавчання.

Для цієї задачі використовували наступні архітектури для опрацювання на датасеті Youtube-100M:

- Fully Connected
- AlexNet
- VGG
- Inception V3
- ResNet-50

Серед яких найкрощою виявилась ResNet-50. Коротко про її архітектурні модифікації:

- Було модифіковано стандарт ResNet-50:
- Видалено крок 2 з перших 7x7 згорткових шарів, щоб кількість активацій не надто відрізнялося в аудіоверсії.
- Змінено середній розмір пулу на 6 × 4 щоб відобразити зміну активацій. Оригінальна мережа має 26M ваг та 3,8B фіч. Аудіо варіант має вагу 30M і 1,9B фіч.

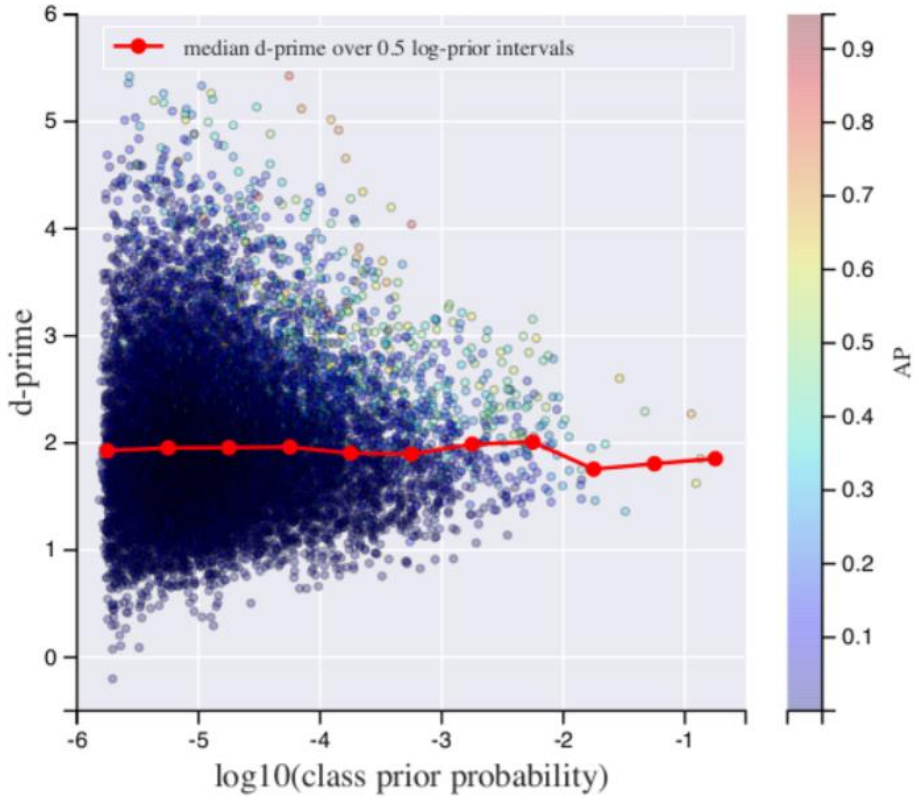


Рис. 2.2. Діаграма розсіювання ймовірністей ResNet-50 для класу d-prime проти log-prior. Кожна точка - це окремий клас із випадкової підмножини з набору 30K розмічених фіч. Колір відображає середню точність.

2.1.1. Огляд реалізації вилучення аудіофіч

Вилучення ознак (аудіофіч) – це процес перетворення звукового сигналу в послідовність векторів, що несуть характерну інформацію про аудіоконтент. Ці вектори використовуються як основа для різних типів алгоритмів аудіоаналізу. Типово для алгоритмів аудіоаналізу на основі функцій, обчислених за вікном Ханна (filter blanks). Засновані на вікні функції можна розглядати як короткий опис часу сигналу для конкретного моменту часу.

Видобуток аудіофіч є дуже важливим питанням для отримання оптимальних результатів у програмі. Вилучення потрібної інформації із звуку збільшує продуктивність системи та зменшує складність наступних алгоритмів. Як правило, для моделі потрібні різні додаткові обробоки, що покращують характеристики навчання. Для класифікаційних завдань існує

широкий спектр застосунків для виділення аудіофіч. Фічі можна розділити на дві категорії: часові і частотні.

У частотній області спектральні дескриптори часто обчислюються за допомогою короткочасного перетворення Фур'є (STFT). Поєднуючи це вимірювання з релевантною інформацією, такою як облік частоти та тимчасового маскування, можна створити слухову спектрограму, яка потім може бути використана для визначення гучності, тембру, початку, удару та темпу, а також висоти та гармонії [6]. На додаток до спектральних дескрипторів, існують також тимчасові дескриптори, які складаються з форми звукової хвилі та її амплітуди, дескрипторів енергії, дескрипторів гармонік, отриманих із синусоїдального гармонічного моделювання сигналу, та перцептивних дескрипторів, обчислених за допомогою тих самих DNN.

Більшість алгоритмів на основі DNN перетворюють оригінальні звукові сигнали в спектрограми перед їх обробкою. Спектрограми забезпечують візуальне представлення частот відносно часу. Методи, що використовують підхід, розподілений за часом [22,29], ділять спектрограму на кадри, щоб створити спектрограму, розподілену за часом. Спектрограма, розподілена за часом, використовується як вхід для CNN, щоб навчити модель розрізняти місцеві особливості на різних етапах часу.

Інший підхід [30] для класифікації звуку розбиває спектрограму вздовж частот, створюючи частотно-розподілену спектрограму. Використання цього підходу дозволяє моделям вивчати функції на основі різних частот.

Хоча моделі на основі спектрограм були дуже успішними, є деякі внутрішні проблеми, які важко вирішити. Зокрема, функція генерації спектрограми не залежить від подальшого процесу класифікації. Краще використовувати перевірене практичне рішення як Tensor Flow. У нашому випадку ми будемо використовувати таку модель, яка використовує неперетворені MFCC. Для цього експерименту ми використовуємо TensorFlow VGGish як екстрактор функцій з даних. Це 4-рівневі CNN, здатні мати справу

зі складними нелінійними відображеннями і можуть обмінюватися вагами на вході, що забезпечує узгодженість вхідних даних [31].

2.1.2. Огляд MFCC для роботи з аудіо класифікацією

Простими словами mel – це одиниця висоти звуку, заснована на сприйнятті цього звуку нашими органами слуху. Як відомо, АЧХ людського вуха навіть віддалено не нагадує пряму, і амплітуда - не зовсім точна міра гучності звуку. Тому, і ввели емпірично підібрані одиниці гучності, наприклад, фон:

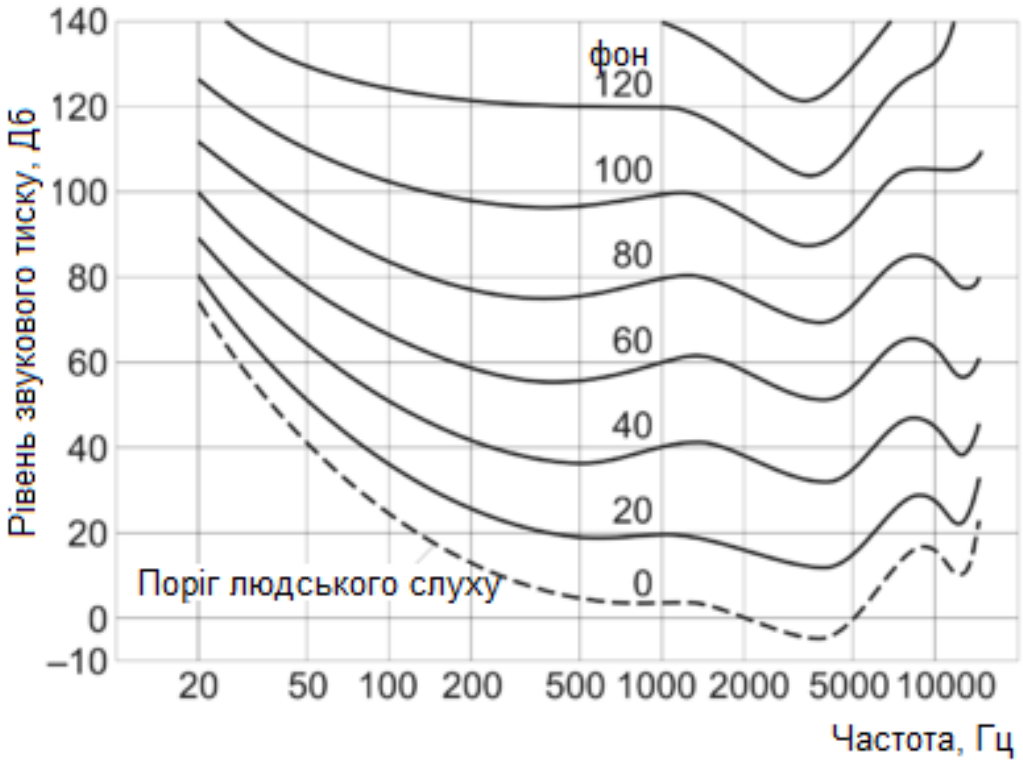


Рис. 2.3. Відображення емпіричної подібності

Аналогічно, сприймається людським слухом висота звуку не зовсім лінійно залежить від його частоти рис 2.4.

Така залежність не претендує на більшу точність, але зате описується простою формулою:

$$m = 1125 \ln(1 + f/700).$$

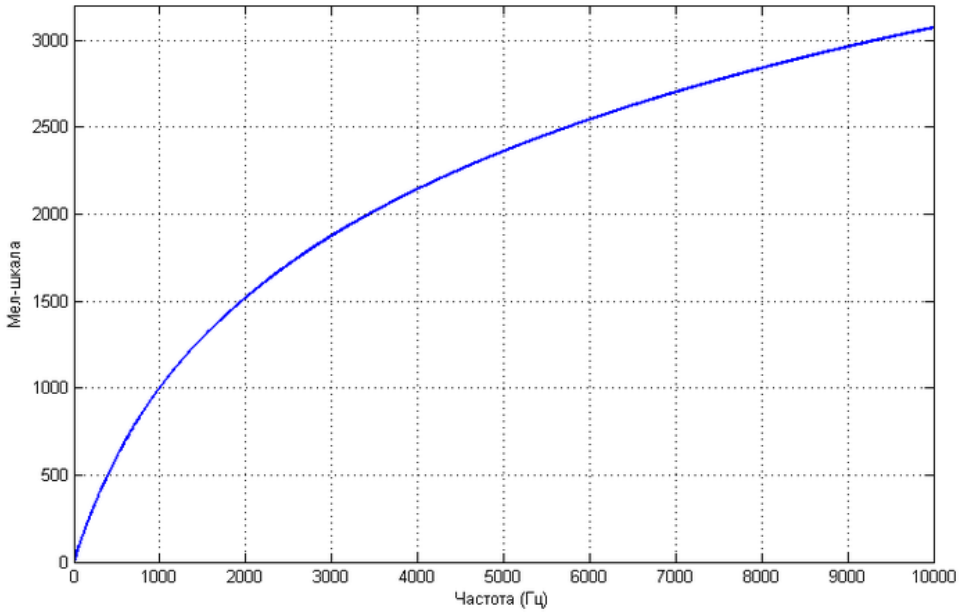


Рис. 2.4. Залежність висоти звуку мел-шкали до частоти

Подібні одиниці виміру часто використовують при вирішенні задач розпізнавання, так як вони дозволяють наблизитися до механізмів людського сприйняття, яке поки що лідирує серед відомих систем розпізнавання мови.

Відповідно до теорії словотворення мова являє собою акустичну хвилю, яка випромінюється системою органів: легкими, бронхами і трахеєю, а потім перетворюється в голосовому тракті. Якщо припустити, що джерела збудження і форма голосового тракту відносно незалежні, мовний апарат людини можна представити у вигляді сукупності генераторів тональних сигналів і шумів, а також фільтрів.

Схематично це можна представити так:

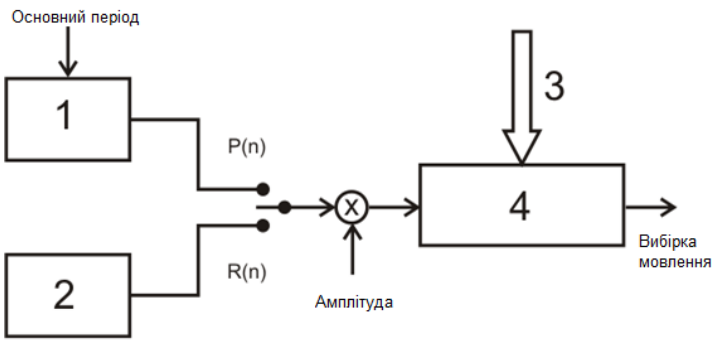


Рис. 2.5. Схема виміру для задачі розпізнавання

Сигнал на виході фільтра (4) можна представити у вигляді згортки:

$$\mathrm{f}(t) = \mathrm{s}(t) \otimes \mathrm{h}(t),$$

де s (t) - початковий вигляд акустичної хвилі, а h (t) - характеристика фільтра (залежить від параметрів голосового тракту)

У частотної області це виглядає так:

$$F(\omega) = S(\omega)H(\omega).$$

Отриману величину слід прологарифмувати, для щоб отримати замість нього суму:

$$\ln[S(\omega)^2 \cdot H^2(\omega)] = \ln S^2(\omega) + \ln H^2(\omega).$$

Тепер нам потрібно перетворити цю суму так, щоб отримати непересічні набори характеристик вихідного сигналу і фільтра. Для цього є кілька варіантів, наприклад зворотне перетворення Фур'є:

$$C(q) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \ln[F(\omega)^2]e^{i\omega q}d\omega.$$

Також в залежності від цілей можна використовувати пряме перетворення Фур'є або дискретне косинусное перетворення

Щоб зрозуміти, як перетворити аудіо в набір коефіцієнтів MFCC, потрібно навести приклад на основі простого короткого звуку.

Для цього візьмемо звук і покажемо її тимчасове представлення:

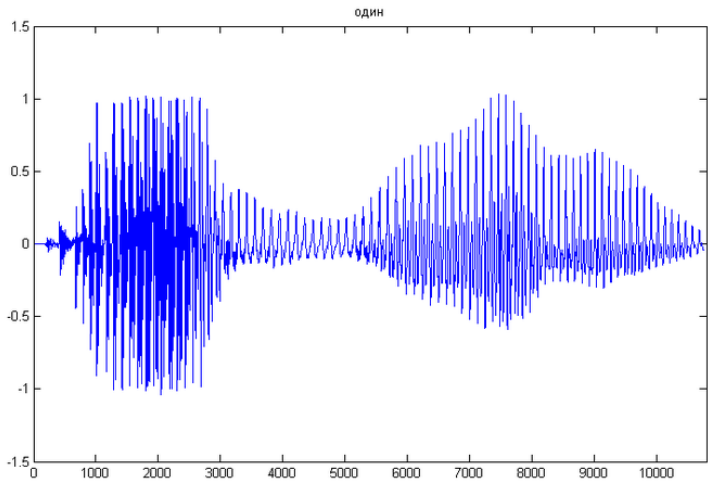


Рис. 2.6. Короткий звуковий сигнал в осцилограмі

Насамперед нам потрібен спектр вихідного сигналу, який ми отримуємо за допомогою перетворення Фур'є. Для простоти прикладу, не будемо розбивати сигнал на частини як це використовується в реаліях [17, 11, 5, 4], тому беремо спектр по всій тимчасової осі:

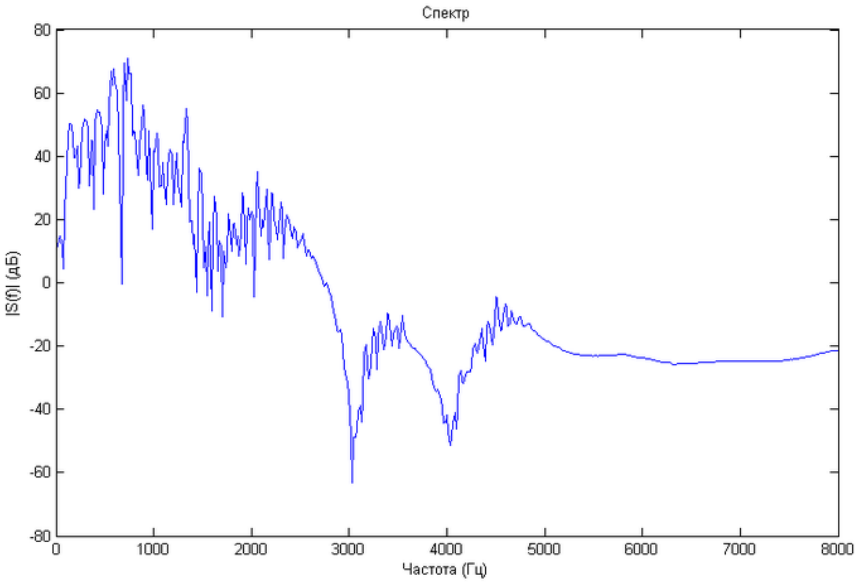


Рис. 2.7. Спектр звукового сигналу на осі часу

Отриманий спектр нам потрібно розташувати на mel-шкалі. Для цього ми використовуємо вікна, рівномірно розташовані на mel-осі.

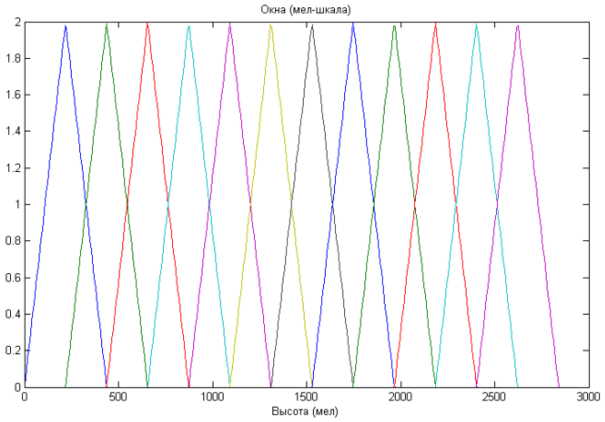


Рис. 2.8. Спектр сигналу розділений вікнами на mel-осі

Якщо перевести цей графік в частотну шкалу:

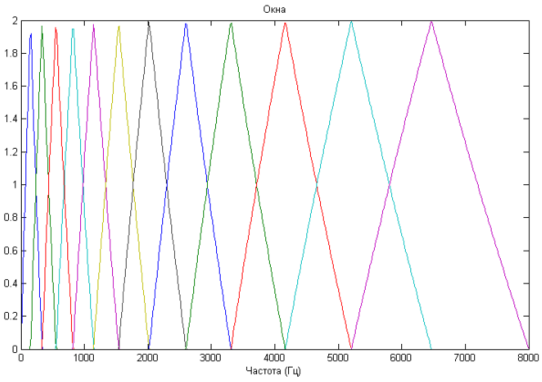


Рис. 2.9. Спектр сигналу на осі частот

На цьому графіку помітно, що вікна «збираються» в області низьких частот, забезпечуючи більш високу «дозвіл» там, де воно необхідне для розпізнавання.

Простим перемножением векторів спектра сигналу і віконної функції знайдемо енергію сигналу, яка потрапляє в кожне з вікон аналізу. Тоді можна отримати деякий набір коефіцієнтів, але це ще не ті MFCC, які можна використа у навчанні. Поки це ще спектральними коефіцієнтами. Тому зводимо їх в квадрат і логарифмуємо. Нам залишилося тільки отримати з них кепстральні фічі, або «спектр спектра». Для цього ми могли б ще раз застосувати перетворення Фур'є, але краще використовувати дискретне косинусне перетворення [18].

В результаті отримуємо послідовність приблизно такого вигляду:

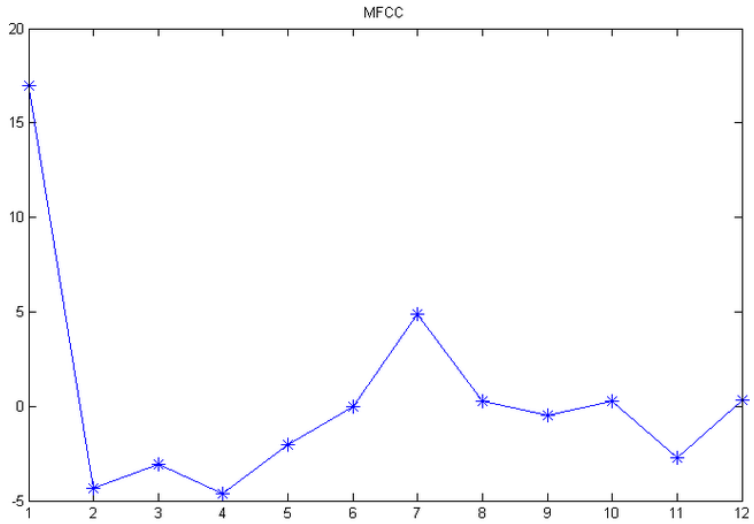


Рис. 2.10. Виділені MFCC після перетворень

Таким чином ми маємо дуже невеликий набір значень, який при розпізнаванні успішно замінює тисячі відліків мовного сигналу. У книгах та роботах [19, 20, 22] пишуть, що для задачі розпізнавання та класифікації можна взяти перші 13 з 24 обчислених коефіцієнтів рандомно. У будь-якому випадку це набагато менший обсяг даних, ніж спектрограма або тимчасове представлення сигналу.

Для кращого результату можна розбити вихідний звук на відрізки невеликої тривалості, і обчислювати коефіцієнти для кожного з них. Також може допомогти «зважування» віконних функцій [21]. Все залежить від алгоритму, для якого потрібно передати результат.

Для класифікації довгих аудіо, серед яких буде і музика, і мовлення найкращим рішенням є MFCCs усіх частот.

2.1.3. Огляд модифікацій MFCCs

В роботі Ruben Gonzalez – Better than MFCC Audio Classification Features було представлено нові типи фіч які можна вилучити і надати для нейромоделі.

Ідея введення нових фіч грунтується на тому що більшість інформації вилучених з аудіо – викидається.

Зокрема, представлений покращений набір функцій MFCC (MFCC +) додає спектральні та часові функції до MFCC. Три запропоновані набори функцій включають принципові спектральні коефіцієнти (PSC), принципові цестральні коефіцієнти (PCC) та принципові спектрально-часові коефіцієнти (PSTC).

Цепстральні коефіцієнти мел-частоти обчислюються за короткочасним перетворенням Фур'є як цепструма спектра деформації розплаву. Частоти коефіцієнтів Фур'є переназначаються на шкалу розплаву за допомогою співвідношення (1) та трикутних вікон, що перекриваються, на всю октаву. Нарешті, кепструм (2) отримують з використанням ремафікованих коефіцієнтів m (n).

$$Mel(f) = 2595 \log_{10}(1 + f/100),$$

$$c(k) = DCT\left\{\log|DFT\{m(n)\}|\right\}.$$

Покращений набір даних MFCC + включає чотири загальні часові ознаки та шість спектральних ознак. Чотири часові характеристики – це нульовий коефіцієнт перетину (ZCR), середньоквадратичне значення (середньоквадратичне значення), короткочасна енергія (E) та енергетичний потік (F). Вони визначаються наступним чином:

$$ZCR = \sum_{k=2}^{k} |\mathrm{sgn}\big(x(k)\big) - \mathrm{sgn}(x(k-1))|,$$

$$sgn(n) = \begin{cases} 1, n > 0 \\ 0, n = 0 \\ -1, n < 0 \end{cases},$$

$$E = \frac{1}{K}\left(\sum_{k=1}^{k} |X(k)|^2\right),$$

$$F = E(n) - E(n-1).$$

Шість використовуваних спектральних характеристик – це пропускна здатність сигналу (BW), спектральний центроїд (SC) та висота тону (P) за допомогою субгармонічного підсумовування [8], висота тону та гармонійність за методом Брегмана [9] та перекіс, який є відсотком енергії в висоті звуку відносно гармонічних частинок.

$$BW = \sqrt{\left(\sum_{k=1}^{k}(k - SC)^2\,|X(k)|^2\right)\Big/\left(\sum_{k=1}^{k}|X(k)|^2\right)},$$

$$SC = \left(\sum_{k=1}^{k} k \times |X(k)|^2\right)\Big/\left(\sum_{k=1}^{k}|X(k)|^2\right),$$

$$P = f : f \geq 0 \wedge \forall g \geq 0, \text{ де } \mathrm{H}(f) \geq \mathrm{H}(g),$$

$$H(f) = \sum_{k=1}^{k} h_k X(k \cdot f).$$

Замість того, щоб просто довільно вибирати ознаки, засновані на суб'єктивному понятті виокремленості, з статистичної точки зору можна ідентифікувати основні компоненти в будь-якому даному наборі даних. Ці основні компоненти гарантовано оптимально відображають базові дані для будь-якої кількості компонентів.

Зазвичай для отримання цих компонентів може бути використаний або аналіз принципових компонентів (PCA), або його еквівалентне перетворення Кархунена-Лоева (KL). На практиці перетворення KL часто апроксимується за допомогою дискретного косинусного перетворення (DCT). Статистично оптимальний набір ознак Принципових спектральних компонентів (PSC) можна відповідно отримати, взявши кілька перших коефіцієнтів DCT спектра, отриманих за допомогою короткочасного перетворення Фур'є (де || представляє комплексну величину):

$$\mathrm{PCC}(k) = \mathrm{DCT}\{\mathrm{DFT}\,\{\mathrm{x(n)}\}\}.$$

Варіацією PSC, яка забезпечує більш рівномірне масштабування набору функцій за допомогою відбілювання спектра, є використання Принципових Цепстральних Компонентів (PCC):

$$\mathrm{PCC}(k) = \mathrm{DCT}\{\log|\mathrm{DFT}\,\{\mathrm{x(n)}\}|\}.$$

Оскільки методи PSC та PCC формуються з одновимірного спектра, вони не можуть зафіксувати еволюцію спектру з часом. Часові характеристики звуків, як відомо, важливі для ідентифікації деяких класів звуку. Відповідно набір функцій PSTC фіксує основну інформацію, що міститься в частотно-часовому розподілі енергії в аудіосигналах. Цей набір функцій отриманий шляхом двовимірного дискретного косинусного перетворення (DCT) спектрограми звукового сигналу.

Для оцінки ефективності запропонованих ознак класифікації звуку було використано п’ять дуже різних наборів даних. Набір даних "Four Audio" складався з 415 окремих записів мови (71), музики (197), оплесків (85) та сміху (62). Вони були отримані з різних джерел, включаючи записи, виступи в прямому ефірі та засоби масової інформації. Всі вони тривали 2,5 секунди та відбирали частоти 44,1 кГц та 16 біт. Набір даних «Frog Calls» складався з 1629 записів 74 різних видів місцевих австралійських викликів жаб [10]. Їх відбирали з частотою 22,05 кГц та 16 бітами, кожна тривала 250 мілісекунд. Набір даних «Комаха» складався із записів звуків, виданих 381 різним видом комах, і були класифіковані за чотирма наступними родинами: Katydid,

Cricket, Cicada та іншими. Всі вони тривали 5 секунд і відбирали частоти 44,1 кГц та 16 біт.

Набір даних «Музичні інструменти» складався з 1345 записів 97 різних музичних інструментів [11]. Це були категорії в один із дванадцяти різних класів: фортепіано, клавесин, щипкова струна, струнна смичка, деревний духовий інструмент, латунь, орган, тимпани, ударно-металеві перкусії, дерев'яні перекутані, неналаштовані ударні та інші. Всі вони були відібрані на частоті 44,1 кГц та 16 біт, кожна тривалістю 500 мілісекунд. Для кожного даного набору даних були сформовані вектори об’єктів різного розміру від 8 до 96 вимірів як перші N об’єктів з кожного з п’яти наборів ознак.

Потім ці вектори оцінювали за допомогою класифікатора k-NN (k = 1), використовуючи десятикратну перехресну перевірку. Оскільки більшість функцій, використаних в експериментах, були отримані за допомогою короткочасного перетворення Фур'є, спочатку потрібно було визначити оптимальний розмір вікна. Для цього класифікатор пройшов навчання з ознаками, виділеними із спектрів Фур'є при різних розмірах вікон. Це було виконано для всіх наборів даних і всіх наборів функцій. Для наборів даних Four Audio, Frog Call та Insect найкращі результати для всіх наборів функцій були отримані з використанням найбільшого розміру вікна - 1024 вибірки. За винятком функцій PSTC, найкращої продуктивності для всіх наборів функцій з використанням наборів даних Instruments and Environment було досягнуто за допомогою вікна з 512 зразками. У випадку з функціями PSTC розмір вікна аналізу становив 256 зразків для набору даних Instruments та 128 зразків для набору даних звуків навколишнього середовища. Причиною цього є те, що кількість векторів, необхідних для навчання, чинить тиск на кількість доступних зразків, з яких формується кожен тренувальний вектор. Це призводить до компромісу між формуючими елементами, що забезпечують кращу спектральну або часову роздільну здатність.

У випадку з наборами даних «Інструменти та навколишнє середовище», збільшення тимчасового дозволу забезпечило кращу продуктивність. Щоб

гарантувати, що класифікатор був однаково навчений у всіх випадках, для будь-якого даного набору даних було використано рівно однакову кількість зразків для формування векторів навчання для всіх наборів ознак. Щоб гарантувати, що ефективність наборів ознак не залежить від кількості навчальних векторів, спочатку було визначено оптимальну кількість навчальних векторів для кожного набору ознак (фіч) та набору даних.

Це було здійснено шляхом послідовного вилучення безлічі векторів із кожного запису не перекриваючись. У всіх випадках найкращі показники були отримані, коли класифікатор навчався з максимальною кількістю векторів навчання, які можна було витягти з наборів даних, що було однаковим для всіх наборів фіч, за винятком PSTC. Для набору даних «Four Audio» найкращу ефективність було отримано із використанням 100 векторів тренувань для кожного запису для всіх наборів фіч, крім набору PSTC фіч, де було використано 20 векторів тренувань.

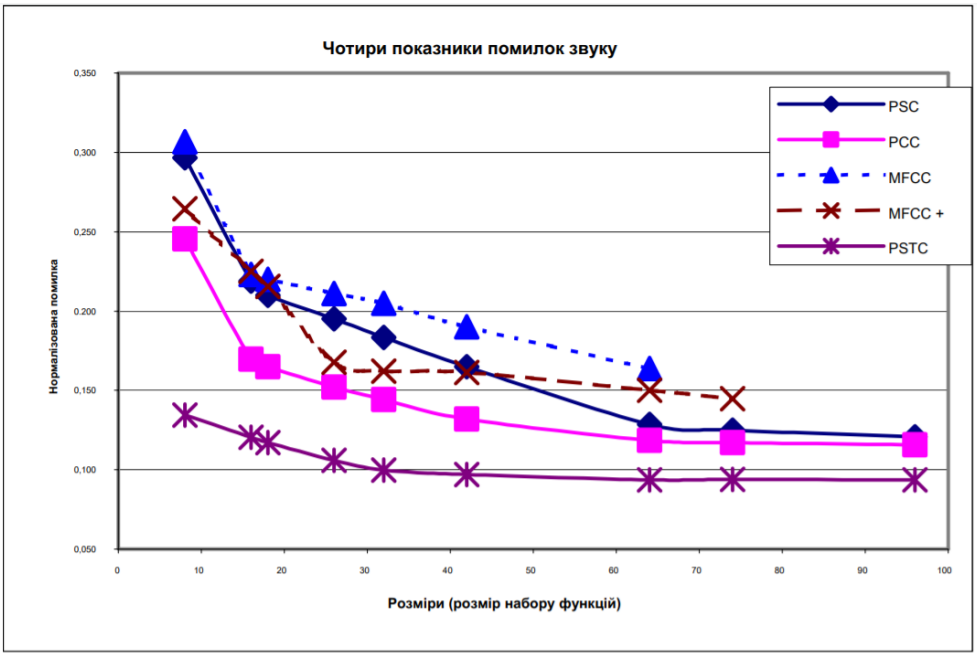


Рис. 2.11. Вилучення базових векторів для різних фіч

Features	8	16	18	26	32	42	64	74	96
PSC	0.297	0.219			0.183		0.129		0.121
PCC	0.245	0.170			0.144		0.119		0.116
MFCC	0.307	0.223	0.220		0.205		0.164		
MFCC+			0.215	0.168		0.161		0.145	
PSTC	0.135	0.121			0.100		0.094		0. 094

Рис. 2.12. Коефіцієнт помилок для різних типів аудіофіч

Результати, показані в рис. 2.11 і в рис. 2.12 показали нормований коефіцієнт помилок класифікації для кожного з наборів ознак для кожного розміру оціненого вектора. Для цього набору даних функції PSTC явно перевершили всі інші завдяки характеристикам PCC, що забезпечують другу найкращу продуктивність. За розмірами партії характеристики PSC працювали ненабагато краще ніж MFCC, але наближалися до PCC фіч при більш високих розмірах вікна, коли було використано достатньо коефіцієнтів. Лише MFCCs мали найгіршу загальну ефективність. Покращені функції MFCC + забезпечували помірну продуктивність при малих розмірах вікна, що не покращувалось при більших розмірах. Через невелику кількість даних, доступних для аналізу в наборі даних Frog Call, із кожного запису можна виділити лише п’ять векторів навчання, використовуючи вікно розміром 1024 вибірки. У випадку функцій PSTC з даних можна було вилучити лише один вектор поїзда, і це серйозно завадило роботі набору фіч PSTC.

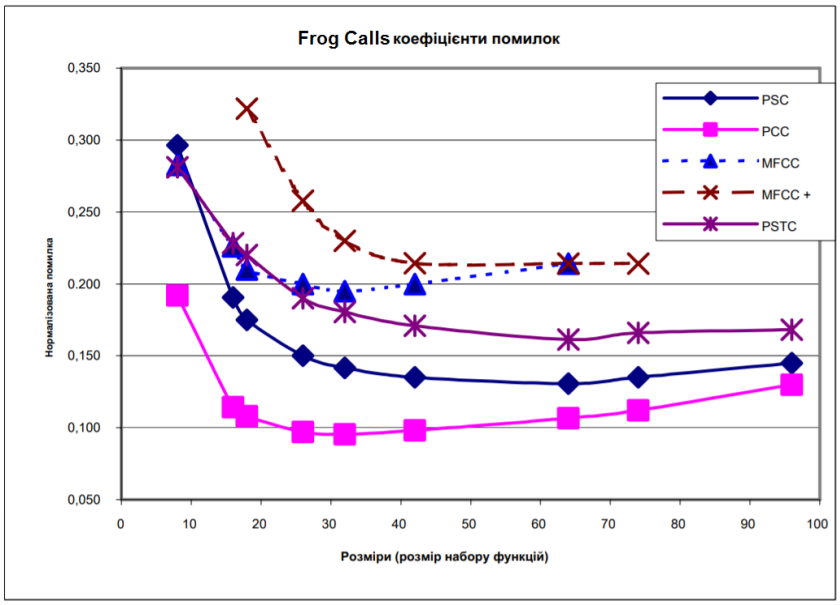


Рис. 2.13. Вилучення базових векторів для датасету Frog Calls

Features	8	16	18	26	32	42	64	74	96
PSC	0.296	0.191			0.142		0.131		0.145
PCC	0.192	0.114			0.095		0.107		0.130
MFCC	0.283	0.226			0.195		0.214		
MFCC+			0.322	0.258		0.214		0.214	
PSTC	0.281	0.228			0.181		0.161		0.168

Рис. 2.14. Нормований коефіцієнт помилок для Frog Calls

Результати наведені в рис. 2.13 і рис. 2.14. Хоча PSTC фічі перевершили MFCC та MFCC +, вони відстали від характеристик PSC та PCC. Фічі PCC знову перевершили характеристики PSC у всіх вимірах. Примітно, що фічі MFCC перевершили розширені фічі MFCC + у всіх, крім найвищих розмірів. Найкраща продуктивність для набору даних звуків комах була отримана за допомогою 50 навчальних векторів (20 у випадку функцій PSTC) та розміру вікна 1024 зразки. Цей набір даних надав результати, подібні до набору даних Frog Call, як це демонструють результати, показані в рис. 2.15 і в рис. 2.16.

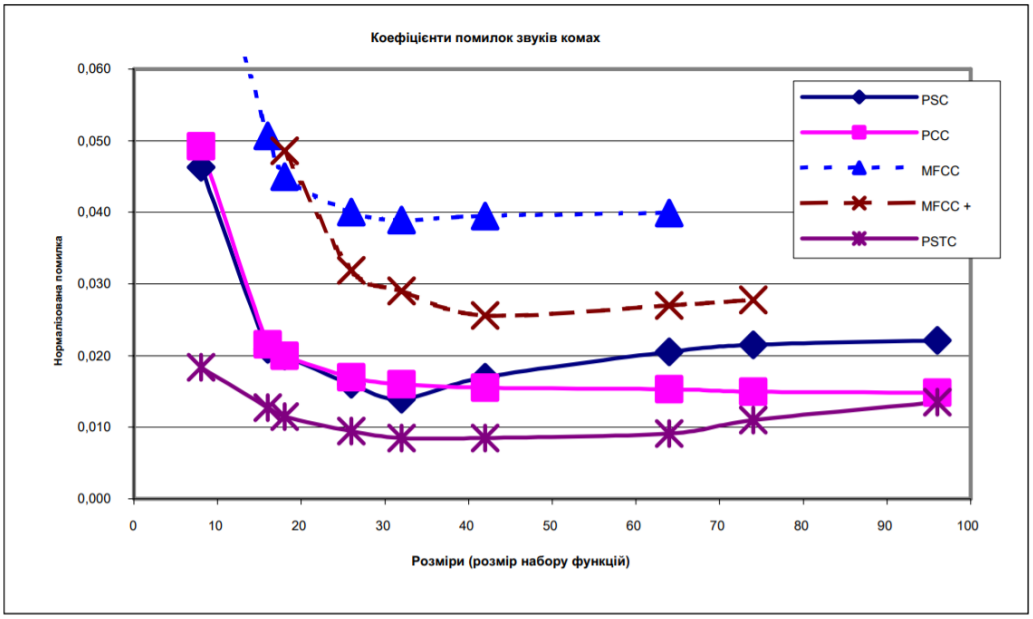


Рис. 2.15. Вилучення базових векторів для звуків комах

Features	8	16	18	26	32	42	64	74	96
PSC	0.046	0.021			0.014		0.021		0.022
PCC	0.049	0.022			0.016		0.015		0.015
MFCC	0.084	0.051			0.039		0.040		
MFCC+			0.049	0.032		0.026		0.028	
PSTC	0.018	0.013			0.008		0.009		0.014

Рис. 2.16. Нормований коефіцієнт помилок для звуків комах

Фічі PTFC знову забезпечили найкращу загальну продуктивність у всіх вимірах. Як і слід було очікувати, MFCC + працював краще, ніж MFCC, за винятком низьких розмірів, але обидва вони знову продемонстрували найгіршу продуктивність.

Характеристики PSC та PCC дали дуже схожі результати, за винятком того, що PSC стали менш конкурентоспроможними при більших розмірах. Результати набору даних інструментів помітно відрізняються від результатів інших наборів даних. Найкраща продуктивність у всіх випадках була отримана за допомогою 50 векторів тренувань та вікна вибірки 512 для всіх наборів функцій, крім функцій PSTC, які використовували 5 векторів тренувань та 256 вікон вибірки.

Покращений MFCC + трохи перевершив функції PSTC, як показано в рис. 2.17 і в рис. 2.18.

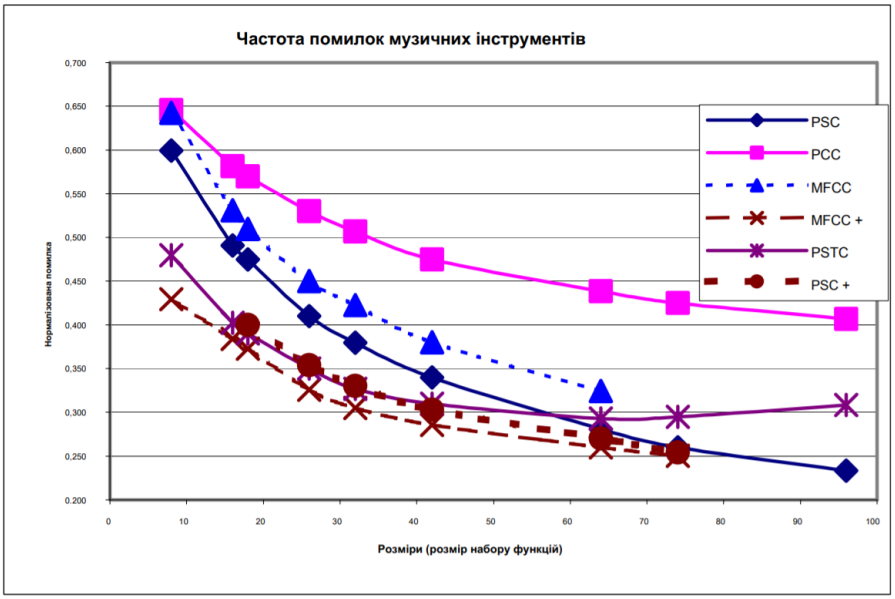


Рис. 2.17. Вилучення базових векторів для датасету музичних інструментів

Features	8	16	18	26	32	42	64	74	96
PSC	0.599	0.491			0.379		0.281		0.233
PCC	0.646	0.581			0.507		0.439		0.407
MFCC	0.642	0.531			0.423		0.325		
MFCC+	0.451		0.373	0.326		0.286		0.250	
PSTC	0.480	0.402			0.327		0.293		0.308
PSC+			0.400	0.354		0.303		0.254	

Рис. 2.18. Нормований коефіцієнт помилок для музичних інструментів

Характерно, що PCC фічі працювали навіть гірше, ніж стандартні MFCC. Однак функції PSC зуміли забезпечити кращу продуктивність, ніж MFCC, для всіх вимірів. Щоб оцінити внесок спектральних та часових особливостей набору даних MFCC + у продуктивність, вони були додані до набору даних PSC, щоб сформувати розширений набір даних PSC +.

Ці вдосконалені PSC + фічі змогли наблизитись, але не перевершити продуктивності MFCC + фіч. Найкраща продуктивність із набором звуків навколишнього середовища була отримана з використанням 10 векторів та 512 зразка вікна у всіх випадках, за винятком PSTC фіч, які використовували 5 векторів навчання та 128 вікон як зразків. У цьому випадку фічі PSC забезпечували найкращу продуктивність, як показано в рис. 2.20 і в рис. 2.21.

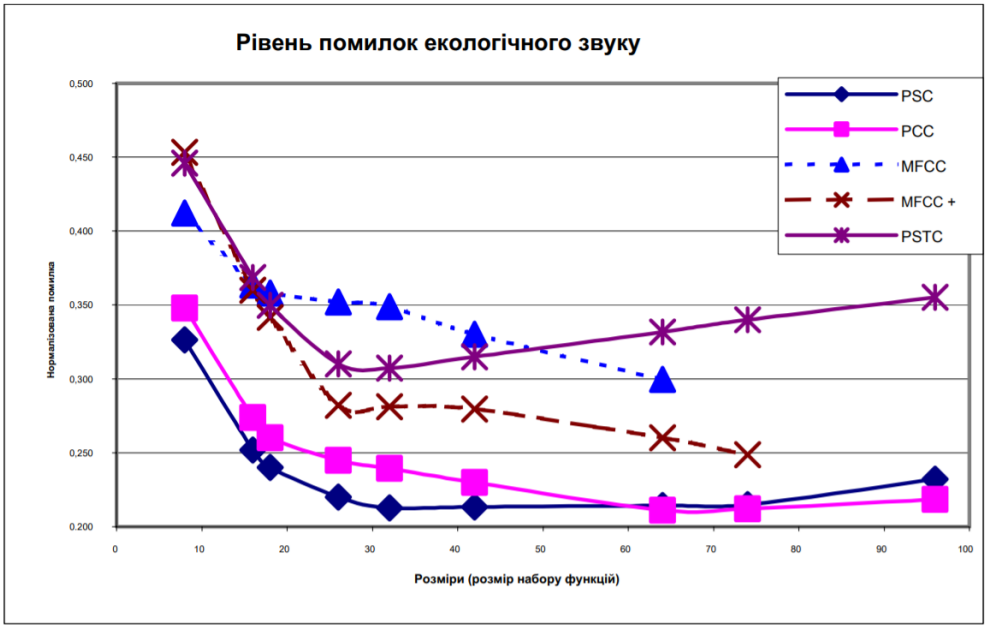


Рис. 2.20. Вилучення базових векторів для екологічного звуку

Features	8	16	18	26	32	42	64	74	96
PSC	0.326	0.252			0.213		0.214		0.232
PCC	0.348	0.274			0.240		0.211		0.219
MFCC	0.412	0.363			0.349		0.300		
MFCC+	0.453		0.342	0.282		0.280		0.249	
PSTC	0.446	0.369			0.307		0.332		0.355

Рис. 2.21. Нормований коефіцієнт помилок для екологічного звуку

PCC фічам вдалося перевершити показники PSC лише при великих розмірах. Характерно, що функції PTFC забезпечували помірну

продуктивність кращу, ніж MFCC, але гіршу, ніж MFCC + при низьких та середніх розмірах, але погіршувались при більш високих розмірах. Незрозуміло, чи це було пов’язано з недостатньою кількістю зразків для отримання достатньої кількості тренувальних векторів з досить високою роздільною здатністю частоти або якоюсь іншою внутрішньою характеристикою набору даних.

Зважаючи на дослід проведений Рубеном, можна зробити висновок, що подальше вдосконалення MFCCs можливе і необхідно точно розуміти область застосування нових типів аудіофіч, так як результати для різних типів задач виявились неоднозначними, але обгрунтованими, тобто вартими подальших досліджень.

2.2. Вибір застосування для вилучення аудіофіч

Алгоритми класифікації аудіоконтенту загалом можна розділити на дві частини: вилучення ознак та класифікація [5]. Частини витягування ознак переважно реалізуються CNN, оскільки вони можуть ефективно витягувати характеристики з необроблених даних [6, 7].

Реалізацію CNN можна згрупувати у два класи залежно від того, як вони попередньо обробляють вхідні звуки: сигнали через форму хвиль [7, 8] або спектрограми [11, 12, 13]. Метод, заснований на формі хвиль, безпосередньо обробляє вхідні дані як 1D масив даних, тоді як реалізація на основі спектрограми повинна перетворювати необроблені аудіосигнали в спектрограми за допомогою перетворення Фур'є.

Порівняно з методом, заснованим на формі хвиль, методи, засновані на спектрограмі, вручну витягують інформацію про частоту та наносять їх на теплову мапу. Іншими словами, сили частот у кожен момент позначаються кольором або яскравістю. Ця попередня обробка може полегшити навчання CNN функцій, пов'язаних з частотою. Для порівняння, алгоритм, заснований на формі хвилі, обробляє вихідні аудіофайли безпосередньо, не залучаючи

будь-яких графіків, які можуть викликати додаткові шуми та / або зробити структуру даних розрідженою.

Крім того, весь алгоритм (вилучення ознак і класифікація) може навчатись або тренуватись разом і налаштовувати разом методи, засновані на спектрограмі, які відповідно будуть контролювати побудову спектрограми.

Метод багатозадачного навчання (MTL) [14] використовується для кількох завдань класифікації аудіо. MTL є фокусом машинного навчання, в якому комплекс навчальних завдань вирішуються одночасно, покращуючи точність класифікацій за рахунок зменшення розриву між навчальними та тестовими похибками [14].

Використовуючи спільний прихований рівень (shared hidden level), нейронні мережі можуть використовувати метод MTL [14] Дослідження показали, що моделі, засновані на MTL-SVM, мають кращу продуктивність, ніж моделі SVM, визначені окремим завданням [15]. Причина, по якій MTL працює, полягає в тому, що ті фактори, які ініціюють зміну даних, можуть бути розподілені між різними завданнями.

Незважаючи на те, що багато методів класифікації аудіо є ефективними для конкретного класу класифікації, дослідники використовували згорткові мережі глибоких переконань (CDBN) для класифікації аудіоданих з високою продуктивністю за кількома завданнями класифікації аудіо [16]. Використання DNN для вилучення цепстральних ознак також було використано для багатьох завдань класифікації аудіо [7].

Дослідники використовують глибокі залишкові мережі (ResNets) разом із механізмом затвора для вилучення представлення об’єктів ознак в аудіоданих. Це було продемонстровано більш ефективно із кількома завданнями класифікації аудіоконтенту та досягло вищої точності порівняно із специфічними моделями завдань, які навчались окремо [17].

Для цього будемо використовувати бібліотеку VGGish яка зарекомендувала себе у схожих задачах класифікації [15, 16]:

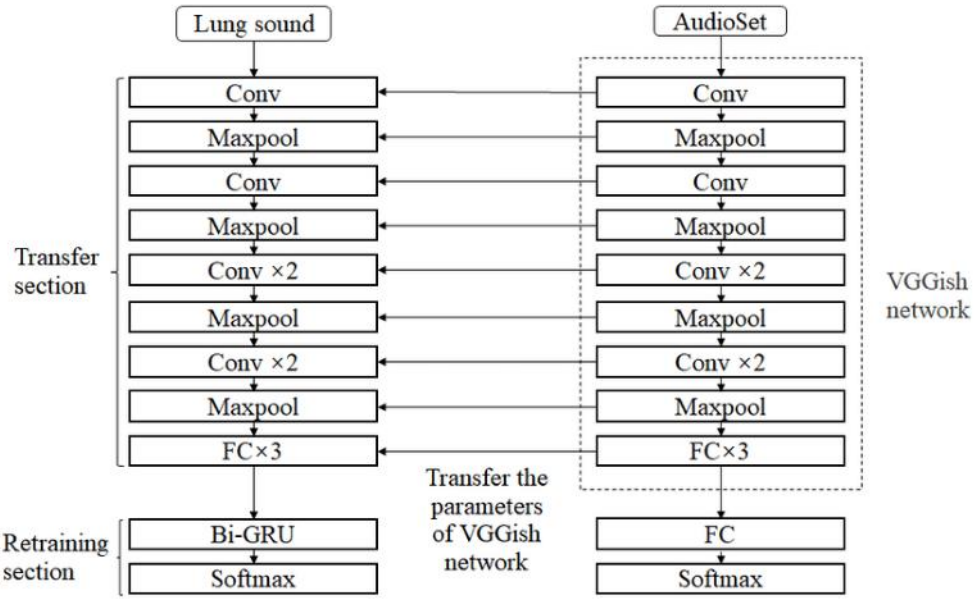


Рис. 2.21. Схема роботи мережі бібліотеки VGGish

На високому рівні конвеєр попередньої обробки робить наступне:

- Неопрацьовані дані модуляції імпульсного коду з файлу wav перетворюються у плаваючі за шкалою [-1,0, +1,0].
- Якщо каналів два, елементи усереднюються для отримання одного каналу.
- Дані передискретизуються лише до 16000 зразків на секунду.
- Дані розбиті на кілька вікон, що перекриваються.
- Вікно Хеммінга застосовується до кожного вікна.
- Спектр потужності розраховується за допомогою швидкого перетворення Фур'є.
- Частоти, що перевищують і нижче певних порогів, знижуються.
- Застосовуються банки фільтрів частоти Mel.
- Нарешті, натуральний логарифм береться з усіх значень.
- Конвеєр попередньої обробки приймає на вхід аудіо на 975 мс (точна довжина введення залежить від частоти дискретизації) і створює масив фігури (96, 64).

2.2.1. Вилучення фіч за допомогою VGGish

VGGish - це попередньо навчена згорткова нейронна мережа від Google. Докладніше можна дивитися у статті та на сторінці GitHub. Як випливає з назви, архітектура цієї мережі натхнена відомими мережами VGG, що використовуються для класифікації зображень. Мережа складається з серії шарів згортки та активації, за яким необов'язково слід максимальний рівень об'єднання. Ця мережа містить 17 шарів загалом.

Ця мережа зберігається статичною під час навчання моделям. В практичній роботі було видалено останні три шари оригінальної моделі VGGish. В якості вхідних даних для завершального етапу було використано найширший рівень із вихідної мережі. Ця модифікована модель VGGish видає подвійний вектор довжиною 12 288. У системах, що не *unix, модель також була квантована у вісім бітів, щоб зменшити її розмір.

2.2.2. Перевикористання глибоких фіч

Це єдиний етап, який оновлюється на основі вхідних даних. Під час навчання класифікатора звуку ця модель навчається з використанням функцій VGGish та міток введення. Під час прогнозування за допомогою цієї мережі робиться лише прямий прохід. Ця спеціальна нейронна мережа - це проста тришарова нейронна мережа. Перші два шари - це щільні шари, по 100 одиниць кожен. Ці шари використовують активацію RELU. Кінцевий шар – softmax. Кількість одиниць у цьому шарі дорівнює кількості міток.

Часто доводиться тренувати та оцінювати більше одного екземпляра моделі на одному наборі даних. Наприклад, цього вимагають як перехресна перевірка K-Folds, так і налаштування гіперпараметрів. У таких випадках більша частина роботи в різних прогонах може бути використана повторно.

Класифікація звуку проводиться за допомогою триступеневого процесу. Робота, виконана на перших двох із цих етапів (попередня обробка сигналу та вилучення функцій VGGish), буде однаковою для всіх циклів і може бути використана повторно.

У наведеному нижче коді використовується набір даних ESC-10 із вступного прикладу для перехресної перевірки у 5 разів:

```
import turicreate as tc
data = tc.load_sframe('./ESC-10')
# Calculate the deep features just once.
data['deep_features'] = tc.sound_classifier.get_deep_features(data['audio'])
accuracies = []
for cur_fold in data['fold'].unique():
    test_set = data.filter_by(cur_fold, 'fold')
    train_set = data.filter_by(cur_fold, 'fold', exclude=True)
    model = tc.sound_classifier.create(train_set,
                                       target='category',
                                       feature='deep_features')
    metrics = model.evaluate(test_set)
    accuracies.append(metrics['accuracy'])
print("Accuracies: {}".format(accuracies))
```

Це дозволяє нам виконувати 5-кратну перехресну перевірку більш ніж удвічі швидше. Для більших наборів даних економія часу буде ще більшою.

2.2.3. Конфігурація нейромереж для аудіо

Спеціалізована нейронна мережа, що використовується великим аудіо класифікатором, складається з ряду щільних шарів. Використання більшої кількості шарів або більше одиниць у кожному шарі може суттєво вплинути на точність. Це також вплине на розмір вашої моделі.

Параметр custom_layer_sizes дозволяє вказати, скільки шарів і кількість одиниць у кожному шарі. Значеннями за замовчуванням для цього параметра є [100, 100], що відповідає двом щільним шарам по 100 одиниць кожен.

Використання меншої кількості загальних одиниць призведе до зменшення моделі, можливо, з мінімальним впливом на точність. Для великої кількості навчальних даних, щоб отримати кращу точність, потрібно використовувати більше шарів та / або більше одиниць. У наведеному нижче коді випробовується кілька різних конфігурацій нейронної мережі та повідомляється про точність перевірки для кожної з них:

```
import turicreate as tc
```

```
data = tc.load_sframe('./ESC-10')
# Calculate the deep features just once.
data['deep_features'] = tc.sound_classifier.get_deep_features(data['audio'])
# Try several different neural network configurations
models = []
network_configurations = ([100, 100], [100], [1000, 1000], [100, 100, 100])
for cur_hyper_parameter in network_configurations:
    cur_model = tc.sound_classifier.create(data, target='category', custom_layer_sizes=cur_hyper_parameter, feature='deep_features')
    models.append(cur_model)
# Report results
for m, p in zip(models, network_configurations):
    print("{}, {}".format(p, m.validation_accuracy))
```

2.3. Вибір моделей для класифікації радіотипів

- Dbof або глибокий утримувач каркасної моделі - це згорткова нейронна мережа. Основною ідеєю є проектування двох шарів у згортковій частині. У першому шарі, шарі проекції, ваги все ще застосовуються до об’єктів, хоча всі вибрані об’єкти мають однаковий параметр. (Рис. 1)
- Модель LSTM як особливий вид RNN, здатний вивчати довгострокові залежності.
- Logistic Regression, яка приймає вхідні дані, передає їх через функцію, яка називається сигмовидною функцією, а потім повертає вихід імовірності від 0 до 1. Ця сигмоїдна функція відповідає за класифікацію вхідних даних. Як писано [24], ця мережа навчається за допомогою Stocastic Gradient Descent (SGD) з логістичними втратами для логістичного шару та перехресними ентропійними втратами для рівня softmax. Градієнти зворотного розповсюдження з верхнього шару тренують вагові вектори проекційного шару дискримінаційно, щоб забезпечити потужне представлення вхідного пакета ознак.

У цьому проекті модель LSTM була побудована за аналогічним підходом до [23]. Забезпечує найкращу ефективність набору перевірок, 2

складених шари LSTM та 10 ітерацій розгортання. Для LstmModel ми змінили базовий рівень навчання на 0,001 відповідно до документації. Ми також змінили значення lstm_cells на 256, оскільки в нього закінчилася оперативна пам'ять. Всі моделі працюють під коробкою. Основна інформація про роботу моделей описана в документації [24].

Спочатку потрібно буде вибрати деяке програмне забезпечення для роботи з нейронними мережами. Першим відповідним рішенням, яке було обране в розділі 1, був аналіз звуку через Python.

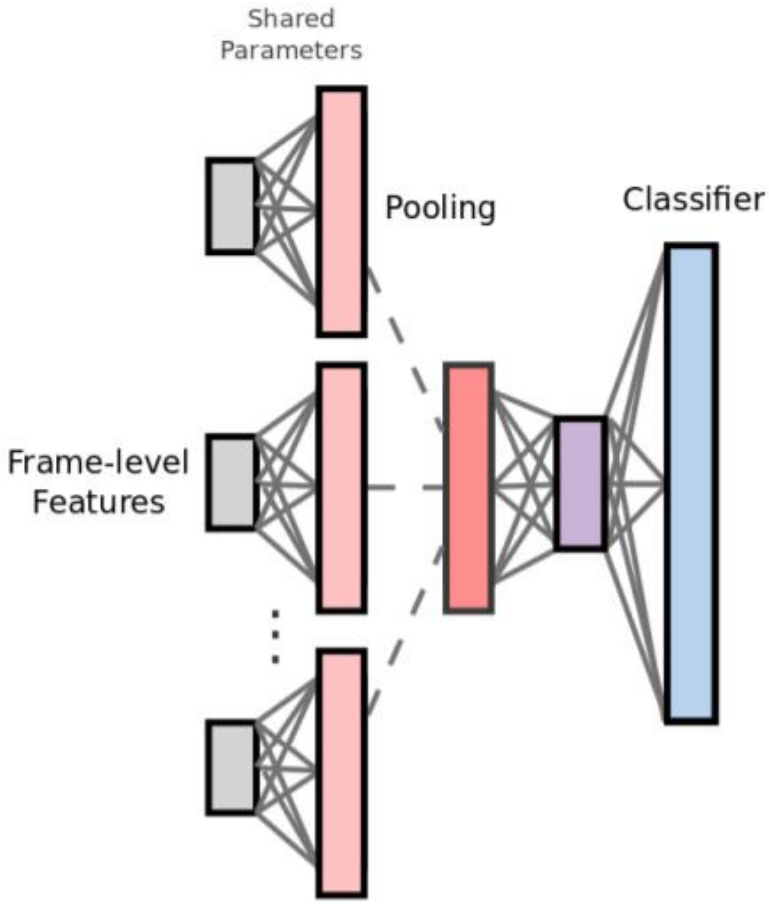


Рис. 2.22. Мережева архітектура підходу DBoF. Вхідні функції спочатку подаються на рівень проекції зі спільними параметрами для всіх функцій. Далі слідує шар об’єднання, який перетворює розріджені коди на рівні об’єкта в подання рівня аудіо. Кілька прихованих шарів та рівень класифікації забезпечують остаточні прогнози рівня звуку.

Основною проблемою машинного навчання є наявність хорошого навчального набору даних. Існує багато наборів даних для розпізнавання мови та класифікації музики, але не так багато для класифікації аудіо. Тому датасет для класифікації довгих аудіо довелось створювати самому.

Також після певного тестування на практиці було виявлено такі проблеми:

- pyAudioAnalysis недостатньо гнучкий. Тут не можливо застосувати різноманітні параметри нейромережі, і деякі з них обчислюються в runtime. Наприклад кількість навчальних експериментів на основі кількості зразків, і це не можливо змінити.
- Набір даних має лише 10 класів, і всі вони не є повністю відповідними.

Наступним рішенням, яке було знайдено, був Google AudioSet. Він заснований на позначених відео сегментах YouTube і може бути завантажений у двох форматах:

1. Файли CSV, що описують для кожного сегмента ідентифікатор відео YouTube, час початку, час закінчення та одну або кілька міток.

2. Витягнуті аудіо функції, які зберігаються як файли TensorFlow Record.

Ці функції сумісні з моделями YouTube-8M. Також це рішення пропонує модель TensorFlow VGGish як функцію витяжки. Він охоплював значну частину наших вимог, але не мав усіх необхіднх розміток для радіотипів.

Тому було прийнято рішення зкомпонувати з існуючого акту моніторингу і 48-годинного запису двох радіотрансяліцій відповідні два датасети.

Також в дипломній роботі представлено алгоритм класифікації з механізмом уваги для вибору класифікаторів аудіоконтенту. Його повна архітектура побудована на рис. 2.23.

Ключовою частиною алгоритму CAB CNN є блок класифікатора на основі уваги (ACB), що містить n класифікаторів та одиницю уваги (рис. 2.23).

З метою покращення як навчальної, так і статистичної ефективності, оригінальний семпл не передається безпосередньо в блок. Натомість

використовується “очищене” подання a, яке генерується подачею вихідних даних у 1D-CNN стек з наступними шарами MaxPooling (рис. 2.24). Це зроблено для того щоб:

- покращити локальні ознаки (фічі);
- зменшити розмір вхідних даних класифікатора АСВ;
- зберегти взаємозв'язок "один на один" (за віссю часу) між вихідними даними та згенерованими поданнями.

Грубо кажучи, оскільки CNN зберігає просторову інформацію аудіосигналу, він розділяє вихідний аудіосигнал на p інтервали (де значення p залежить від архітектури CNN, шарів MaxPooling та довжини вхідного файлу) і застосовуючи те саме перетворення для вилучення ознак. Ці інтервали можуть мати певне перекриття, що залежить від архітектури рівнів CNN та MaxPooling.

На Рис.3, ми можемо помітити, що перший вузол шару MaxPooling охоплює перші три входи масиву необроблених даних, тоді як другий вузол охоплює з другого по п’ятий. Незважаючи на те, що перекриття може спричинити деяку неясність того, що представляють витягнуті ознаки, вони не мають ніякого впливу на роботу алгоритму.

Оскільки алгоритм вилучення ознак зберігає просторову інформацію, ми можемо перерахувати його вихідні вектори об’єктів у порядку часу:

$$a = [a_1, a_2, \cdots, a_t, \cdots, a_p],$$

де at подання t-го інтервалу часу у вихідних даних. Для кожного at у списку, АСВ виводить середньозважене середнє значення ймовірності вектора, згенерованим класифікаторами.

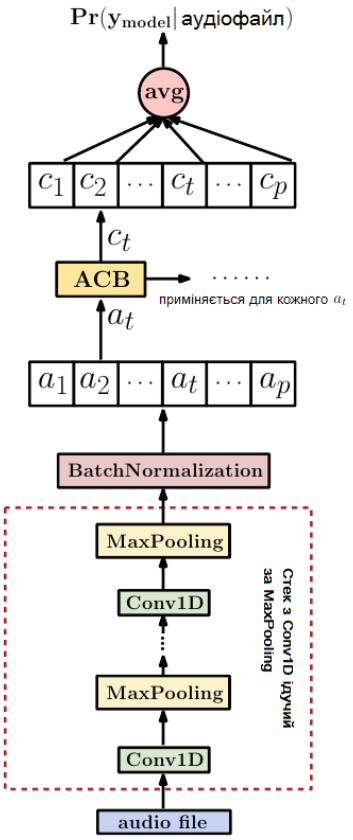


Рис. 2.23. Оригінальний аудіофайл спочатку подається стек шарів CNN та MaxPooling для отримання "очищеного" подання a. Далі застосовується batch-нормалізація для полегшення навчання моделі. На кожен раз інтервалу t, ACB (детальний дизайн представлений на Рис 2.2) обробляє подання кожного at і виводить ймовірність класів ct. Нарешті, імовірність вихідного класу становить незважене середнє ct.

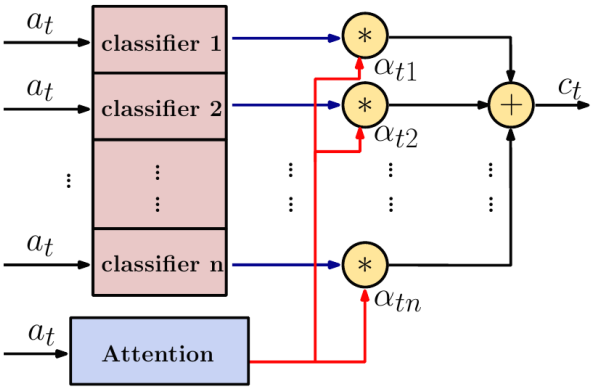


Рис 2.24 (Блок класифікатора на основі уваги (ACB) в момент часу t. Блок складається з масиву в n одиниць класифікації та механізму уваги. Блок отримує подання (at) аудіосигналу в момент часу t, а потім кожен

класифікатор реалізує процес класифікації, а одиниця уваги (attn) видає важливі ваги ati для кожного класифікатора. Тоді контекстний вектор ct дорівнює сумі зважених виходів класифікаторів.)

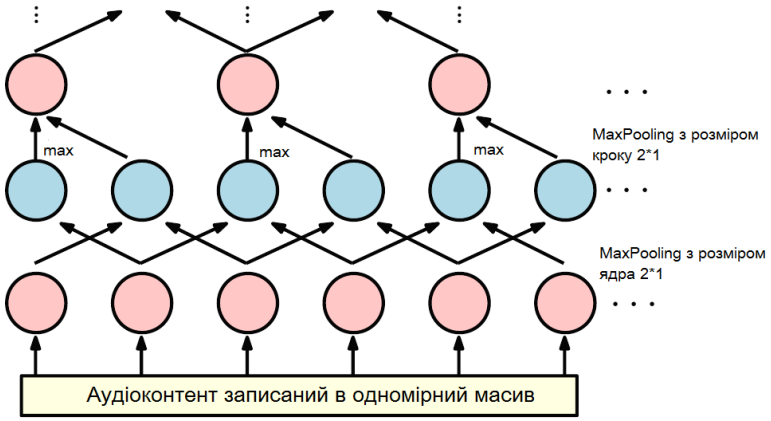


Рис 2.25 (1D-CNN з наступним шаром MaxPooling. Ядро 1D-CNN може витягувати ознаки з вихідних даних з дуже високою ефективністю параметрів. Шар MaxPooling може потім переганяти дані, подані нижніми шарами, і тим самим зменшує вихідний обсяг. В даному алгоритмі, застосовано стек такої структури для рекурсивного фільтрування інформації. У цьому прикладі, ознаки, корисні для подальшого завдання класифікації, зберігаються, тоді як недоречна інформація видаляється.)

Більш детально, припустимо, що ACB має n класифікаторів, і існує m класів для розрізнення. Потім, при отриманні at, класифікатор i створює вектор імовірності $c_{ti} \in R_m$ для i = 1, 2, … n, і механізм уваги генерує ваги важливості:

$$a_t = [a_{t1}, a_{t2}, \cdots, a_{tn}].$$

Після цього блок виводить

$$c_t = \sum_{i=1}^{n} \alpha_{ti}\, c_{ti}\ .$$

Конструкцію важливих ваг можна пояснити двома способами. По-перше, вони показують, наскільки важливим є класифікатор при класифікації звуку в момент часу t. Крім того, вони надають впевненість, що кожен класифікатор дає правильний результат.

Використовуючи цей механізм уваги, ми звільняємо одного класифікатора від необхідності розрізняти великий набір ознак, пов’язаних із завданням класифікації аудіоконтенту. Натомість одному класифікатору потрібно зосередитись лише на певному типі ознак.

Більш детально, припустимо, що нам потрібно класифікувати деякі аудіозаписи за типом аудіоконтенту (реклама, пісня, ефір). Люда може використовувати наступну стратегію:

1. визначити, чи присутні деякі певні особливості (сигнал, швидка надиктовка для реклами; мелодичність, повторюваність для пісні; діалогове мовлення, розмірений темп для ефіру);

2. якщо функція присутня і виключно належить до певного типу аудіоконтенту, тоді ми можемо сказати, що аудіо є відповідним типом.

Даний алгоритм працює подібним чином. Зокрема, блок уваги визначає, які ознаки присутні, і дозволяє відповідним класифікаторам робити класифікацію. Це робиться шляхом присвоєння тим класифікаторам високоважливих ваг. Таким чином, кожному класифікатору потрібно зосередитись лише на невеликій підмножині всіх доступних ознак, які, таким чином, можна легко навчити модель та мають високу точність прогнозування.

Віддаючи на обробку at до класифікатора ACB ми отримуємо список ct представляючих передбачені ймовірності класів у момент часу t. Вкінці, просто беремо незважене середнє за всі ct робимо фінальне передбачення:

$$\Pr(y_{model}) = \sum_{t \geq 1} c_t \, .$$

При чому незважене середнє передбачає пріоритет: ознаки, що відповідають певному класу, мають однакову ймовірність бути активними на кожному інтервалі часу.

2.4. Деталізація датасету

Датасет для радіотипів був представлений у двох форматах. Дескрипторна частина була надана у вигляді xlsx файлу як акт моніторингу

радіо рис. 2.4, друга частина – це відповідно файли .mp3 тривалістю 24 години кожен, які були переведені у необхідний wav формат, який за допомогою веб-сервісу був розподілений на аудіосегменти згідно до акту моніторингу та опрацьований на трьох вище зазначених нейромоделях: lstm, dbof, logistic з пакету Youtube-8M.

Датасети представляють онлайн записи двох регіональних радіо, розмічені посекундно рис. 2.26. У наступному розділі описано обробку звукової частини датасету за допомогою інструментів matlab та бібліотеки python audio.

Мітки до даного датасету згідно з пунктом 2 першого розділу були поділені на три категорії та записані в файл cvs (рис. 2.27).

Згідно до міток було переписано акт моніторингу радіотрансляцій (рис. 2.28)

	A	B
1	index,mid,display_name	
2	0,/m/09x0r,"Broadcast speech"	//передача
3	1,/m/05zppz,"Musical material"	//пісенний матеріал
4	2,/m/0ytgt,"Advertising"	//реклама
5		
6	1,/m/02zsn,"Formal"	//ефір (одна людина)
7	2,/m/01h8n0,"Podcast"	//підкаст-діалог (дві людини)
8	3,/m/05p4gz,"Conversation"	//група людей
9		
10	0,/m/06bz3,"Ukrainian"	//українська
11	1,/m/07hvw1,"Other"	//іноземна

Рис. 2.26. Розмітка радіотипів для нейромережі

АКТ здійснення планового моніторингу програм та передач Дочірнього підприємства "Телерадіокомпанія "Регіон-Плюс", м. Краматорськ, 107,8 МГц

02.02.2017

Дата час	Назва	Тип	Мо	Власний продук	Вітчизняний продук	Примітка	Довжи
0:00:00	Оформлення ефіру	Інше	рос	Так	Так	Русское радио Украина	0:00:05
0:00:05	Відбивка передачі	Розваж	рос	Так	Так	Ночной дозор на Русском радио	0:00:10
0:00:15	Пісенний матеріал	Розваж	укр	Так	Так	Державний гімн України	0:01:38
0:01:53	Оформлення ефіру	Інше	рос	Так	Так	Зима на Русском радио Украина	0:00:07
0:02:00	Пісенний матеріал	Розваж	рос	Так	Так	Спектакль окончен гаснет свет	0:03:50
0:05:50	Оформлення ефіру	Інше	рос	Так	Так	Зима на Русском радио	0:00:14
0:06:04	Пісенний матеріал	Розваж	рос	Так	Так	Она живет мечтая она просто святая	0:03:37
0:09:41	Оформлення ефіру	Інше	рос	Так	Так	Зима на Русском радио	0:00:10
0:09:51	Пісенний матеріал	Розваж	рос	Так	Так	Слайд в темное и ничего	0:03:40
0:13:31	Оформлення ефіру	Інше	рос	Так	Так	Зима Русское радио Украина	0:00:14
0:13:45	Пісенний матеріал	Розваж	рос	Так	Так	Она сумасшедшая но она моя	0:02:54
0:16:39	Оформлення ефіру	Інше	рос	Так	Так	Русское радио Украина	0:00:10
0:16:49	Пісенний матеріал	Розваж	укр	Так	Так	Така малесенька іскорка	0:03:12
0:20:01	Оформлення ефіру	Інше	рос	Так	Так	Зима Все будет хорошо	0:00:12
0:20:13	Пісенний матеріал	Розваж	укр	Так	Так	На Мері я наштовхнувся у клубі	0:03:23
0:23:36	Оформлення ефіру	Інше	рос	Так	Так	Реклама	0:00:07
0:23:43	Реклама	Інше	укр	Ні	Так	Блок реклами Русского радио	0:00:22
0:24:05	Оформлення ефіру	Інше	рос	Так	Так	Реклама на Русском радио	0:00:09
0:24:14	Пісенний матеріал	Розваж	рос	Так	Так	И мы с друзьями зажигаем в баре	0:01:33
0:25:47	Позивний	Інше	рос	Так	Так	Русское радио	0:00:03
0:25:50	Пісенний матеріал	Розваж	рос	Так	Так	И до рассвета пусть горит любовь	0:01:36
0:27:26	Оформлення ефіру	Інше	рос	Так	Так	Все будет хорошо Русское радио Зима	0:00:11
0:27:37	Пісенний матеріал	Розваж	рос	Так	Так	Только не плачь сердце не прячь	0:03:15
0:30:52	Оформлення ефіру	Інше	рос	Так	Так	Русское радио Украина	0:00:09

Рис. 2.27. Акт моніторингу радіо

	A	B	C	D	E
148	3:33:28	Musical	Formal	Ukrainian	0:00:48
149	3:34:16	Broadcast	Conversation	Other	0:00:12
150	3:34:28	Musical	Formal	Other	0:02:50
151	3:37:18	Broadcast	Conversation	Other	0:00:12
152	3:37:30	Advrtising	Formal	Other	0:03:41
153	3:41:11	Broadcast	Conversation	Ukrainian	0:00:08
154	3:41:19	Advrtising	Formal	Other	0:03:21
155	3:44:40	Broadcast	Conversation	Other	0:00:08
156	3:44:48	Musical	Formal	Other	0:02:42
157	3:47:30	Broadcast	Conversation	Other	0:00:16
158	3:47:46	Musical	Formal	Ukrainian	0:02:52
159	3:50:38	Broadcast	Conversation	Other	0:00:09
160	3:50:47	Musical	Formal	Other	0:03:04
161	3:53:51	Broadcast	Conversation	Other	0:00:12
162	3:54:03	Musical	Formal	Ukrainian	0:02:38
163	3:56:41	Broadcast	Conversation	Other	0:00:10
164	3:56:51	Musical	Formal	Other	0:03:29
165	4:00:20	Broadcast	Conversation	Other	0:00:07
166	4:00:27	Advrtising	Formal	Other	0:00:05
167	4:00:32	Broadcast	Conversation	Other	0:00:09
168	4:00:41	Musical	Formal	Other	0:03:06
169	4:03:47	Advrtising	Conversation	Other	0:00:13
170	4:04:00	Musical	Formal	Ukrainian	0:03:11
171	4:07:11	Advrtising	Conversation	Other	0:00:09
172	4:07:20	Musical	Formal	Other	0:03:40
173	4:11:00	Передача	Просв	Ukrainian	0:02:04
174	4:13:04	Broadcast	Conversation	Other	0:00:14
175	4:13:18	Musical	Formal	Ukrainian	0:03:16
176	4:16:34	Broadcast	Conversation	Other	0:00:09
177	4:16:43	Musical	Conversation	Other	0:03:10
178	4:19:53	Broadcast	Conversation	Other	0:00:14

Рис. 2.28. Приклад перетвореного акту моніторингу

Висновки до розділу

У даному розділі було обрано моделі для класифікації радіотипів як довгого аудіоконтенту. Було описано їх архітектуру, особливості та застосування по відношенню до усього процесу: від вилучення та відбору ознак, або аудіофіч до їх класифікації і роботі з довгими аудіоданими. Було оглянуто датасет, а саме – визначено його складові і на основі першого розділу опрацьовано та розмічено.

Через доступність та велику базу практичних застосувань і тестувань було обрано програмний інтерфейс Youtube-8M та наступні три моделі для подальшої роботи: LSTM, Dbof та frame-level logistic model. Надано їх архітектурні складові. Для подільших навчань у галузі класифікації аудіо було розглянуто модель CAB-CNN яка використовує класифікатор уваги, ймовірно вона зможе показати кращі результати, якщо кафедра буде у цьому зацікавлена.

Для вилучення аудіофіч і їх вибір з них 13-ти глибинних (основних) для подальшої передачі класифікатору обрано бібліотеку VGGish, яка реалізована на python і має велику кількість переваг порівняно з імплементацією DTaoo та yamnet.

Middleware для роботи зі звуком обрано pyAudioAnalysis через гнучкість та реалізацію в необхідному стеку.

РОЗДІЛ 3. РОЗРОБКА АРХІТЕКТУРИ ДЕМОНСТРАЦІЙНОГО СЕРЕДОВИЩА ТА РЕАЛІЗАЦІЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ

3.1. Програмна модель

Інтелектуальна система повністю написана на мові програмування Python. Існує два способи взаємодії через PyAudio: через мікрофон або зразок чи набір зразків аудіоматеріалів у форматі wav 16 бітного розміру.

Для цього треба встановити певні залежності для роботи з ІС. 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-e

Feature_extractor.py інтерфейсу Youtube-8M:

```
import os
import sys
import tarfile
import numpy
from six.moves import urllib
import tensorflow as tf
INCEPTION_TF_GRAPH = 'http://download.tensorflow.org/models/image/image
net/inception-2015-12-05.tgz'
YT8M_PCA_MAT = 'http://data.yt8m.org/yt8m_pca.tgz'
MODEL_DIR = os.path.join(os.getenv('HOME'), 'yt8m')
class YouTube8MFeatureExtractor(object):
  """Extracts YouTube8M features for RGB frames.
  First time constructing this class will create directory `yt8m` insid
e your
  home directory, and will download inception model (85 MB) and YouTube
8M PCA
  matrix (15 MB). If you want to use another directory, then pass it to
argument
  `model_dir` of constructor.
```

```
        If the model_dir exist and contains the necessary files, then files w
ill be
        re-used without download.
        Usage Example:
            from PIL import Image
            import numpy
            # Instantiate extractor. Slow if called first time on your machin
e, as it
            # needs to download 100 MB.
            extractor = YouTube8MFeatureExtractor()
            image_file = os.path.join(extractor._model_dir, 'cropped_panda.jp
g')
            im = numpy.array(Image.open(image_file))
            features = extractor.extract_rgb_frame_features(im)
        ** Note: OpenCV reverses the order of channels (i.e. orders channels
as BGR
        instead of RGB). If you are using OpenCV, then you must do:
                                          im      =      im[:,      :,      ::-
1]  # Reverses order on last (i.e. channel) dimension.
        then call `extractor.extract_rgb_frame_features(im)`
        """
        def __init__(self, model_dir=MODEL_DIR):
          # Create MODEL_DIR if not created.
          self._model_dir = model_dir
          if not os.path.exists(model_dir):
            os.makedirs(model_dir)

          # Load PCA Matrix.
          download_path = self._maybe_download(YT8M_PCA_MAT)
          pca_mean = os.path.join(self._model_dir, 'mean.npy')
          if not os.path.exists(pca_mean):
            tarfile.open(download_path, 'r:gz').extractall(model_dir)
          self._load_pca()
          # Load Inception Network
          download_path = self._maybe_download(INCEPTION_TF_GRAPH)
          inception_proto_file = os.path.join(self._model_dir,
                                              'classify_image_graph_def.pb')
          if not os.path.exists(inception_proto_file):
            tarfile.open(download_path, 'r:gz').extractall(model_dir)
          self._load_inception(inception_proto_file)
        def extract_rgb_frame_features(self, frame_rgb, apply_pca=True):
          """Applies the YouTube8M feature extraction over an RGB frame.
```

```
        This passes `frame_rgb` to inception3 model, extracting hidden layer
        activations and passing it to the YouTube8M PCA transformation.

        Args:
          frame_rgb: numpy array of uint8 with shape (height, width, channe
ls) where
            channels must be 3 (RGB), and height and weight can be anything
, as the
            inception model will resize.
          apply_pca: If not set, PCA transformation will be skipped.
        Returns:
                      Output   of   inception   from   `frame_rgb`   (2048-
D) and optionally passed into
          YouTube8M PCA transformation (1024-D).
        """
        assert len(frame_rgb.shape) == 3
        assert frame_rgb.shape[2] == 3  # 3 channels (R, G, B)
        with self._inception_graph.as_default():
          if apply_pca:
            frame_features = self.session.run(
              'pca_final_feature:0', feed_dict={'DecodeJpeg:0': frame_rgb})
          else:
            frame_features = self.session.run(
               'pool_3/_reshape:0', feed_dict={'DecodeJpeg:0': frame_rgb})
            frame_features = frame_features[0]
        return frame_features
```

3.2. Архітектура проекту

Під час навчання вхідними даними для наших моделей фіксований шар 224 × 224. Єдиною попередньою обробкою, яку ми робимо, є віднімання середнього значення коефіцієнта, обчисленого на навчальному наборі, з кожної функції. Функція проходить через стос згорткових (конверсійних) шарів, де ми використовуємо фільтри з дуже малим сприйнятливим полем: 3 × 3 (що є найменшим розміром для охоплення поняття ліворуч / праворуч, вгору / вниз, по центру). В одній з конфігурацій ми також використовуємо фільтри згортки 1 × 1, які можна розглядати як лінійне перетворення вхідних каналів (за яким слід нелінійність). Швидкість згортки зафіксована на 1 ознаці;

просторове заповнення конв. введення шару є таким, що просторова роздільна здатність зберігається після згортки, тобто відступ - 1 функція для 3 × 3 конв. шари. Просторове об'єднання здійснюється за допомогою п'яти шарів максимального об'єднання, які йдуть за деякими конв. шари (не всі шари конв. супроводжуються об'єднанням макс.). Макс-пул виконується через вікно 2 × 2 з кроком 2.

За стосом згорткових шарів (який має різну глибину в різних архітектурах) слідують три повністю підключених (FC) шари: перші два мають по 4096 каналів, третій виконує класифікацію ILSVRC по 1000 напрямках і, таким чином, містить 1000 каналів (один для кожного класу).

Кінцевим шаром є шар soft-max. Конфігурація повністю підключених шарів однакова у всіх мережах. Всі приховані шари оснащені нелінійністю ректифікації (ReLU (Krizhevsky et al., 2012)). Ми зазначаємо, що жодна з наших мереж (за винятком однієї) не містить нормалізації локальної нормалізації відповіді (LRN) (Крижевський та ін., 2012): як це буде показано в розділі. 4, така нормалізація не покращує продуктивність набору даних ILSVRC, але призводить до збільшення споживання пам'яті та часу обчислень. Там, де це можливо, параметри рівня LRN - це параметри (Krizhevsky et al., 2012).

Наступним завданням було з’ясувати, як працює інтерфейс YouTube-8M. Він призначений для роботи з відео, але, на щастя, може працювати і з аудіо. Ця бібліотека є досить гнучкою, але вона має жорстко закодовану кількість моделей для класифікації.

YouTube-8M може працювати з даними двох типів: агреговані фічі та фічі для кадру. Google AudioSet може надавати дані як аудіофічі, як було вже зазначено раніше.

3.3. pyAudioAnalysis як middleware суб‘єктів процесу класифікації

PyAudioAnalysis можна використовувати для вилучення звукових функцій, підготовки та застосування аудіо класифікаторів, сегментування

аудіопотоку за допомогою контрольованих або неконтрольованих методологій.

Бібліотека написана на Python, яка є мовою програмування високого рівня, яка викликає все більший інтерес, особливо в науковому та науковому середовищі протягом останніх кількох років. Python досить привабливий для програм обчислювального аналізу сигналів, головним чином завдяки тому, що він забезпечує оптимальний баланс функцій програмування високого рівня та низького рівня: менше кодування без важливого обчислювального навантаження.

Початкова проблема високих обчислювальних вимог частково вирішується застосуванням процедур оптимізації на об'єктах вищого рівня. Крім того, у порівнянні з Matlab або іншими подібними рішеннями, Python безкоштовний і може вести до автономних додатків без потреби у величезних попередньо встановлених двійкових файлах та віртуальних середовищах. Ще однією великою перевагою Python є те, що існує величезна кількість бібліотек, що забезпечують функціональні можливості, пов'язані з науковим програмуванням. У таблиці 1 представлений список відповідних бібліотек аудіоаналізу, реалізованих у Python, C / C ++ та Matlab. Рис. 1 ілюструє концептуальну схему бібліотеки, тоді як на рис. 3.1 показані деякі знімки екрану з використання бібліотеки. pyAudioAnalysis реалізує такі функціональні можливості:

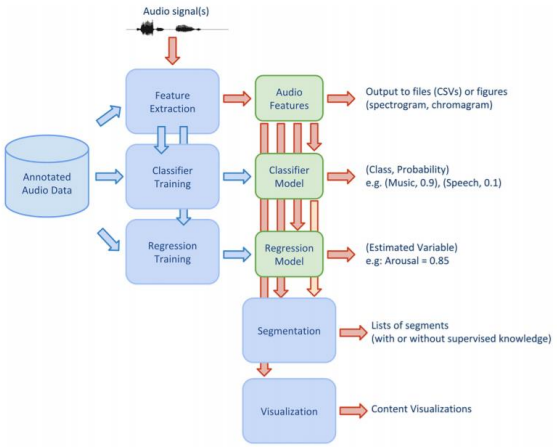


Рис. 3.1. Спектр звуковий сигналу на осі часу

Вилучення функцій: у бібліотеці реалізовано кілька звукових функцій як з часової, так і з частотної області.

Класифікація: контрольовані знання (тобто анотовані записи) використовуються для навчання класифікаторів. Процедура перехресної перевірки також реалізована для того, щоб оцінити оптимальний параметр класифікатора (наприклад, параметр витрат у машинах підтримки вектора або кількість найближчих сусідів, що використовуються в класифікаторі kNN). Результатом роботи цієї функції є модель класифікатора, яку можна зберігати у файлі. Крім того, обгортки, які класифікують невідомий звуковий файл (або набір аудіофайлів), також надаються в контексті рис. 3.2.

Регресія: моделі, які відображають звукові функції до реальних змінних, також можна навчити в контрольованому контексті. Знову ж таки, застосовується перехресна перевірка для оцінки найкращих параметрів моделей регресії.

Сегментація: у бібліотеці реалізовані наступні контрольовані або неконтрольовані завдання сегментації: сегментація та класифікація виправленого розміру, видалення тиші, діаризація динаміків та мініатюра звуку. За потреби, навчені моделі використовуються для класифікації аудіо сегментів за попередньо визначеними класами або для оцінки однієї або декількох вивчених змінних (регресія).

Візуалізація: для даної колекції аудіозаписів pyAudioAnalysis може використовуватися для отримання візуалізації взаємозв’язків вмісту між цими записами.

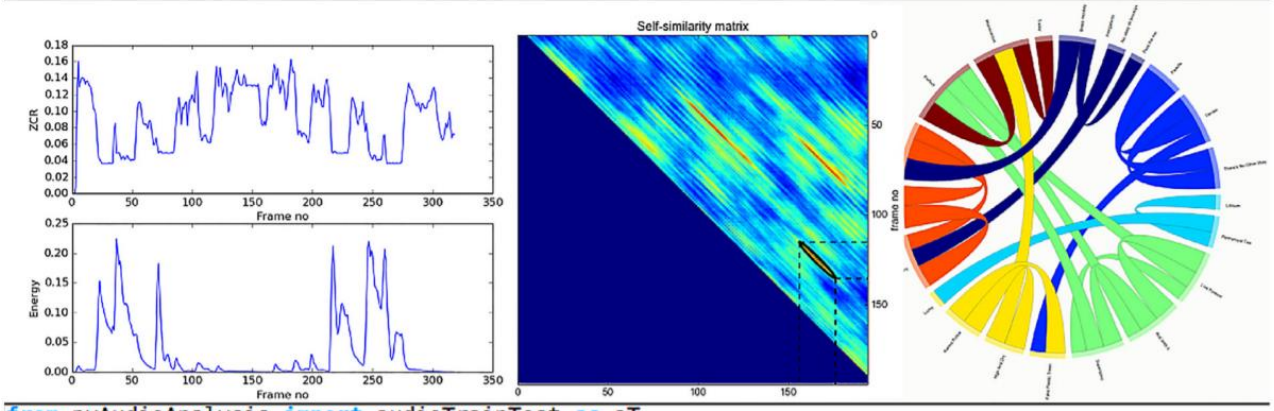
Вилучення загальних характеристик та концептуальні компоненти машинного навчання пов’язані між собою, щоб сформувати повні рішення щодо класифікації та сегментації звуку.

Для вирішення широко використовуваних завдань аудіоаналізу застосовуються як найсучасніші, так і базові методи. Також пропонуються попередньо навчені моделі для деяких контрольованих завдань (наприклад,

класифікація мовлення-музики, класифікація музичного жанру та виявлення подій фільму).

Усі передбачені функціональні можливості написані з використанням чіткого і простого коду, щоб концептуальні алгоритмічні кроки могли бути чітко представлені в контексті навчального процесу.

```
from pyAudioAnalysis import audioBasicIO
from pyAudioAnalysis import audioFeatureExtraction
import matplotlib.pyplot as plt
[Fs, x] = audioBasicIO.readAudioFile("data/doremi.wav");
F = audioFeatureExtraction.stFeatureExtraction(x, Fs, 0.050*Fs, 0.025*Fs);
plt.subplot(2,1,1); plt.plot(F[0,:]); plt.xlabel('Frame no'); plt.ylabel('ZCR');
plt.subplot(2,1,2); plt.plot(F[1,:]); plt.xlabel('Frame no'); plt.ylabel('Energy'); plt.show()
```




```
from pyAudioAnalysis import audioTrainTest as aT
aT.featureAndTrain(["Silence/","SpeechMale/","SpeechFemale/","ObjectsOther/","Music/"],1.0, 1.0,
 aT.shortTermWindow, aT.shortTermStep, "svm", "svm5Classes")
```

Рис. 3.2. Класифікація звукового файлу

3.4. Візуалізація процесу трансформації звуку на практиці

Для того щоб отримати звук з мікрофону було використано PyAudio. Він надає певний інтерфейс для зручної роботи зі звуком, як було зазначено раніше, ми будемо використовувати модель TensorFlow VGGish як функцію витягування аудіофіч. Ось коротке пояснення процесу трансформації:

Для візуалізації був використаний приклад “собачого лаю” із набору даних UrbanSound.

Перший крок – перепрофілювання звуку до 16 кГц моно:

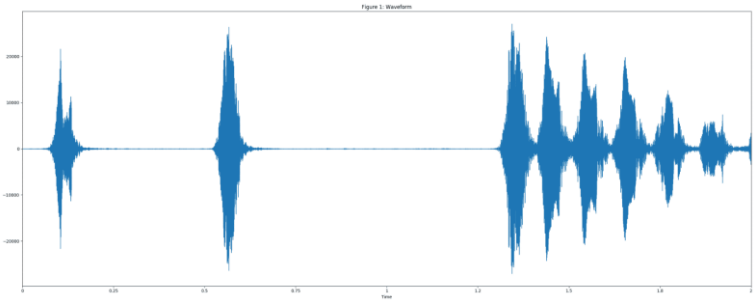


Рис. 3.3. Класифікація невідомого файлу

Другий – обчислення спектрограми, використовуючи величини короткочасного перетворення Фур'є з розміром вікна 25 мс, переходом вікна 10 мс та періодичним вікном Ханна. Третій – обчислення спектрограми розрідженого звуку, відобразивши спектрограму на 4 мел-коефіцієнтній спектрограмі часу:

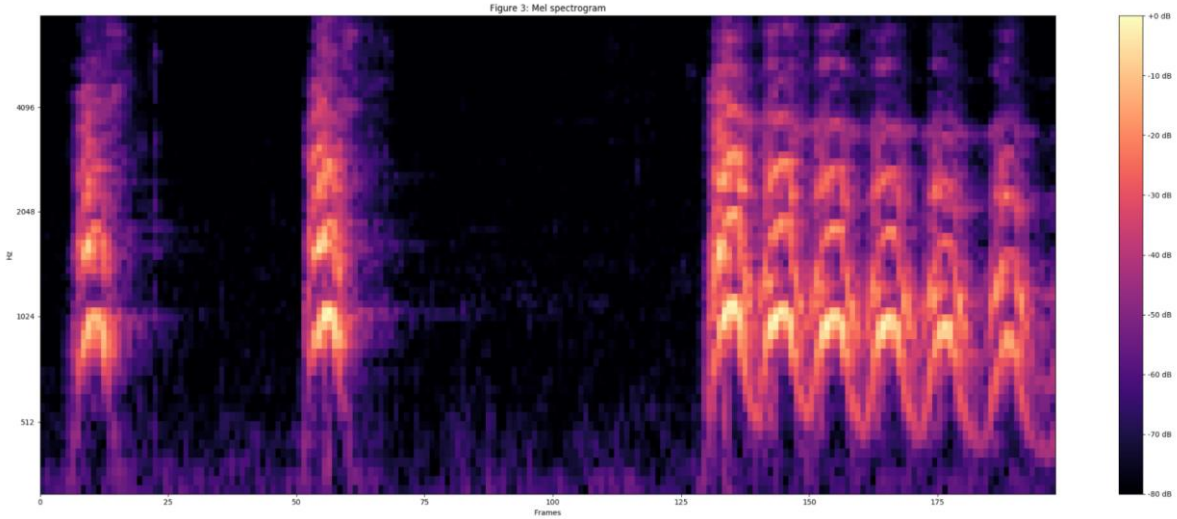


Рис. 3.4. Класифікація невідомого файлу

Далі – обчислення стабілізованої логарифмічної спектрограми, застосовуючи log (мел-спектр + 0,01), де використовується зсув, щоб уникнути нульового логарифму:

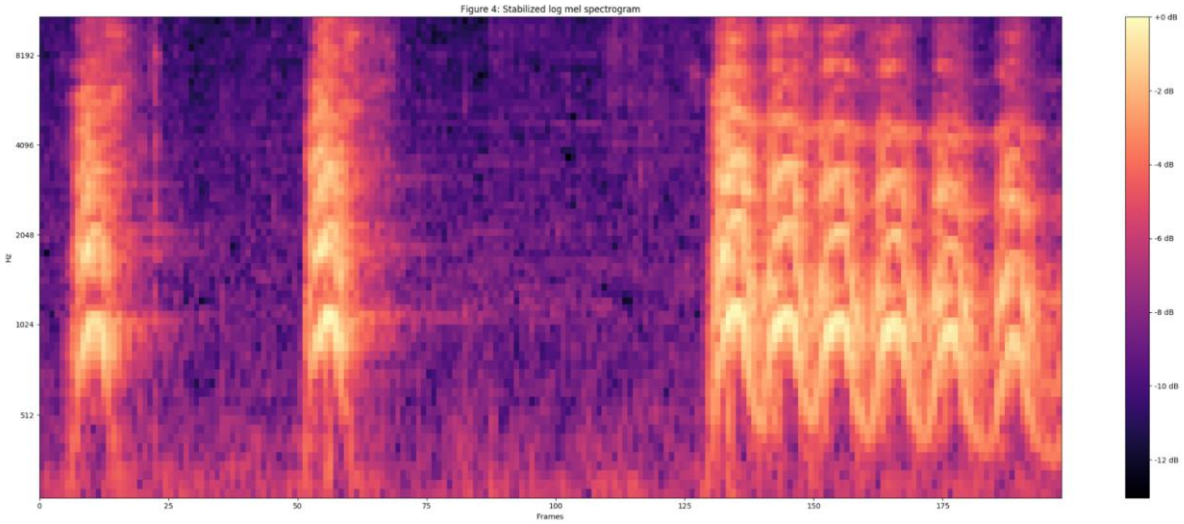


Рис. 3.5. Класифікація невідомого файлу

Потім ці аудіофічі оформляються у неперекриваючі семпли по 0,96 секунди, де кожен семпл охоплює 64 діапазони розмежування та 96 кадрів по 10 мс кожен.

Потім ці приклади подаються у модель VGGish для вилучення аудіофіч.

3.5. Опис інтерфейсу захоплення аудіопотоку

Для реалізації захоплення звуку python capture.py запускає процес, який постійно буде захоплювати дані з вашого мікрофона. Він буде передавати дані для класифікації кожні 5-7 секунд (за замовчуванням). Ви побачите результати так само, як в попередньому прикладі. Ви можете запустити його з параметром --save_path = / path_to_samples_dir /, в цьому випадку всі захоплені дані будуть збережені у вказаній папці в форматі .WAV. Ця функція корисна, якщо ви хочете спробувати різні моделі з тими ж зразками. Використовуйте параметр --help, щоб отримати додаткову інформацію.

```
import os
import time
import json
import threading
import logging.config
import datetime
import numpy as np
from collections import deque
```

```
from scipy.io import wavfile
from devicehive_webconfig import Server, Handler
from audio.captor import Captor
from audio.processor import WavProcessor, format_predictions
from web.routes import routes
from log_config import LOGGING
logging.config.dictConfig(LOGGING)
logger = logging.getLogger('audio_analysis.daemon')

class DeviceHiveHandler(Handler):
    _device = None

    def handle_connect(self):
        self._device = self.api.put_device(self._device_id)
        super(DeviceHiveHandler, self).handle_connect()

    def send(self, data):
        if isinstance(data, str):
            notification = data
        else:
            try:
                notification = json.dumps(data)
            except TypeError:
                notification = str(data)
        self._device.send_notification(notification)

class Daemon(Server):
    _process_thread = None
    _process_buf = None
    _ask_data_event = None
    _shutdown_event = None
    _captor = None
    _sample_rate = 16000
    _processor_sleep_time = 0.01

    events_queue = None

    def __init__(self, *args, **kwargs):
        min_time = kwargs.pop('min_capture_time', 5)
        max_time = kwargs.pop('max_capture_time', 5)
        self._save_path = kwargs.pop('save_path', None)

        super(Daemon, self).__init__(*args, **kwargs)
```

```
                self.events_queue = deque(maxlen=10)
                self._ask_data_event = threading.Event()
                self._shutdown_event = threading.Event()
                 self._process_thread = threading.Thread(target=self._process_lo
op,
                                                        name='processor')
                self._process_thread.setDaemon(True)

               self._captor = Captor(min_time, max_time, self._ask_data_event,
                                     self._process, self._shutdown_event)

            def _start_capture(self):
                logger.info('Start captor')
                self._captor.start()
            def _start_process(self):
                logger.info('Start processor loop')
                self._process_thread.start()
            def _process(self, data):
                self._process_buf = np.frombuffer(data, dtype=np.int16)
            def _on_startup(self):
                self._start_process()
                self._start_capture()
            def _on_shutdown(self):
                self._shutdown_event.set()
            def _process_loop(self):
                with WavProcessor() as proc:
                    self._ask_data_event.set()
                    while self.is_running:
                        if self._process_buf is None:
                            # Waiting for data to process
                            time.sleep(self._processor_sleep_time)
                            continue
                        self._ask_data_event.clear()
                        if self._save_path:
                            f_path = os.path.join(
                                 self._save_path, 'record_{:.0f}.wav'.format(tim
e.time())
                            )
                             wavfile.write(f_path, self._sample_rate, self._proc
ess_buf)
                            logger.info('"{}" saved'.format(f_path))
                        logger.info('Start processing')
```

```
                    predictions = proc.get_predictions(
                        self._sample_rate, self._process_buf)
                    formatted = format_predictions(predictions)
                    logger.info('Predictions: {}'.format(formatted))
                     self.events_queue.append((datetime.datetime.now(), form
atted))
                   self._send_dh(predictions)
                    logger.info('Stop processing')
                    self._process_buf = None
                    self._ask_data_event.set()
        def _send_dh(self, data):
            if not self.dh_status.connected:
                logger.error('Devicehive is not connected')
                return
            self.deviceHive.handler.send(data)
    if __name__ == '__main__':
        server = Daemon(DeviceHiveHandler, routes=routes)
        server.start()
```

3.6. Результати експериментального дослідження

Для машинного навчання GPU є більш підходящим вибором, ніж CPU. Ви можете знайти більше інформації про це серед списку джерел [27, 28]. Тож цей пункт можна пропустити в дипломній роботі і перейти безпосередньо до нашого налаштування. Для експериментів було використано ПК з одним NVIDIA GTX 970 2 Гб.

У випадку з довгими аудіоданними час тренувань насправді не мав значення. Слід зазначити, що 2-3 годин навчання було достатньо для прийняття початкового рішення щодо обраної моделі та її точності.

Звичайно, вимогами є отримати якомога більшу точність. Але для підготовки більш складної моделі (потенційно кращої точності) потрібно більше оперативної пам’яті (відеопам’яті у випадку з графічним процесором), щоб вмістити її.

Повний список моделей YouTube-8M з описом доступний джерелах [27, 28]. Оскільки наші навчальні дані були у форматі вікон Ханна, слід було використовувати моделі рівня вікон. Google AudioSet надає нам набір даних,

розділений на три частини: збалансований, незбалансований та оцінка. Ви можете отримати більше інформації про них в джерелах [28,29].

Для навчання та оцінки була використана модифікована версія YouTube-8M. Він доступний в списку використаних джерел [40].

Команда для збалансованого навчання виглядає наступним чином:

```
python train.py –
train_data_pattern=/path_to_data/audioset_v1_embeddings/bal_train/*.tfrecord –
num_epochs=100 –learning_rate_decay_examples=400000 –feature_names=audio_embedding –
feature_sizes=128 –frame_features –batch_size=512 –num_classes=527 –
train_dir=/path_to_logs –model=ModelName
```

Для Lstm було зміиено базовий рівень навчання на 0,001, як пропонується в документації. Також змінено значення lstm_cells за замовчуванням на 256, оскільки на практиці не вистачало оперативної пам'яті для більшої кількості.

Було отимано наступні результати навчання надані в таблиці 3.1.

Таблиця. 3.1. Оцінка моделей класифікації

№	.tfrecord size (Mb) 16-bit stream	Model	Studying time, min	Last step evaluation, %	Test accuracy on untrained data,%
1	87.4	logistic	23,5	56.53	54.0
		dbof	136,2	85.40	80.9
		lstm	140,1	88.14	88.1
2	101.2	logistic	36,9	52.98	50.4
		dbof	163,5	80.07	75.9
		lstm	168,0	88.55	87.5

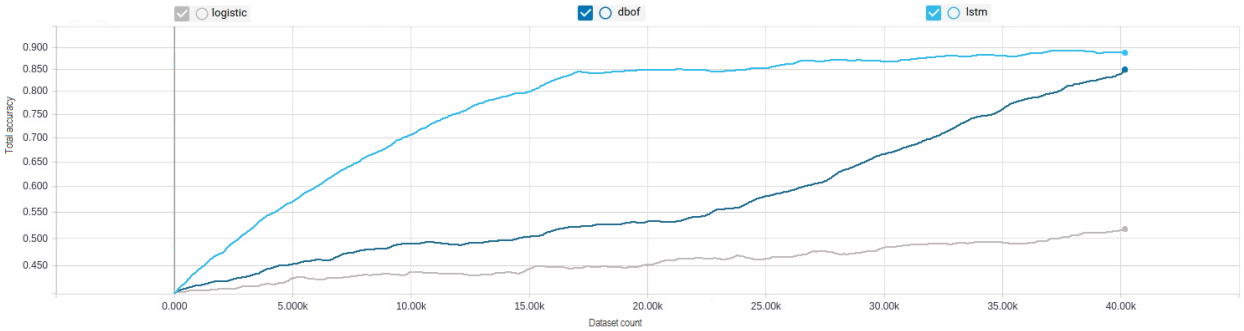


Рис. 3.6 Класифікація невідомого файлу

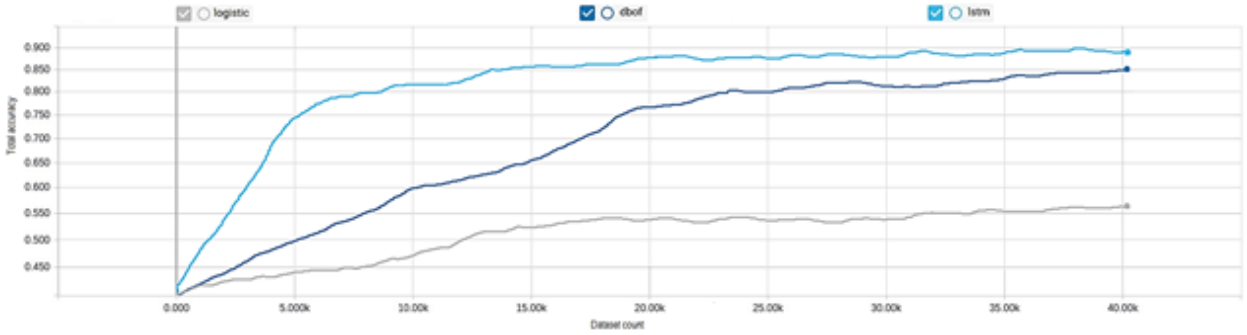


Рис. 3.7 Класифікація невідомого файлу

Як видно з таблиці 3.1, було отримано хороші результати на етапі навчання проте це не означає, що будуть хороші результати при повному оцінюванні:

Також було отримано результати для незбалансованого навчання. У ньому набагато більше зразків, тому було змінено кількість навчальних епох до 10 (принаймні, слід змінити до 5, тому що для навчання потрібен значний час).

3.6.1. Логи навчань

Якщо ви хочете ознайомитися з журналами навчання, ви можете завантажити та витягти train_logs.tar.gz. Потім необхідно запустити tensorboard –logdir / path_to_train_logs / і перейти до http://127.0.0.1:6006

YouTube-8M приймає багато параметрів, і багато з них впливають на тренувальний процес.

Наприклад, ви можете налаштувати швидкість навчання та кількість епох, які сильно змінять навчальний процес. Також є три різні функції для розрахунку втрат та багато інших корисних змінних, які ви можете налаштувати та змінити для покращення результатів.

TensorFlow - це дуже гнучкий інструмент, який може бути корисним у багатьох програмах машинного навчання, таких як розпізнавання зображень та звуку.

Висновки до розділу

У даному розділі було створено ІС та проведено навчання обраних з розділу 2 моделей: DBoF, LSTM, Logistic для класифікації аудіоконтенту з інтерфейсу Youtube-8M. Було протестовано дані моделі, проведено аналіз отриманих моделей. Для вилучення та відбору аудіофіч використано бібліотеку VGGish після чого за допомогою python у векторному вигляді передано до раніше згаданих моделей для їх класифікації. Було розроблено датасет, а саме – розмічено дані та розбито на сегменти, або семпли у форматі wav 16 біт.

Було представлено логи навчань моделей, лістинги інтерфейсу для захоплення аудіо з мікрофону та класифікації семплів з файлового менеджера. Надано команди python для збалансованого та незбалансовано навчання. Описано час та точність на тестовій вибірці, створено відповідні графіки.

РОЗДІЛ 4. МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП-ПРОЄКТУ

4.1. Опис ідеї проєкту

Базовою стратегією стартапу є створення системи класифікації аудіоконтенту, а саме радіотипів для контролю радіоквот в Україні. Однак, створення та ринкове впровадження стартап-проектів відзначається підвищеною мірою ризику, ринково успішними стає лише невелика частка.

У даному розділі наведена розробка проектної пропозиції для проекту, що було створено у рамках магістерської дисертації. Очевидно, що результати даної дисертаційної роботи є важливими з громадської та економічної точок зору. Дана робота є інноваційною та потрібною в державному та регіональному управлінні. У загальному, суттю роботи є оцінка величини іноземних мов, типів радіопередач, реклами, що потім може бути розглянутим при продовженні контракту з радіокомпаніями з метою уникнення корупції.

Таблиця 4.1. Опис ідеї стартап-проєкту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Інтелектуальна система класифікації радіотипів	Використання для моніторингу власної продукції радіокомпаніями	Точність радіокласифікації для актів моніторингу, рішення проблеми корупції на радіо
	Контроль радіоквот	
	Використання державними службовцями	

Таблиця 4.2. Визначення сильних, слабких та нейтральних характеристик ідеї проекту.

№ н/п	Техніко-Економічні характеристики ідеї	(потенційні) товари/концепції конкурентів			W (слабка сторона)	N (нейтральна сторона	S (сильна сторона)
		Мій проект	MPEG-7	Службовець			
1	Вартість налаштування (у грн.)	750	5000	3600	-	-	+
2	Складність технічної реалізації	Відсутня	Наявна	Відсутня	-	+	-

3	Точність роботи нейромоделей	Висока	Низька	Висока	-	-	+
4	Технології, що застосовані	Старі покращені	Нові	Нові	-	+	-
5	Робота додатку для обробки отриманих результатів	Ні, на даний момент мобільний додаток працює лише на Android	Так	Так	+	-	-

Закінчення таблиці 4.2

Отже, можна зробити висновок, що головними перевагами для користувача є вартість та точність роботи. Нейтральними сторонами є складність технічної реалізації та технології, що були застосовані під час розробки. Складність технічної реалізації, а інакшими словами її відсутність одна з основних переваг серед аналогів, оскільки чим точніший результат моніторингу, тим вигіднішим буде продукт на ринку. З технологіями, що використовуються в класифікації ситуація аналогічна, чим новіші технології використовуються, тим дорожче буде кінцева вартість ІС.

Проте, можна помітити, що у нейромережі окрім сильних та нейтральних сторін є й слабка сторона, а саме: навчання на нових даних та обробка результатів класифікації, на даний момент часу ІС потребує уваги саме в цих областях. Оскільки проект тільки виходить на ринок і немає можливості охопити всі області, то проект розроблений як основа для подальшої інтеграції та покращення.

4.2. Технологічний аудит ідеї проєкту

Визначення технологічної здійсненності ідеї проекту передбачає аналіз таких складових (табл. 4.3):

- за якою технологією буде виготовлено продукт згідно ідеї проекту?
- чи існують такі технології, чи їх потрібно розробити/доробити?

- чи доступні такі технології авторам проекту?

В таблиці 4.3 проводиться аудит технології, за допомогою якої можна реалізувати ідею проекту.

Таблиця 4.3. Технологічна здійсненність ідеї проєкту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Інтелектуальна система класифікації радіотипів	Розробка моделей для класифікації довгих аудіоданих	Нова технологія	Доступна технологія
2		Розробка моделей вилучення аудіофіч	Існуюча технологія	Доступна технологія
3		Розробка алгоритму роботи класифікації і сегментації онлайн	Нова технологія	Доступна технологія
Обрана технологія реалізації ідеї проекту: за рахунок розробки алгоритму з використанням нейронних мереж				

Проаналізувавши таблицю 4.3. було обрано пункт під номером 1, оскільки використання систем штучного інтелекту не лише покращить точність класифікації, а й збільшить швидкість моніторингу радіопотоку, що в свою чергу дозволить даному проекту конкурувати на ринку. Окрім цього, використання нейронних мереж нове рішення, а отже даний продукт на ринку буде єдиний.

4.3. Аналіз ринкових можливостей запуску стартап-проєкту

Таблиця 4.4. Попередня характеристика потенційного ринку

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	6
2	Загальний обсяг продаж, грн/ум.од	5000-10000
3	Динаміка ринку (якісна оцінка)	Стагнація
4	Наявність обмежень для входу (вказати характер обмежень)	Висока конкуренція
5	Специфічні вимоги до стандартизації та сертифікації	Відповідність стандартам ГОСТ 14254-96, ГОСТ 12.1.019-79, ГОСТ 26104-89, ГОСТ 12.1.004-9 та ГОСТ 12.2.007.0-75.

6	Середня норма рентабельності в галузі (або по ринку), %	34%

Закінчення таблиці 4.4

Провівши аналіз попиту, можна зробити висновок, що коефіцієнт рентабельності достатньо високий. Оскільки, середня норма рентабельності складає більш ніж 30%, що майже вдвічі вигідніше за депозити в банку, а це в свою чергу, найбільше приваблює інвесторів.

Далі визначаємо потенційні групи клієнтів, їх характеристики, та формуємо орієнтовний перелік вимог до товару для кожної групи (табл. 4.5).

Таблиця 4.5. Характеристика потенційних клієнтів стартап-проєкту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Точність виявлення радіоактивного джерела	Ліквадатори аварій на АЕС, інспектори радіаційної безпеки	Перевага надається новим алгоритмам з покращеними характеристиками	Низька ціна та висока точність кінцевого продукту
2	Виявлення низькоактивних джерел радіації	Інспектори радіаційної безпеки	Перевага надається сигналізаторам або детекторам	Низька ціна та висока точність кінцевого продукту
3	Побудова радіаційних карт в режимі реального часу	Інспектори радіаційної безпеки та будь-які заінтересовані особи	Перевага надається дозиметрам, що можуть підключатись до телефонів	Точність побудови та низька ціна

Потреби, що формують ринок є точність виявлення радіоактивного джерела, можливість виявлення низькоактивних джерел радіації та побудова радіаційних карт в режимі реального часу. При цьому потенційні клієнти потребують нижчої ціни пристрою та високої точності роботи

Ринкові можливості – це сприятливі обставини, які підприємство може використовувати для отримання переваг. Як приклад ринкових можливостей можна привести погіршення позицій конкурентів, різке зростання попиту, появу нових технологій виробництва продукції, зростання рівня доходів

населення і т. п. Слід зазначити, що можливостями з погляду SWOT-аналізу є не всі можливості, які існують на ринку, а тільки ті, які можна використовувати

Проведемо аналіз факторів ринкового середовища, що сприяють ринковому впровадженню (табл. 4.6) проекту, та факторів, що йому перешкоджають (табл. 4.7).

Таблиця 4.6. Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Потреба в розробці точніших алгоритмів виявлення радіації	Знаходження або переманювання кваліфікованих кадрів для написання алгоритму	Підвищення робітникам заробітної плати для утримання їх на робочих місцях
2	Потреба в крос-платформленості пристрою	Програма не повинна працювати тільки на одному програмному забезпеченні	Відведення додаткового часу та ресурсів для вирішення цієї проблеми
3	Потреба в сучасних технологіях розробки	Використання сучасних компонентів сприятиме підвищенні ефективності системи в цілому	Співпраця з виробниками напівпровідників, взаємовигода від співпраці
4	Потреба роботи пристрою у режимі спектрометру	Програма має працювати у різних режимах для задоволення потреб усіх користувачів	Написання програмістами компанії оновлених алгоритмів
5	Потреба роботи пристрою у режимі детектору		

Отже, при виведені проект на ринок необхідно враховувати усі загрози та ризики. Основними ризиками даного проекту може бути потреба у різних режимах роботи та точності роботи алгоритмів, використання сучасних алгоритмів роботи дозволить швидко реагувати на можливі загрози.

Розглянемо можливості даного проекту (табл. 4.7)

Таблиця 4.7. Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Зростання конкурентів	Зростання конкурентноспроможних розробок	Покращення характеристик системи та розробка нових функцій

2	Поява схожих алгоритмів виявлення радіації у конкурентів	Конкуренти можуть розробити аналогічні або дуже схожі алгоритми роботи	Оформлення заявки на патент для власної розробки
3	Стрімкий розвиток технологій	Розвиток нових сучасних технологій	Перехід на сучасні технології

Закінчення таблиці 4.7

Разом з розширенням ринку та плином часу фактори можливостей будуть збільшуватись, разом з тим будуть збільшуватись і загрози, тому необхідно швидко реагувати за можливості і загрози.

Проведемо аналіз пропозиції (табл. 4.8) та визначимо загальні риси конкуренції на ринку:

Таблиця 4.8.Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1.Тип конкуренції: Олігополія	На ринку налічується не багато фірм, що випускають дані пристрої.	Необхідно слідкувати за якістю елементів пристрою та точністю роботи алгоритмів
2. За рівнем конкурентної боротьби: національний	Користувачами даного пристрою можуть бути фірми з усього світі. Доставка товару не відіграє особливої ролі для користувачів	Необхідно розширювати сегмент користувачів
3. За галузевою ознакою: внутрішньогалузева	Основною галуззю є радіаційний контроль.	Необхідно розширювати функціональні можливості даного пристрою
4. Конкуренція за видами товарів: товарно-видова	Спостерігається конкуренція між схожими алгоритмами	Підвищення точності та ефективності алгоритмів
5. За характером конкурентних переваг: цінова	Ціни на дані пристрої не високі, за допомогою нового алгоритму та нових компонентів збільшується якість продукції	Необхідність використання нових технологій та якісних елементів для дозиметричного пристрою.
6. За інтенсивністю: марочна	Велику роль відіграє репутація компанії, що виготовляє пристрій	Збільшити просування товару, покращити взаємодію з користувачами

Провівши ступеневий аналіз на ринку, можна зробити висновок, що на ринку достатньо конкурентоспроможних пристроїв, проте як вже було зазначено вище, використання елементів штучного інтелекту для радіаційного

контролю дає змогу покращити швидкодію та точність алгоритму, при цьому не збільшуючи кінцеву вартість пристрою.

Таблиця 4.9 Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари замінники
	Навести перелік прямих конкурентів	Визначити бар'єри входження в ринок	Визначити фактори сили постачальників	Визначити фактори сили споживачів	Фактори загроз з боку замінників
Висновки: ринок є достатньо заповнений конкурентоспроможними пристроями, конкуренція висока, оскільки дані пристрої добре зарекомендували себе в якості систем контролю радіації. Завадою до входу на ринок також є значна сума капіталовкладень.	Atom Fast, Smart Geiger PRO, Терра-П, Gamma Sapiens, Ecotest VIP, дозиметр-радіометр МКС-АТ6130 та МКС-17Д «Зяблик»	Висока репутація конкурентних фірм; необхідний розмір інвестицій та необхідний час для проходження сертифікацій якості та безпечності пристрою.	Зазвичай постачальники не диктують умови співпраці	Користувачам важлива низька ціна та висока точність роботи	Точніші алгоритми для виявлення радіоактивних джеоел, які можуть надати товари-замінники.

Провівши аналіз конкуренції в галузі за М. Портером видно, що є достатня кількість прямих конкурентів, що гарно зарекомендували себе на ринку, тому необхідно постійно слідкувати за якістю та точністю роботи свого пристрою. Також, необхідно постійно покращувати алгоритми роботи та технології.

На основі аналізу конкуренції в галузі, що наведено в таблиці 4.9, а також із урахуванням характеристик ідеї проекту, які були розглянуті в таблиці 4.2, вимог споживачів до товару (табл. 4.5.) та факторів маркетингового середовища (табл. 4.6, 4.7) визначимо та обґрунтуємо перелік факторів конкурентоспроможності. Аналіз конкурентоспроможності представлено в таблиці 4.10.

Таблиця 4.10 Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Швидкодія роботи алгоритму виявлення радіації та його точність	Використання нових алгоритмів та систем, що підвищують швидкодію та точність алгоритму
2	Цінова політика	Ціна на продукт буде значно нижчою ніж у конкурентів
3	Крос-платформленість	Зв’язок дозиметру не тільки з Android, а й з IOS. В подальшому робота зв’язок пристрою з ПК та ноутбуками.
4	Зворотній зв’язок	Користувачі потребуються консультування як в технічних так і в експлуатаційних питаннях
5	Репутація	У зв’язку з тим, що пристрій буде використовуватись для радіаційної безпеки, то репутація є важливим фактором під час вибору пристрою

Як можна побачити з таблиці було обґрунтовано основні п’ять факторів конкурентоспроможності, основними з яких стали: цінова політика, швидкодія роботи алгоритму та його точність і репутація компанії.

Проведемо порівняльний аналіз сильних та слабих сторін факторів конкурентоспроможності (табл. 4.11).

Таблиця 4.11 Порівняльний аналіз сильних та слабких сторін

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з власним пристроєм «RadioInstant»						
			-3	-2	-1	0	+1	+2	+3
1	Швидкодія роботи алгоритму та його точність	20						+	
2	Цінова політика	16			+				
3	Крос-платформленість	15						+	
4	Зворотній зв’язок	17				+			
5	Репутація	13					+		

З таблиць 4.10 та 4.11 бачимо, що фактори конкурентоспроможності суттєві та мають великий позитивний внесок при впроваджені на ринок

пристрою «RadioInstant». Основною перевагою даного пристрою є висока точність алгоритму та низька ціна.

Далі проведемо SWOT-аналіз стартап-проекту [4], що наведено в таблиці 4.12.

Таблиця 4.12 SWOT-аналіз стартап-проекту

Сильні сторони: 1. Гнучка цінова політика. 2. Можливість побудови радіаційних карт в режимі реального часу. 3. Висока точність алгоритму. 4. Виявлення низькоактивних джерел радіації	Слабкі сторони: 1.Низька репутація пристрою на початку впровадження проекту. 2. Необхідність значного початкового капіталовкладення (кредит в банку або сторонні інвестиції)
Можливості: 1. Стрімке поширення товару на ринку за рахунок доступності технології. 2. Розробка нових функцій пристрою	Загрози: 1. Розробка кращих алгоритмів виявлення радіації. 2. Мінливість ринку.

Для успішності впровадження стартап-проекту на ринку необхідно враховувати появу ризиків та слабких сторін проекту. Наприклад, для покращення репутації пристрою необхідно враховувати побажання користувачів та постійно оновлювати алгоритми роботи.

На основі SWOT-аналізу розробимо альтернативи ринкової поведінки для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок.

Визначені альтернативи аналізуються з точки зору строків та ймовірності отримання ресурсів (табл. 4.13).

Таблиця 4.13 Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Створити прототип пристрою контролю радіації	Доставка елементів для розробки прототипу не займе багато часу	14-22 днів
2	Використання сучасних чуттєвих елементів з вже існуючих дозиметрів	Для цього необхідно замовити велику партію сучасних дозиметрів на мікроконтролерних платах, потім необхідно їх розібрати	20-38 днів

		та підключити сучасні чуттєві елементи до власної плати.	

Закінчення таблиці 4.13

Краще за все використовувати підхід, що наведено у першому пункті, оскільки час його реалізації до трьох тижнів. Використання сучасних чуттєвих елементів з вже існуючих дозиметрів не найкращий варіант, оскільки виникають ризики складності підключення чуттєвих елементів та пошук альтернативних варіантів підключення. Також використання сучасних чуттєвих елементів значно підвищить ціну пристрою. З зазначених альтернатив обираємо стратегію компенсації слабких сторін стартапу наявними ринковими можливостями.

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів.

Таблиця 4.14 Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Побудова радіаційних карт	Висока	Середній	Не інтенсивна	Середня складність
2	Виявлення низькоактивних джерел радіації	Висока	Середній	Не інтенсивна	Середня складність
Цільова група: інспекторів з радіаційної безпеки та працівники АЕС					

За результатами аналізу потенційних груп споживачів було обрано працівників АЕС та інспекторів з радіаційної безпеки. Цей вибір аргументовано тим, що для даних груп користувачів конкуренція не дуже висока та невисока складність входу на ринок для даних груп користувачів. Оскільки важливим фактором залишається ціна продукції, тоді при відповідності ціновим очікуванням користувача та якістю роботи алгоритмів потрапити до даного сегменту буде неважко.

Для роботи в обраних сегментах ринку необхідно сформувати базову стратегію розвитку (табл. 4.15).

Таблиця 4.15 Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Створення прототипу пристрою контролю радіації	Швидке налаштування виробництва	Можливо закупляти вже готові автономні дозиметри та розробляти для них власні алгоритми контролю радіації, але в такому випадку кінцевий продукт від конкурентів буде відрізнятись лише ціною	Стратегія спеціалізації

Визначена базова стратегія розвитку проекту – стратегія спеціалізації, оскільки ця стратегія передбачає концентрацію на потребах одного цільового сегменту, без прагнення охопити увесь ринок. Мета тут полягає в задоволенні потреб вибраного цільового сегменту краще, ніж конкуренти. Така стратегія може спиратися як на диференціацію, так і на лідерство по витратах, або і на те, і на інше, але тільки у рамках цільового сегменту [3]. Тоді, точність роботи алгоритму дає можливість встановлювати вищу ціну на продукцію, так як споживачі готові їх сприйняти.

Наступним кроком є вибір стратегії конкурентної поведінки (табл. 4.16).

Таблиця 4.16 Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
1	Ні	Компанія буде забирати існуючих клієнтів у конкурентів	Ні, плануються власні інноваційні розробки	Наступальна стратегія

На основі проведеного аналізу для вибору стратегії конкурентної поведінки була обрана наступна стратегія — наступальна стратегія. Наступальна стратегія припускає збільшення своєї частки ринку. При цьому

переслідувана мета полягає в подальшому підвищенні прибутковості роботи компанії на ринку за рахунок максимального використання ефекту масштабу. Наступальна стратегія припускає активну інноваційну політику компанії. Вона постійно атакує власні ж досягнення, збільшуючи розрив між собою і основними конкурентами [3].

На основі вимог споживачів з обраних сегментів до постачальника та до продукту, а також в залежності від обраної базової стратегії розвитку та стратегії конкурентної поведінки розробимо стратегію позиціонування (табл. 4.17).

Таблиця 4.17 Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1	Постійне вдосконалення продукту враховуючи побажання споживачів	Стратегія спеціалізації	Висока точність роботи пристрою та формування прихильності користувачів	Зворотній зв'язок із виробником, технічна підтримка, якість та точність
2	Обслуговування	Стратегія спеціалізації	Легке обслуговування	Простота в використанні, точність роботи та швидкодія алгоритму
3	Якість	Стратегія спеціалізації	Швидкість, гнучкість	Ціна, якість

Отже, окрім високої точності роботи алгоритму та низької ціни користувач також потребує легкості в обслуговуванні, простоти в використанні та постійного оновлення функцій продукту.

Під час розробки маркетингової програми першим кроком є розробка маркетингової концепції товару, який отримає споживач. У таблиці 4.18 підсумуємо результати аналізу конкурентоспроможності товару.

Таблиця 4.18 Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Точність роботи алгоритму	Точність визначення радіоактивних джерел, можливість визначення низькоактивних радіоактивних джерел	Можливість підвищити точність роботи алгоритму за рахунок використання систем штучного інтелекту, підвищення швидкодії алгориму
2	Відмовостійкість	Стабільність роботи приладу	Висока стабільність системи роботи системи за рахунок використання систем штучного інтелекту

Очевидними вигодами даного товару є стабільність роботи пристрою, точність визначення джерел радіації та можливість визначення низькоактивних радіоактивних джерел, що надає перевагу даному пристрою у порівнянні з конкурентами. В подальшому, можливі зміни у режимах роботи пристрою (детектор, радіометр) та використання оновлених технологій, що дозволить вивести проект на високий рівень.

Наступним кроком є розробка трирівневої маркетингової моделі товару, де уточнюється ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання (табл. 4.19).

Таблиця 4.19 Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові	
I. Товар за задумом	Детектор на базі апаратної плати Arduino Uno, чуттєвим елементом якого є трубки Гейгера-Мюллера, даний пристрій з’єднується с мобільним додатком через Wi-Fi модуль.	
II. Товар у реальному виконанні	Властивості/характеристики	
	1. Новий підхід до алгоритму	Застосування систем штучного інтелекту для написання алгоритму виявлення радіації
	2. Простота технічної реалізації	Використання радянських трубок Гейгера-Мюллера для підключення до Arduino Uno
	3. Додаткове програмне забезпечення	Додавання нових можливостей

	Якість: відповідає нормам ГОСТ 14254-96, ГОСТ 12.1.019-79, ГОСТ 26104-89, ГОСТ 12.1.004-9 та ГОСТ 12.2.007.0-75.
	Пакування: готовий пристрій має вигляд пластмасового непрозорого циліндру.
ІІІ. Товар із підкріпленням	До продажу з введенням у роботу
	Після продажу – технічна підтримка, гарантійне обслуговування
Потенційний товар буде захищено від копіювання за рахунок логотипу та патент на розроблений алгоритм	

Закінчення таблиці 4.19

Далі визначимо цінові межі, якими необхідно керуватися при встановленні ціни на потенційний товар, це передбачає аналіз цін товарів конкурентів, та доходів споживачів продукту (табл. 4.20).

Таблиця 4.20 Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	1000-5000	1500-10000	25000	500-750

Якщо збільшувати кількість користувачів, то можна трохи понизити ціну товару. В будь-якому випадку ціна буде меншою, ніж у товарів аналогів.

Наступним кроком є визначення оптимальної системи збуту, в межах якого приймається рішення (табл. 4.21):

Таблиця 4.21 Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має Виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Попереднє замовлення з підписанням контракту	Доставка в строки, контроль за уникненням пошкоджень.	Пряма	Пряма
2	Оптові замовлення			

У зв’язку з тим, що обслуговується вузьконаправлений сегмент ринку доцільніше використовувати прямий канал збуту. Використання прямого каналу збуту дає можливість більш якісно контролювати ціни на ринку та не тратити доходи на оплату роботи посередників.

Останньою складової маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (табл. 4.22).

Таблиця 4.22 Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Огляд продукту, перспективи на майбутнє	Форма зворотного зв'язку в мобільному додатку	Встановлення мети позиціювання. Розробка стратегії позиціювання. Розробка комплексу маркетингу. Оцінка ефективності позиціювання.	Розповсюдження інформації про продукт	Підкреслення переваг продукту

Було обрано ключові позиції: маркетингові дослідження, встановлення мети позиціювання, розробка стратегії позиціювання, розробка тактики позиціювання, розробка комплексу маркетингу, оцінка ефективності позиціювання. Розглянуто завдання рекламного повідомлення та концепцію рекламного звернення.

Висновки до розділу

У розділі як стартап проект пропонується інтелектуальна система класифікації типів радіоконтенту. Алгоритм класифікації радіотипів написано з елементами штучного інтелекту, а саме з трьома нейромоделями, що дає змогу значно підвищити швидкодію та точність моніторингу радіо. Також використання системи штучного інтелекту дає можливість визначення відповідності радіоквотам. Даний метод потребує складної технічної реалізації та тривалого часу навчання. Тому пропонується використання lstm

мережі для покращення точності та бібліотеки VGGish для вилучення аудіофіч.

Основними споживачами приладу будуть держслужбовці та державні органи. Під час проведення аналізу ринку потенційного стартап-проекту було встановлено, що даний пристрій буде користуватись попитом у користувачів обраного сегменту, також, використання інноваційного алгоритму класифікації довгих аудіоданих може привернути увагу власників телерадіокомпаній для моніторингу власного продукту.

Сильними сторонами проекту є можливість побудови CI/CD системи для замовників.

Серед можливих конкурентів, можна виділити два інші компанії з розробки штучного інтелекту.

Для реклами та збільшення обсягів збуту пропонується надавати додаткові послуги користувачам з подальшого обслуговування та гарантії на 6 місяців після покупки. Також можна робити рекламні акції та оптові знижки задля приваблення користувачів та збільшення обсягів продажу. Доцільно вважати, що необхідно вкласти у стартап-проект суму близько $23000.

ВИСНОВКИ

В дипломній роботі розглянуто використання нейромереж для створення інтелектуальної системи. Роботу присвячено вирішенню задачі класифікації довгого аудіоконтенту як предметної області, вибору найбільш оптимального методу висконання для мережі згідно до її структури та характеру використання, інтеграції мережі у систему прийняття рішень для тестування програмної реалізації веб-додатку, обраних моделей та демонстраційного середовища згідно до предметної області.

У роботі було розроблено інтелектуальну систему, яка відповідає предметній області демонстраційної задачі, запропоновані методи її програмної реалізації та визначено характер використання нейромережі. Було описано типові алгоритми для побудови ІС – виділення аудіофіч за допомогою згорткових нейромереж, надання значень для класифікації радіотипам, або умов, та на основі отриманої інформації про характер використання мережі обрано найбільш оптимальний як і у плані виконання, так і кінцевої точності модель для класифікації.

Було виконано всі кроки для переходу нейронних мереж від обробки даних до класифікації кінцевих ознак. Ми отримали результати тестів для двох різних наборів даних, створених за допомогою бібліотеки VGGish, і класифікували їх із зміненими параметрами для роботи з довгими аудіозаписами за допомогою інтерфейсу Youtube-8M. Отримані дані свідчать про те, що моделі CNN lstm та DDN dbof є найкращими для такого роду класифікації аудіовмісту. Їх моделі представлені в документації Youtube-8M. Варіанти передачі також були надані при навчанні моделей.

Однією з головних невирішених проблем у цій статті є сегментація звукового потоку в режимі реального часу. Тож подальша робота буде спрямована на вдосконалення моделей, що надаються бібліотекою, тестування продуктивності моделі за допомогою класифікатора уваги та процесу сегментації аудіо.

ПЕРЕЛІК ПОСИЛАНЬ

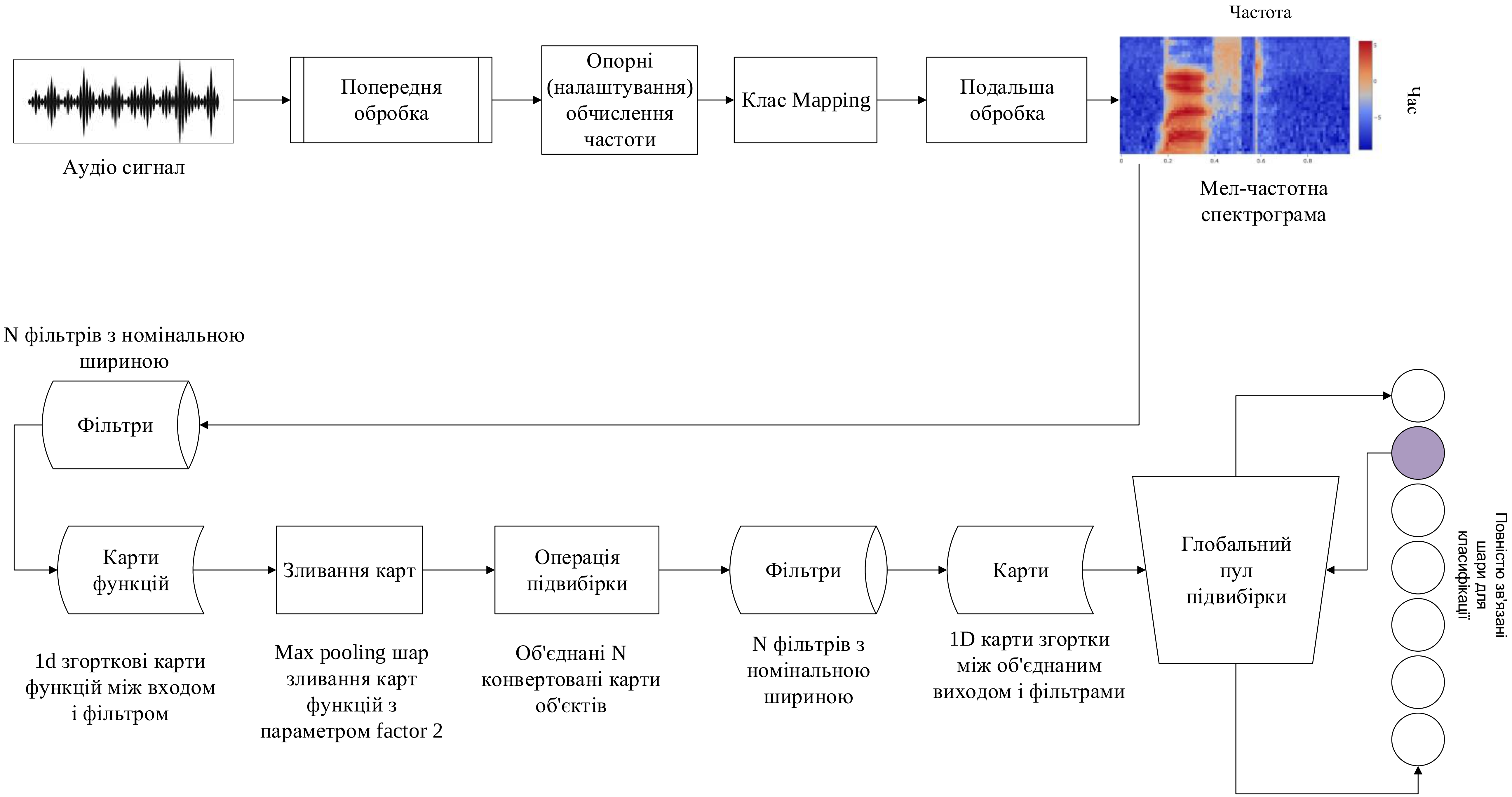
1. Darwiche A. Modeling and Reasoning with Bayesian Networks / Adnan Darwiche., 2009. – 562 с.
2. Pear J. Fusion, propagation, and structuring in belief networks / Judea Pear. // Artificial Intelligence. – 1986. – №29. – С.241–288.
3. Nielsen T. Bayesian Networks and Decision Graphs / Thomas Nielsen. – New York: Springer-Verlag, 2007. – 268 с. – (2). – (Information Science and Statistics series).
4. Fenton N. Managing Risk in the Modern World: Applications of Bayesian Networks / N. Fenton, M. Neil. – London: London Mathematical Society, 2007. – 168 с. – (Knowledge Transfer Report from the London Mathematical Society and the Knowledge Transfer Network for Industrial Mathematics).
5. Comley J. Minimum Message Length and Generalized Bayesian Nets with Asymmetric Languages / J. Comley, D. Dowe // Advances in Minimum Description Length: Theory and Applications / J. Comley, D. Dowe. – Cambridge, Massachusetts, 2005. – (MIT Press). – (Neural information processing). – С.265–294.
6. The BUGS project: Evolution, critique and future directions [Електронний ресурс] / D.Lunn, D. Spiegelhalter, A. Thomas, N. Best // Statistics in Medicine. – 2009. – Режим доступу до ресурсу: https://doi.org/10.1002%2Fsim.3680.
7. Korb K. Bayesian Artificial Intelligence / K. Korb, A. Nicholson., 2010. – 221 с. – (CRC Computer Science & Data Analysis). – (2).
8. Jason G. Game Engine Architecture / Gregory Jason., 2018. – 1240 с. – (3).
9. Robert N. Game Programming Patterns / Nystrom Robert., 2014. – 354 с. – (1).
10. Eric L. Foundations of Game Engine Development, Volume 1: Mathematics / Lengyel Eric., 2016. – 200 с. – (1).
11. Ian M. Game Physics Engine Development: How to Build a Robust Commercial-Grade Physics Engine for your Game / Millington Ian., 2010. – 552 с. – (2).
12. Sanglard F. Game Engine Black Book: Doom / Fabien Sanglard., 2018. – 427 с.

13. A visual attention based convolutional neural network for image classification. // IEEE.2016.
14. Z. Qawaqneh, AA Mallouh, BD Barkana, A framework for a neural network and a transformed MFCC for the classification of the speaker and his statics // Knowledge-based systems 115 (2017) 5-14.
15. Intelligent system of multimodal efficient data analysis // Neural Networks 63 (2015) 104-116.
16. C. Pedersen, J. Diederich, Classification of accents using reference vector machines // IEEE // ACIS International Conference on Computer and Information Science (ICIS), IEEE, 2007, pp. 444 -449.
17. Continuous speech recognition based on wavelet transform // зв'язку (ICSPCS), IEEE, 2016, с. 1-8.
18. Ye, Fundamentals of implementation of the competitive level model using recurrent neural networks lotka-volterra // (2010) 494-507.
19. Z. Fu, G. Lu, K. M. Ting, D. Zhang, Optimizing cepstral features for audio classification, in: International Joint Conference on Artificial Intelligence, 2013.
20. M. Espi, M. Fujimoto, K. Kinoshita, T. Nakatani, Exploiting spectrotemporal locality in deep learning based acoustic event detection, Journal on Audio, Speech, and Music Processing 2015 (1) (2015) 26.
21. W. Lim, D. Jang, T. Lee, Speech emotion recognition using convolutional and recurrent neural networks, in: 2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA), IEEE, 2016, pp. 1–4.
22. Y. Leng, C. Sun, X. Xu, Q. Yuan, S. Xing, H. Wan, J. Wang, D. Li, Employing unlabeled data to improve the classification performance of SVM, and its application in audio event classification, Knowledge-Based Systems 98 (2016) 117–129.
23. Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, R. Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron J. Weiss, Kevin Wilson, CNN architectures for

large-scale audio classification, Google, Inc., New York, NY, and Mountain View, CA, USA.

24. J. H. Hansen, G. Liu, Unsupervised accent classification for deep data fusion of accent and language information, Speech Communication 78 (2016) 19 – 33.
25. Sami Abu-El-Haija YouTube-8M: A Large-Scale Video Classification Benchmark (2019).
26. YouTube-8M documentation // URL: http://research.google.com/youtube8m/
27. YouTube-8M Tensorflow // URL: https://github.com/google/youtube-8m#overview-of-models.
28. CNN architectures for large-scale audio classification, Google inc., (10 Jan 2017).
29. Devhive: Using VGGish for training tensorflow models // URL: https://docs.devicehive.com/blog/using-gpus-for-training-tensorflow-models

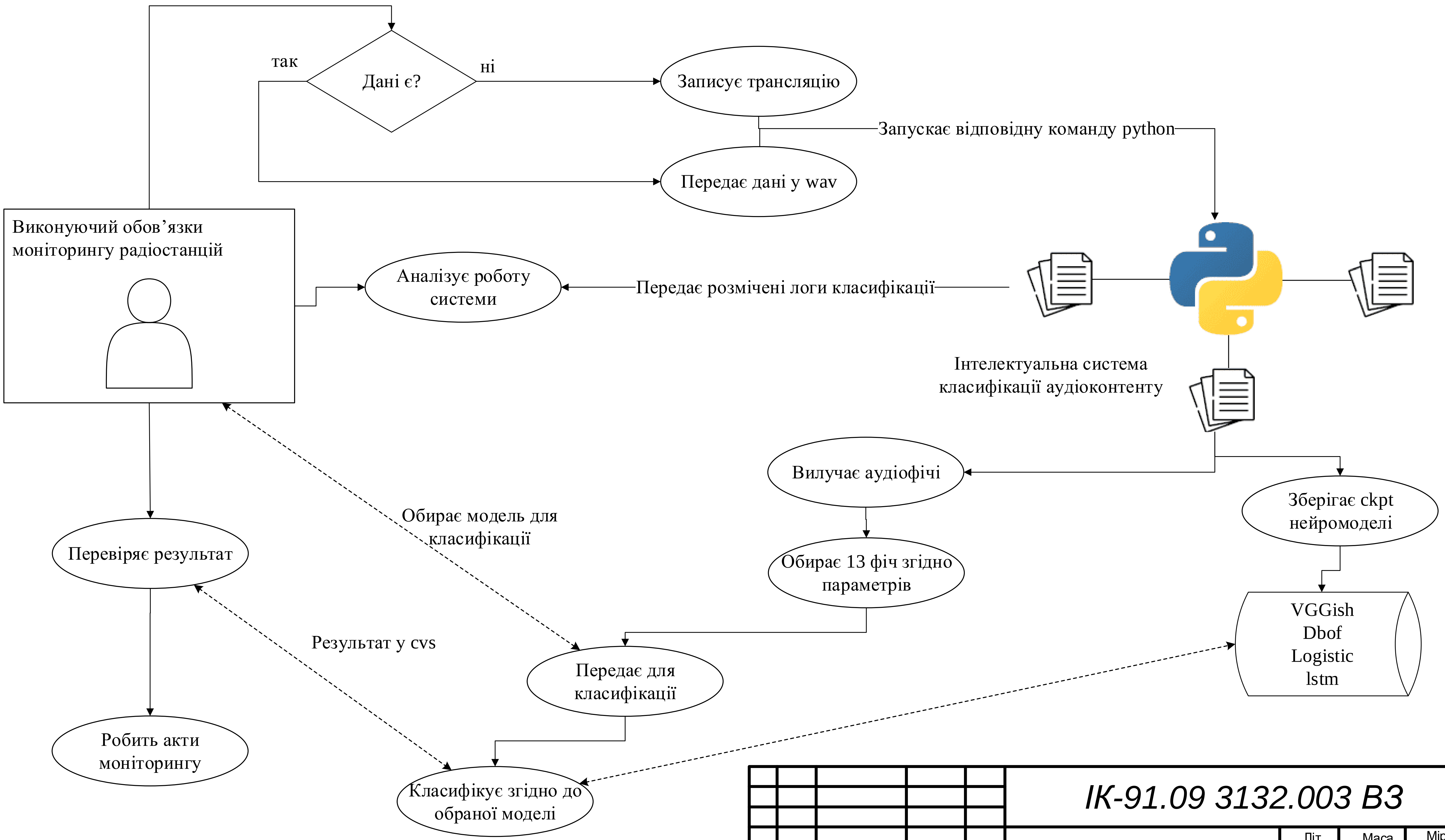
ІК-91.09 3132.002 ВЗ



Зм.	Лист	№ докум	Підпис	Дата	ІК-91.09 3132.002 ВЗ
Розроб.		Жиляєв А.М.			Схема системи класифікації аудіо
Перев.		Олійник В.В.			Літ. / Маса / Мірило
					Лист 1 / Листів 1
Н.контр		Пасько В.П			Кафедра Технічної кібернетики
Затв.		Пархомей І.Р.			Група ІК-91мп

Підпис і дата
Інв. № дубл.
Взам. інв. №
Підпис і дата
Інв. № ориг.

ІК-91.09 3132.003 ВЗ



Підпис і дата | Інв. № дубл. | Взам. інв. № | Підпис і дата | Інв. № ориг.

Зм.	Лист	№ докум	Підпис	Дата	
Розроб.	Жіляєв А.М.				
Перев.	Олійник В.В.				
Н.контр	Пасько В.П				
Затв.	Пархомей І.Р.				

ІК-91.09 3132.003 ВЗ

Схема системи аудіокласифікації

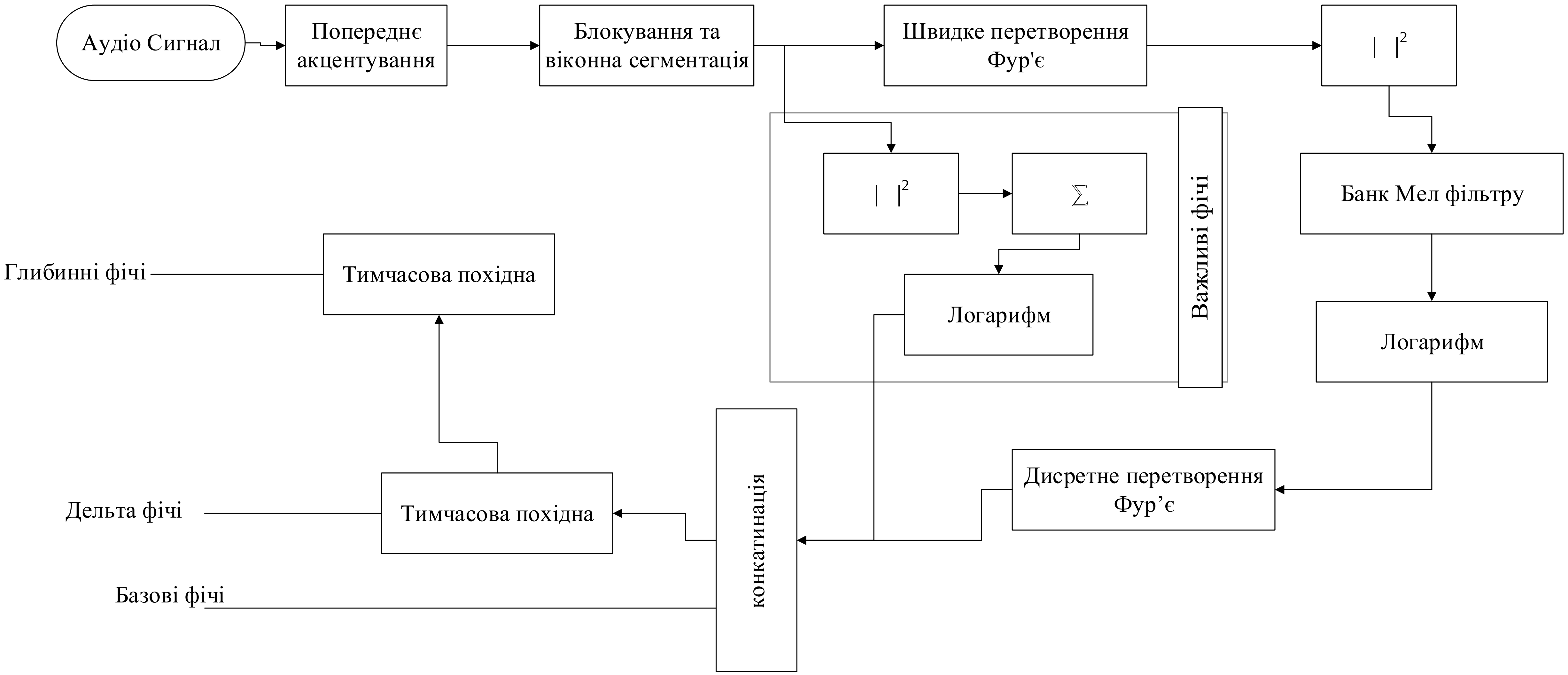
Літ.	Маса	Мірило

Лист 1 | Листів 1

Кафедра Технічної кібернетики

Група ІК-91мп

ІК-91.09 3132.004 ВЗ



Підпис і дата
Інв. № дубл.
Взам. інв. №
Підпис і дата
Інв. № ориг.

Зм.	Лист	№ докум	Підпис	Дата
Розроб.		Жіляєв А.М.		
Перев.		Олійник В.В.		
Н.контр		Пасько В.П		
Затв.		Пархомей І.Р.		

ІК-91.09 3132.004 ВЗ

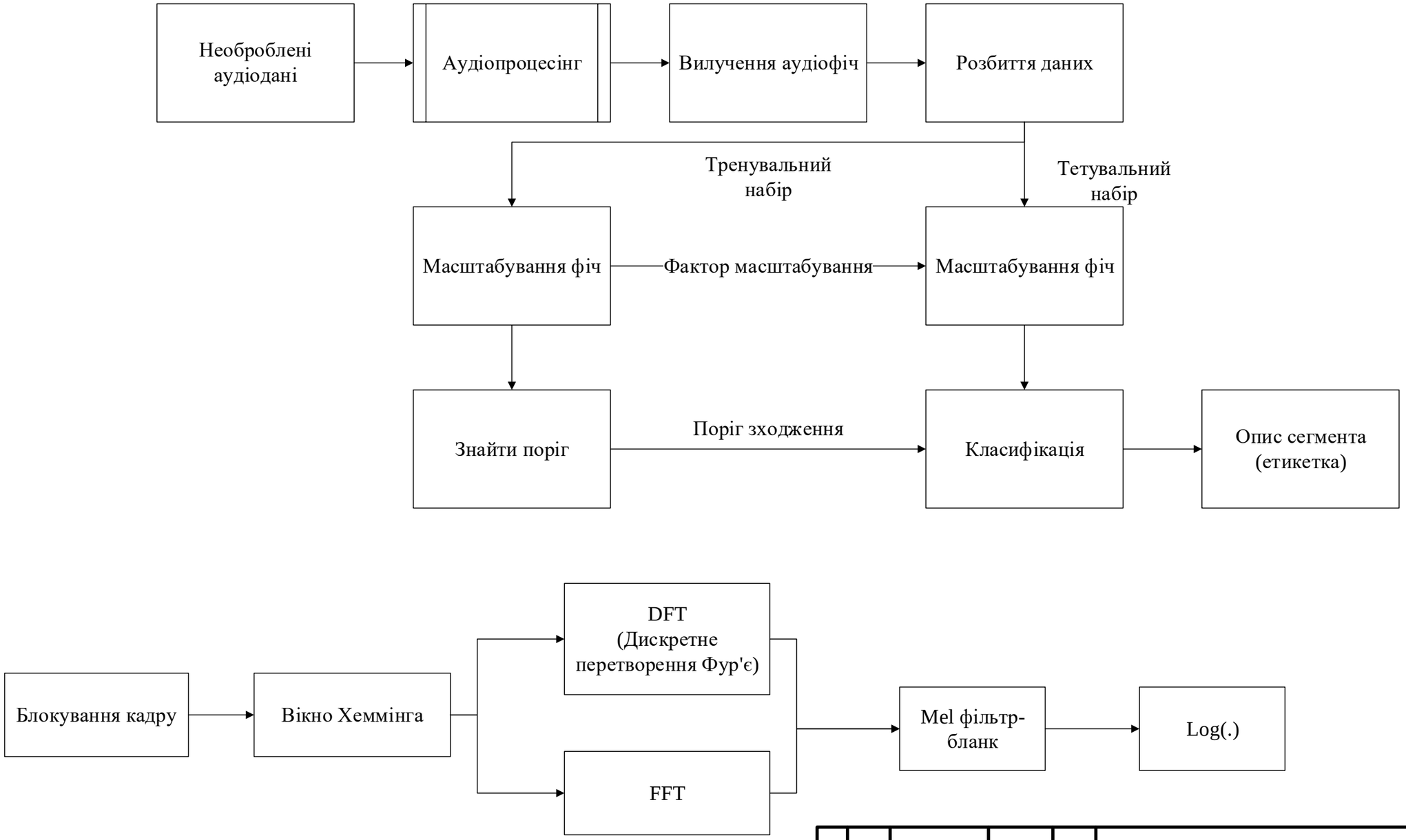
Трансформація аудіопотоку

Літ. | Маса | Мірило

Лист 1 | Листів 1

Кафедра Технічної кібернетики

Група ІК-91мп

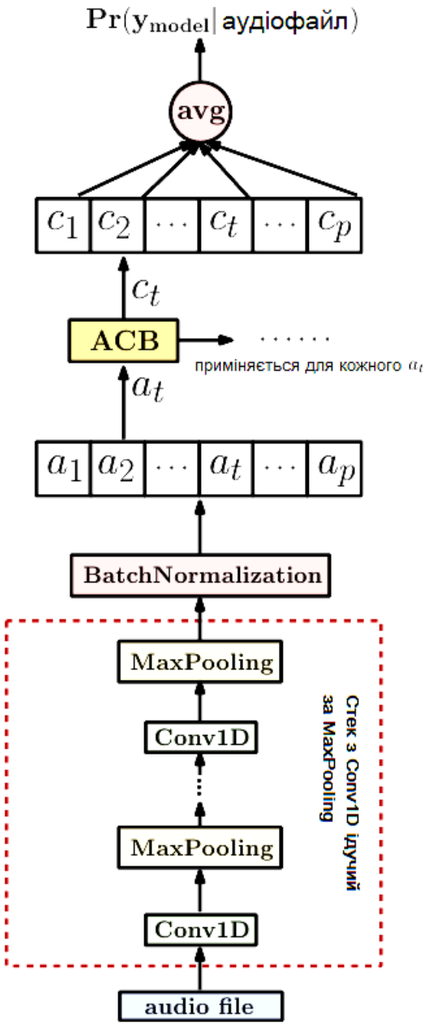


Аудіо обробка

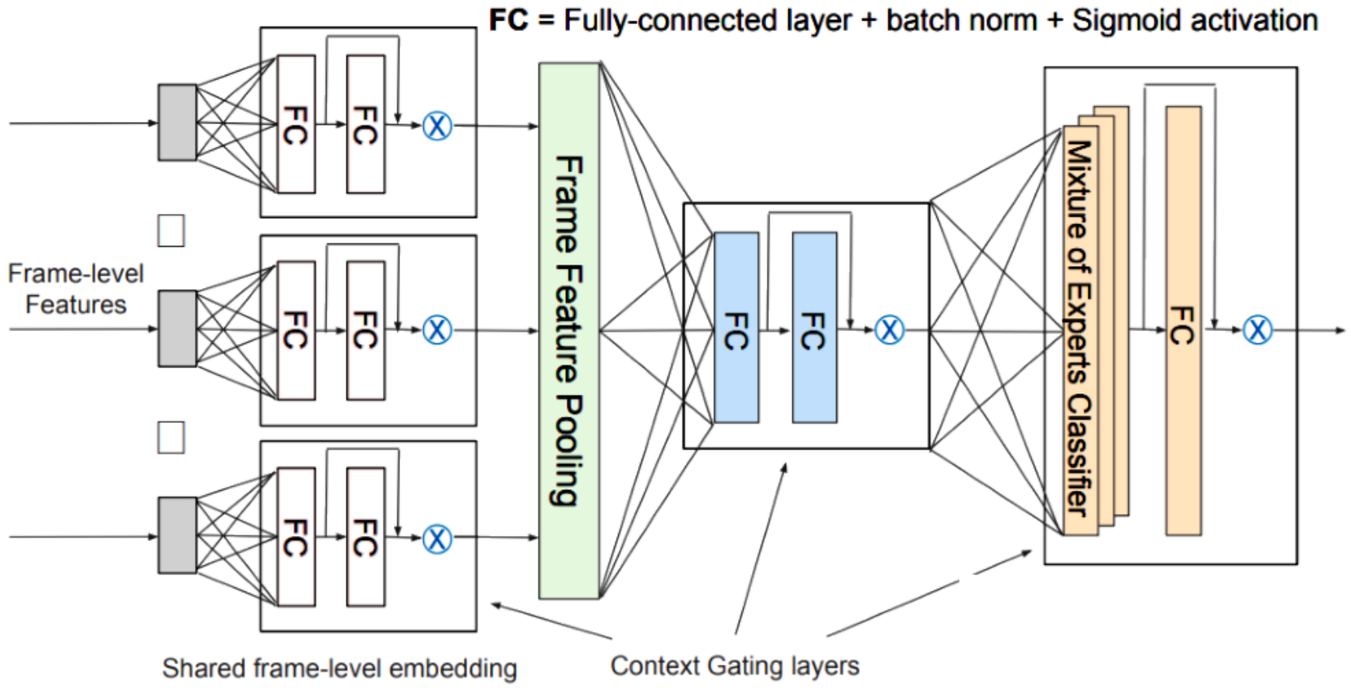
Підпис і дата
Інв. № дубл.
Взам. інв. №
Підпис і дата
Інв. № ориг.

					ІК-91.09 3132.005 ВЗ	
Зм.	Лист	№ докум	Підпис	Дата	Блок схема вилучення аудіофіч	Літ. / Маса / Мірило
Розроб.		Жіляєв А.М.				
Перев.		Олійник В.В.				Лист 1 / Листів 1
Н.контр		Пасько В.П			Кафедра Технічної кібернетики	Група ІК-91мп
Затв.		Пархомей І.Р.				

Схема моделей нейромереж
CAB-CNN та DBoF



CAB-CNN Модель

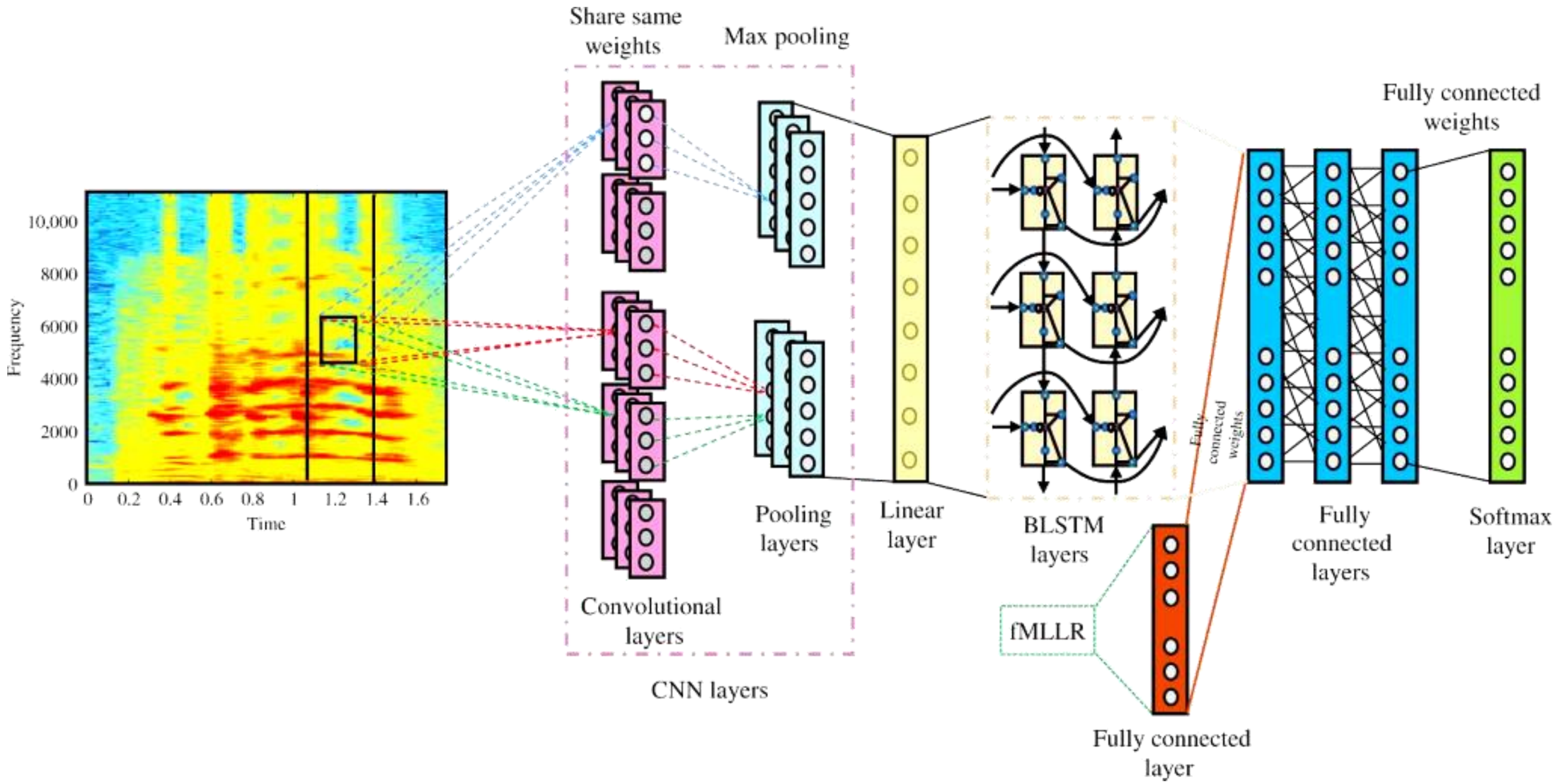


DBoF модель інтерфейсу Youtube-8m

Демонстраційний плакат № 1
до магістерської дисертації на тему
„Інтелектуальна система класифікації аудіоконтенту”

Розробив: Жіляєв А.М.
Прийняв: к.т.н., старший викладач Олійник В.В.

Схема моделі нейромережі BLSTM



Демонстраційний плакат № 2
до магістерської дисертації на тему
„Інтелектуальна система класифікації аудіоконтенту”

Розробив: Жіляєв А.М.
Прийняв: к.т.н., старший викладач Олійник В.В.