НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА ШТУЧНОГО ІНТЕЛЕКТУ

На правах рукопису
УДК 004.852

До захисту допущено
В.о. завідувача кафедри ШІ
О.І. ЧУМАЧЕНКО
«___» _____ 2022 р.

Магістерська дисертація

на здобуття ступеня магістра за спеціальністю 122 «Комп'ютерні науки»
на тему: «Визначення емоційного забарвлення голосу за допомогою технік Deep Learning у реальному часі»

Виконав:
студент 2 курсу, групи КІ- з11мп
Загарницький Дмитро Валерійович
_____

Керівник:Шаповал Наталія Віталіївна.
_____

Рецензент:
_____

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

Київ
2022

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА ШТУЧНОГО ІНТЕЛЕКТУ

Рівень вищої освіти — другий (магістерський)
Спеціальність (ОПП) — 122 «Комп'ютерні науки» («Системи і методи штучного інтелекту»)

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри ШІ
О.І. ЧУМАЧЕНКО
«___» _____ 2022 р.

ЗАВДАННЯ
на магістерську дисертацію студенту Загарницькому Дмитру Валерійовичу

1. Тема дисертації:«Визначення емоційного забарвлення голосу за допомогою технік Deep Learning у реальному часі», науковий керівник дисертації Шаповал Наталія Віталіївна, затверджені наказом по університету від «02» листопада 2022 р. № 4040-с.

2. Термін подання студентом дисертації: 12.12.2022 р.

3. Об’єкт дослідження: аудіозаписи голосів.

4. Предмет дослідження: моделі машинного навчання для визначення емоційного забарвлення голосу у реальному часі.

5. Перелік завдань, які потрібно зробити:

1) здійснити огляд літератури за темою роботи;
2) дослідити актуальність обраної теми;

3) ознайомитись із існуючими методами та моделями розпізнавання емоційного забарвлення голосу;

4) провести аналіз вхідних даних, попередньо провести їх обробку, примінивши та проаналізувавши різні підходи;

5) реалізувати навчання CNN моделі на вхідних даних за допомогою технології transfer learning, для перевірки можливості класифікації емоційного забарвлення на основі спектрограм голосу у форматі RGB;

6) провести експеримент, що засвідчує працеспроможність запропонованої моделі, виконати аналіз результатів;

7) провести аналіз ринкових можливотсей запуску стартап проєкту;

8) зробити висновки;

9) підготувати ілюстративний матеріал;

10) оформити пояснювальну записку.

6. Перелік ілюстративного матеріалу.

7. Дата видачі завдання:1 вересня 2022 року.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1.	Вивчення літератури за темою роботи.	02.09.2022 – 14.09.2022	Виконано
2.	Підготовка першого розділу.	15.09.2022 – 29.09.2022	Виконано
3.	Підготовка другого розділу.	30.09.2022 – 18.10.2022	Виконано
4.	Розробка програмного продукту.	20.10.2022 – 05.11.2022	Виконано
5.	Підготовка третього розділу	06.11.2022 – 18.11.2022	Виконано
6.	Підготовка частини стартап-проєкту	19.11.2022 –	Виконано

		22.11.2022	
7.	Концептуальні висновки. Перспективи розвитку отриманих рішень	23.11.2022 – 25.11.2022	Виконано
8.	Оформлення пояснювальної записки	26.11.2022 – 03.12.2022	Виконано

Студент Загарницький Дмитро

Науковий керівник дисертації Наталія ШАПОВАЛ

РЕФЕРАТ

Робота об’ємом 88 сторінок, яка містить 18 рисунків, 10 таблиць, 22 джерела за переліком посилань та 3 додатки.

Метою даної роботи є дослідження сучасних методів визначення емоційного забарвлення голосу та створення моделі для його аналізу у реальному часі.

Об’єкт дослідження - аудіозаписи голосів.

Предмет дослідження - моделі машинного навчання для визначення емоційного забарвлення голосу у реальному часі.

Результатом виконання роботи став детальний аналіз роботи чотирьох моделей глибоко навчання, навчених на даних різної якості. Також було створено програмне забезпечення, яке у реальному часі аналізує емоційне забарвлення голосу, використовуючи одну з натренованих моделей.

Ключові слова: Artificial Neural Network, Learning rate, SER, LVA, RNN, LSTM, GRU, CNN, Частота дискретизації, Смуга пропускання, Компандування.

ABSTRACT

Work of 88 pages, which contains 18 figures, 10 tables, 22 references and 3 appendices.

The purpose of this work is to research modern methods of Speech Emotion Recognition and to create a model for its real-time analysis.

The object of research is audio recordings of voices.

The subject of research is the machine learning models to determine the emotional color of voice in real time.

The result of the work was a detailed analysis of the work of four deep learning models trained on data of different quality. We also created software that analyzes the emotional color of the voice in real time using one of the trained models.

Keywords: Artificial Neural Network, Learning rate, SER, LVA, RNN, LSTM, GRU, CNN, Sampling frequency, Bandwidth, Companding.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Artificial Neural Network - штучна нейронна мережа, яка побудована з послідовно з’єднаних шарів штучних нейронів.

Learning rate - значення, яке корегує швидкість навчання моделі.

SER - розпізнавання емоційних аспектів мовлення.

LVA - багаторівневий аналіз голосу.

RNN - рекурентна нейронна мережа.

LSTM - підвид рекурентної нейронної мережі, який використовує принцип довготривалої короткострокової пам’яті

GRU - підвид рекурентної нейронної мережі, який використовує принцип вентильного рекурентного вузла.

CNN - згорткова нейронна мережа.

Частота дискретизації - визначає кількість записів у секунду.

Смуга пропускання - діапазон частот, у межах якого амплітудно-частотна характеристика забезпечує передачу сигналу з мінімальним викривленням його форми.

Компандування - техніка стиснення, а потім розширення (або декомпресії) аналогового або цифрового сигналу.

ВСТУП

Актуальність теми: Природним орієнтиром для систем AI є поведінка людини. У соціальному житті людини емоційний інтелект важливий для успішного та ефективного спілкування. Людина має природну здатність розуміти і реагувати на емоції своїх партнерів по спілкуванню за допомогою голосу та виразу обличчя. Давньою метою AI було створення моделей, які можуть розпізнавати, інтерпретувати та висловлювати емоції.

Вивчення настроїв взаємодії людини з людиною (HHI) може допомогти машинам ідентифікувати та реагувати на невербальне спілкування людини, що робить досвід HRI більш природним. Колл-центр – це багатий ресурс комунікаційних даних. Численні дзвінки записуються щодня, щоб оцінити якість взаємодії між CSR та клієнтами. Вивчення виразів настроїв у добре навчених CSR під час спілкування може допомогти AI зрозуміти не тільки те, що говорить користувач, а й те, як він/вона це говорить, щоб взаємодія відчувалася більш людською.

Але застосувати розпізнавання емоційного забарвлення голосу можна використати не тільки у взаємодії з клієнтами, а й для покращення роботи чат бота, різних застосунків для слідкування за своїм емоційним здоров’ям тощо. Тому дану тему вважаємо вкрай актуальною, тим паче, якщо мова йде про аналіз емоцій у реальному часі, коли у системи з’являєтсья можливість підлаштовуватись під користувача тут і зараз.

Метою роботи є дослідження сучасних методів визначення емоційного забарвлення голосу та створення моделі для його аналізу у реальному часі.

Завдання роботи:

- Розглянути існуючі на сьогоднішній день підходи для розпізнавання емоційного забарвлення голосу.
- Провести аналіз вхідних даних, попередньо провести їх обробку, примінивши та проаналізувавши різні підходи.
- Реалізувати навчання CNN моделі на вхідних даних за допомогою технології transfer learning, для перевірки можливості класифікації емоційного забарвлення на основі спектрограм голосу у форматі RGB.
- Провести аналіз результатів та оцінку ефективності роботи моделі при аналізі даних у реальному часі

Об’єкт дослідження - аудіозаписи голосу.

Предмет дослідження - моделі машинного навчання для визначення емоційного забарвлення голосу у реальному часі.

Методом дослідження є опрацювання джерел, що містять теоретичну інформацію щодо роботи систем визначення емоційного забарвлення голосу та використання у них методів машинного навчання; аналіз та підготовка набору даних, навчання обраних алгоритмів, тестування їх ефективності.

У цій роботі виявлення емоцій розглядається як проблема класифікації, оскільки метою є з'ясування, яка емоція (клас) превалює у голосу в момент часу.

Наукова новизна одержаних результатів полягає у тому, що було досліджено класифікацію емоційного забарвлення саме аудіо версії голосу, адже сучасні роботи у більшості орієнтовані на класифікацію саме письмового тексту. До того ж, був обраний підхід, який, завдяки своїй швидкості, дозволяє проводити аналіз емоційного забарвлення аудіо у реальному часі, а саме класифікація спектральних фото аудіо за допомогою CNN.

Практичне значення одержаних результатів полягає у тому, що було аргументовано можливість обробки даних та аналізу емоційної складової голосу у реальному час.

РОЗДІЛ 1 КОНЦЕПЦІЇ, ЩО СТОСУЮТЬСЯ ЗАСТОСУВАННЯ АЛГОРИТМІВ МАШИННОГО НАВЧАННЯ ДЛЯ ВИЯВЛЕННЯ ЕМОЦІЙНОГО ЗАБАРВЛЕННЯ ГОЛОСУ

Аналіз настроїв – це техніка для аналізу емоцій, що містяться в текстах. Більшість досліджень аналізу настроїв спрямовані на тексти, що містять суб'єктивні думки, такі як твіти в Twitter та оглядові документи.

Багато з цих досліджень класифікують емоції з точки зору письменників. З іншого боку, дослідження новинних статей, які в основному стосуються об'єктивних подій, класифікують емоції з точки зору читача. Наприклад, Lin et al. поділили емоції читачів щодо новин на щасливі, гнівні, сумні, здивовані, сердечні, приголомшливі, нудні та корисні. Вони припустили, що найпоширеніша емоція, обрана користувачами, які читали статтю, є правильною відповіддю, і визначили емоції новинних статей за допомогою SVM [1]. Лі та ін. класифікував емоції читача на зворушливість, співпереживання, нудьгу, гнів, розвагу, смуток, здивування та теплоту. [2]

Вони сформулювали проблему аналізу настроїв як проблему класифікації кількох міток і запропонували тематичну модель для оцінки ваги різних документів для кожної емоції. Циптаді та ін. класифікував емоції читачів на горді, гнівні, сумні, радісні, налякані, веселі, натхненні та здивовані. Вони показали, що ефективність класифікації наївного байєсового класифікатора та логістичної регресії була покращена шляхом застосування методу передвибірки SMOTE, щоб полегшити проблему незбалансованих даних. На додаток до досліджень, що класифікують цілі новинні статті, деякі дослідження класифікували заголовки новинних статей, а інші класифікували речення статей новин. [3]

У дослідженнях щодо класифікації речень, наприклад, Бховмік. класифікував емоції читачів на гнів, огиду, страх, щастя, смуток і здивування. Вони попросили кількох людей коментувати речення 5015 новинних статей. Вони підтвердили, що рівень згоди можна покращити, усунувши здивування, а також інтегрувавши гнів і огиду. Крім того, вони оцінили ефективність класифікації з декількома мітками за ADTboost:MH. [4]

Аналогічно, виключення здивування та інтеграція гніву та відрази покращило продуктивність моделі. Лі та ін. змоделювали залежність міток, припускаючи, що одне речення, ймовірно, матиме подібні мітки, такі як ненависть і гнів, а контекстну залежність речень, припускаючи, що речення в одному контексті, ймовірно, будуть мати однакові мітки, використовуючи факторний графік.

Найкращу продуктивність показала модель із використанням двох сусідніх речень, а не документа чи абзацу як контексту. Хоча це було дослідження оглядових документів, Zhang сформулював проблему аналізу настроїв як проблему маркування послідовності речень. Вони запропонували метод визначення етикеток емоцій (позитивних, негативних і нейтральних) за допомогою CRF. З експериментальних результатів активного навчання метод маркування документа з найменшою середньою ймовірністю першої половини речень у документі найбільш ефективно покращив продуктивність моделі.

Останніми роками в області обробки природної мови привертають увагу підходи до тонкого налаштування мовної моделі, попередньо вивченої з величезними немаркованими текстовими даними. BERT є типовим методом передтренінгових мовних репрезентацій. Ефективність методу з використанням BERT також була підтверджена в задачі аналізу настроїв. Однак ці моделі ідентифікують емоцію цілого огляду або окремого речення.

Розглядаючи класифікацію емоцій у кожному реченні новинної статті, необхідно враховувати не лише кожне речення, а й контекстну інформацію навколо них. Більшість традиційних досліджень з аналізу настроїв для новинних статей класифікують емоції з більшою деталізації з точки зору читачів.

1.1 Аналіз настроїв на основі тексту

Аналіз настроїв зосереджений насамперед на обробці тексту і в основному складається з класифікаторів на основі правил, які використовують великі лексикони настроїв, або методів, керовані даними, які припускають наявність великих анотованих корпусів. Лексика сентиментів – це перелік лексичних ознак (наприклад, слів), які зазвичай позначаються відповідно до їх семантичної орієнтації як позитивні чи негативні. Широко використовувані лексикони включають лексикони на основі бінарної полярності, такі як Harvard General Inquirer, Linguistic Inquiry і Word Count, а також лексикони на основі валентності, такі як AFINN, SentiWordNet і SnticNet. Використовуючи цю лексику, дослідники можуть застосовувати власні правила або використовувати існуюче моделювання на основі правил, наприклад VADER, для аналізу настроїв. [5]

Однією великою перевагою моделі, заснованої на правилах, є те, що ці підходи не вимагають навчальних даних і узагальнюються на декілька доменів. Однак, оскільки слова анотуються на основі їх безконтекстної семантичної орієнтації, може виникнути неоднозначність слів, коли слово має кілька значень. Наприклад, такі слова, як «переможений» та «приголомшений», класифікуються як «позитивні» в Bing, а у AFINN є негативними. Хоча алгоритм, заснований на правилах, як відомо, є шумним і обмеженим, лексикон настроїв є корисним компонентом для будь-якого складного алгоритму виявлення настроїв і є одним з основних ресурсів, з яких можна почати.

Інший основний напрямок роботи в аналізі настроїв складається з методів, керованих даними, заснованих на великому наборі даних, анотованих для полярності. Найбільш широко використовувані набори даних

включають корпус MPQA, який являє собою колекцію анотованих вручну статей новин, оглядів фільмів із бінароною полярністю, колекцію газетних заголовків, анотованих для полярності. З великими анотованими наборами даних були застосовані контрольовані класифікатори. Такі підходи відходять від сліпого використання ключових слів і підрахунку спільного використання слів, а скоріше покладаються на неявні особливості, пов’язані з великими семантичними базами знань.

1.2 Аналіз настроїв на основі аудіо

Вокальна експресія є основним носієм афективних сигналів у людському спілкуванні. Мовлення як сигнали містить кілька функцій, які можуть витягувати лінгвістичну, специфічну для мовця інформацію та емоційну. У цьому розділі розглядається пов’язана робота щодо аналізу настроїв на основі аудіо та мультимодального з’єднання. Дослідження аналізу настроїв на основі мовлення були зосереджені на визначенні відповідних акустичних особливостей. Може бути використане програмне забезпечення з відкритим вихідним кодом, таке як OpenEAR, openSMILE, JAudio інструментарій або пакети бібліотек, щоб отримати функції. Ці ознаки разом із деякими їх статистичними похідними тісно пов’язані з вокальними просодичними характеристиками, такими як тон, гучність, висота, інтонація, флексія, тривалість тощо. [6]

Контрольовані чи неконтрольовані класифікатори можуть бути встановлені на основі статистичні похідні від цих ознак. Моделі послідовності можуть бути встановлені на основі банків фільтрів, кепстральних коефіцієнтів Mel-frequency (MFCC) або інших низькорівневих дескрипторів, виділених із необробленого мовлення без розробки функцій. Однак цей підхід зазвичай вимагає високоефективних обчислень і великих анотованих аудіофайлів. Мультимодальний аналіз настроїв почав привертати увагу нещодавно через необмежене мультимодальне джерело інформації в Інтернеті, наприклад відео та аудіо.

Більша частина мультимодального аналізу настроїв сьогодні зосереджена на монологічних відео. В останні кілька років розпізнавання настроїв у розмові почало набувати дослідницького інтересу, оскільки

відтворення людської взаємодії вимагає глибокого розуміння розмови, а настрої відіграють ключову роль у розмові.

Існуючі набори даних розмови зазвичай записуються в контрольованому середовищі, наприклад в лабораторії, і сегментуються на висловлювання, транскрибуються в текст і коментуються мітками емоцій або настроїв вручну. Широко використовуваний набір даних включає AMI Meeting Corpus, IEMOCAP, SEMAINE та AVEC. Нещодавно було розроблено кілька моделей рекурентної нейронної мережі (RNN) для виявлення емоцій у розмовах, напр. Діалог-RNN або ICON. Однак вони менш точні у виявленні емоцій для висловлювань з емоційним зсувом, а дані навчання вимагають інформації про мовця. [7]

Моделі розмов не використовуються в нашому аналізі настроїв полярності через якість даних і підхід, який використовується для отримання навчальних даних. В основі будь-якого мультимодального механізму аналізу настроїв лежить мультимодальний синтез. Мультимодальне злиття об’єднує всі окремі модальності в єдине об’єднане уявлення. Характеристики витягуються з даних кожної модальності незалежно. Об’єднання на рівні рішення подає особливості кожної модальності в окремі класифікатори, а потім об’єднує їхні рішення. Об’єднання на рівні ознак об’єднує вектори ознак, отримані з усіх модальностей, і передає отриманий довгий вектор у контрольований класифікатор. Нещодавні дослідження мультимодального злиття для розпізнавання настроїв були проведені або на рівні функції, або на рівні рішення.

1.3 Важливість SER у нашому житті

Розпізнавання емоцій голосу, або speech emotion recognition (SER) — це завдання розпізнавання емоційних аспектів мовлення незалежно від семантичного змісту. Хоча люди можуть ефективно виконувати це завдання як природну частину мовного спілкування, здатність виконувати це автоматично за допомогою програмованих пристроїв все ще є предметом досліджень.

Дослідження систем автоматичного розпізнавання емоцій спрямовані на створення ефективних методів виявлення емоцій у режимі реального часу користувачів мобільних телефонів, операторів колл-центру та клієнтів, водіїв автомобілів, пілотів та багатьох інших користувачів спілкування «людина-машина». Додавання емоцій до машин було визнано критичним фактором, що змушує машини виглядати та діяти схоже на людину. [8]

Традиційно машинне навчання (ML) передбачає обчислення ознак (features) із необроблених даних (наприклад, мова, зображення, відео, ЕКГ, ЕЕГ). Ознаки використовуються для навчання моделі, яка вчиться класифікувати їх. Поширеною проблемою, з якою стикається цей підхід, є вибір цих ознак. Загалом невідомо, які ознаки можуть призвести до найбільш ефективної класифікації даних у різні категорії (або класи). Деяке розуміння можна отримати, протестувавши велику кількість різних ознак, об’єднавши різні ознаки в загальний вектор ознак або застосувавши різні методи вибору ознак. Якість створених вручну елементів може мати значний вплив на продуктивність класифікації.

Завдяки появі класифікаторів на базі глибоких нейронних мереж (DNN) було запропоновано елегантне рішення, яке обходить проблему оптимального вибору ознак. Ідея полягає у використанні мережі, яка приймає

необроблені дані як вхідні дані та генерує мітку класу як вихідні дані. Немає потреби ні обирати вручну ознаки, ні визначати, які параметри є оптимальними з точки зору класифікації. Ціною цього дуже зручного рішення є набагато більші вимоги до мічених зразків даних порівняно зі звичайними методами класифікації.

У багатьох випадках, і це включає SER, лише мінімальні дані доступні для цілей навчання. Як показано в цьому дослідженні, проблему обмежених навчальних даних значною мірою можна подолати підходом, відомим як трансферне навчання. Воно використовує існуючу мережу, попередньо навчену на великих даних, для вирішення загальної проблеми класифікації. Потім ця мережа додатково навчається з використанням невеликої кількості доступних даних для вирішення більш конкретного завдання.

Враховуючи, що на даний момент найпотужніші попередньо навчені нейронні мережі навчені для класифікації зображень, щоб застосувати ці мережі до проблеми SER, сигнал голосу потрібно перетворити у формат зображення [9]. Це дослідження описує кроки, пов’язані з переходом від голосу до зображення; у ньому пояснюються процедури навчання та тестування, а також умови, які необхідно виконати, щоб досягти розпізнавання емоцій у режимі реального часу з безперервної потокової мови.

1.4 Метод LVA

Найбільш відомий метод класифікації емоційного забарвлення голосу був розроблений Ізраїльською компанією Nemesysco і називається LVA (Layered Voice Analysis). Цей метод був розроблений для кількісної та якісної оцінки емоційної та розумової напруги в голосі людини. Наразі він використовується для виявлення особливостей емоційного реагування людини під час проведення інтерв'ю. Дана технологія може бути використана як додатковий інструментарій для визначення ступеня достовірності інформації, що подається респондентом, оцінки ризиків, пов'язаних з людським фактором.

Технологію LVA винайшов молодий ізраїльський учений Амір Ліберман у 1997 році. Створене ним програмне забезпечення постійно допрацьовується у співпраці з фахівцями з різних галузей (психології, психофізіології, криміналістики, фонетики, акустики, психіатрії та ін.) та адаптується до ринків різних країн. Верифікація та апробація даного методу проводились у різних країнах поліграфологами, нейрофізіологами, психіатрами, психологами.

Технологія LVA дозволяє досліджувати різні психічні стани людини, фіксуючи приховані емоційні сигнали та моменти підвищеного когнітивного навантаження, що виявляються у його голосі. Ця інформація дає уявлення у тому, наскільки напружено думає респондент у відповіді певні питання, що його турбує, які теми є йому чутливими. Своїх результатів метод досягає, використовуючи авторські методи статистичного аналізу безлічі вокальних параметрів, що класифікуються програмним забезпеченням як показники стресу, хвилювання, збентеження, невпевненості, прагнення приховати якусь інформацію.

Існує кілька варіантів програмного забезпечення, орієнтованого на конкретні практичні області QA5, RA7, LVA-і та інші. Деякі з цих систем ґрунтуються на тому, що респондентові задаються спеціально підготовлені питання, а комп'ютер реєструє та аналізує його голосові реакції. Інші системи аналізують весь голосовий трафік при вільній (не структурованій) розмові та виділяють важливі (за заздалегідь визначеними критеріями) розмови. Ці системи застосовуються в контактних центрах, мережах загального користування, в силових структурах. Важливо, що це метод дозволяє оцінювати довільно не контрольовані людиною голосові реакції. Для його реалізації абсолютно не важливо, якою мовою і що говорить людина, важливо – як! Тому цей метод практично не має мовних та культурних обмежень. Звичайно, для кожної країни потрібно розробляти питання конкретною мовою. [10]

Звичайно, що алгоритм цього методу є таємницею компанії, тому його не можна опробувати чи протестувати.

1.5 RNN у SER

Звуковий сигнал є нічим іншим, як послідовністю чисел, через що логічно буде його аналізувати за допомогою рекурентних нейронних мереж, а саме LSTM та GRU.

LSTM (Long Short-Term Memory) — це тип рекурентної нейронної мережі, яка зазвичай використовується для таких завдань, як перетворення тексту в мовлення або обробка природної мови. Вони мають повторюваний стан, який оновлюється кожного разу, коли нові дані надходять через мережу. Таким чином, LSTM має пам'ять.

GRU (Gated Recurrent Unit), у свою чергу, є вдосконаленою версією стандартної рекурентної нейронної мережі. Але щоб вирішити проблему зникнення градієнта стандартної RNN, GRU використовує, так звані, ворота оновлення та скидання. По суті, це два вектори, які вирішують, яку інформацію слід передати на вихід. Їхня особливість полягає в тому, що їх можна навчити зберігати інформацію яка проходила через них раніше і не має прямого відношення до прогнозу, не видаляючи її.

Хоча більшість проведених досліджень зосереджено на отриманні емоційно релевантної інформації незалежно від висловлювань, мережі пам’яті на основі RNN із механізмами привернення уваги були запропоновані та успішно використані для захоплення історичних аспектів розмови та запиту до банку пам’яті для отримання відповідної інформації, необхідної для виявлення емоцій [11]. Завдяки повторюваній структурі моделей RNN, які обробляють вхідні дані, моделі RNN останнім часом були першим вибором серед дослідників для завдань моделювання послідовності, таких як розпізнавання мовлення та прогнозування емоцій, порівняно зі звичайними прихованими моделями Маркова (HMM). Крім того, доведено, що покращені

варіанти RNN, включаючи комірки LSTM і GRU, здатні передавати коротко- та довгострокову контекстну інформацію під час розмови [12]. Таким чином, властива моделям RNN з осередками пам’яті здатність відстежувати стан пам’яті в послідовних даних дає вагомі підстави для включення моделей RNN у розпізнавання емоцій багатосторонніх розмов для кращого використання контекстної інформації в розмовах. Оскільки на людську природу природним чином впливають різні емоції, які виникають у контексті розмови, важливо, щоб модель демонструвала здатність виявляти складні емоції зі значною точністю з кращим використанням контекстної інформації в розмовах. Нещодавня поява в дослідницькому співтоваристві фільтрації емоційно важливої інформації з висловлювань і введення розмовного контексту значно підвищила точність. У підході в [13] використовується мережа пам’яті на основі RNN, включаючи механізм привернення уваги з кількома стрибками, який вводить власний і міжособистісний вплив у глобальну пам’ять розмови, щоб отримати афективні підсумки контексту. DialogueRNN та його варіанти (BiDialogRNN, BiDialogRNN з увагою) — це нещодавно створені найсучасніші моделі, які використовуються для отримання прогнозів емоцій із розмов, враховуючи контекстну інформацію, враховуючи глобальний контекст розмови, стани мовця та стани емоцій за допомогою три окремі блоки Gated Recurrent Units (GRU) [14]. Незважаючи на те, що продуктивність моделі була досліджена з точки зору текстової модальності та тримодального сценарію (текст, аудіо, візуальний), не було повідомлено про роботу окремої продуктивності аудіомодальності.

З останніх досліджень, можна побачити, що наразі вдалося досягти непоганої точності за допомогою аналізу голосу рекурентними нейронними мережами [15]

Але так як нашим завданням є побудова системи, яка зможе проводити аналіз емоцій голосу у реальному часі - було використано інший підхід, а саме аналіз спектрограм голосу за допомогою CNN.

1.6 Висновки до першого розділу

Як видно з проаналізованих матеріалів, тема SER вже давно досліджується і є досить актуальною через її потребу у реальному житті та бізнесі. Наразі SER знаходить своє використання у таких сферах як: обробка запитів клієнтів, відтворення людського голосу, слідкування за емоціями та станом працівників небезпечних професій, поліцейська справа та аналіз співробітників на кшталт брехні. Були проведені дослідження і як для аналізу емоцій семантичним підходом, тобто через аналіз фактичних слів, які були сказані, чи написані, так і для аналізу голосу через звуковий сигнал. Наразі найпоширенішим серед бізнесу є підхід LVA, розроблений Ізраїльською компанією Nemesysco, через його чудову репутацію. Але, у свою чергу, їх продукт є комерційним і його неможливо дослідити. Що стосується наукового світу, то там лідером у SER є RNN та її підвиди (LSTM та GRU). У цій роботі було обрано підхід до SER у реальному часі за допомогою аналізу спектрограм голосу за допомогою згорткових нейронних мереж.

РОЗДІЛ 2 ОБРОБКА ДАНИХ

Для нашого дослідження було переглянуто декілька наборів даних, але зупинилися на EMO-DB так як він відображає найбільш різноманітні записи як чоловічих, так і жіночих голосів, до того ж, усі актори різного віку. База даних EMO-DB — це вільнодоступна німецька емоційна база даних. База даних створена Інститутом комунікації Технічного університету, Берлін, Німеччина. У записі даних брали участь десять професійних спікерів (п'ять чоловіків і п'ять жінок). Всього в базі даних 390 висловлювань. База даних EMODB містить сім емоцій: 1) гнів; 2) нудьга; 3) тривожність; 4) щастя; 5) смуток; 6) огида; 7) нейтральний. Дані були записані з частотою дискретизації 48 кГц, а потім знижена до 16 кГц.

Кожен актор виявляв 7 емоцій для 10 різних висловлювань (5 коротких і 5 довгих) з емоційно нейтральним мовним змістом. У деяких записах доповідачі надавали більше однієї версії одного висловлювання. Після перевірки на основі тестів на аудіювання, проведених 10 оцінювачами, до бази даних було включено лише зразки мовлення, які отримали суб’єктивний рівень розпізнавання >80%. У таблиці 1 узагальнено вміст EMO-DB з точки зору кількості записаних зразків мовлення (висловлювань), загальної тривалості емоційного мовлення для кожної емоції та кількості згенерованих зображень спектрограми (RGB) для кожної емоції.

Таблиця 2.1. Інформація про набір даних.

Емоція	Кількість зразків	Загальна тривалість (сек.)	Кількість згенерованих зображень
Злість	129	335	27220
Нудьга	79	220	18125
Огида	38	127	11010
Страх	55	123	5463
Задоволення	58	152	12400
Нейтральна	78	184	14590
Сум	53	210	18455
Загально	390	1207	111425

Що стосується згорткової нейронної мережі, то було вирішено піти шляхом використання підходу transfer learning, тобто донавчання вже претренованної моделі, так як він зазвичай дає набагато кращі результати ніж навчання моделі з нуля (у випадках лімітованості у даних або ресурсах). За основу була взята модель AlexNet, а саме її реалізація у fastai.

2.1 Завантаження та обробка набору даних

Так як EMO-DB є відкритим набором даних, є дуже багато ресурсів, звідки його можна завантажити. Був обраний набір даних, який розміщений на Kaggle. [16]

Сам набір даних являє собою аудіофайли у яких в назві закодований актор, номер запису та емоція.

Враховуючи, що доступні обчислювальні ресурси були обмежені, і була доступна лише невелика база даних емоційно позначених зразків мовлення, метою було визначити обчислювально ефективний підхід, який міг би працювати з невеликим набором навчальних даних. Було розроблено систему SER у наступному вигляді. Для кожного блоку був розрахований масив спектрограм, який перетворювався у формат зображення RGB і передавався як вхідні дані до попередньо навченої CNN. Після відносно короткого навчання навчена CNN була готова класифікувати емоції з нерозміченої (потокової) мови, використовуючи той самий процес перетворення мови в зображення. У представлених тут експериментах продуктивність SER перевірялася з використанням двох різних частот дискретизації (16 і 8 кГц) і процедури компандування μ-law. Система SER була реалізована за допомогою мови програмування Python та робочою станцією слугував Google Colab Pro із відеокартою NVIDIA® Tesla® K80 та 12 ГБ оперативної пам’яті та інколи NVIDIA® Tesla® P100 та 24 ГБ оперативної пам’яті, так як ресурси на цьому сервісі можуть змінюватися з часом.

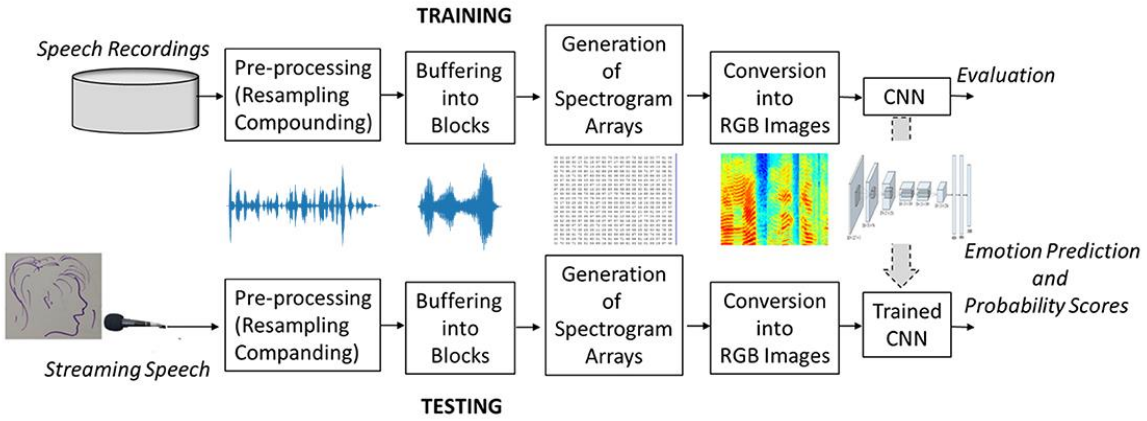


Рисунок 2.1. Послідовність дій у системі SER, яка описана у цій роботі

У традиційних вузькосмугових системах передачі даних смуга пропускання мовного сигналу була обмежена, щоб збільшити швидкість передачі. У телефонії, наприклад, діапазон частот мови раніше обмежувався діапазоном від 300 Гц до 3,4 кГц. Цього було достатньо, щоб забезпечити базовий рівень розбірливості мови, але ціною високої якості голосу. Цілком ймовірно, що таке серйозне скорочення пропускної здатності призвело до значного зменшення емоційної інформації, що передається мовцями.

Щоб перевірити цю можливість, модель глибокого навчання, яка використовується у цій роботі, буде навчено з двома різними частотами дискретизації: початкова 16 кГц відповідає широкій смузі пропускання 8 кГц, а зменшена частота дискретизації 8 кГц відповідає вузькій смузі частот 4 кГц. Також систему SER було навчано з u-law компандуванням та без нього.

U-law, μ-law або mu-law компандування — стандартне стиснення сигналу в цифровому телекомунікації. Це одна з двох стандартних версій G.711. Цей алгоритм компандування використовується в телекомунікаціях у

Північній Америці та Японії для оптимізації динамічного діапазону аналогового аудіосигналу перед його оцифруванням. [17]

Динамічний діапазон — це відношення найгучнішого звуку без спотворень до фонового шуму.

Це кодування зменшує динамічний діапазон сигналу, а отже, підвищує ефективність кодування та призводить до більшого співвідношення сигнал/спотворення, ніж лінійне кодування для даних бітів. Кодек u-law стискає звуки, такі як людська мова або інші цифрові сигнали, до 8 біт під час передачі в телекомунікаційній системі – системі телефонії. Це забезпечує чіткіші звуки, зберігаючи той самий приблизний рівень шуму.

Алгоритм u-law використовується як у старих аналогових, так і в нових цифрових системах. В аналогових системах він використовується після отримання звуку цифровою комп'ютерною системою. Ця зміна здійснюється за допомогою нелінійного підсилювача посилення. Якщо сигнал вже є цифровим, немає потреби його додатково стискати, оскільки 8-бітний розмір файлу даних є ідеальним розміром для цифрового файлу, і більшість комп'ютерів розпізнає його за розміром символу.

Цей алгоритм використовується в деяких стандартних мовах програмування, які використовують його для створення та зберігання звуків

Враховуючи оригінальні зразки мовлення x, компресія відрізків голосу F(x) була розрахована як

$$F(x) = \frac{\ln(1+\mu|x|)}{\log(1+\mu)^{sgn(x)}}$$

Тоді як реконструйовані мовні зразки x˜ були розраховані як

$$\tilde{x} = F^{-1}\left(F\left(x\right)\right) = sgn\left(F\left(x\right)\right)\frac{\left((1+\mu)^{|F(x)|}-1\right)}{\mu}$$

Значення параметра стиснення μ було встановлено на 255 (стандарт у США та Японії) [18].

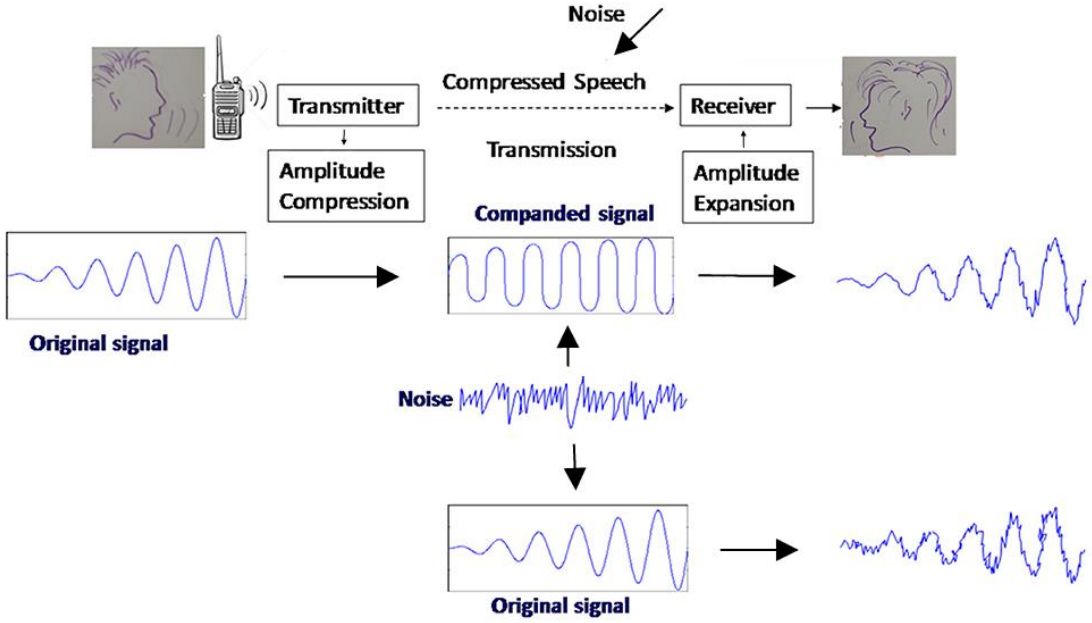


Рисунок 2.2. Вплив u-law компандування на шум.

Потокове або записане мовлення було буферизовано в блоки по 1 с для проведення поблочної обробки. Між наступними блоками застосовувався короткий крок тривалістю 10 мс. Рівні амплітуди були нормалізовані в діапазоні від -1 до 1.

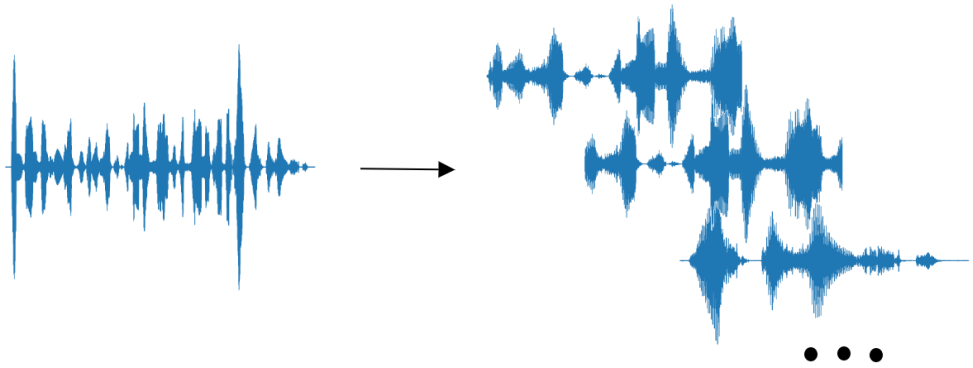


Рисунок 2.3. Розбиття аудіофайлу на блоки

Процедура, яка використовується для генерації масивів спектрограм, проілюстрована на рисунку 5. Короткочасне перетворення Фур'є було виконано для кожного 1-секундного блоку мовних сигналів.

Після чого результат короткочасного перетворення Фур'є перетворюється у спектрограму. Спектрограма - це спосіб візуального представлення гучності або амплітуди сигналу, оскільки вона змінюється з часом на різних частотах. Вісь у перетворюється на логарифмічний масштаб, а розмірність кольору перетворюється на децибели (це можна вважати логарифмічним масштабом амплітуди). Це тому, що люди можуть сприймати лише дуже малий і концентрований діапазон частот і амплітуд.

Дослідження показали, що люди не сприймають частоти в лінійному масштабі. Людина краще виявляє відмінності на нижчих частотах, ніж на вищих. Наприклад, вона може легко визначити різницю між 500 і 1000 Гц, але навряд чи зможе визначити різницю між 10 000 і 10 500 Гц, навіть якщо відстань між двома парами однакова.

У 1937 році Стівенс, Фолькманн і Ньюманн запропонували таку одиницю висоти, щоб однакові відстані у висоті звуку звучали однаково далеко для слухача. Це називається шкалою Мела. Виконується математична операцію над частотами, щоб перетворити їх у шкалу mel. [19]

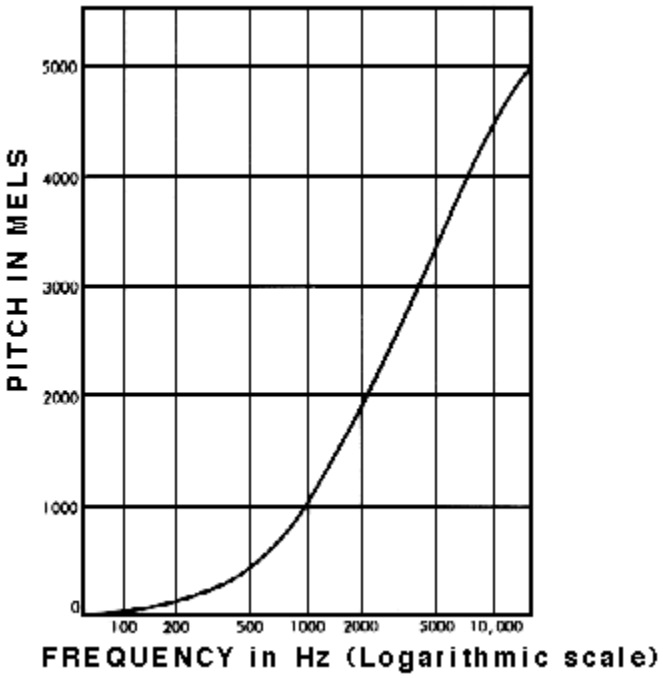


Рисунок 2.4. Mel шкала

Повний шлях, який проходить фрагмент голосу відображений на рисунку 5.

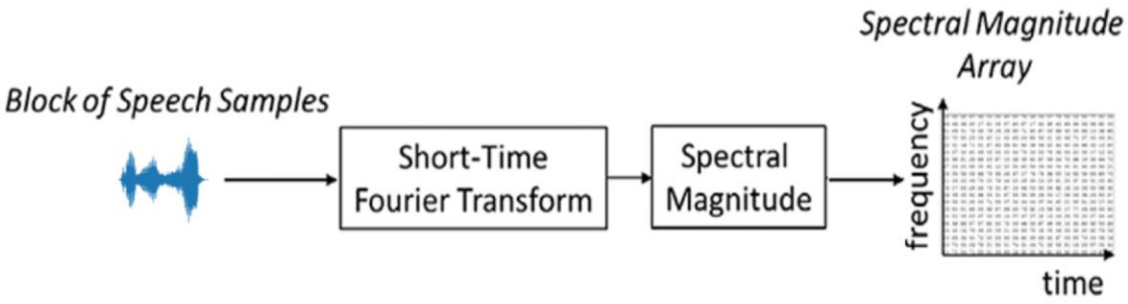


Рисунок 2.5. Генерація масивів спектрограм

Масиви спектральних величин 257 × 251 дійсних чисел були перетворені у формат кольорового зображення RGB, представлений трьома масивами кольорових компонентів

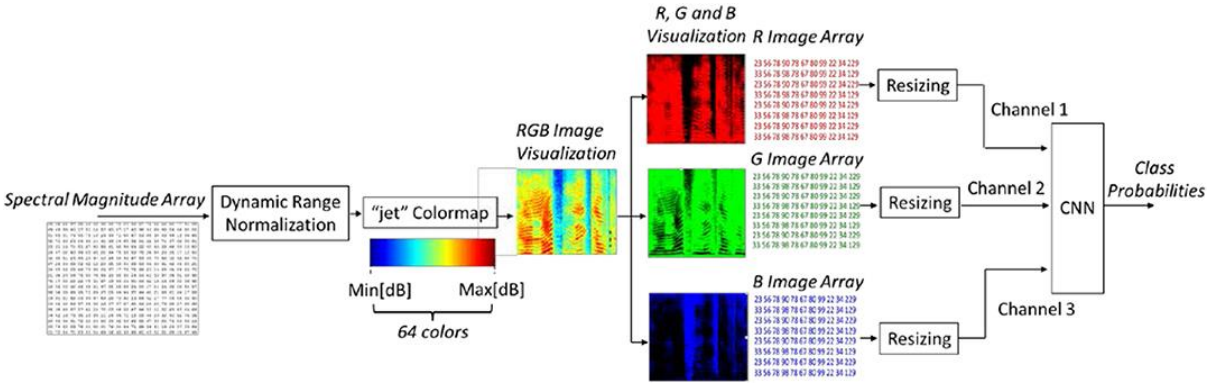


Рисунок 2.6. Перетворення масивів спектральних амплітуд у масиви зображень R, G та B.

Перетворення масивів спектральних амплітуд у RGB зображення відбувається за допомогою бібліотеки matplotlib та функції color_map. Після чого фотографія конвертується у об’єкт Image бібліотеки Pillow, яка обробляє

її та змінює її розмір до потрібних нам 256x256 пікселів. Після чого зображення розбиваються по папкам з назвою емоцію, яку вони відображають.

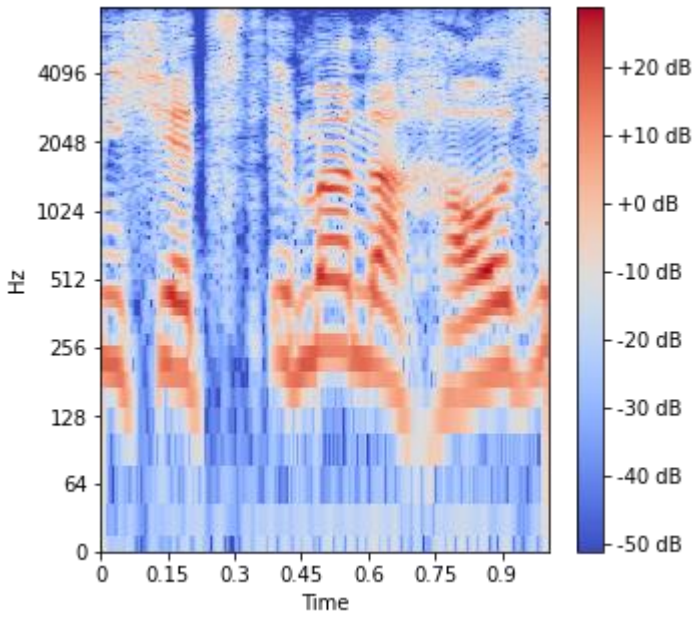


Рисунок 2.7. Приклад логарифмічної спектрограми

На рисунку 7 видно приклад, як звичніше “відчувати” звук людині, а саме у логарифмічному масштабі, де нижчі звуки набагато відчутніші, ніж звуки високих частот. Але для моделі глибокого навчання можна використати mel шкалу, що і було зроблено. Як приклад - один відрізок звуку з емоцією гнів у mel масштабі на рисунку 8.

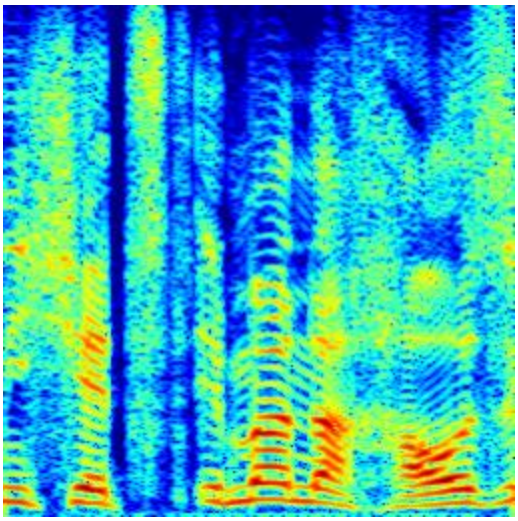


Рисунок 2.8. Приклад mel спектрограми

Для роботи моделі, нам потрібно було розташувати дані у правильному форматі, де кожне фото просортороване по відповідній папці.

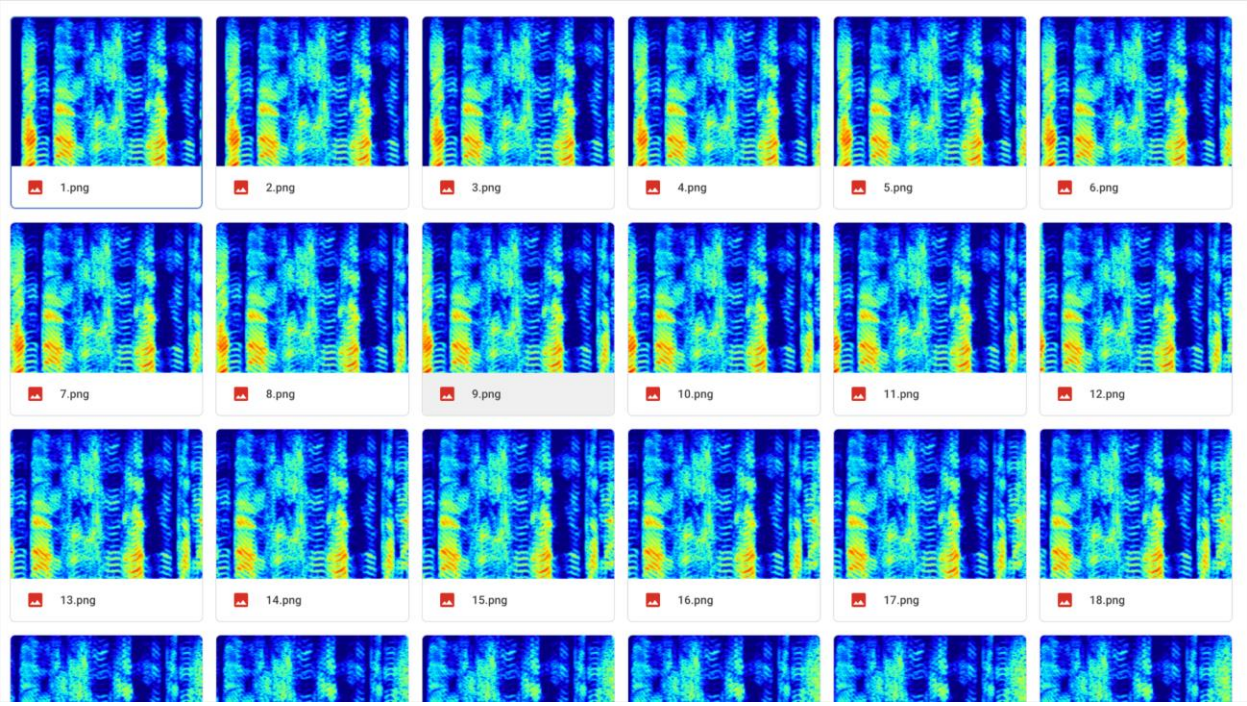


Рисунок 2.9. Дані у папці з емоцією Anger.

Такі набори даних були створені для 4-х експериментів:

- Експеримент 1: частота дискретизації 16 кГц (смуга пропускання = 8 кГц). Названня далі у роботі ЧД=16, СП=8, U-Law=No.
- Експеримент 2: частота дискретизації 8 кГц (смуга пропускання = 4 кГц). Названня далі у роботі ЧД=8, СП=4, U-Law=No.
- Експеримент 3: частота дискретизації 16 кГц, μ-law компандування. Названня далі у роботі ЧД=16, СП=8, U-Law=Yes.
- Експеримент 4: частота дискретизації 8 кГц, μ-law компандування. Названня далі у роботі ЧД=8, СП=4, U-Law=Yes.

Зниження частоти дискретизації було досягнуто відкиданням кожного другого запису у звуковій послідовності. Ця функціональність вже реалізована у бібліотеці librosa.

Що стосується μ-law компандування, то функція для нього була написана на основі формули, яка була зазначена вище.

Весь код обробки даних, від аудіофайлу до фотографій спектрограм аудіовідрізків довжиною 1с, може бути знайдений у Додатку А.

2.2 Висновки до другого розділу

Був опрацьований набір даних EMO-DB. Він є досить релевантним, так як наявні наступні позитивні ознаки: різноманіття віку акторів, наявність різних чоловічих та жіночих голосів, декілька варіацій запису одної емоції, які потім відсортовувалися незалежними респондентами та достатньо кількість записів.

Кожен запис проходив наступну обробку: Змінювалася частота дискретизація на 8кГц та застосовувалося u-law компандування (у двох експериментах з чотирьох), нарізався на блоки по 1 секунді з кроком у 10 мс, тобто з одного аудіо на 3 секунди виходило 300 фрагментів. Після чого до кожного фрагменту застосовувалося короткочасне перетворення Фур'є, результат якого перетворювався у масиви спектральних велечин, з яких вже створювалися RGB зображення. Весь набір даних склав більше ніж 111000 зображень, які були розділені на тренувальні та тестові дані.

РОЗДІЛ 3 НАВЧАННЯ МОДЕЛІ ТА ЕКСПЕРИМЕНТИ

Проведемо 4 експерименти, які були описані раніше. Декілька слів важливо сказати про те, навіщо вони. Усі використання SER у реальному житті пов’язані з різною якістю звуку. Як зазначалося раніше, у традиційних вузькосмугових системах передачі даних смуга пропускання мовного сигналу була обмежена, щоб збільшити швидкість передачі. Тобто аналіз голосу, записаного у звуковій студії буде істотно відрізнятися від аналізу голосу, який отримано під час телефонного зв’язку, або у випадку поганого з’єднання.

Що стосується u-law компандування, то воно використане для того аби перевірити можливість аналізу емоцій у голосі людини, якщо звук був додатково оброблений компресором.

Також нас цікавить не тільки якість класифікації емоцій, а й час, який піде на обробку звуку різної якості аби впевнитись у можливості роботи моделі у реальному часі.

3.1 Модель AlexNet та її навчання

AlexNet — це згорточна нейронна мережа (CNN), представлена Крижевським та ін. (2012). Його було попередньо навчено на понад 1,2 мільйонах зображень із набору даних ImageNet Стенфордського університету, щоб розрізняти 1000 категорій об’єктів. Він складається з 3-канального вхідного шару, що дозволяє вводити три 2-вимірні масиви, кожен розміром 256 × 256 пікселів. Після вхідного рівня йдуть п’ять згорткових шарів (Conv1-Conv5), кожен з яких має шар максимального об’єднання та нормалізації (рис. 6). Двовимірні вихідні об’єкти з останнього згорткового шару Conv5 перетворюються в одновимірні вектори та подаються в три повністю з’єднані шари (fc6-fc8). У той час як згорткові шари витягують характерні ознаки з вхідних даних, повністю пов’язані шари вивчають параметри моделі класифікації даних. Експоненціальна функція SoftMax відображає вихідні значення fc8 у нормалізований вектор дійсних значень, які потрапляють у діапазон [0,1] і в сумі дають 1. Ці значення надаються на вихідному рівні та представляють ймовірності кожного класу. Остаточна класифікаційна позначка присвоюється класу, який отримав найвищий бал вірогідності.

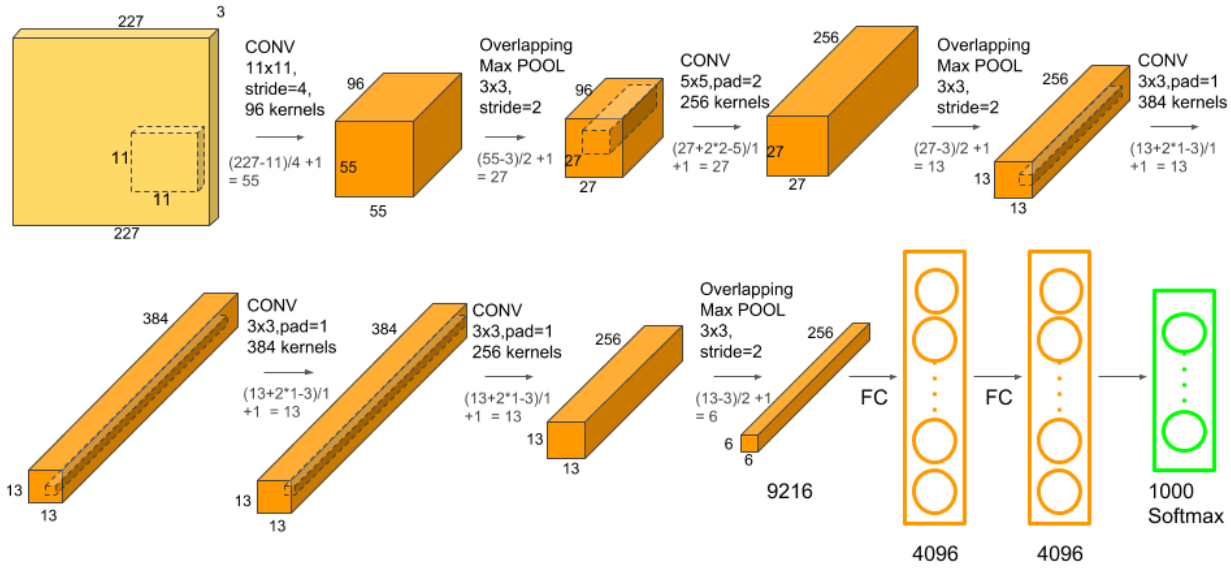


Рисунок 3.1. Структура AlexNet

Після адаптації до класифікації семи емоцій AlexNet було донавчено на позначених даних. Щоб досягти швидшого навчання в нових модифікованих шарах і повільнішого в старих шарах, початкова швидкість навчання (learning rate) була встановлена на невелике значення, а значення weight learning rate та bias learning rate були збільшені лише для кінцевих шарів. Оскільки мережа вже була попередньо навчена, процес навчання був набагато швидшим і “можливим” порівняно з тим, що потрібно було б під час навчання тієї самої структури мережі з нуля. Однак можливо, що за наявності необхідних ресурсів навчання з нуля могло б привести до кращих результатів. Хоча в останні роки AlexNet конкурували зі значно складнішими мережевими структурами [20].

У transfer learning процес навчання має на меті досягти найвищого впливу навчання на кінцевих, повністю пов’язаних рівнях мережі, залишаючи попередні рівні майже недоторканими. Процес навчання саме

останніх шарів, у той самий час залишаючи початкові шари недоторканими, незивається fine-tune.

Було взято реалізацію моделі на бібліотеці FastAI. Для донавчання були використані наступні гіперпараметри.

Таблиця 3.1. Гіперпараметри донавчання моделі

Гіперпараметр	Значення
Алгоритм оптимізації	SGDM
Розмір батчу	128
Кількість епох	80
Швидкість навчання	0.01

Для навчання дані були розбиті на 80/20, де 80% це тренувальні дані та 20% це тестувальні. Також автоматично 20% тренувальних даних були валідаційними і використовувалися моделлю під час навчання.

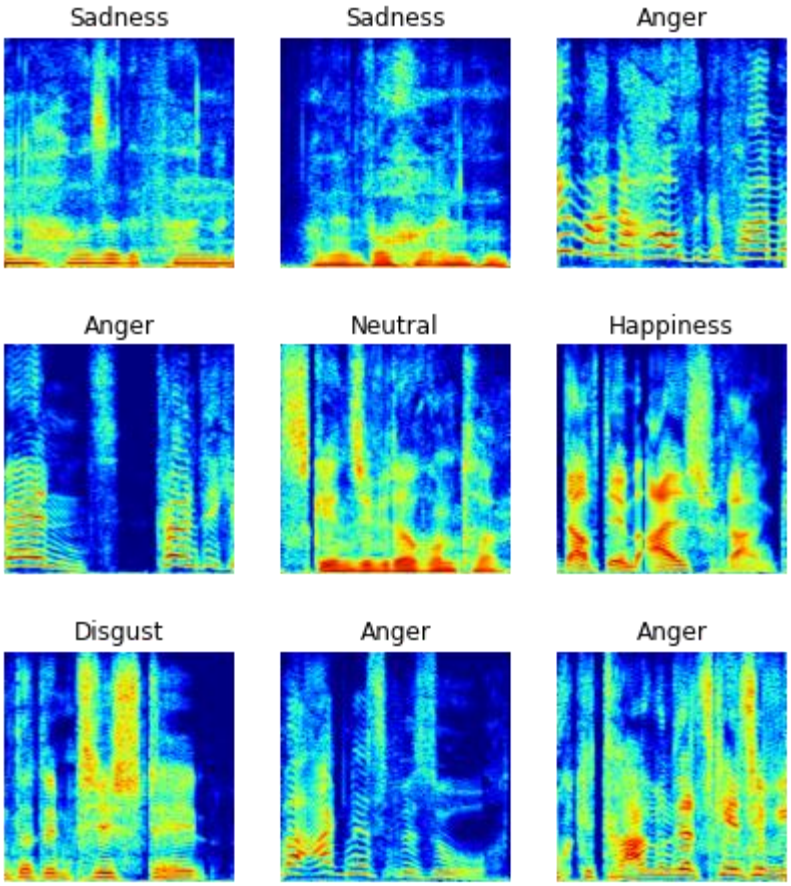


Рисунок 3.2. Приклад знімків з одного батчу.

Що стосується learning rate, то він був обраний за допомогою функції .find_lr() у бібліотеці FastAI. Логіка за цією функцією стоїть наступна - алгоритм проводить кілька ітерацій навчання. Починаючи з дуже низького початкового learning rate і змінюючи його в кожній міні-серії, поки не буде досягнуте дуже високого learning rate. Записується втрата (loss) на кожній ітерації, після чого обирається оптимальне значення learning rate.

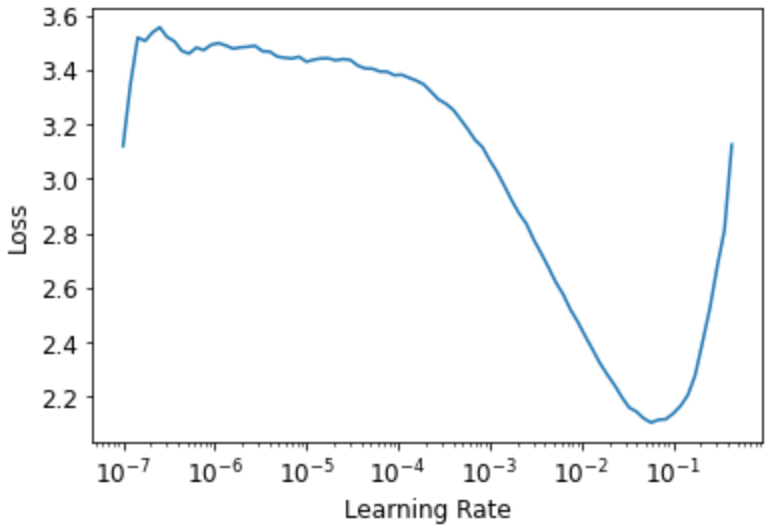


Рисунок 3.3. Пошук оптимального learning rate

Але хоч ітерації і є досить короткими, пошук оптимального learning rate зайняв доволі багато часу (1 годину). Проте, це значно заощадить нам час у майбутньому, так як нам не доведеться вручну підбирати це значення.

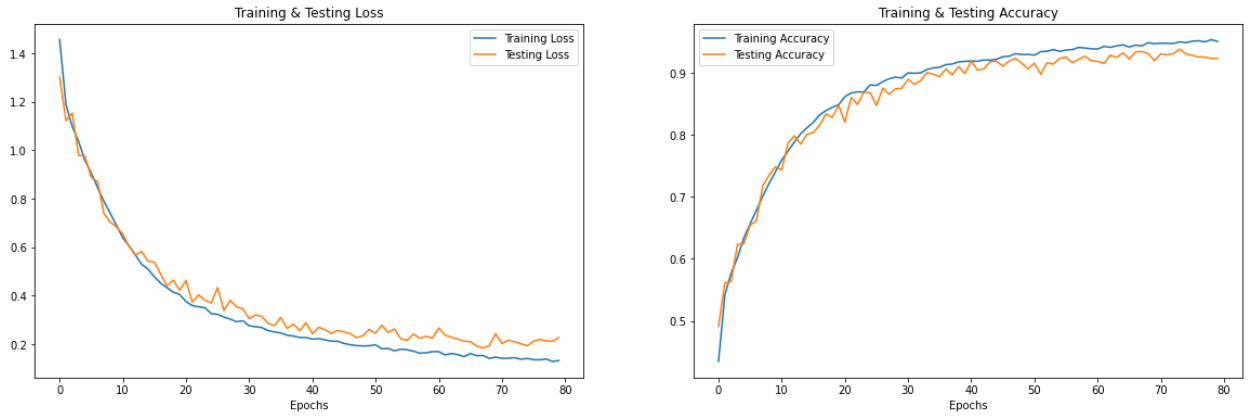


Рисунок 3.4. Навчання та валідація моделі.

На рисунку 14 зображено навчання моделі для даних з частотою дискретизації 16 кГц та без компандування u-law. Також були проведені навчання і 3-х інших моделей для 3-х інших експериментів.

Донавчання на 111000 зображень зайняло 8 годин, і тому transfer learning підхід є досить популярним у наш час, бо навчати усю модель з самого початку було б дуже ресурсомістко і по часу, і по потужностям системи. До того ж, завдяки тому, що згорткові шари, які займаються виявленням “ознак” на фото вже навчені, нам залишається тільки натренувати ваги останніх шарів, які й класифікують емоції.

Для запобігання перенавчання моделі було використано early stopping, принцип якого полягає у тому, що навчання зупиняється на моменті, коли зміна точності моделі на валідаційних даних стає меншою за виставлене значення. В нашому випадку значення було виставлене як min_delta=0.001.

Весь код для створення та тренування моделі може бути знайдений у Додатку Б.

3.2 Метрики оцінки моделей

Для оцінки якості нашої моделі використовуються наступні метрики precision, recall, accuracy та f-score. Їх розрахунок базується на таких поняттях як True Positive, True Negative, False Positive та False Negative:

	$y = 1$	$y = 0$
$\hat{y} = 1$	True Positive (TP)	False Positive (FP)
$\hat{y} = 0$	False Negative (FN)	True Negative (TN)

Рисунок 3.5. Таблиця співвідношень передбачених та реальних класів [21]

TP - це правильно класифіковані емоції. Наприклад якщо фактичне значення класу вказує на те, що емоція голосу - це “Страх” і передбачений клас говорить нам те саме.

TN - це правильно класифіковані емоції окрім класу “Страх”.

False Positive та False Negative виникають, коли фактичний клас суперечить передбачуваному класу.

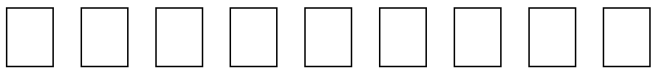
FP - коли фактичний клас - будь-який окрім “Страх”, а модель класифікує зображення, як “Страх”.

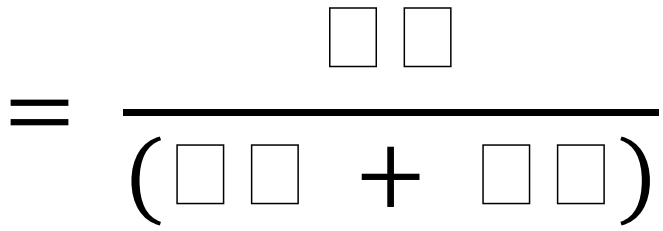
FN - Коли фактична емоція “Страх”, але передбачений клас - будь-яка інша емоція.

Метрики precision та recall базуються на значеннях True Positive, False Positive та False Negative.

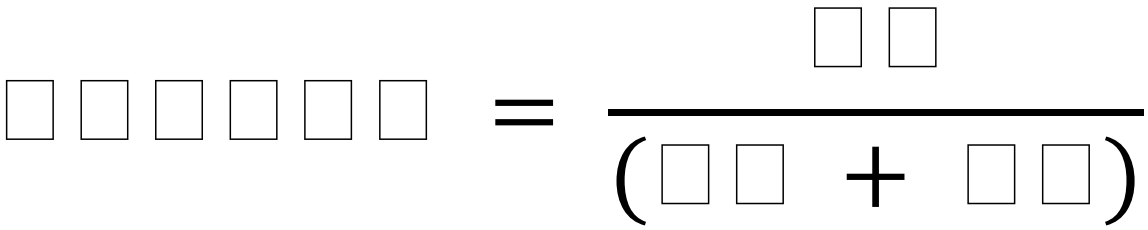
Precision - це відношення правильно передбачених класів до загальної кількості класів, які були передбачені. Дана метрика несе найбільшу інформативність у випадку, якщо важливо аби усі передбачення були точними і важливо уникати помилкових передбачень ціною втрачання деяких

вірних передбачень. Формула метрики наведена нижче.



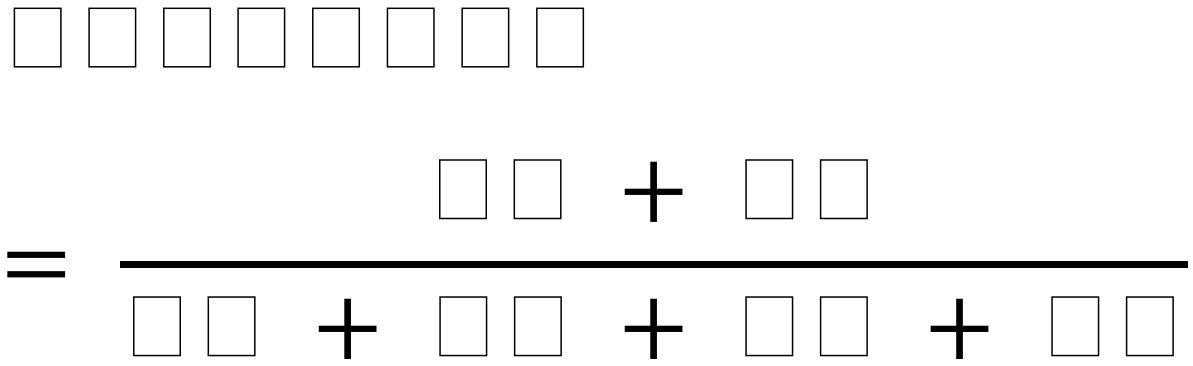


Recall - у свою чергу, це відношення вірно класифікованих значень одного класу до всіх вірно передбачених класів. Ця міра несе найбільшу користь у випадку, якщо нам не бажано пропустити усі інші класи окрім того, для якого рахується метрика. Найчастіше приклад важливості даної метрики описують через класифікацію людських хвороб: добре, якщо буде відправлено більше людей на додатковий огляд, класифікувавши у них хворобу, і погано, якщо буде пропущено людей у яких дійсно є хвороба, але класифікували їх стан як “здоровий”. Формула метрики наведена нижче.

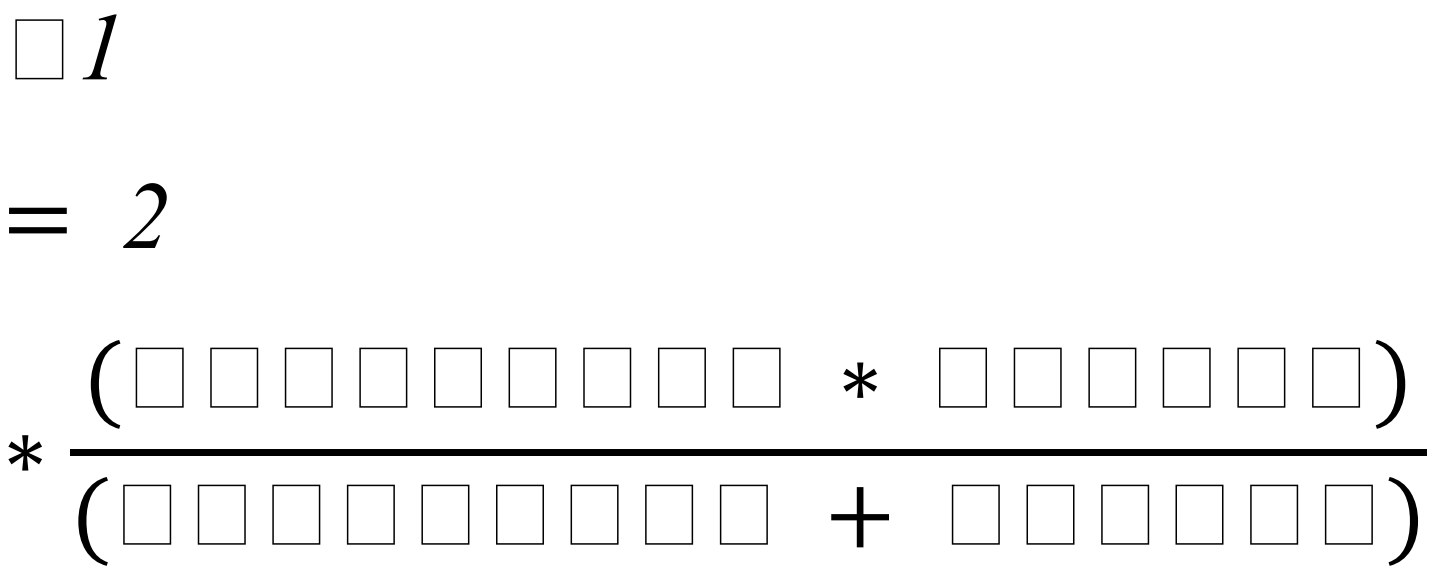


Accuracy - є найбільш інтуїтивно зрозумілою метрикою, і це відношення правильно передбачених спостережень до загальної кількості спостережень. Проте accuracy є релевантною метрикою лише у випадку, коли у нас є симетричні набори даних, де кількість класів майже однакова. Тому

нам доведеться також дивитися на інші параметри, щоб оцінити продуктивність нашої моделі.



F1 - визначається як середнє гармонійне між precision та recall. Таким чином, ця оцінка враховує як False Positive, так і False Negative результати. Інтуїтивно це не так легко зрозуміти, як precision чи accuracy, але F1 зазвичай більш корисний, ніж accuracy, особливо якщо у вас нерівномірний розподіл класів. Accuracy працює найкраще, якщо помилкові спрацьовування та помилково негативні результати мають однакову вартість. Якщо вартість False Positive і False Negative результатів сильно відрізняється, краще дивитися як на Precision, так і на Recall. [22]



3.3 Експерименти з порівнянням якості звуку

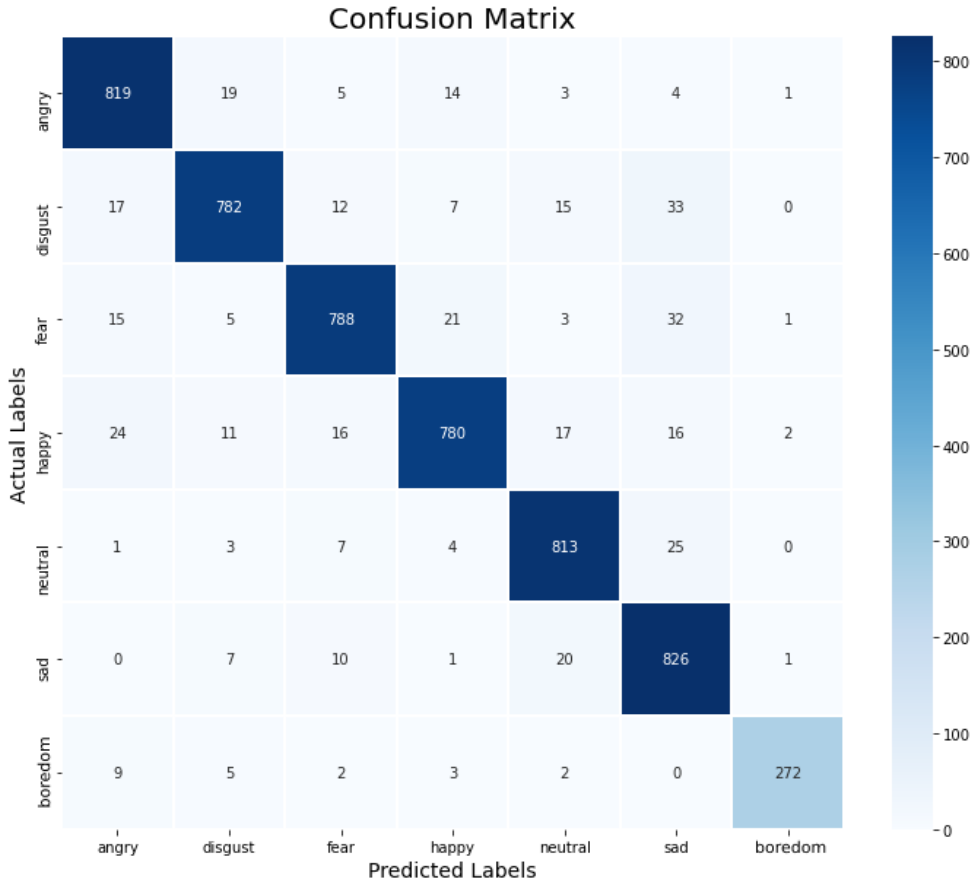


Рисунок 3.6. Confusion matrix для експерименту ЧД=16, СП=8, U-Law=No

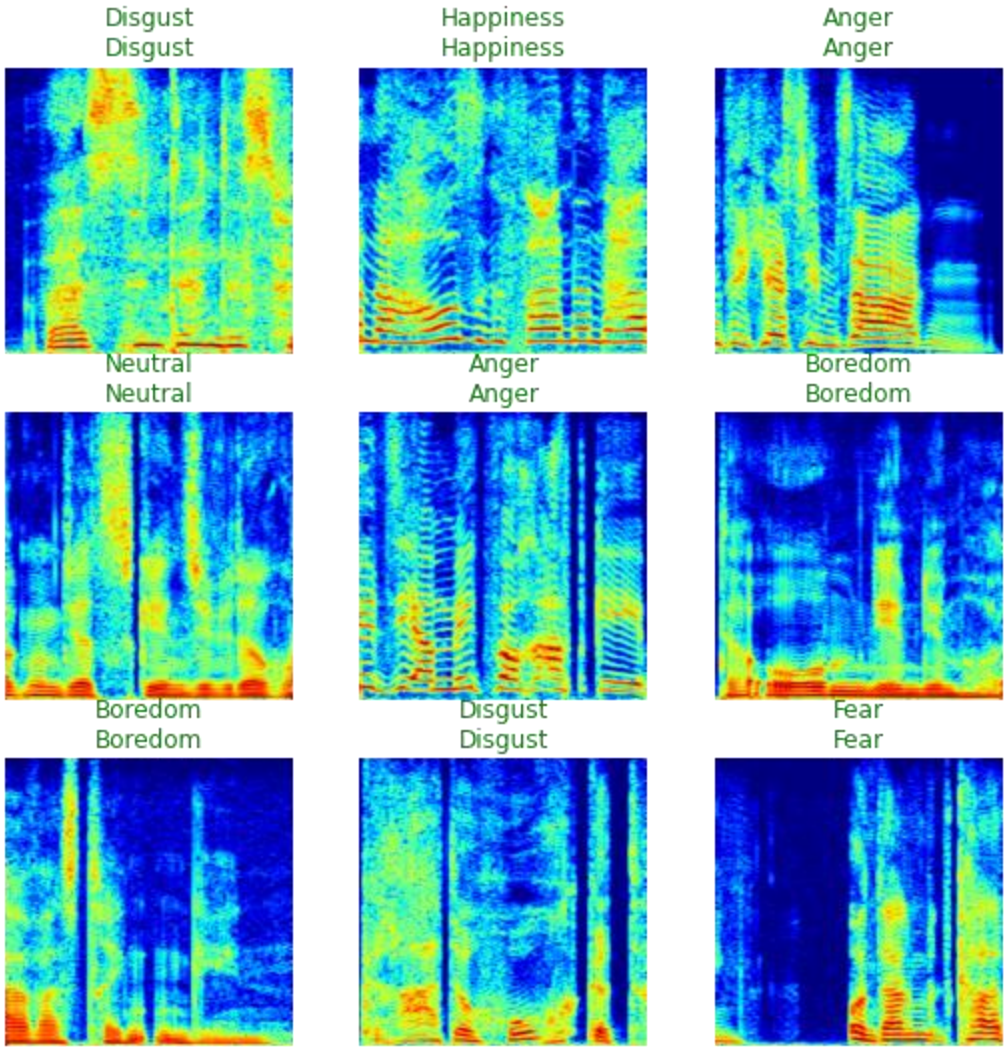


Рисунок 3.7. Приклад валідації класифікації моделі

Зразки даних тестування ніколи не використовувалися під час процедури навчання мережі. Експерименти не залежали від мовця та статі.

Метриками для оцінки виступили Accuracy, Precision, Recall та F-score.

Застосування SER в режимі реального часу було досягнуто шляхом поблочної обробки. Для кожного блоку була створена класифікаційна мітка, яка вказує на одну з семи категорій емоційного класу. Після навчання моделі були проведені наступні експерименти:

Таблиця 3.2. Опис експериментів

Експеримент	Частота дискретизації (kHz)	Смуга пропускання (kHz)	Компандування u-law
1	16	8	No
2	8	4	No
3	16	8	Yes
4	8	4	Yes

Для подальших назв експериментів вони будуть кодуватися як ЧД=16, СП=8, U-Law=Yes

Таблиця 3.3. Результати експериментів

Експеримент	Weighted precision (%)	Weighted recall (%)	Weighted F-score (%)	Weighted accuracy (%)
ЧД=16, СП=8, U-Law=No	90.33	90.9	89.5	90.56
ЧД=8, СП=4, U-Law=No	86.4	87.5	85.6	86.4

ЧД=16, СП=8, U-Law=Yes	85.27	86.73	84.7	85.275
ЧД=8, СП=4, U-Law=Yes	82.91	85.6	82.23	82.9

Як можна побачити, зниження якості звуку істотно впливає на якість класифікації натренованими моделями, хоча у будь-якому разу точність у 82% є досить істотною і у випадку значної різниці у швидкості обробки звуку - можна використовувати саме метод зі зниженою частотою дискретизації та компандуванням.

Таблиця 3.4. Середній час обчислення в мілісекундах (мс)

Експеримент	Час інференсу	Створення ознак			Загальний час
		Створення масивів спектрограм	Перетворення у RGB зображення	Загальний час створення ознак	
ЧД=16, СП=8, U-Law=No	18.7	8.0	3.6	11.6	30.3
ЧД=8, СП=4, U-Law=No	18.6	4.7	3.6	8.3	26.9

ЧД=16, СП=8, U-Law=Yes	18.4	7.5	3.6	11.1	29.5
ЧД=8, СП=4, U-Law=Yes	18.5	4.6	3.6	8.2	26.7

3.4 Висновки до третього розділу

Другою важливою складовою цієї роботи є, звісно, модель. Був обраний підхід саме донавчання моделі, замість повної її побудови та навчання з нуля, аби з обмеженими ресурсами досягти максимального результату у короткий термін. З декількох навчених згорткових нейронних мереж було обрано AlexNet, так як у нас вже був досвід її використання раніше і вона показувала чудові результати у задачах класифікації зображень. Принцип transfer learning полягає у тому, що залишаються вже навчені шари без змін, у даному випадку це згорткові шари, які відповідають за виявлення ознак; а ваги тренуємо саме у останніх шарах, які відповідають за класифікацію. Більшість гіперпараметрів було обрано згідно рекомендацій, які надають автори цієї моделі, окрім learning rate, який було визначено пробуючи різні значення на невеликих навчальних вибірках, та кількість епох, яка підбиралася згідно швидкості зміни функції втрат.

Задля оцінки якості навченого класифікатора, застосовувалося 4 метрики, а саме: presision, recall, F1-score та accuracy.

Навчання однієї моделі зайняло майже 8 годин, а підбір learning rate близько однієї години, але на виході отримано класифікатор з дуже високою точністю на валідаційних даних.

Була проведена оцінка точності всіх чотирьох моделей і відобразили їх результати у таблиці. Як можна побачити основна модель ЧД=16, СП=8, U-Law=No показала високу взважену точність у 90.56%. Валідація моделі проходила на даних, які до цього не використовувалися для тренування або валідації під час тренування моделі.

Можна побачити, що внесення змін у звукові дані, а саме зниження частоти дискретизації, смуги пропускання та застосування u-law

компандування негативно впливає на точність моделей, що було очікувано. Але мінімальна точність, яка була отримана для моделі ЧД=8, СП=8, U-Law=Yes, а саме 82.9% є істотною точністю, яка дозволяє розглядати використання моделі саме на таких даних, якщо це буде потрібно у зв’язку з різними проблемами передачі звуку між системами. Що стосується різниці у часу інференсу для даних різної якості, то вона не є критичною, а саме 4 мс, тому для побудови нашого програмного забезпечення було вибрано саме модель ЧД=16, СП=8, U-Law=No.

РОЗДІЛ 4 РОЗРОБКА СТАРТАП ПРОЄКТУ

Задум стартап проєкту полягає у створенні мобільного додатку, який може бути використаний будь-ким через завантаження на смартфон. Його основною функціональністю буде під’єднання до мікрофону та аудіо користувача і миттєвий аналіз емоційної складової голосу, який проходить через ці канали. Також, по закінченню запису буде доступна статистика з відсотками кожної емоції за час аудіо чи розмови.

Додаток планується під дві системи: Android (з розробкою на мові Kotlin), та Apple (розроблений на мові Swift). Прототип програмного забезпечення буде розроблений на мові програмування Python.

4.1 Розробка програмного забезпечення

Задля створення програмного забезпечення на основі моделі, яка була навчена раніше, нам потрібно описати основні завдання, які воно буде виконувати.

Головною задачею програмного забезпечення буде приймати звуковий сигнал з мікрофона, обробляти його і на виході видавати класифіковану емоцію у реальному часі.

На рисунку №№ ви можете побачити діаграму, яка відображає усі мікросервіси нашого програмного забезпечення.

Воно було реалізоване на мові програмування Python, а моделлю слугувала донавчена нейронна мережа AlexNet, яка була створена для класифікації звуку з частотою дискретизації у 16 кГц, смугою пропускання 8 кГц та без u-law компандування.

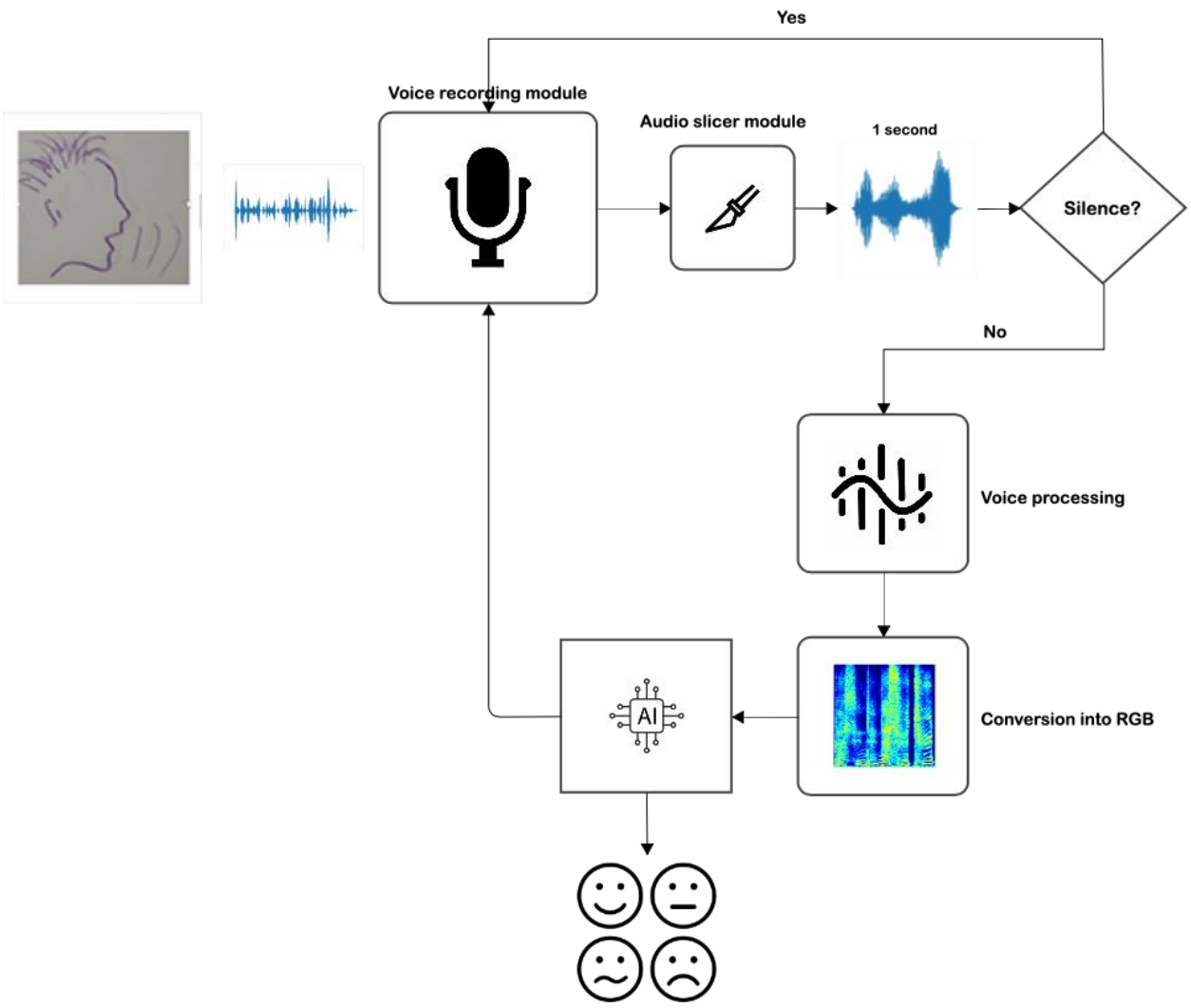


Рисунок 4.1. Діаграма програмного забезпечення

Саме програмне забезпечення було модульно реалізовано на персональному комп’ютері з наступними характеристиками: CPU Intel Core i7-9870H 2.6 GHz, RAM 16GB 2667 MHz, GPU Radeon 5300M 4GB.

У нашому програмному забезпеченні було використано наступні бібліотеки:

- fastai - для завантаження та використання нетренованої моделі.
- librosa - для обробки звуку.
- numpy - для роботи з масивами чисел.

- pyaudio - для підключення до мікрофону комп’ютера і зчитування звуку.
- matplotlib.pyplot - для перетворення спектрограми на RGB зображення.

По стандарту мікрофон записує аудіо з частотою дискредитації у 44100 Hz, тому для моделі перед обробкою звуку понижується його частоту дискретизації до 16000 Hz.

Після запуску програмного забезпечення і надання дозволу на використання мікрофону, програма починає зчитувати аудіо сигнал. Він ділиться на частини по 1 секунді. Після чого над ними проводиться обробка, яка була описана у цій роботі, а саме приміняється stft та створюється mel спектрограма. Спектрограма перетворюється у RGB зображення за допомогою color_map від matplotlib. Саме зображення вже подається на вхід до моделі, яка на виході видає класифіковану емоцію.

На рисунку №№ ви можете побачити приклад запуску програми мною і класифікацію емоційного забарвлення мого голосу.

Рисунок 4.2. Робота програми класифікації емоцій у реальному часі

Проте важливо відмітити, що на процес інференсу уходить в середньому 112 мс, це пов’язано з роботою з файловою системою, адже перед направленням зображення у модель воно локально зберігається. Тому у

кінцевому результаті для кожної 1 секунди аудіо втрачається 0.1 секунда корисного звуку. Але це є допустимою втратою сигналу, так як істотно більша частина корисного звуку успішно аналізується системою.

4.2 Аналіз ринкової стратегії проекту

Далі у роботі розглянуто групи потенційних користувачів продукту і виділено цільових користувачів.

Таблиця 4.1 – Огляд цільових користувачів

№ п/п	Цільова група	Потреба у продукті	Попит в межі цільової аудиторії	Конкуренція у сфері	Легкість входу у сигмент
1	Підприємці	Низька	Низький	Низька	Важко
2	Великі компанії	Середня: можливість додаткового аналізу емоцій співробітників	Середня	Низька	Важко
3	Маленькі компанії.	Низька	Низький	Низька	Важко
4	Звичайні користувачі смартфонів	Велика	Середня	Низька	Середнє
Які цільові групи користувачів було обрано: 2, 4					

Було проаналізовано цільові групи користувачів продукту і за результатами аналізу було обрано групи для яких буде запропоновано

продукт. Також було обрано стратегію диференційованого маркетингу, яка полягає у одночасній роботі з декількома групами цільових користувачів.

Було сформовано стратегію розвитку для продукту. А саме постійно оновлення і покращення якості продукту на основі відгуків користувачів та підлаштування продукту під запити бізнесу у окремій версії PRO. Ключовим конкурентоспроможним фактором буде швидкодія та якість продукту. Базовою ж стратегією розвитку буде концентрований маркетинг, який дозволяє охопити велику кількість користувачів та постійно приводити нових клієнтів.

Що стосується конкурентної поведінки, то була обрана наступна стратегія. Так як продукт не є першопрохідцем на ринку - компанія буде шукати нових клієнтів та переманювати вже існуючих у конкурентів. Також компанія буде запозичувати нововведення у конкурентів та удосконалювати свій продукт.

Для продукту був розроблений комплекс асоціацій, на основі стратегій, які були описані раніше. За цими асоціаціями користувачі мають ідентифікувати продукт та торговельну марку. Головною вимогою до товару є легкість у сприйнятті та користуванні, а саме - зручність інтерфейсу. Також важливою є точність та надійність роботи продукту. Основний упор буде робитись на різницю з конкурентами, а саме доступність та легкість у використанні.

4.3 Розроблення маркетингової програми стартап-проекту

Був сформований опис товару, який отримає користувач, з точки зору маркетингу на основі аналізу конкурентоспроможності продукту.

Таблиця 4.2 – Ключові переваги товару

№ п/п	Запит клієнтів	Що пропонує товар	Переваги над конкурентами
1	Швидкія та надійність	Швидка оцінка емоцій голосу з досить простою архітектурою, яка є досить надійною	Швидкодія та простота в експлуатації
2	Зручність	Простий інтерфейс не перевантажений функціоналом	Простий функціонал який покриває запити клієнтів
3	Практичність продукту та точність класифікації	Точність є близькою до 90% і виводиться у реальному часі	Результат виводиться одразу у реальному часі

Трирівнева маркетингова модель:

1-й рівень. Описується основна ідея товару і проблема, яку він вирішує.

2-й рівень. Опис реальних характеристик товару, які будуть імплементовані під час його створення. Наприклад: ціна, дизайн, властивості, тощо.

3-й рівень. Додаткові послуги та знижки, які стимулюють до купівлі продукту.

Таблиця 4.3 – Опис трьох рівнів моделі товару

Рівень	Опис		
I. Опис товару	Класифікація емоцій у голосі людини у реальному часі		
II. Продукт у реальному житті	Властивості/характеристики	М/Нм	Вр/Тх/Тл/Е/Ор
	1. Інтуітивність інтерфейсу. 2.Точність. 3. Низька ціна.	1.Нм 2.Нм 3.Нм	Технічна Технологічна Вартісна
	Якість: тестування незалежними аудиторами		
	Пакування: відсутнє		
	Марка: VOICY		
III Товар з підкріпленням	Продукт буде пропонувати знижки новим користувачам, та сезонні знижки всім на святах. Також буде розроблена версія PLUS з додатковим функціоналом, а саме класифікація додаткових емоцій та взаємодія з пристроями по типу розумних годинників для поєднання аналізу голосу та показників стресу.		

Цінові межі будуть встановлені на основі вартості інших продуктів у AppStore та Google Play, так як у конкурентів відсутня інформація про ціну у відкритому доступі.

Таблиця 4.4 – Встановлення ціни

№ п/п	Рівень цін на подібні товари	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	- $	1.99$ / місяць	У двох груп різний рівень доходів	Безкоштовно (показ реклами та стимуляція купівлі додаткових послуг) - 20$ / місяць

Оптимальною системою збуту є магазини програмного забезпечення, такі як App Store та Google Play, згадані раніше. Адже через них ми, по-перше, отримуємо доступ до більшості користувачів смартфонів, а також маємо можливість зручно оформлювати підписку.

Останньою частиною програми маркетингу є визначення концепції маркетингових комунікацій з клієнтом. Каналом комунікації буде тільки інтернет, через свою простоту та доступність. Ключовими позиціями будуть: легкість використання продукту, низька ціна та точність. Основним

завданням рекламного матеріалу буде показати функціонал програмного забезпечення та виділити переваги над конкурентами.

4.4 Висновки до четвертого розділу

Був розроблений стартап проєкт на основі створеної системи SER.

Програмне забезпечення для оцінки емоційного забарвлення голосу було розроблено на мові програмування Python і складається з декількох модулів, а саме: зчитування звуку з мікрофону, обробка звуку та створення RGB зображення з mel спектрограми, модель.

Під час його тестування було помічено, що через взаємодію з файловою системою для збереження RGB зображення і подальшого зчитування його моделлю витрачається в середньому 112 мс на одну обробку. З цього слідує, що на кожну секунду витрачається 0.1 секунда корисного аудіо, але як відмічалося раніше - це є допустимою втратою сигналу, так як істотно більша частина корисного звуку успішно аналізується системою.

Програмне забезпечення чудово себе показало, стабільно та точно опрацьовуючи голос у реальному часі.

Також був прописаний повний маркетинговий план для створення, презентації та просування продукту до клієнтів. Було виділено основні (цільові) групи клієнтів, а саме: великий бізнес, у якого є потреба в оцінці емоційного стану своїх співробітників; а також звичайні користувачі смартфонів яким було б цікаво використати подібний продукт для оцінки свого голосу та голосу інших людей, а також отримання статистики по своїм розмовам.

ВИСНОВКИ

У роботі були розглянуті сучасні підходи до аналізу голосу людини, а саме виділення емоцій з нього. Наразі є два основні підходи - це перетворення голосу у текст та проведення семантичного аналізу сказаного людиною, та аналіз звуку як сигналу за допомогою рекурентних нейронних мереж. Для аналізу емоцій у голосі у реальному часі було обрано підхід з перетворення аудіо у спектрограму, після чого вона вже відправлялася на вхід до нетренованої CNN, а саме AlexNet.

У роботі використано набір даних EMO-DB який складається з записів голосів декількох акторів різного віку та статі, які промовляли речення з різними емоціями. До кожного аудіофайлу примінялися наступні дії: змінювалася частота дискретизація на 8кГц та застосовувалося u-law компандування (у двох експериментах з чотирьох), нарізався на блоки по 1 секунді з кроком у 10 мс, тобто з одного аудіо тривалістю 3 секунди виходило 300 фрагментів. Після чого до кожного фрагменту застосовувалося короткочасне перетворення Фур’є, результат якого перетворювався у масиви спектральних величин, з яких вже створювалися RGB зображення.

На отриманих даних була донавчена модель AlexNet за допомогою технології transfer learning. Після навчання було проведено 4 експерименти, які мали на меті показати вплив зміни якості звуку на точність роботи моделі.

Підсумовуючи, було виявлено, що використання CNN для класифікації емоцій голосу дає точність у 90.56% і швидкість інференсу достатня для аналізу емоцій у реальному часі. Також було показано, що обидва фактори, зменшення смуги пропускання мовлення та впровадження процедури компандування мовлення μ-law, мають шкідливий вплив на результати SER. За рахунок зменшення частоти дискретизації з 16 до 8 кГц (тобто, зменшення

смуги пропускання з 8 до 4 кГц) спостерігалося невелике зниження середньої точності SER на (близько 3.3%). Процедура компандування зменшила результат на схожу величину (приблизно на 5.1%), а сукупний вплив обох факторів призвело до зниження приблизно на 7.5% порівняно з базовими результатами. У всіх експериментальних випадках SER виконувався в режимі реального часу з емоційними мітками, що генерувалися кожну 1 секунду. Різниця у часі інференсу для даних різної якості виявилася не суттєвою, а саме 4 мс.

Також було створено програмне забезпечення, яке у реальному часі аналізує емоційне забарвлення голосу у реальному часі на базі моделі, яка була навчена на даних з наступними характеристиками ЧД=16, СП=8, U-Law=No. Воно добре себе показало, стабільно і точно аналізуючи голос мовця у реальному часі.

Програмне забезпечення є основою стартап проєкту для якого була описаний повна маркетингова стратегія. Було прописано список цільових користувачів, стратегія розвитку продукту, комплекс асоціацій з продуктом, ключові переваги товару, а також описана трирівнева маркетингова модель. Сам продукт позиціонується як програмний додаток для смартфону, який можна завантажити у App Store та Google Play.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Cortes, C. and Vapnik, V. Support-Vector Networks. Machine Learning, 1995 – P. 273-297
2. Theoretical model based on TRAM, Lin et al., 2007 – P. 1-3
3. Emotion Classification Based On Public Opinion Analysis On Online News, Ciptadi, A. S. Girsang, 2019 – P. 1-8
4. Reader perspective emotion analysis in text through ensemble based multi-label classification framework, Bhowmick et al., 2010 – P. 4-8.
5. Pennebaker, James & Francis, M. & Booth, R. Linguistic Inquiry and Word Count, 2001 – P. 71
6. About openSMILE – [Електронний ресурс]. – Режим доступу: https://audeering.github.io/opensmile/about.html
7. The AMI Meeting Corpus, Simone Ashby, 2005 – P. 1-4.
8. Gustavo Aguilar, Viktor Rozgic, Weiran Wang, and Chao Wang. 2019. Multimodal and multi-view models for emotion recognition – 2020 – P. 1–6.
9. Столар, М. Н., Лех, М., Болія, Р. Б., і Скіннер, М.«Розпізнавання емоцій голосу у реальному часі за допомогою класифікації зображень RGB і transfer learning», 2017. 1–6.
10. LVA Technology – [Електронний ресурс]. – Режим доступу: https://www.nemesysco.com/lva-technology/
11. Jiao W, Lyu MR, King I. Real-time emotion recognition via attention gated hierarchical memory network, 2019 – P. 12-14
12. Lieskovská E, Jakubec M, Jarina R, Chmulík M. A review on speech emotion recognition using deep learning and attention mechanism, 2021 – P. 10

13. Hazarika D, Poria S, Mihalcea R, Cambria E, Zimmermann R (2020) ICoN: Interactive conversational memory network for multimodal emotion detection, 2018 – P. 2594-2604.
14. Majumder N, Poria S, Hazarika D, Mihalcea R, Gelbukh A, Cambria E DialogueRNN: An Attentive RNN for Emotion Detection in Conversations. Proceedings of the AAAI Conference on Artificial Intelligence, vol 33, 2019 – P. 6818–6825.
15. A voice-based real-time emotion detection technique using recurrent neural network empowered feature modelling – [Електронний ресурс]. – Режим доступу: https://link.springer.com/article/10.1007/s11042-022-13363-4
16. EMO-DB Dataset – [Електронний ресурс]. – Режим доступу: https://www.kaggle.com/datasets/piyushagni5/berlin-database-of-emotional-speech-emodb
17. U-Law (Explained) – [Електронний ресурс]. – Режим доступу: https://www.liveagent.com/customer-support-glossary/u-law/
18. Методи кодування сигналу – [Електронний ресурс]. – Режим доступу: https://www.cisco.com/c/en/us/support/docs/voice/h323/8123-waveform-coding.html
19. Understanding of Mel spectrogram – [Електронний ресурс]. – Режим доступу: https://medium.com/analytics-vidhya/understanding-the-mel-spectrogram-fca2afa2ce53
20. Жегеді, С., Луі, В., Джіа, І., Серманет, П., Рід, С., Ангелов, Д. та ін. «Поглиблюючись із CNN», у матеріалах Конференції IEEE з комп’ютерного бачення та розпізнавання образів, 2015 – P 1–9.

21. Confusion matrix – [Електронний ресурс]. – Режим доступу: https://www.researchgate.net/profile/Farid-Morsidi/publication/301515757/figure/tbl1/AS:613988828184603@1523397752854/Confusion-Matrix-between-cluster-labels-TP-true-positive-FP-false-positive-TN-true.png
22. Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures – [Електронний ресурс]. – Режим доступу: https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/

ДОДАТОК А

```
import numpy as np
import pandas as pd
import glob
import librosa
from scipy.io import wavfile
import matplotlib.pyplot as plt
from PIL import Image
import io

audios = glob.glob("/content/drive/MyDrive/KPI/Master Diploma/code/wav" + '/*.*')

label2name = {
   "L": "Boredom",
   "A": "Fear",
   "E": "Disgust",
   "F": "Happiness",
   "T": "Sadness",
   "W": "Anger",
   "N": "Neutral"
}

#16kHz sr and 8kHz bandwidth
```

```
    path_to_folder = "/content/drive/MyDrive/KPI/Master Diploma/code/16_8_data"

    test_counter = 0
    counter = 0
    for file in audios:
        file_name = file.split("/")[-1]
        emotion = label2name[file_name[5]]
        samples, sample_rate = librosa.load(file, sr=16000)
        # separated_samples = []
        for window in range(0, len(samples), int(0.01*16000)):
            signal = samples[window:window+16000]
            if len(signal) < 16000:
                break
            D = np.abs(librosa.stft(signal, n_fft=512, hop_length=64))**2
            logS = librosa.power_to_db(abs(D))
            buf = io.BytesIO()
            plt.imsave(buf, logS[::-1], cmap='jet')
            buf.seek(0)
            im = Image.open(buf)
            im = im.resize((256, 256))

            if test_counter == 4:
                im.save(f"{path_to_folder}/test/{emotion}/{counter}.png")
                test_counter = 0
            else:
                im.save(f"{path_to_folder}/train/{emotion}/{counter}.png")
```

```
            test_counter += 1


        counter+=1



    #8kHz sr and 4kHz bandwidth
    path_to_folder = "/content/drive/MyDrive/KPI/Master Diploma/code/8_4_data"

    test_counter = 0
    counter = 0
    for file in audios:
        file_name = file.split("/")[-1]
        emotion = label2name[file_name[5]]
        samples, sample_rate = librosa.load(file, sr=8000)
        # separated_samples = []
        for window in range(0, len(samples), int(0.01*8000)):
            signal = samples[window:window+8000]
            if len(signal) < 8000:
                break
            D = np.abs(librosa.stft(signal, n_fft=512, hop_length=64))**2
            logS = librosa.power_to_db(abs(D))
            buf = io.BytesIO()
            plt.imsave(buf, logS[::-1], cmap='jet')
            buf.seek(0)
```

```
            im = Image.open(buf)
            im = im.resize((256, 256))

            if test_counter == 4:
                im.save(f"{path_to_folder}/test/{emotion}/{counter}.png")
                test_counter = 0
            else:
                im.save(f"{path_to_folder}/train/{emotion}/{counter}.png")
                test_counter += 1
            counter+=1



        #16kHz sr and 8kHz bandwidth U-Law
        path_to_folder = "/content/drive/MyDrive/KPI/Master
Diploma/code/16_8_data_mu_law"

        test_counter = 0
        counter = 0
        for file in audios:
            file_name = file.split("/")[-1]
            emotion = label2name[file_name[5]]
            samples, sample_rate = librosa.load(file, sr=16000)
            # separated_samples = []
            for window in range(0, len(samples), int(0.01*16000)):
                signal = samples[window:window+16000]
                if len(signal) < 16000:
                    break
```

```
            signal = librosa.mu_compress(signal, mu=255, quantize=True)
            D = np.abs(librosa.stft(signal, n_fft=512, hop_length=64))**2
            logS = librosa.power_to_db(abs(D))
            buf = io.BytesIO()
            plt.imsave(buf, logS[::-1], cmap='jet')
            buf.seek(0)
            im = Image.open(buf)
            im = im.resize((256, 256))

            if test_counter == 4:
                im.save(f"{path_to_folder}/test/{emotion}/{counter}.png")
                test_counter = 0
            else:
                im.save(f"{path_to_folder}/train/{emotion}/{counter}.png")
                test_counter += 1
            counter+=1



        #8kHz sr and 4kHz bandwidth U-Law
        path_to_folder = "/content/drive/MyDrive/KPI/Master
Diploma/code/8_4_data_mu_law"

        test_counter = 0
        counter = 0
        for file in audios:
            file_name = file.split("/")[-1]
            emotion = label2name[file_name[5]]
```

```
samples, sample_rate = librosa.load(file, sr=8000)
# separated_samples = []
for window in range(0, len(samples), int(0.01*8000)):
    signal = samples[window:window+8000]
    if len(signal) < 8000:
        break
    signal = librosa.mu_compress(signal, mu=255, quantize=True)
    D = np.abs(librosa.stft(signal, n_fft=512, hop_length=64))**2
    logS = librosa.power_to_db(abs(D))
    buf = io.BytesIO()
    plt.imsave(buf, logS[::-1], cmap='jet')
    buf.seek(0)
    im = Image.open(buf)
    im = im.resize((256, 256))

    if test_counter == 4:
        im.save(f"{path_to_folder}/test/{emotion}/{counter}.png")
        test_counter = 0
    else:
        im.save(f"{path_to_folder}/train/{emotion}/{counter}.png")
        test_counter += 1
    counter+=1
```

ДОДАТОК Б

```
import matplotlib.pyplot as plt
import librosa
import librosa.display
import numpy as np

from fastai import *
from fastai.vision.all import *
from fastai.vision.data import ImageDataLoaders
from fastai.tabular.all import *
from fastai.text.all import *
from fastai.vision.widgets import *



#16kHz 8kHz
path_to_folder = "/content/drive/MyDrive/KPI/Master Diploma/code/16_8_data/train/"
dls = ImageDataLoaders.from_folder(path_to_folder, valid_pct=0.2, seed=21, num_workers=0)

learn = cnn_learner(dls, models.alexnet, loss_func=CrossEntropyLossFlat(), metrics=accuracy)

lr_min, lr_steep = learn.lr_find()
print(f"Minimum/10: {lr_min:.2e}, steepest point: {lr_steep:.2e}")
```

```
        history = learn.fit(80, float(f"{lr_steep:.2e}"),
cbs=EarlyStoppingCallback(monitor='accuracy', min_delta=0.001, patience=5))
        learn.show_results()
        learn.freeze()
        learn.export('/content/drive/MyDrive/KPI/Master
Diploma/code/speech_16_8_1.pkl')

        path_to_folder = "/content/drive/MyDrive/KPI/Master
Diploma/code/16_8_data/test/"
        dls = ImageDataLoaders.from_folder(path_to_folder, valid_pct=0.2,
seed=21, num_workers=0)
        predicted = learn.predict(dls)
        print(learn.predict('/content/test.png'))



        #8kHz 4kHz
        path_to_folder = "/content/drive/MyDrive/KPI/Master
Diploma/code/8_4_data/train/"
        dls = ImageDataLoaders.from_folder(path_to_folder, valid_pct=0.2,
seed=21, num_workers=0)

        learn = cnn_learner(dls, models.alexnet, loss_func=CrossEntropyLossFlat(),
metrics=accuracy)

        lr_min, lr_steep = learn.lr_find()
        print(f"Minimum/10: {lr_min:.2e}, steepest point: {lr_steep:.2e}")
```

```
        history = learn.fit(80, float(f"{lr_steep:.2e}"),
cbs=EarlyStoppingCallback(monitor='accuracy', min_delta=0.001, patience=5))
        learn.show_results()
        learn.freeze()
        learn.export('/content/drive/MyDrive/KPI/Master
Diploma/code/speech_8_4_1.pkl')

        path_to_folder = "/content/drive/MyDrive/KPI/Master
Diploma/code/8_4_data/test/"
        dls = ImageDataLoaders.from_folder(path_to_folder, valid_pct=0.2,
seed=21, num_workers=0)
        predicted = learn.predict(dls)
        print(learn.predict('/content/test.png'))



        #16kHz 8kHz U-Law
        path_to_folder = "/content/drive/MyDrive/KPI/Master
Diploma/code/16_8_mu_law_data/train/"
        dls = ImageDataLoaders.from_folder(path_to_folder, valid_pct=0.2,
seed=21, num_workers=0)

        learn = cnn_learner(dls, models.alexnet, loss_func=CrossEntropyLossFlat(),
metrics=accuracy)

        lr_min, lr_steep = learn.lr_find()
        print(f"Minimum/10: {lr_min:.2e}, steepest point: {lr_steep:.2e}")
```

```
        history = learn.fit(80, float(f"{lr_steep:.2e}"),
cbs=EarlyStoppingCallback(monitor='accuracy', min_delta=0.001, patience=5))
        learn.show_results()
        learn.freeze()
        learn.export('/content/drive/MyDrive/KPI/Master
Diploma/code/speech_16_8_mu_law_1.pkl')

        path_to_folder = "/content/drive/MyDrive/KPI/Master
Diploma/code/16_8_mu_law_data/test/"
        dls = ImageDataLoaders.from_folder(path_to_folder, valid_pct=0.2,
seed=21, num_workers=0)
        predicted = learn.predict(dls)
        print(learn.predict('/content/test.png'))



        #8kHz 4kHz U-Law
        path_to_folder = "/content/drive/MyDrive/KPI/Master
Diploma/code/8_4_mu_law_data/train/"
        dls = ImageDataLoaders.from_folder(path_to_folder, valid_pct=0.2,
seed=21, num_workers=0)

        learn = cnn_learner(dls, models.alexnet, loss_func=CrossEntropyLossFlat(),
metrics=accuracy)

        lr_min, lr_steep = learn.lr_find()
        print(f"Minimum/10: {lr_min:.2e}, steepest point: {lr_steep:.2e}")
```

```
        history = learn.fit(80, float(f"{lr_steep:.2e}"),
cbs=EarlyStoppingCallback(monitor='accuracy', min_delta=0.001, patience=5))
        learn.show_results()
        learn.freeze()
        learn.export('/content/drive/MyDrive/KPI/Master
Diploma/code/speech_8_4_mu_law_1.pkl')

        path_to_folder = "/content/drive/MyDrive/KPI/Master
Diploma/code/8_4_mu_law_data/test/"
        dls = ImageDataLoaders.from_folder(path_to_folder, valid_pct=0.2,
seed=21, num_workers=0)
        predicted = learn.predict(dls)
        print(learn.predict('/content/test.png'))
```

ДОДАТОК В

```
import matplotlib.pyplot as plt
import librosa
import librosa.display

from fastai import *
from fastai.vision.all import *
from fastai.vision.data import ImageDataLoaders
from fastai.tabular.all import *
from fastai.text.all import *

import warnings
warnings.filterwarnings("ignore")

import numpy as np
import pyaudio
import time
import librosa

import keyboard


model = load_learner('/Users/dimka/Downloads/speech_01.pkl')

class AudioProcessing(object):
```

```
def __init__(self):
    self.FORMAT = pyaudio.paFloat32
    self.CHANNELS = 1
    self.RATE = 16000
    self.CHUNK = 16000
    self.p = None
    self.stream = None

def start(self):
    self.p = pyaudio.PyAudio()
    self.stream = self.p.open(format=self.FORMAT,
                              channels=self.CHANNELS,
                              rate=self.RATE,
                              input=True,
                              output=False,
                              stream_callback=self.callback,
                              frames_per_buffer=self.CHUNK)

def stop(self):
    self.stream.close()
    self.p.terminate()

def callback(self, in_data, frame_count, time_info, flag):
    numpy_array = np.frombuffer(in_data, dtype=np.float32)
    data_16 = librosa.resample(numpy_array, self.RATE, 16000)
    D = np.abs(librosa.stft(data_16, n_fft=512, hop_length=64))**2
    logS = librosa.power_to_db(abs(D))
```

```
        buf = io.BytesIO()
        plt.imsave(buf, logS[::-1], cmap='jet')
        buf.seek(0)
        im = Image.open(buf)
        im = im.resize((256, 256))
        im.save(f"test.png")
        prediction = model.predict("test.png")
        print(prediction[0])
        return None, pyaudio.paContinue

    def mainloop(self):
        while (self.stream.is_active()):

            if keyboard.is_pressed('q'):
                self.stream = 0

audio = AudioProcessing()
audio.start()
audio.mainloop()
audio.stop()
```