МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

КУРСОВА РОБОТА

з дисципліни

“МАШИННЕ НАВЧАННЯ”

на тему: **Розпізнавання емоцій голосу**

Студентки групи КН-317
спеціальності 122
“Комп’ютерні науки”
Долинської Х. І.
Керівник: Сивоконь О. О.

Кількість балів: _____ Оцінка: _____

Члени комісії:

_____
(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)

_____
(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)

_____
(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)

Львів – 2024

ЗМІСТ

1. ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

SER - Speech Emotion Recognition (Розпізнавання мовних емоцій)
NN - Neural Networks (Нейронна мережа)
GMM - Gaussian Mixture Model (Модель гаусівської суміші)
K-NN - K-Nearest Neighbor (К-найближчих сусідів)
HMM - Hidden Markov Model (Прихована модель Маркова)
SVM - Support Vector Machine (Метод опорних векторів)
MLP - Multilayer Perceptron (Багатошаровий персептрон)
CNN - Convolutional Neural Network (Згорткова нейронна мережа)
DCNN - Deep Convolutional Neural Network (Глибока згорткова нейронна мережа)
ResNet - Residual Network (Залишкова нейронна мережа)
RNN - Recurrent Neural Network (Рекурентна нейронна мережа)
BES - Berlin Emotional Speech
PES - Polish Emotional Speech
LDC - Linguistic Data Consortium
UGA - University of Georgia
RAVDESS - Ryerson Audio-Visual Database of Emotional Speech and Song
IEMOCAP - Interactive Emotional Dyadic Motion Capture
CHEAVD - Combined Human and Audiovisual Emotion Database
SES - Spanish Emotional Speech
OAA - One-Against-All
MFCC - Mel-Frequency Cepstral Coefficients
LPCC - Linear Predictive Cepstral Coefficients
STFT - Short-Time Fourier Transform
CWT - Continuous Wavelet Transform
fCWT-trained - Classifier trained with features from CWT
DA - Data Augmentation
RCS - Random Cropping and Scaling
WGN - White Gaussian Noise
MFCC - Mel Frequency Cepstrum Coefficient
STFT - Short-time Fourier transform

2. ВСТУП

Протягом останнього десятиліття проблематика розпізнавання емоцій у голосі набуває особливого значення. Основна складність полягає в тому, щоб зробити комп'ютер здатним розпізнавати не лише семантичне значення висловлювання, а й емоційний фон, що його супроводжує.

Актуальність вивчення емоцій у мовленні визначається не лише загальною важливістю емоційного інтелекту в сучасному світі, а й конкретними проблемами, які вирішуються через це дослідження. Одна з ключових сутностей проблеми полягає у розробці систем, що можуть аналізувати і розпізнавати емоції в мовленні людини. Це важливо для розуміння та покращення комунікації, розвитку психологічних досліджень та створення більш ефективних інтерфейсів взаємодії з комп'ютерами та іншими технологіями.

Мета даної роботи полягає в розробці та вивченні системи розпізнавання емоцій у голосі людини.

Завдяки цим дослідженням планується досягти наступної **цілі**: розробити та вдосконалити систему, яка може точно визначати емоційний стан людини за голосом. Це допоможе визначити, чи не є промова монотонною, та зрозуміти, де необхідно змінити емоцію голосу задля покращення промови в цілому.

Перш за все, планується дослідити різні методи та підходи до створення систем аналізу емоцій у мовленні. Далі – обрати найвідповідніший набір даних та найбільш підходящий метод, після чого розробити та навчити модель машинного навчання на основі обраного набору даних, який містить емоційно виразне мовлення.

Об'єктом дослідження даної роботи є процес розпізнавання емоцій у мовленні людини. Цей процес виникає внаслідок взаємодії різних аспектів мовленнєвої активності, таких як інтонація, темп, виразність та використання мовних засобів.

Предметом дослідження є алгоритми та методи, які дають змогу реалізувати процес розпізнавання емоцій у мовленні людини.

Соціальна значущість проблеми виявляється в тому, що розробка систем розпізнавання емоцій у мовленні може мати велике значення для українського суспільства. Наприклад, такі системи можуть застосовуватися у психологічних консультаціях, медичній діагностиці, в сфері освіти для аналізу емоційної атмосфери у класах тощо. Зокрема, такі системи необхідні для аналізу емоційності промови людини з метою її (промови) покращення. Наприклад, це

може сприяти покращенню монотонної промови шляхом ідентифікації емоційних аспектів та їх впливу на інтонацію та ритм мовлення. Наприклад, система може виявити, що промова особи має відсутність різноманітності в інтонації та виразності, що може вказувати на монотонність. Така інформація буде корисною для системи, яка надає рекомендації щодо зміни інтонації або акцентування на певних словах чи фразах для підвищення емоційної виразності промови. Це допоможе покращити якість та емоційну виразність мовлення особи, зробивши її промову більш цікавою та зрозумілою для аудиторії.

Дослідження емоцій у мовленні також має велике значення для подальшого розвитку області обробки природної мови та інтелектуальних систем. Розробка ефективних алгоритмів розпізнавання емоцій може сприяти створенню більш гнучких та інтуїтивно зрозумілих систем штучного інтелекту, що можуть бути використані у різних галузях, від розважальних до підприємницьких.

Розв'язання конкретних часткових питань у цій області досліджень може сприяти значним змінам у науці, в тому числі у психології, лінгвістиці, інформаційних технологіях тощо. Наприклад, вивчення взаємозв'язку між інтонацією та емоційним виразом може принести нові уявлення про функціонування мовної системи та її емоційні аспекти.

Отже, дана робота може мати значне **практичне значення**, зокрема, розроблена система розпізнавання емоцій у мовленні може бути використана для аналізу емоційності промови людини та її подальшого покращення. Крім того, отримані результати можуть бути використані для подальшого наукового дослідження в галузі обробки природної мови, дослідження впливу емоційного забарвлення на сприйняття мовлення, розвиток нових методів та алгоритмів для розпізнавання та аналізу емоцій у мовленні.

3. АНАЛІЗ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

Для успішної роботи системи розпізнавання мовних емоцій (SER) необхідно вирішити три ключові проблеми: вибір хорошої бази даних емоційного мовлення, виділення ефективних ознак, розробка надійних класифікаторів за допомогою алгоритмів машинного навчання [3].

При виборі відповідного набору даних необхідно враховувати наступні критерії: ступінь природності емоцій, розмір бази даних і кількість доступних емоцій [1]. Існує три основні типи наборів даних:

- зіграні емоції: емоції, відтворені за певними сценаріями, наприклад, під час акторської гри;
- викликані емоції: емоції, що виникають у штучно створених ситуаціях, наприклад, реакції на музику, відео, рекламу, тощо;
- природні спонтанні емоції: емоції, які виникають у повсякденному житті, наприклад, реакції на реальні події, чи записи, витягнуті з реаліті-шоу.

Оскільки завданням роботи є натренувати модель для визначення емоції голосу людини під час її промови, найбільш відповідним варіантом для навчання моделі буде використання бази даних з природними спонтанними емоціями. Такі дані краще відображають реальні вирази емоцій у мовленні, що дозволить моделі краще розпізнавати та класифікувати емоції в голосі людини під час промови. Проте, недоліком таких записів є те, що вони можуть бути спотворені фоновим шумом [2] та містити незбалансовані емоційні категорії [1].

Наступною проблемою, яку потрібно вирішити для роботи системи, є виділення ефективних ознак. Оскільки типовий набір значущих емоцій складається з 300 різних емоційних станів, це ускладнює їх класифікацію [13]. Як зазначається в статті “Опитування щодо розпізнавання мовленнєвих емоцій: особливості, схеми класифікації та бази даних”, дану проблему можна вирішити “[…] згідно з “Теорією палітри”: будь-яку емоцію можна розкласти на первинні емоції, подібно до того, як будь-який колір є комбінацією кількох основних кольорів.” [14] (переклад – ХД). Ці первинні емоції включають нейтральність, радість, гнів, відразу, страх, смуток, стрес і здивування [14].

Останньою проблемою для імплементації системи SER є розробка надійних класифікаторів. У різних наукових роботах розглядалися різні підходи до вирішення цієї проблеми.

Наприклад, у статті “Implementation and Comparison of Speech Emotion Recognition System Using Gaussian Mixture Model (GMM) and K-Nearest Neighbor

(K-NN) Techniques” [6] описані моделі гаусівської суміші (GMM) та K-найближчих сусідів (K-NN). Автори отримали дві моделі, кожна з яких класифікувала певні емоції краще, ніж інша. Наприклад, GMM з високою точністю розпізнає емоції злості (92%), суму (89%) та нейтральні емоції (73%), натомість модель K-NN краще розпізнає емоції щастя (90%). Варто зауважити, що обидві моделі погано визначають емоції здивування (25% для обох моделей) та страху (50% для GMM та 25% для K-NN). Внаслідок проведених досліджень автори зробили висновок, що для імплементації більш точної системи доцільно буде поєднати ці дві моделі, оскільки одна з них ефективніше визначає певні емоції, тоді як інша є більш ефективною у визначенні інших.

У статті “A Two-Stage Hierarchical Bilingual Emotion Recognition System Using a Hidden Markov Model and Neural Networks” [7] розглядається прихована модель Маркова (HMM). Автори використовують два набори даних (берлінська та польська) для тренування моделі. Метою їхнього дослідження є доведення необхідності знання мови для визначення типу емоції. Зокрема, вони розглянули два сценарії класифікації емоцій: змішали всі речення з обох баз даних в один набір даних в першому випадку та використали ієрархічну методологію, в якій спочатку визначали мову запису в другому. Результатами проведених експериментів є точність класифікації емоцій 57.64% для першого підходу та 93.06% для другого. Використовуючи другий підхід, автори збільшили набір емоцій до 6, що призвело до часткової втрати точності (89,13%), проте дозволило класифікувати більше емоцій.

Ще один метод для розпізнавання емоцій описаний у статті “Speech Emotion Recognition using Support Vector Machine” [8]. У роботі було розглянуто дві стратегії класифікації методом опорних векторів (SVM): One-Against-All та Gender dependent; для витягнення ознак з мовленнєвих висловлювань були використані алгоритми MFCC та LPCC; для тренування моделей було використано два набори даних: LDC та UGA. Під час аналізу отриманих результатів було виявлено, що виявлення ознак за допомогою MFCC забезпечило вищу точність порівняно з LPCC (85,085% та 73,125% відповідно); використання набору даних LDC підвищило точність класифікації відносно набору UGA (90,08% та 65,97% для тренувальних даних відповідно); точність класифікатора, залежного від статі вища за класифікатор OAA (84,42% та 72,785% відповідно). Таким чином, найкращою комбінацією виявилося використання класифікатора, залежного від статі, разом з алгоритмом MFCC та набором даних LDC з точністю моделі 82,94%.

У статті “Human-Computer Interaction for Recognizing Speech Emotions Using Multilayer Perceptron Classifier” [9] описується ще один метод розпізнавання емоцій: багатошаровий персептрон (MLP). Автори ускладнили модель, збільшивши розмір прихованого шару (розмір вхідного шару 100, а прихованого - 750x750x750). Попри це ускладнення, автори залишили інші параметри для тренування на низькому рівні, що призвело до швидкого тренування моделі, яке займало кілька хвилин.

Наступні методи класифікації емоцій використовуються у статті “Speech Emotion Recognition Using Deep Convolutional Neural Networks Improved by the Fast Continuous Wavelet Transform” [10]. У роботі було реалізовано та проаналізовано дві моделі DCNN: STFT та fCWT; для доповнення даних (DA) були розглянуті техніки RCS, WGN, а також відсутність DA; для тренування моделей було використано два набори даних: eNTERFACE05 та EMO-DB. У результаті проведених досліджень виявилося, що найкращим поєднанням для моделі класифікації емоцій є fCWT класифікатор, техніка доповнення даних RCS5 та набір даних eNTERFACE05 з досягнутою точністю 76,6%.

Ще один метод розглядається у статті “Speaker to Emotion: Domain Adaptation for Speech Emotion Recognition with Residual Adapters” [11]. Автори розглядають модель ResNet20, для попереднього навчання вибрали набір даних VoxCeleb2, а для навчання - IEMOCAP. З метою покращення результатів після навчання моделі автори замінили FC-шар мережі і налаштували всю мережу за цільовим корпусом емоцій. Оскільки результат не виявився кращим, автори припустили, що кількість параметрів для навчання надто велика для невеликого обсягу даних.

У статті “Automatic Speech Emotion Recognition Using Recurrent Neural Networks with Local Attention” [12] автори розглянули різні архітектури RNN. Найкращою серед них виявилася модель RNN – weighted pool with attention, точність якої дорівнює 63,5%. Для тренування моделі було обрано набір даних IEMOCAP. Однією з переваг отриманої моделі є отримання результату у вигляді вектора ознак на рівні висловлювання. Це означає, що модель не визначає конкретну емоцію, а аналізує повне речення (чи аудіоряд) та класифікує повний спектр використаних емоцій.

Для візуалізації переваг та недоліків кожного з розглянутих методів зобразимо таблицю:

Таблиця 1. Аналіз методів розпізнавання емоцій, описаних у розглянутих наукових роботах

Метод	**Джерело**	**Набір даних**	**Набір емоцій**	**Точність**	**Переваги методу**	**Недоліки методу**
GMM	[6]	BES	6	66%	- з високою точністю розрізняє емоції злості та суму;	- погано розрізняє інші емоції з набору; - зі збільшенням набору емоцій зростає час тренування;
K-NN	[6]	BES	6	51,67%	- добре розрізняє емоції щастя і злості; - швидко тренується;	- погано розрізняє інші емоції з набору;
HMM	[7]	BES, PES	4	93,06%	- класифікує емоції з високою точністю;	- розрізняє малий набір емоцій; - тренування моделі часозатратне;
HMM	[7]	BES, PES	6	89,13%	- класифікує емоції з досить високою точністю;	- тренування моделі дуже часозатратне;
SVM	[8]	LDC, UGA	4	82,94%	- розрізняє набір емоцій з практично однаковою точністю;	- розрізняє малий набір емоцій;

MLP	[9]	RAVDESS	8	81%	- класифікує багато емоцій; - швидко тренується;	- невисока точність класифікації;
DCNN	[10]	eNTERFACE05	7	76,6%	- класифікує значний набір емоцій; - використання CWT дозволяє фіксувати як часові, так і спектральні особливості;	- модель потребує величезної кількості даних, що, у свою чергу, унеможливлює класифікацію емоцій в реальному часі; - низька точність моделі;
ResNet	[11]	VoxCeleb2, IEMOCAP	9	72,73%	- класифікує великий набір емоцій; - модель не потребує повного повторення тренування при додаванні нових доменів;	- низька точність моделі;
RNN	[12]	IEMOCAP	4	63,5%	- модель здатна виділяти релевантні ознаки аудіодоріжки та ігнорувати нерелевантні, що дозволяє зберігати динаміку мовлення та розуміти його зв'язки;	- розрізняє малий набір емоцій; - низька точність моделі;

Завдяки таблиці можемо легко порівняти різні методи, описані в інших роботах. Зокрема, можемо зробити висновки, використання яких методів дозволяє досягти найбільшої точності, швидкості тренування, ефективності класифікації тощо. Також можемо побачити переваги та недоліки кожного методу в порівнянні з іншими, що допоможе визначити, які методи для класифікації емоцій варто розглядати в першу чергу.

4. АНАЛІЗ МАТЕРІАЛІВ ТА МЕТОДІВ

Проаналізувавши методи, використані в різних наукових роботах та переглянувши набори даних, які автори використовували для досягнення своїх цілей, перейдемо до вирішення проблем для успішної роботи системи SER, враховуючи потреби та вимоги даного дослідження.

Враховуючи інформацію, зазначену в аналізі літературних джерел, можемо виділити наступні ознаки, які потрібно враховувати при виборі набору даних:

- Тип даних (зіграні, викликані, природні спонтанні емоції). Як було згадано вище, для поставленої задачі найкраще підходитимуть спонтанні природні емоції. Проте їх складніше класифікувати, ніж, наприклад, зіграні емоції, що ускладнює задачу.
- Кількість емоцій. Порівнюючи різні методи, описані в наукових роботах, бачимо, що зі збільшенням емоцій падає точність моделі. Проте класифікація більшої кількості емоцій дозволяє системі точніше визначати емоційний стан людини за голосом. Наприклад, замість простого розпізнавання основних емоцій, таких як щастя або злість, система може виявляти більш дрібні емоції, такі як здивування чи захоплення.
- Перелік набору емоцій. Оскільки метою завдання є розпізнавання загального спектру емоцій, важливо бачити, що набір даних містить різні емоції, зокрема і позитивні, і негативні, і нейтральні, а не лише різновид одних, наприклад, негативних.
- Розмір набору даних. Також важливо враховувати кількість даних, оскільки від неї залежить можливість ефективного навчання моделі та її загальна продуктивність.
- Мови, з яких складається набір даних. Найкраще підходять набори даних, що містять англійську мову, оскільки англійська є мовою, яка широко використовується на міжнародному рівні.

Проаналізувавши таблицю 1, можемо виділити наступні набори даних, які обиралися для вирішення задачі класифікації емоцій: IEMOCAP, RAVDESS, eNTERFACE05, BES (Emo-DB) та PES. Також можна виділити інші набори даних, які часто використовуються для вирішення задачі SER: SES, CREMA-D, LSSED, MSP-Podcast, AESI, EmoFilm, Quechua-SER, nEMO та OGVC.

Для зручнішого порівняння наборів даних зобразимо ознаки кожного з них у вигляді таблиці:

Таблиця 2. Аналіз наборів даних для вирішення задачі SER

Назва набору даних	**Тип даних**	**К-сть емоцій**	**Перелік набору емоцій**	**Контент**	**Розмір набору даних**	**Мова**
IEMOCAP	викликані емоції	9	щастя, злість, здивування, нейтральність, страх, сум, засмучення, розчарування	302 відеозаписи від 2 спікерів	-	англійська
RAVDESS	зіграні емоції	8	щастя, злість, здивування, нейтральність, спокій, огида, страх, сум	7 356 записів від 24 акторів	24.8 Гб	англійська
RAVDESS Audio only	зіграні емоції	8	щастя, злість, здивування, нейтральність, спокій, огида, страх, сум	1 440 записів від 24 акторів	0,45 Гб	англійська
eNTERFACE05	викликані емоції	6	гнів, страх, здивування, щастя, сум, відраза	Записи від 42 спікерів, які представляють 14 різних	0,8 Гб	німецька

				національностей		
BES (Emo-DB)	зіграні емоції	7	гнів, нудьга, відраза, тривога/страх, радість, смуток, нейтральність	494 висловлювань від 10 спікерів	0,04 Гб	німецька
PES	природні спонтанні емоції	7	радість, смуток, гнів, страх, огиду, здивування, очікування	235 висловів, взятих, в основному, з прямих ефірів і програм, таких як реаліті-шоу	-	польська
SES	зіграні емоції	5	гнів, радість, смуток, здивування, нейтральність	30 слів, 15 коротких речень, 3 абзаци	-	іспанська
CREMA-D	зіграні емоції	6	злість, відраза, страх, щастя, нейтральність, сум	7 442 записи від 91 актора різного етнічного походження	-	англійська

LSSED	природні спонтанні емоції	11	злість, нейтральність, страх, щастя, сум, розчарування, нудьга, відраза, захоплення, здивування, страх	147 025 речень, виголошених 820 особами	-	англійська
MSP-Podcast	природні спонтанні емоції	8	гнів, сум, щастя, здивування, страх, відраза, презирство, нейтральність	62 140 розмовних зворотів, витягнутих з подкастів	-	англійська
AESI	викликані емоції	5	гнів, страх, щастя, сум, нейтральність	696 записаних висловлювань від 20 носіїв мови	-	грецька
EmoFilm	зіграні емоції	5	гнів, презирство, щастя, страх, смуток	1 115 аудіозаписів з 43 фільмів	-	англійська, італійська, іспанська

Quechua-SER	викликані емоції	9	радість, сум, нудьга, страх, мрійливість, спокій, збудження, злість, нейтральність	15 годин записів висловлювань	-	кечуа
nEmo	викликані емоції	6	гнів, страх, щастя, сум, здивування, нейтральність	4 481 запис	0,53 Гб	польська
OGVC	природні спонтанні та зіграні емоції	9	страх, здивування, сум, відраза, гнів, очікування, радість, прийняття, нейтральність	9 114 спонтанних висловлювань і 2656 акторських висловлювань від 4 акторів (двох чоловіків і двох жінок).	-	японська

Серед описаних в таблиці наборів можна виділити наступні набори даних: IEMOCAP, RAVDESS, LSSED та MSP-Podcast. Останні два набори містять природні спонтанні емоції, що краще підходить для поставленої задачі, проте доступ до них обмежений, що унеможливлює їх використання. Порівнюючи IEMOCAP та RAVDESS, важко визначити, який з них кращий. Оскільки IEMOCAP містить відеодані, а RAVDESS і відео, і аудіо, а для поставленого завдання необхідні лише аудіодані, задля зручності використовуватимемо лише аудіодані мовлення набору RAVDESS.

RAVDESS Emotional speech audio [16] - набір даних, що містить лише аудіофайли мовлення з набору даних RAVDESS. Це записи 24 акторів, серед яких 12 жінок та 12 чоловіків; кожен актор брав участь у 60-ти випробуваннях та озвучив два лексично підібрані твердження з нейтральним північноамериканським акцентом. Отже, загалом маємо 1440 аудіофайлів, що містять наступні емоції, виражені у мовленні: спокій, радість, смуток, гнів, страх, здивування та огида. Кожен емоційний вираз має дві рівні інтенсивності - звичайний і виражений, а також додатковий нейтральний вираз.

Кожен з файлів набору має унікальне ім'я. Ім'я файлу складається з 7-значного цифрового ідентифікатора (наприклад, 03-01-06-01-02-01-12.wav). Ці ідентифікатори визначають наступні характеристики запису:

- Модальність:
    1) повна аудіосистема
    2) лише відео
    3) лише аудіо
- Голосовий канал:
    1) мова
    2) пісня
- Емоція:
    1) нейтральність
    2) спокій
    3) радість
    4) смуток
    5) гнів
    6) страх
    7) огида
    8) здивування
- Емоційна інтенсивність:
    1) звичайна

2) виражена
   * для "нейтральної" емоції немає сильної інтенсивності

- Висловлення:
  1) "Kids are talking by the door"
  2) "Dogs are sitting by the door"
- Спроба:
  1) перша
  2) друга
- Актор (від 1 до 24):
  a) непарний номер - чоловіки
  b) парний номер - жінки

Таким чином, з назви аудіофайлу можна витягнути всі характеристики запису. Наприклад, файл з назвою 03-01-06-01-02-01-12.wav має наступні характеристики: лише аудіо (03), промова (01), страх (06), нормальна інтенсивність (01), висловлювання "Dogs are sitting by the door" (02), перша спроба (01), 12-й актор (12), жінка, оскільки ідентифікаційний номер актора парний.

Основними перевагами RAVDESS Emotional speech audio є:

1) Розмаїтість емоцій: як було сказано вище, даний набір містить 8 емоцій, а саме: спокій, радість, смуток, гнів, страх, здивування та огида. У порівнянні з іншими наборами цей містить помірну кількість емоцій, що дозволить створити модель, здатну розпізнавати широкий спектр людських емоцій. Даний набір емоцій - середина між надто спрощеними наборами, де представлено лише кілька основних емоцій, і надто складними, де включено понад десяток різних емоційних станів, що ускладнює навчання моделі та її застосування у реальних умовах.
2) Рівномірний розподіл даних: набір містить однакову кількість усіх емоцій, окрім нейтральної, оскільки кожна емоція має два види інтенсивності: звичайна та виражена, а нейтральність - лише звичайну. Також в наборі порівну записів голосу чоловіків та жінок. Це забезпечує збалансоване представлення кожної емоції та статі, що дозволяє уникнути упередженості моделі під час її навчання і підвищує точність та надійність її роботи. Деталі розподілу емоцій можна глянути на рисунку нижче:

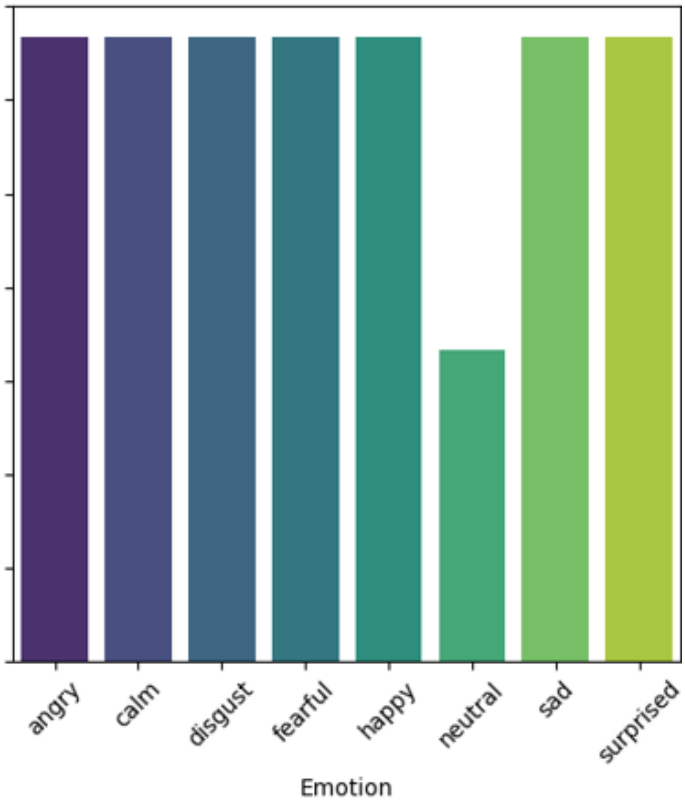

Рис. 1. Кількість аудіодоріжок для кожної емоції

3) Якісні аудіодоріжки: аудіофайли в наборі записані в студійних умовах з високою якістю звуку, що забезпечує чітке та розбірливе звучання кожної емоції, що, у свою чергу, підвищує точність розпізнавання емоцій. На рисунку нижче зображено середню тривалість аудіо кожної емоції:

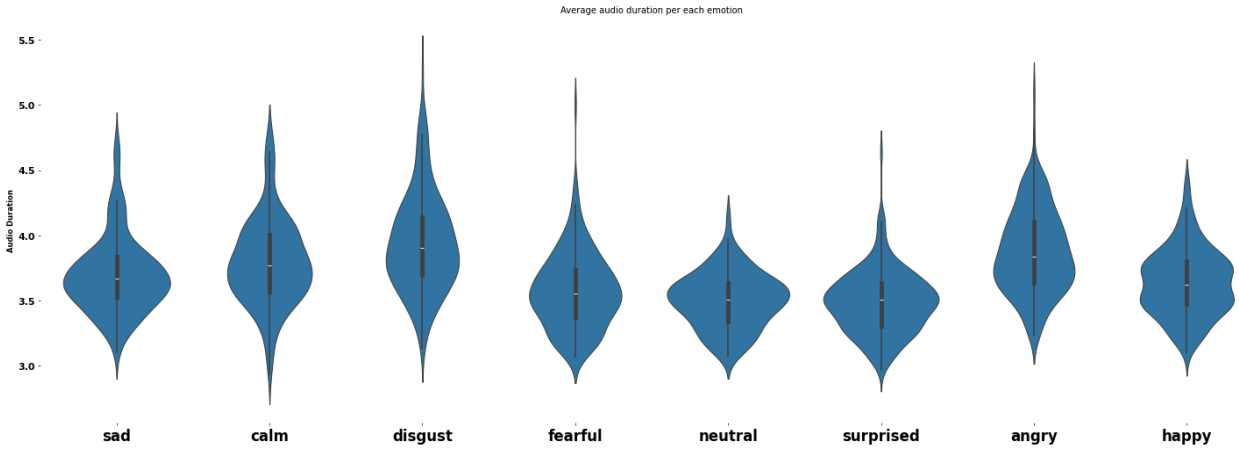

Рис. 2. Середня тривалість аудіо для кожної з емоцій набору

4) Помірний обсяг даних: Набір містить 1440 аудіофайли, що є помірною кількістю даних для їх класифікації. Можливо, для покращення досягнутої точності буде доцільно доповнити набір обробленими даними з повного набору RAVDESS, проте поки цих даних достатньо.

Для того, щоб навчити модель добре розпізнавати емоції, з набору будуть взяті лише аудіодоріжка та емоція запису. Оскільки в реальному використанні на вхід модель отримуватиме лише аудіо, а на вихід повинна буде видати емоцію, використання такої інформації, як статі, інтенсивності, голосового каналу, спроби немає сенсу. У подальшому можна спробувати подавати на вхід не лише аудіо, а й сказане висловлювання, проте це ускладнює модель.

Для ефективного розпізнавання емоцій з голосу не варто подавати на вхід моделі сирі аудіодоріжки. Натомість, аудіо необхідно перетворити у відповідні форми, які можуть бути легше оброблені моделлю. Основні види таких перетворень включають [17]:

1. Mel-spectrogram: графічне представлення аудіосигналу у вигляді спектру частот, що змінюються з часом. Мел-спектрограми використовують мел-шкалу, яка краще відповідає людському слуху, що дозволяє моделі краще захоплювати звукові особливості, важливі для розпізнавання емоцій.
2. MFCC: набір числових характеристик, що описують короткострокові спектральні особливості звуку. MFCC широко використовуються у задачах розпізнавання мовлення та звуку, оскільки вони ефективно представляють основні характеристики голосу.
3. Spectrogram (спектрограма): тривимірне зображення, яке показує частоти, амплітуди та час. Спектрограма відображає інтенсивність різних частот звукового сигналу протягом часу і може використовуватися для виявлення унікальних звукових особливостей.
4. STFT: часово-частотні перетворення, що розкладають сигнал на компоненти частоти з врахуванням часу, дозволяючи аналізувати зміни частотних характеристик у часі.

Ці форми перетворення аудіосигналу дозволяють моделі глибокого навчання ефективніше обробляти та аналізувати звукові дані, покращуючи точність та швидкість розпізнавання емоцій у голосі.

Серед усіх перетворень для тренування моделі було обрано мел-спектрограму, оскільки вона виділяє найважливіші частотні компоненти сигналу та зменшує кількість даних, з якими працює модель, що покращує результати моделі, знижує обчислювальні витрати та пришвидшує процес навчання. Варто також звернути увагу, що спектрограми несуть більше інформації, ніж слова текстової транскрипції, для розпізнавання емоцій людини. Деталі підготовки даних до тренування з використанням цього методу перетворення аудіосигналу будуть описані в наступному розділі.

Перейдемо до вибору методу розпізнавання емоцій. Проаналізувавши методи, описані в різних літературних джерелах, для реалізації моделі розпізнавання емоцій було обрано метод CNN.

Архітектура CNN є комбінацією трьох компонентів:

1) згорткові шари (Convolutional layers): містять певну кількість фільтрів, що застосовуються до вхідних даних. Кожен фільтр сканує вхідні дані, використовуючи метод точкового добутку та для створення певної кількості feature maps в одному згортковому шарі.
2) об'єднувальні шари (Pooling layers): використовуються для скорочення або зменшення feature maps об'єктів. Існує декілька схем зменшення розмірності: max pooling, min pooling, mean pooling, average pooling тощо.
3) повністю з’єднані шари (Fully connected layers): використовуються для вилучення глобальних ознак, які подаються до класифікатора SoftMax для визначення ймовірності для кожного класу.

CNN розташовує всі ці шари в ієрархічній структурі: згорткові шари (CL), об'єднувальні шари (PL), а потім повністю з’єднані шари (FCL), за якими слідує класифікатор SoftMax. [18]

Для кращого розуміння архітектури CNN наведемо приклад моделі, що складається з 2 згорткових, 2 об'єднувальних та 1 повністю з'єднаного шарів. ЇЇ візуалізацію можна побачити на Рис. 3.

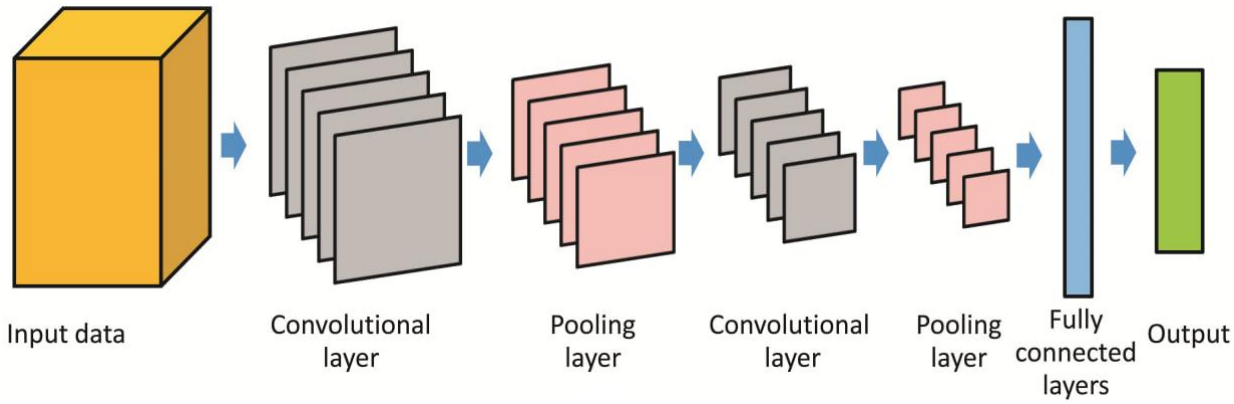

Рис. 3. CNN з 2 згортковими шарами, 2 об'єднувальними шарами та повністю з'єднаним шаром [19]

Перевагами використання даного методу є:

- вилучення високорівневих ознак з низькорівневої необробленої інформації про пікселі. Тобто, CNN може автоматично виявляти локальні патерни у вхідних мел-спектрограмах, такі як зміни в амплітуді та частоті, що є важливими для розпізнавання емоцій;

- завдяки шарам пулінгу, CNN є стійкою до змін масштабу та зсувів у вхідних даних, що робить модель більш гнучкою до різних варіантів аудіо сигналів;
- CNN можуть адаптуватися до складних і багатовимірних вхідних даних, таких як мел-спектрограми, завдяки своїй здатності до зменшення розмірності через згорткові та пулінгові шари.

Ці переваги роблять CNN хорошим вибором для завдань розпізнавання емоцій на основі аудіоданих.

5. ЕКСПЕРИМЕНТИ

Почнемо з підготовки даних до їх використання на тренуванні моделі. Після завантаження набору, дані було поділено на три частини: тренувальна, тестова, валідаційна. Випадковим чином було обрано по 2 актори для тестового і валідаційного набору, серед яких по одній людині кожної статі (1 чоловік, 1 жінка). Інші 20 акторів були розподілені до тренувального набору. Такий розподіл даних можна пояснити наступним фактором: при класифікації емоцій в реальному світі модель буде “чути” голос та манери звучання конкретної людини вперше, а отже, використання записів одного актора і в тренувальній, і в тестовій чи валідаційній частинах є недоцільним. Також варто врахувати рівномірність розподілу даних: нерівномірний розподіл даних негативно впливатиме на здатність моделі рівно розпізнавати емоції. Також необхідно додати записи кожного типу емоції до тестової та валідаційної частин, щоб мати можливість визначити точність розпізнавання кожної з емоцій.

Деталі розподілу емоцій для кожного з наборів можна розглянути на Рис. 4

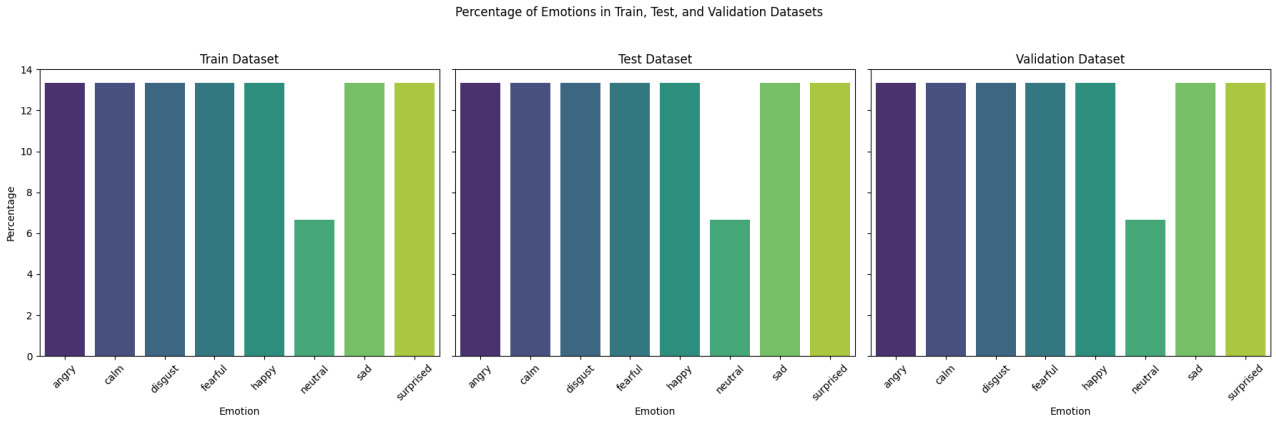

Рис. 4. Розподіл даних кожної емоції для тренувального, тестового та валідаційного наборів.

Після поділу даних на окремі набори, з назви кожного запису було витягнуто його ознаки та характеристики, зокрема: шлях до місця, де зберігається конкретний запис, модальність запису, його голосовий канал, відтворена емоція, емоційна інтенсивність, висловлення, спроба, актор та стать актора.

Переглянути вигляд отриманого набору можна в Таблиці 3. У ній описано кілька екземплярів записів тренувального набору. Тестовий та валідаційний набори мають такий же вигляд за винятком наповнення кожного з наборів екземплярами записів та їхньою кількістю. Як бачимо, в тренувальному наборі розміщено 1200 записів, а в тестовому і валідаційному - по 120.

Таблиця 3. Вигляд тренувального набору після його обробки

index	audio_file	Modality	Vocal_channel	Emotion	Emotional_intensity	Statement	Repetition	Actor	Gender
0	train_dataset/Actor_14/03-01-04-02-01-01-14.wav	audio-only	speech	sad	strong	Kids are talking by the door	1st repetition	14	Female
1	train_dataset/Actor_14/03-01-02-02-01-02-14.wav	audio-only	speech	calm	strong	Kids are talking by the door	2nd repetition	14	Female
2	train_dataset/Actor_14/03-01-07-02-01-02-14.wav	audio-only	speech	disgust	strong	Kids are talking by the door	2nd repetition	14	Female
3	train_dataset/Actor_14/03-01-06-01-02-02-14.wav	audio-only	speech	fearful	normal	Dogs are sitting by the door	2nd repetition	14	Female
…	…	…	…	…	…	…	…	…	…
1199	train_dataset/Actor_24/03-01-08-02-02-02-24.wav	audio-only	speech	surprised	strong	Dogs are sitting by the door	2nd repetition	24	Female

Як бачимо з Таблиці 3, набір містить багато деталей запису, зокрема не тільки відтворену емоцію, а й її інтенсивність, сказане висловлення, номер спроби, ідентифікатор актора та його стать. У Розділі 4 було вказано, що використання таких даних є непотрібним для моделі. Тому такі дані було вилучено з наборів. Також варто зауважити, що модальність та голосовий канал аудіо однакові для всіх екземплярів, тому зберігати їх також не потрібно, як і враховувати при тренування моделі, оскільки вони не допомагатимуть при розпізнаванні емоцій. Отже, після вилучення всіх непотрібних ознак в наборах залишилися лише шлях до аудіозапису та виявлена емоція. Самі записи, що знаходяться за вказаними в наборі шляхами було перетворено в мел-спектрограму, як було вказано у Розділі 4. Перейдемо до опису деталей підготовки даних до тренування.

Спочатку візуалізуємо екземпляр звукових даних за допомогою waveplot - графічного зображення аудіосигналу в часі, що показує, як змінюється амплітуда звукового сигналу протягом часу. Це базова візуалізація, яка допомагає зрозуміти загальну структуру звукового файлу. Приклад такого зображення можна побачити на Рис. 5.

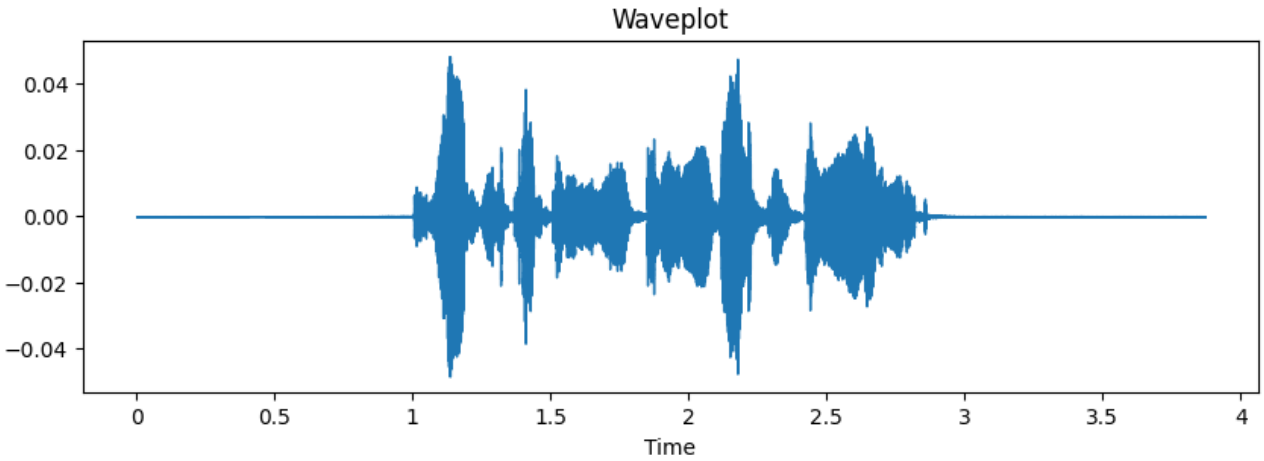

Рис. 5. Графічне зображення одного з екземплярів аудіозапису тренувального набору

Перейдемо до побудови мел-спектрограми (надалі спектрограми). На Рис. 6 зображено приклад спектрограми для одного з аудіозаписів. На ньому видно, як змінюється спектральна енергія звукового сигналу протягом часу. Спектрограма використовує мел-шкалу для групування частот, що відповідає сприйняттю звуку людиною. Кожен піксель на спектрограми відображає енергію для певного діапазону частот у певний момент часу. Для побудови спектрограми було використано бібліотеку `librosa`.

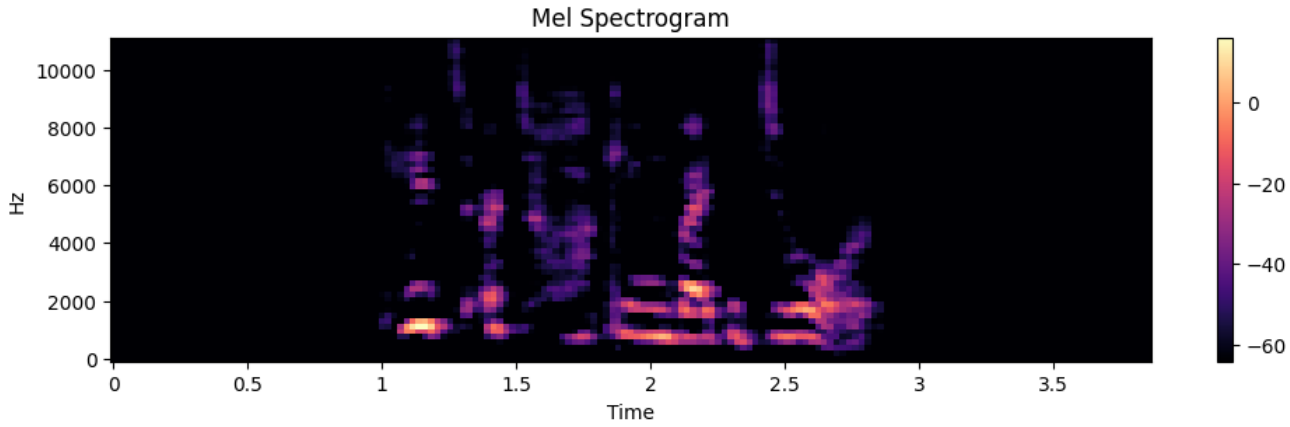

Рис. 6. Мел-спектрограма одного з екземплярів аудіозапису тренувального набору

Як було згадано в Розділі 4, задля покращення точності моделі поточний набір даних можна розширити, додавши більше даних. Проте не обов'язково шукати дані з інших наборів. Ще одним варіантом доповнення набору є аугментація даних.

Аугментація даних - це процес створення нових даних на основі існуючих шляхом застосування випадкових перетворень, які викликають зміни в оригінальних даних [20]. Цей процес використовується для збільшення кількості даних та різноманітності в наборі даних, що може допомогти покращити загальну продуктивність моделі та її здатність узагальнювати.

У даному випадку, аугментація даних включатиме три основні етапи:

1) Зміщення та доповнення: зміщення аудіосигналу в часі та його доповнення до певної довжини, що може допомогти моделі краще вчитися з записів різної довжини та з різними часовими зміщеннями. Переглянути вигляд аудіосигналу після застосувань змін даного етапу можна на Рис. 8.
2) Зміна швидкості та висоти звуку: може допомогти моделі краще вчитися з аудіосигналів з різними швидкостями та висотами звуку. Переглянути вигляд аудіосигналу після застосувань змін даного етапу можна на Рис. 9.
3) Зашумлення: додавання випадкового шуму до аудіосигналу, що може допомогти моделі краще вчитися з аудіосигналів в умовах шуму. Переглянути вигляд аудіосигналу після застосувань змін даного етапу можна на Рис. 10.

Для кожного запису було застосовано наступні набори етапів аугментації: перший та другий, перший та третій. Внаслідок цього поточномий набір вдалося збільшити з 1200 до 3600 записів. Варто зауважити, що початкові дані

входять до новоствореного набору. Вигляд початкового запису зображено на Рис. 7, на Рис. 8-10 - записи після застосування аугментації.

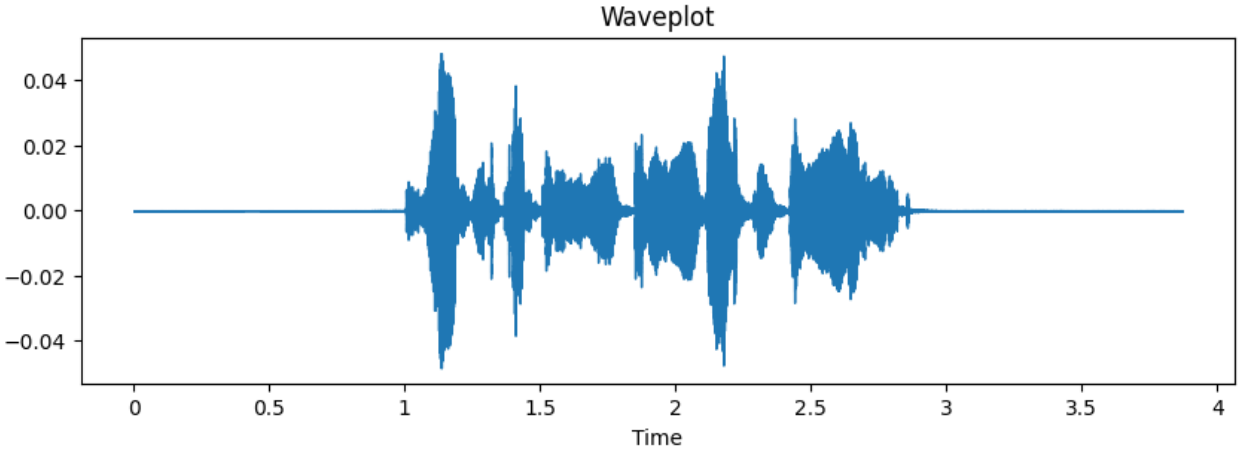

Рис. 7. Графічне зображення початкового екземпляру аудіозапису тренувального набору

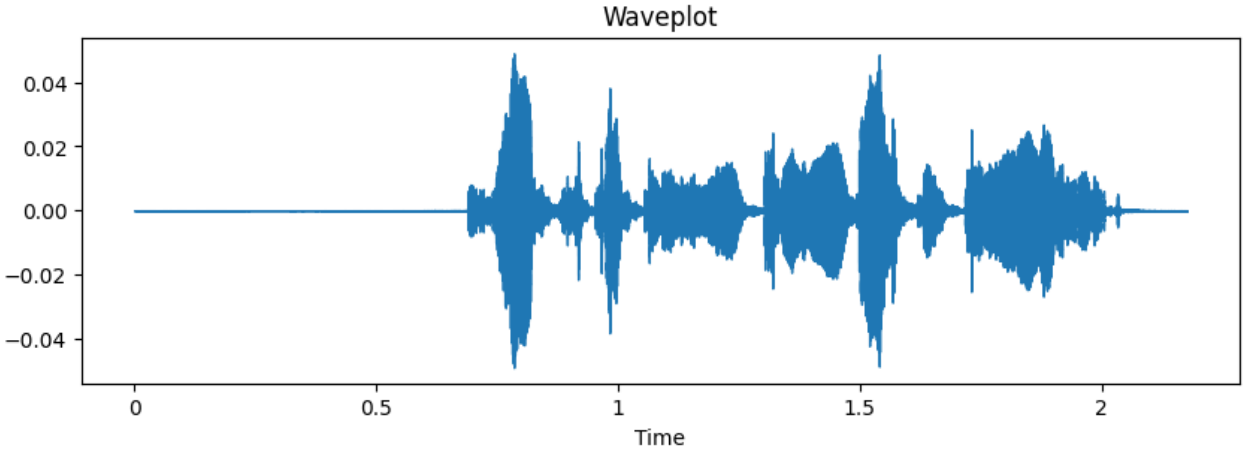

Рис. 8. Графічне зображення екземпляру аудіозапису після зміщення та доповнення

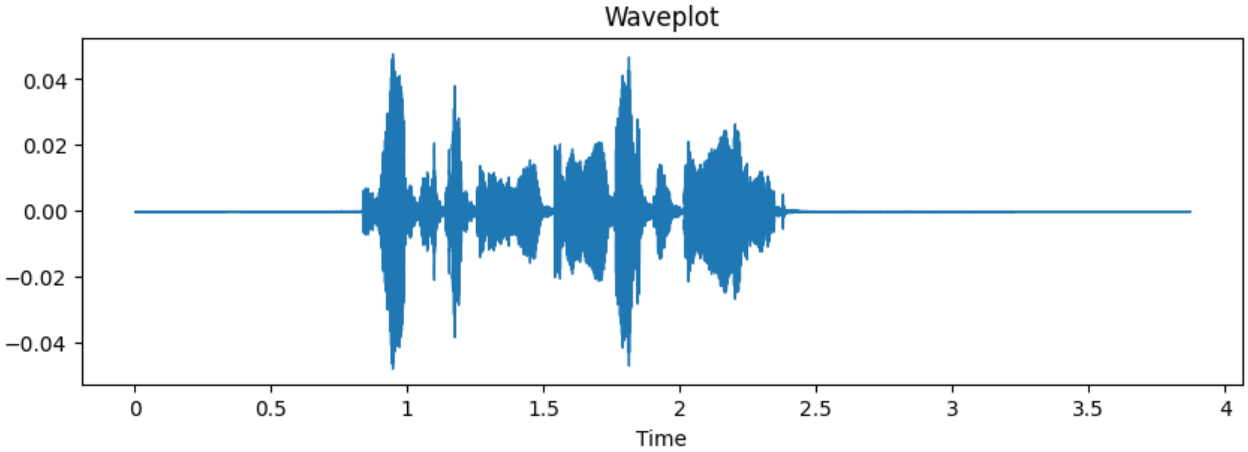

Рис. 9. Графічне зображення екземпляру аудіозапису після зміни швидкості та висоти звуку

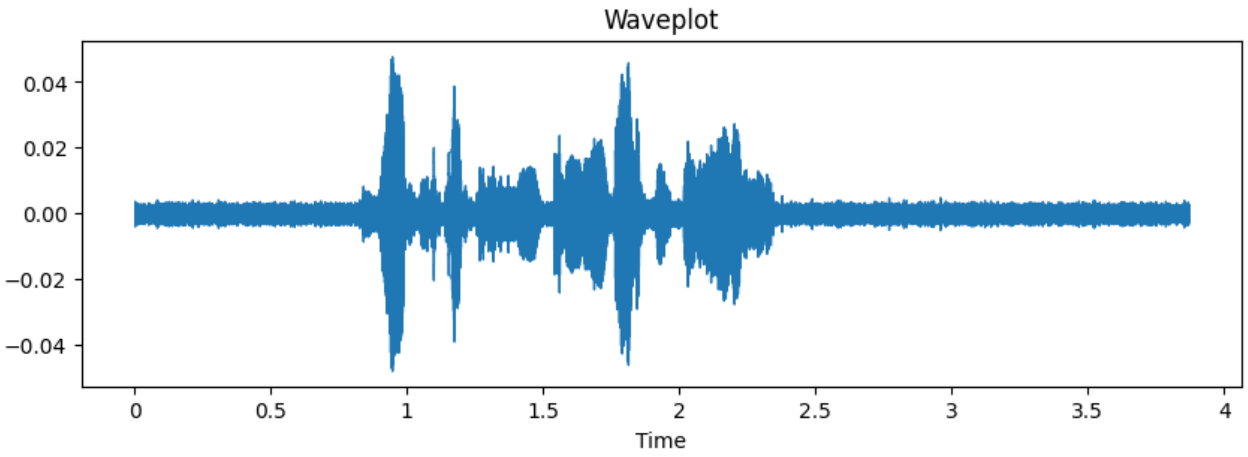

Рис. 10. Графічне зображення екземпляру аудіозапису після зашумлення

Перейдемо до побудови архітектури моделі. Крім основних шарів моделі CNN, описаних у Розділі 4, були додані шари нормалізації (Batch Normalization) та відсічки (Dropout), оскільки їх використання сприяє стабільному та ефективному процесу навчання, запобігаючи перенавчанню та покращуючи загальну узагальненість моделі.

Отже, модель складається з чотирьох основних блоків, кожен з яких містить згортковий шар (Conv2D), шар нормалізації (BatchNormalization), шар активації ReLU (Activation), шар пулінгу (MaxPooling2D) та шар випадкового відключення (Dropout). Згорткові шари використовуються для виявлення корисних ознак з вхідних даних, нормалізації - для стабілізації процесу навчання, активації ReLU - для додавання нелінійності до моделі, максимального пулінгу - для зменшення розмірності даних, а шари випадкового відключення - для запобігання перенавчанню.

Після цих блоків, дані вирівнюються та проходять через два повнозв'язних шари (Dense), які призначені для виявлення складних шаблонів в даних. На останньому шарі використовується функція активації softmax для виведення ймовірностей для кожної з емоцій.

Очікується, що архітектура цієї моделі добре підійде для задачі класифікації емоцій, оскільки вона може виявляти корисні ознаки на різних рівнях абстракції та покаже високі результати точності.

Для візуалізації архітектури отриманої моделі виведемо результат підсумовування моделі (model.summary):

```
Model: "model_melspec"
_________________________________________________________________
 Layer (type)                                Output Shape              Param #
=================================================================
 input_1 (InputLayer)                        [(None, 60, 94, 1)]       0
 conv2d (Conv2D)                             (None, 60, 94, 32)        1312
 batch_normalization (BatchNormalization)    (None, 60, 94, 32)        128
 activation (Activation)                     (None, 60, 94, 32)        0
 max_pooling2d (MaxPooling2D)                (None, 30, 47, 32)        0
 dropout (Dropout)                           (None, 30, 47, 32)        0
 conv2d_1 (Conv2D)                           (None, 30, 47, 32)        40992
 batch_normalization_1 (BatchNormalization)  (None, 30, 47, 32)        128
 activation_1 (Activation)                   (None, 30, 47, 32)        0
 max_pooling2d_1 (MaxPooling2D)              (None, 15, 23, 32)        0
 dropout_1 (Dropout)                         (None, 15, 23, 32)        0
 conv2d_2 (Conv2D)                           (None, 15, 23, 32)        40992
 batch_normalization_2 (BatchNormalization)  (None, 15, 23, 32)        128
 activation_2 (Activation)                   (None, 15, 23, 32)        0
 max_pooling2d_2 (MaxPooling2D)              (None, 7, 11, 32)         0
 dropout_2 (Dropout)                         (None, 7, 11, 32)         0
 conv2d_3 (Conv2D)                           (None, 7, 11, 32)         40992
 batch_normalization_3 (BatchNormalization)  (None, 7, 11, 32)         128
 activation_3 (Activation)                   (None, 7, 11, 32)         0
 max_pooling2d_3 (MaxPooling2D)              (None, 3, 5, 32)          0
 dropout_3 (Dropout)                         (None, 3, 5, 32)          0
 flatten (Flatten)                           (None, 480)               0
 dense (Dense)                               (None, 64)                30784
 dense_1 (Dense)                             (None, 256)               16640
 emotion_output (Dense)                      (None, 8)                 2056
=================================================================
Total params: 174280 (680.78 KB)
Trainable params: 174024 (679.78 KB)
Non-trainable params: 256 (1.00 KB)
_________________________________________________________________
```

Процес навчання моделі включатиме наступні етапи:

- Ініціалізація моделі: створення моделі з використанням визначеної архітектури. У нашому випадку модель очікуватиме вхідні дані у формі двовимірних Mel-spectrogram.
- Компіляція моделі: використання оптимізатора Adam для оптимізації ваг моделі та метрики функції втрат "categorical_crossentropy" для вимірювання помилки моделі під час навчання.
- Навчання моделі: навчання відбудеться на наборі початкових тренувальних даних та даних, отриманих після аугментації. Тобто, ми отримаємо дві моделі, натреновані на різних наборах. У процесі навчання використовуватимуться такі методи, як раннє зупинення

(EarlyStopping) та збереження кращої моделі (ModelCheckpoint), щоб запобігти перенавчанню та забезпечити, що модель, яка дає найкращі результати на валідаційних даних, зберігається.

- Валідація моделі: перевірка моделі на валідаційному наборі даних. Це допоможе переконатися, що модель не лише добре працює на тренувальних даних, але й може узагальнити свої висновки на нових даних, які вона не бачила раніше.
- Тестування моделі: тестування моделі на тестовому наборі даних, щоб перевірити, як вона працює на даних, які вона ніколи не бачила.

Кількість епох дорівнюватиме 100, розмір блоку (batch_size) - 32.

Перейдемо до отриманих результатів тренування моделі на початкових даних тренувального набору. Як бачимо на Рис. 11, навчання перервалося на 77 епосі. Найбільша точність тренування, якої вдалося досягти, рівна 0.96, точність класифікації валідаційного набору сягнула 0.79. Аналізуючи отримані тренувальні та валідаційні результати, можемо припустити, що модель перенавчилася. Це означає, що модель занадто добре “запам’ятала” тренувальні дані та погано узагальнює їх на нових даних. Це може призвести до погіршення її продуктивності на тестовому наборі даних.

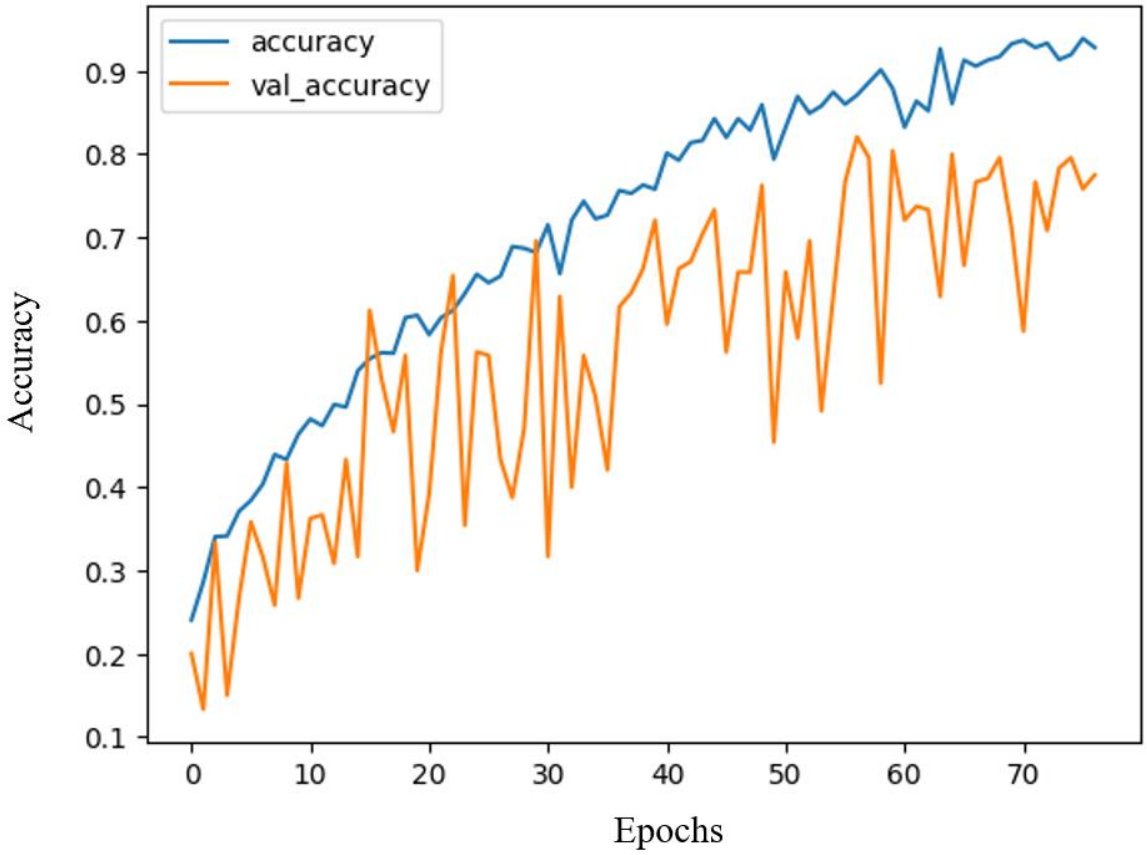

Рис. 11. Результат точності тренування моделі на початковому тренувальному наборі даних на різних епохах

Аналізуючи отримані результати тренування моделі на аугментованих даних тренувального набору, можемо припустити, що модель навчилася краще, оскільки точність класифікації валідаційного набору не перестає збільшуватися з плином епох. Також варто зауважити, що найбільша точність тренування даної моделі, якої вдалося досягти, рівна 0.91, проте точність класифікації валідаційного набору досягла максимального результату (1.0). На Рис. 12, бачимо, що навчання перервалося на 49 епосі.

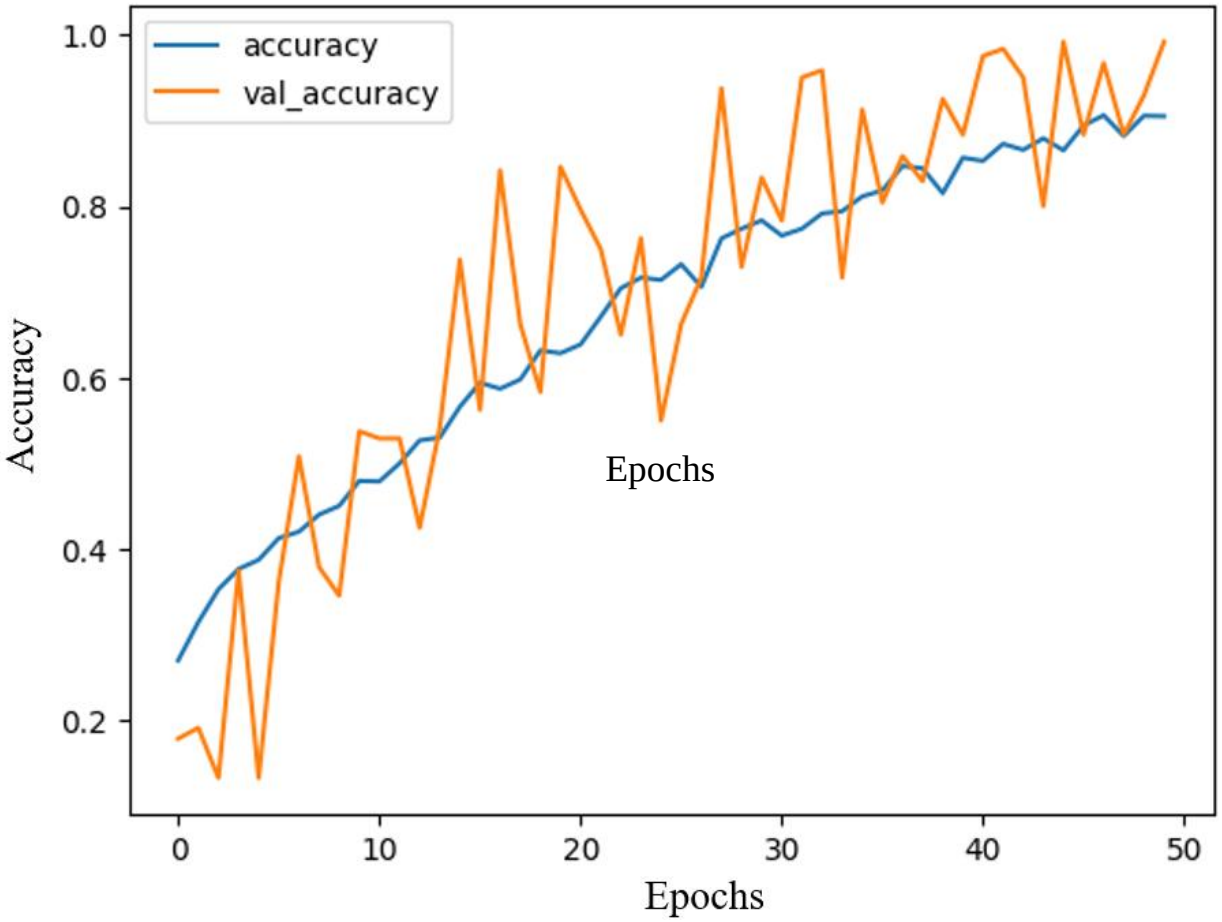

Рис. 12. Результат точності тренування моделі на наборі аугментованих даних на різних епохах

Отримані результати навчання обох моделей свідчать про наступне: точність валідаційного набору аугментованого набору даних набагато вища, ніж моделі, натренованої на початковому наборі. Звідси можемо припустити, що друга модель краще класифікує емоції на тестовому наборі, ніж перша. Детальний опис результатів тестування моделі на нових (тестових) даних розміщено в наступному розділі.

6. ОБГОВОРЕННЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

Протестувавши отримані моделі, на даних, які вони ще не бачили, було отримано різні результати. Для того, щоб оцінити роботу моделей, було використано та обраховано наступні метрики:

- Точність (Accuracy): відношення правильно передбачених випадків до загальної кількості випадків. Ця метрика використовується для оцінки загальної продуктивності моделі.
- Точність (Precision): відношення правильно передбачених позитивних випадків до загальної кількості передбачених позитивних випадків. Висока точність вказує на меншу кількість помилково позитивних результатів.
- Повнота (Recall): відношення правильно передбачених позитивних випадків до загальної кількості дійсно позитивних випадків. Висока повнота вказує на меншу кількість помилково негативних результатів.
- F1-метрика (F1-score): вимірює узагальнену оцінку ефективності моделі, поєднуючи accuracy та recall.

Для кращої візуалізації розпізнавання емоцій було побудовано матрицю відповідностей. Матриця відповідностей (confusion matrix), є інструментом для оцінки продуктивності алгоритмів класифікації. Вона дозволяє візуалізувати якість моделі шляхом порівняння фактичних класів з класами, передбаченими моделлю.

Основні елементи матриці відповідностей:

- True Positives (TP): Кількість випадків, які належать до певного класу і були правильно визначені як цей клас.
- True Negatives (TN): Кількість випадків, які не належать до певного класу і були правильно визначені не як цей клас.
- False Positives (FP): Кількість випадків, які не належать до певного класу, але були помилково визначені як цей клас.
- False Negatives (FN): Кількість випадків, які належать до певного класу, але були помилково визначені як такі, що не належать до цього класу.

Завдяки цим елементам можна визначити описані вище метрики за наступними формулами:

$$Accuracy = \frac{TN + TP}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall}$$

Варто зазначити, що матриця відповідностей надає цінну інформацію про те, як модель класифікує кожен клас, і дозволяє візуально легко розпізнати, чи є певні класи, які модель розрізняє краще або гірше та чи можливі зміщення в моделі та напрямки для подальшого поліпшення. У нашому випадку, матриця відповідностей відображає ефективність моделі класифікації у розпізнаванні різних емоцій. Кожен стовпчик матриці відповідає передбаченому класу, а кожен рядок - реальному класу. Це дозволяє нам бачити, які емоції модель розпізнає правильно, а які - ні.

Проаналізуємо метрики, отримані при тестуванні моделі на початковому наборі даних. Їх можна побачити на Рис. 12. Як бачимо, загальна точність, якої вдалося досягти, рівна 0.72. Також по отриманих метриках бачимо, що модель добре розрізняє емоції щастя, здивування, спокою та страху, трохи гірше - емоції злості, суму та нейтральності. Найгірше модель впізнає емоцію огиди.

	precision	recall	f1-score	support
calm	0.78	0.75	0.77	24
disgust	0.57	0.96	0.72	24
neutral	0.60	0.75	0.67	24
surprised	0.95	0.75	0.84	24
fearful	0.74	0.83	0.78	24
sad	0.60	0.50	0.55	12
angry	0.67	0.42	0.51	24
happy	1.00	0.67	0.80	24
accuracy			0.72	180
macro avg	0.74	0.70	0.70	180
weighted avg	0.75	0.72	0.71	180

Рис. 12. Метрики тестування моделі, натренованої на початковому наборі даних

Матрицю, побудовану для моделі, натренованої на початкових даних, можна побачити на Рис. 13.

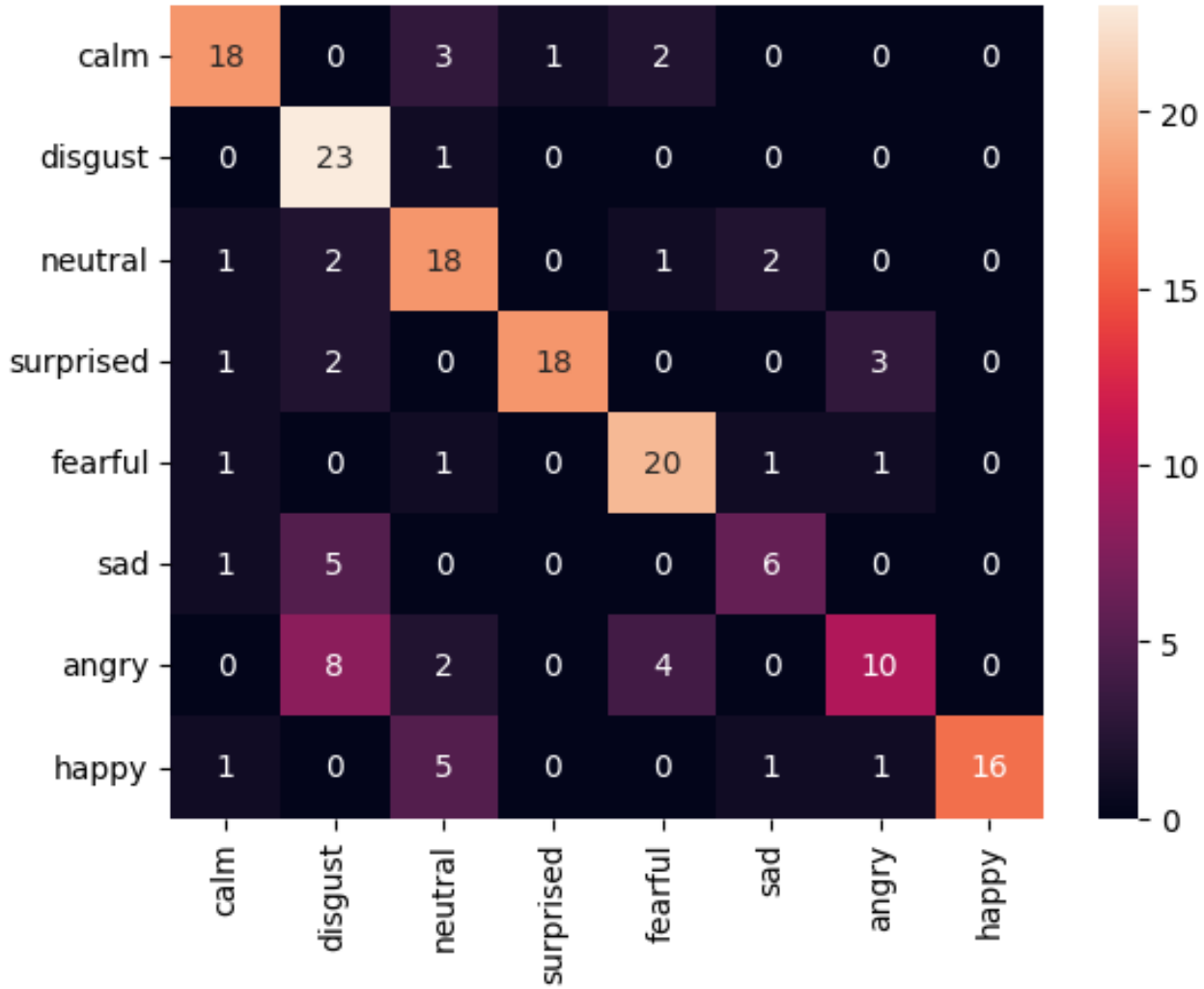

Рис. 13. Матриця відповідностей розпізнавання емоцій моделі, натренованої на початковому наборі даних

Тепер розглянемо метрики, зображені на Рис. 14. Це результати, отримані при тестуванні моделі на аугментованих даних. Як бачимо, загальна точність, якої вдалося досягти, дорівнює 0.81, що на 9% краще, ніж результат попередньої моделі. Натренована модель добре розрізняє емоції здивування, спокою, щастя, нейтральності та огиди, проте погано - емоції страху, суму та злості.

	precision	recall	f1-score	support
calm	0.95	0.75	0.84	24
disgust	0.83	0.83	0.83	24
neutral	0.89	0.67	0.76	24
surprised	1.00	0.88	0.93	24
fearful	0.71	0.83	0.77	24
sad	0.61	0.92	0.73	12
angry	0.58	0.62	0.60	24
happy	0.92	1.00	0.96	24
accuracy			0.81	180
macro avg	0.81	0.81	0.80	180
weighted avg	0.83	0.81	0.81	180

Рис. 14. Метрики тестування моделі, натренованої на початковому наборі даних

Матрицю відповідностей, побудовану для моделі, натренованої на аугментованому наборі даних, можна побачити на Рис. 15.

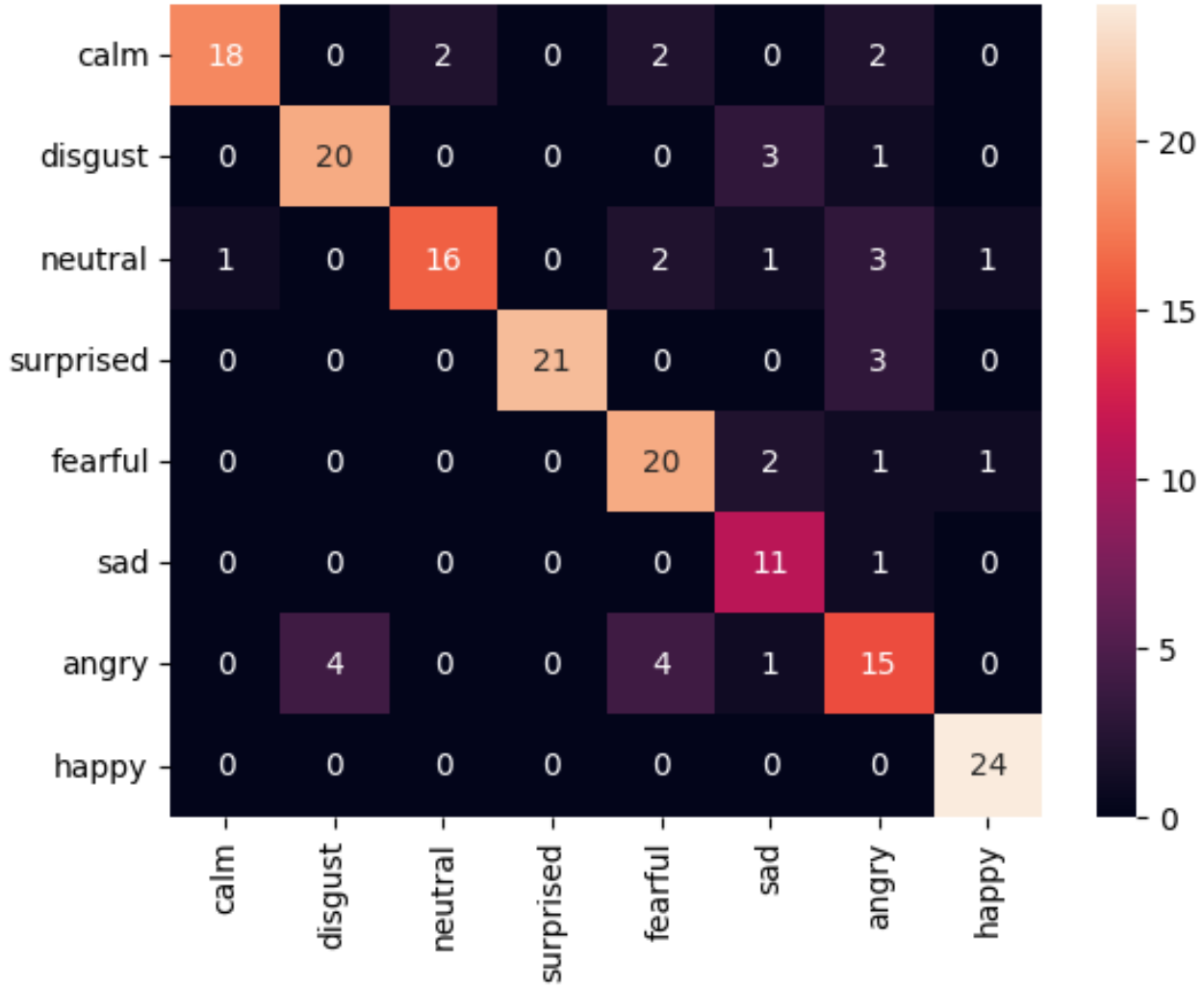

Рис. 15. Матриця відповідностей розпізнавання емоцій моделі, натренованої на аугментованому наборі даних

Як бачимо, модель, натренована на аугментованих даних, показала кращі результати за всіма метриками. Це свідчить про те, що аугментація даних допомогла моделі краще узагальнити і працювати з новими даними. Проте варто врахувати, що для навчання CNN моделі потрібно набагато більше даних. Оскільки актори між наборами розподіляються випадковим чином, складно передбачити, який результат тренування ми отримаємо. Як в тестовий, так і у валідаційний набори можуть попасти записи актора, не схожі з іншими записами, що призведе до неправильної оцінки роботи моделі. Результати, отримані при іншому розподілі акторів можуть бути як гіршими, так і кращими, оскільки ми оцінюємо роботу моделі на надто малій вибірці даних. Тобто, для покращення результатів роботи моделі необхідно пропорційно збільшити набір акторів для кожного з наборів даних. Підтвердженням правильності цієї ідеї є те, що модель, натренована на аугментованому, тобто розширеному наборі даних, показала кращі результати, тому прогнозовано, що зі збільшенням

екземплярів записів модель покаже кращі результати, а результати валідації та тестування будуть більш усередненими. Проблемою значного збільшення набору є те, що тренування моделі займатиме чимало часу. Тренування моделей, описаних у цій роботі, зайняли мало часу, до 2-ох хвилин на все тренування для обох випадків, хоч і перший набір утричі менший від другого, аугментованого. Тому можна припустити, що при помірному збільшенні набору даних час виконання не підніметься до критичного.

7. ВИСНОВКИ

У результаті проведеного дослідження було отримано дві моделі, одна з яких було натренована на початкових даних, тобто мел-спектрограмах, отриманих з початкових аудіозаписів набору, а друга - на доповненому наборі з аугментованих даних. Результати показали, що друга модель розпізнає емоції краще, ніж перша. Найбільша точність, якої вдалося досягти, дорівнює 81%. Це результат, отриманий при тестуванні моделі, натренованої на аугментованих даних, що є на 9% кращим результатом, ніж результати першої моделі. Звідси можна зробити висновок, що для покращення отриманих результатів можна збільшити поточний набір даних, взявши їх з повного набору RAVDESS та обробивши відеозаписи в аудіозаписи. Загалом, моделі глибокого навчання, такі як CNN, зазвичай потребують більшої кількості даних для ефективного навчання.

У ході роботи було розглянуто багато методів розпізнавання емоцій, також було розроблено власну архітектуру моделі на основі CNN та натреновано її на двох наборах даних, один з яких містив мел-спектрограми, створені з аудіозаписів, взятих з набору даних RAVDESS, а другий - мел-спектрограми, отримані з аугментованих даних першого набору. Проведений аналіз вже існуючих результатів дав змогу зробити багато висновків щодо того, як працює система розпізнавання емоцій в цілому. Розроблені моделі показали непоганий результат класифікації, зокрема 81% точності при класифікації 8 емоцій. Отже, можемо зробити висновок, що в даній роботі вдалося виконати поставлені мету та завдання роботи. При розподілі тренувального, тестового та валідаційного наборів було враховано сфери застосування моделі, зокрема визначення, чи не є промова монотонною, та подальше передбачення того, де необхідно змінити емоцію голосу задля покращення промови в цілому. Для цього до тестового та валідаційного наборів було віднесено всі дані кількох акторів, щоб модель “чула” записи цих людей вперше.

Загалом, точність, якої вдалося досягти є хорошим результатом у порівнянні з розглянутими джерелами. Крім цього, в роботі було описано спосіб покращення досягнення точності, розширивши поточний набор даних.

Таким чином, застосування CNN для розпізнавання емоцій голосу людини є хорошим варіантом. Отримані результати все ще можна покращити, збільшивши набір даних. У майбутньому ця модель може стати елементом системи, що аналізуватиме промову людини та робитиме пропозиції щодо її вдосконалення.

8. СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] Lieskovská, E.; Jakubec, M.; Jarina, R.; Chmulík, M. A Review on Speech Emotion Recognition Using Deep Learning and Attention Mechanism. Electronics 2021, 10, 1163. https://doi.org/10.3390/electronics10101163

[2] Kamińska, D., Sapiński, T. & Anbarjafari, G. Efficiency of chosen speech descriptors in relation to emotion recognition. J AUDIO SPEECH MUSIC PROC. 2017, 3 (2017). https://doi.org/10.1186/s13636-017-0100-x

[3] Kerkeni, Leila & Serrestou, Youssef & Raoof, Kosai & Cléder, Catherine & Mahjoub, Mohamed & Mbarki, Mohamed. (2019). Automatic Speech Emotion Recognition Using Machine Learning. 10.5772/intechopen.84856.

[4] T. M. Wani, T. S. Gunawan, S. A. A. Qadri, M. Kartiwi and E. Ambikairajah, "A Comprehensive Review of Speech Emotion Recognition Systems," in IEEE Access, vol. 9, pp. 47795-47814, 2021, doi: 10.1109/ACCESS.2021.3068045.

[5] Kerkeni, L.; Serrestou, Y.; Mbarki, M.; Raoof, K. and Mahjoub, M. (2018). Speech Emotion Recognition: Methods and Cases Study. In Proceedings of the 10th International Conference on Agents and Artificial Intelligence - Volume 1: ICAART; ISBN 978-989-758-275-2; ISSN 2184-433X, SciTePress, pages 175-182. DOI: 10.5220/0006611601750182

[6] Lanjewar, Rahul & Mathurkar, Swarup & Patel, Nilesh. (2015). Implementation and Comparison of Speech Emotion Recognition System Using Gaussian Mixture Model (GMM) and K- Nearest Neighbor (K-NN) Techniques. Procedia Computer Science. 49. 50-57. 10.1016/j.procs.2015.04.226.

[7] Deriche, M., Abo absa, A.H. “A Two-Stage Hierarchical Bilingual Emotion Recognition System Using a Hidden Markov Model and Neural Networks”. Arab J Sci Eng 42, 5231–5249 (2017). https://doi.org/10.1007/s13369-017-2742-5

[8] Manas Jain and Shruthi Narayan and Pratibha Balaji and Bharath K P and Abhijit Bhowmick and Karthik R and Rajesh Kumar Muthu, “Speech Emotion Recognition using Support Vector Machine”, 2020.

[9] Alnuaim AA, Zakariah M, Shukla PK, Alhadlaq A, Hatamleh WA, Tarazi H, Sureshbabu R, Ratna R. Human-Computer Interaction for Recognizing Speech Emotions Using Multilayer Perceptron Classifier. J Healthc Eng. 2022 Mar 28;2022:6005446. doi: 10.1155/2022/6005446. PMID: 35388315; PMCID: PMC8979705.

[10] Zwol, Björn & Langezaal, Mathijs & Arts, Lukas & Gatt, Albert & van den Broek, Egon L.. (2023). Speech Emotion Recognition Using Deep Convolutional

Neural Networks Improved by the Fast Continuous Wavelet Transform. 10.3233/AISE230012.

[11] Y. Xi, P. Li, Y. Song, Y. Jiang and L. Dai, "Speaker to Emotion: Domain Adaptation for Speech Emotion Recognition with Residual Adapters," 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Lanzhou, China, 2019, pp. 513-518, doi: 10.1109/APSIPAASC47483.2019.9023339.

[12] Mirsamadi, Seyedmahdad & Barsoum, Emad & Zhang, Cha. (2017). Automatic Speech Emotion Recognition Using Recurrent Neural Networks with Local Attention. 10.1109/ICASSP.2017.7952552.

[13] Akshay S. Utane, S. L. Nalbalwar. EMOTION RECOGNITION through SPEECH. 2nd National Conference on Innovative Paradigms in Engineering and Technology (NCIPET 2013). NCIPET, 1 (November 2013), 0-0.

[14] Ayadi, M.M., Kamel, M.S., & Karray, F. Survey on speech emotion recognition: Features, classification schemes, and databases. 2011. Pattern Recognit., 44, 572-587.

[15] Fan, Weiquan & Xu, Xiangmin & Xing, Xiaofen & Chen, Weidong & Huang, Dongyan. (2021). LSSED: a large-scale dataset and benchmark for speech emotion recognition.

[16] Livingstone SR, Russo FA (2018) The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English. PLoS ONE 13(5): e0196391. https://doi.org/10.1371/journal.pone.0196391.

[17] Niko Laskaris. How to apply machine learning and deep learning methods to audio analysis. 2019.

[18] Mustaqeem; Kwon, S. A CNN-Assisted Enhanced Audio Signal Processing for Speech Emotion Recognition. Sensors 2020, 20, 183. https://doi.org/10.3390/s20010183

[19] Monkam, Patrice & Qi, Shouliang & Ma, He & Gao, Weiming & Yao, Yudong & Qian, Wei. (2019). Detection and Classification of Pulmonary Nodules Using Convolutional Neural Networks: A Survey. IEEE Access. PP. 1-1. 10.1109/ACCESS.2019.2920980.

[20] Olga Chernytska. Complete Guide to Data Augmentation for Computer Vision. 2021. Towards Data Science.

9. ДОДАТКИ

ДОДАТОК 1

У цьому додатко наведено код, що використовувався в цій роботі.

```
    !pip install opendatasets --upgrade --quiet
    !pip install --upgrade librosa
    # !pip install --upgrade resampy
    # Base Libraries
    import os
    import shutil
    import random
    import librosa
    import numpy as np
    import pandas as pd
    import seaborn as sns
    import tensorflow as tf
    import opendatasets as od
    import matplotlib.pyplot as plt

    # Preprocessing and Image
    import cv2
    import librosa.display
    import IPython.display as ipd
    from sklearn.model_selection import train_test_split
    from keras.utils import image_dataset_from_directory
    from tensorflow.keras.models import load_model
    from tensorflow.keras.preprocessing import image

    import warnings
    warnings.filterwarnings("ignore")
    dataset =
'https://www.kaggle.com/datasets/uwrfkaggler/ravdess-emotional-
speech-audio/data'
    # Using opendatasets let's download the data sets
    od.download(dataset, force=True)
    label_to_char = {
        'Modality': {
            1: "full-AV",
            2: "video-only",
            3: "audio-only"
        },
```

```
    'Vocal_channel': {
        1: "speech",
        2: "song"
    },
    'Emotion': {
        1: "neutral",
        2: "calm",
        3: "happy",
        4: "sad",
        5: "angry",
        6: "fearful",
        7: "disgust",
        8: "surprised"
    },
    'Emotional_intensity': {
        1: "normal",
        2: "strong"
    },
    'Statement': {
        1: "Kids are talking by the door",
        2: "Dogs are sitting by the door"
    },
    'Repetition': {
        1: "1st repetition",
        2: "2nd repetition"
    },
    # 'Actor': parts[6],
    'Gender': {
        1: "Male",
        2: "Female"
    }
}

char_to_label = {
    'Modality': {
        "full-AV": 1,
        "video-only": 2,
        "audio-only": 3
    },
    'Vocal_channel': {
        "speech": 1,
        "song": 2
    },
    'Emotion': {
```

```
            "neutral": 1,
            "calm": 2,
            "happy": 3,
            "sad": 4,
            "angry": 5,
            "fearful": 6,
            "disgust": 7,
            "surprised": 8
        },
        'Emotional_intensity': {
            "normal": 1,
            "strong": 2
        },
        'Statement': {
            "Kids are talking by the door": 1,
            "Dogs are sitting by the door": 2
        },
        'Repetition': {
            "1st repetition": 1,
            "2nd repetition": 2
        },
        # 'Actor': parts[6],
        'Gender': {
            "Male": 1,
            "Female": 2
        }
    }
    # Define paths
    data_dir = './ravdess-emotional-speech-audio'
    train_dir = './train_dataset'
    test_dir = './test_dataset'
    val_dir = './val_dataset'

    # Create directories if they don't exist
    os.makedirs(train_dir, exist_ok=True)
    os.makedirs(test_dir, exist_ok=True)
    os.makedirs(val_dir, exist_ok=True)
    content_list = [folder for folder in os.listdir(data_dir) if
os.path.isdir(os.path.join(data_dir, folder))]
    content_list.remove('audio_speech_actors_01-24')
    content_list
    # Randomly select 2 actors for test and 2 for validation
    test_male, test_female = False, False
    val_male, val_female = False, False
```

```
    test_actors, val_actors, train_actors = [], [], []
    random.shuffle(content_list)
    for item in content_list:
      name_items = item.split('_')
      actor_id = int(name_items[1])
      if actor_id % 2 == 0:
        if not test_female:
          test_actors.append(item)
          test_female = True
        elif not val_female:
          val_actors.append(item)
          val_female = True
        else:
          train_actors.append(item)
      else:
        if not test_male:
          test_actors.append(item)
          test_male = True
        elif not val_male:
          val_actors.append(item)
          val_male = True
        else:
          train_actors.append(item)
    print("Train_actors:", train_actors)
    print("Test_actors:", test_actors)
    print("Val_actors:", val_actors)
    def copy_directory(src, dst):
        if os.path.exists(dst):
            shutil.rmtree(dst)
        shutil.copytree(src, dst)

    # Move actor folders to respective datasets
    for actor_folder in content_list:
        source_path = os.path.join(data_dir, actor_folder)
        if actor_folder in test_actors:
            destination_path = os.path.join(test_dir,
actor_folder)
        elif actor_folder in val_actors:
            destination_path = os.path.join(val_dir, actor_folder)
        else:
            destination_path = os.path.join(train_dir,
actor_folder)
        copy_directory(source_path, destination_path)
```

```
    # Function to extract labels from filename
    def extract_labels(filename):
        filename_without_extension = os.path.splitext(filename)[0]
        parts = filename_without_extension.split('-')
        int_parts = [int(part) for part in parts]
        return {
            'Modality': char_to_label['Modality'][parts[0]],
            'Vocal_channel':
char_to_label['Vocal_channel'][parts[1]],
            'Emotion': char_to_label['Emotion'][parts[2]],
            'Emotional_intensity':
char_to_label['Emotional_intensity'][parts[3]],
            'Statement': char_to_label['Statement'][parts[4]],
            'Repetition': char_to_label['Repetition'][parts[5]],
            'Actor': parts[6],
            'Gender': char_to_label['Gender'][2] if int(parts[6])
% 2 == 0 else char_to_label['Gender'][1]
        }

    def extract_chars(filename):
        filename_without_extension = os.path.splitext(filename)[0]
        parts = filename_without_extension.split('-')
        int_parts = [int(part) for part in parts]
        return {
            'Modality': label_to_char['Modality'][int_parts[0]],
            'Vocal_channel':
label_to_char['Vocal_channel'][int_parts[1]],
            'Emotion': label_to_char['Emotion'][int_parts[2]],
            'Emotional_intensity':
label_to_char['Emotional_intensity'][int_parts[3]],
            'Statement': label_to_char['Statement'][int_parts[4]],
            'Repetition':
label_to_char['Repetition'][int_parts[5]],
            'Actor': int_parts[6],
            'Gender': label_to_char['Gender'][2] if
int(int_parts[6]) % 2 == 0 else label_to_char['Gender'][1]
        }

    # Function to read audio files and extract labels
    # def load_dataset(data_dir):
    #     audio_files = []
    #     labels = []
    #     for root, dirs, files in os.walk(data_dir):
    #         for file in files:
```

```
    #             if file.endswith(".wav"):
    #                 audio_files.append(os.path.join(root, file))
    #                 labels.append(extract_labels(file))
    #     labels_df = pd.DataFrame(labels)
    #     labels_df['Gender'] = labels_df['Gender'].astype(str)
    #     return pd.DataFrame({'audio_file': audio_files,
**labels_df})


    def load_dataset(data_dir):
        audio_files = []
        chars = []
        for root, dirs, files in os.walk(data_dir):
            for file in files:
                if file.endswith(".wav"):
                    audio_files.append(os.path.join(root, file))
                    chars.append(extract_chars(file))
        chars_df = pd.DataFrame(chars)
        for col in ['Modality', 'Vocal_channel', 'Emotion',
'Emotional_intensity', 'Statement', 'Repetition', 'Gender']:
            chars_df[col] = chars_df[col].astype(str)
        # Ensure 'Actor' is an integer
        chars_df['Actor'] = chars_df['Actor'].astype(int)
        return pd.DataFrame({'audio_file': audio_files,
**chars_df})

    # Load datasets
    train_dataset = load_dataset(train_dir)
    test_dataset = load_dataset(test_dir)
    val_dataset = load_dataset(val_dir)
    print("Train dataset:")
    train_dataset.head()
    datasets = [train_dataset, test_dataset, val_dataset]
    dataset_names = ['Train', 'Test', 'Validation']
    emotion_counts = []
    for dataset in datasets:
      emotion_counts_i =
dataset['Emotion'].value_counts(normalize=True) * 100
      emotion_counts.append(emotion_counts_i.sort_index())

    fig, axes = plt.subplots(1, 3, figsize=(18, 6), sharey=True)
    for i, (counts, name) in enumerate(zip(emotion_counts,
dataset_names)):
```

```
        sns.barplot(x=counts.index, y=counts.values,
palette="viridis", ax=axes[i])
        axes[i].set_title(f'{name} Dataset')
        axes[i].set_xlabel('Emotion')
        axes[i].set_ylabel('Percentage' if i == 0 else '')
        axes[i].tick_params(axis='x', rotation=45)

    plt.suptitle('Percentage of Emotions in Train, Test, and
Validation Datasets')
    plt.tight_layout(rect=[0, 0, 1, 0.95])
    plt.show()

    # plt.figure(figsize=(5, 3))
    # sns.barplot(x=emotion_counts.index, y=emotion_counts.values,
palette="viridis")
    # plt.title('Percentage of Emotions in the Train Dataset')
    # plt.xlabel('Emotion')
    # plt.ylabel('Percentage')
    # plt.xticks(rotation=45)
    # plt.show()
    def create_waveplot(data, sr):
        plt.figure(figsize=(10, 3))
        librosa.display.waveshow(data, sr=sr)
        plt.title('Waveplot')
    #     plt.show()

    def create_mfcc(data, sr):
        plt.figure(figsize=(12, 3))
        mfcc = librosa.feature.mfcc(y=data, sr=sr, n_mfcc=30)
        librosa.display.specshow(mfcc, x_axis='time')
        plt.colorbar()
        plt.title('MFCC')
    #     plt.show()

    def create_melspectrogram(data, sr):
        plt.figure(figsize=(12, 3))
        melspec = librosa.feature.melspectrogram(y=data, n_mels =
60)
        logspec = librosa.amplitude_to_db(melspec)
        librosa.display.specshow(logspec, sr=sr, x_axis='time',
y_axis='hz')
        plt.title('Mel Spectrogram')
        plt.colorbar()
    #     plt.show()
```

```
    idx = 0
    for root, dirs, files in os.walk(data_dir):
      for file in files:
        if file.endswith('.wav'):
          file_path = os.path.join(root, file)
          data, sampling_rate = librosa.load(file_path)
          create_waveplot(data, sampling_rate)
          # create_mfcc(data, sampling_rate)
          create_melspectrogram(data, sampling_rate)
          ipd.Audio(file_path)
          idx += 1
          # plt.show()
        if idx == 1:
          break
      if idx == 1:
          break
    datasets = [train_dataset, test_dataset, val_dataset]
    for idx in range(len(datasets)):
      datasets[idx] = datasets[idx].drop(['Modality',
'Vocal_channel', 'Emotional_intensity', 'Statement', 'Repetition',
'Gender', 'Actor'], axis=1)
    train_dataset, test_dataset, val_dataset = datasets
    train_dataset.head()
    !pip install --upgrade resampy
    import librosa
    import resampy
    from sklearn.preprocessing import normalize, LabelEncoder
    from sklearn.model_selection import train_test_split
    from tensorflow.keras.utils import to_categorical
    def padding_and_offset(path, sr=16000, input_length=48000):
      data, _ = librosa.load(path, sr=sr, res_type='kaiser_fast')
      if len(data) > input_length:
        max_offset = len(data) - input_length
        offset = np.random.randint(max_offset)
        data = data[offset:(input_length+offset)]
      else:
        if input_length > len(data):
          max_offset = input_length - len(data)
          offset = np.random.randint(max_offset)
        else:
          offset = 0
        data = np.pad(data, (offset, input_length - len(data) -
offset), "constant")
      return data
```

```
    def speed_pitch(data):
      length_change = np.random.uniform(low=0.8, high = 1)
      speed_fac = 1.2  / length_change
      tmp =
np.interp(np.arange(0,len(data),speed_fac),np.arange(0,len(data)),
data)
      minlen = min(data.shape[0], tmp.shape[0])
      data *= 0
      data[0:minlen] = tmp[0:minlen]
      return data

    def noise(data):
      noise_amp = 0.05*np.random.uniform()*np.amax(data)
      data = data.astype('float64') + noise_amp *
np.random.normal(size=data.shape[0])
      return data
    from tqdm import tqdm

    keys = ['Train', 'Test', 'Validation']
    data_x, data_y = {}, {}
    # aug_x, aug_y = {}, {}
    aug_x, aug_y = [], []

    for idx, dataset in enumerate(datasets):
      x_data, y_data = [], []
      x_aug, y_aug = [], []
      for r in tqdm(dataset.values):
        x = padding_and_offset(r[0])
        x_data.append(x)
        y_data.append(r[1])

        x_aug.append(speed_pitch(x))
        x_aug.append(noise(x))

        y_aug.extend([r[1], r[1]])

      x_data, y_data = np.array(x_data), np.array(y_data)
      x_aug, y_aug = np.array(x_aug), np.array(y_aug)
      data_x[keys[idx]], data_y[keys[idx]] = x_data, y_data
      # aug_x[keys[idx]], aug_y[keys[idx]] = x_aug, y_aug
      aug_x.append(x_aug)
      aug_y.append(y_aug)
```

```
    def encode(e_data, df=train_dataset):
      e_encoder = LabelEncoder()
      e_encoder = e_encoder.fit(list(df['Emotion'].unique()))
      e_encoded = to_categorical(e_encoder.transform(e_data))
      return e_encoded
    x_train, x_test, x_val, y_train, y_test, y_val =
data_x['Train'], data_x['Test'], data_x['Validation'],
data_y['Train'], data_y['Test'], data_y['Validation']
    e_y = []
    for e in y_train:
        e_y.append(e)
    e_y_train = encode(e_y)

    e_y = []
    for e in y_test:
        e_y.append(e)
    e_y_test = encode(e_y)

    e_y = []
    for e in y_val:
        e_y.append(e)
    e_y_val = encode(e_y)

    e_y_train_aug = []
    for e in y_aug:
        e_y_train_aug.append(e)
    e_y_train_aug = encode(e_y_train_aug)
    x_train_aug = np.concatenate((x_train, x_aug), axis=0)
    e_y_train_aug = np.concatenate((e_y_train, e_y_train_aug),
axis=0)

    np.random.seed(42)
    np.random.shuffle(x_train_aug)
    np.random.seed(42)
    np.random.shuffle(e_y_train_aug)
    def get_melspec(data):
      melspec = librosa.feature.melspectrogram(y=data, n_mels=60)
      logspec = librosa.amplitude_to_db(melspec)
      logspec = np.expand_dims(logspec, axis=-1)
      return logspec
    x_train_melspec = []
    x_test_melspec = []
    x_val_melspec = []
    x_train_aug_melspec = []
```

```
    for i in x_train:
      melspec = get_melspec(i)
      x_train_melspec.append(melspec)

    for i in x_test:
      melspec = get_melspec(i)
      x_test_melspec.append(melspec)

    for i in x_val:
      melspec = get_melspec(i)
      x_val_melspec.append(melspec)

    for i in x_train_aug:
      melspec = get_melspec(i)
      x_train_aug_melspec.append(melspec)

    x_train_melspec, x_test_melspec, x_val_melspec,
x_train_aug_melspec = np.array(x_train_melspec),
np.array(x_test_melspec), np.array(x_val_melspec),
np.array(x_train_aug_melspec)
    from tensorflow.keras.layers import Input, Conv1D, Conv2D,
MaxPool1D, MaxPool2D, GlobalMaxPool1D, Dropout, Dense, Flatten,
BatchNormalization, Activation
    from tensorflow.keras import Model
    from tensorflow.keras.optimizers import Adam
    from tensorflow.keras.callbacks import EarlyStopping,
ModelCheckpoint
    def get_2d_melspec_model(shape, n_emotions=8, n_sex=2):
        inputs = Input(shape=shape)
        x = Conv2D(32, (4,10), padding="same")(inputs)
        x = BatchNormalization()(x)
        x = Activation("relu")(x)
        x = MaxPool2D()(x)
        x = Dropout(rate=0.2)(x)

        x = Conv2D(32, (4,10), padding="same")(x)
        x = BatchNormalization()(x)
        x = Activation("relu")(x)
        x = MaxPool2D()(x)
        x = Dropout(rate=0.2)(x)

        x = Conv2D(32, (4,10), padding="same")(x)
        x = BatchNormalization()(x)
```

```
        x = Activation("relu")(x)
        x = MaxPool2D()(x)
        x = Dropout(rate=0.2)(x)

        x = Conv2D(32, (4,10), padding="same")(x)
        x = BatchNormalization()(x)
        x = Activation("relu")(x)
        x = MaxPool2D()(x)
        x = Dropout(rate=0.2)(x)

        x = Flatten()(x)
        x = Dense(64, activation='relu')(x)
        x = Dense(256, activation='relu')(x)

        emotion_output = Dense(n_emotions, activation='softmax',
name='emotion_output')(x)
        sex_output = Dense(n_sex, activation='sigmoid',
name='sex_output')(x)

        model = Model(inputs, [emotion_output, sex_output])
        return model
    def get_2d_melspec_model(shape, n_emotions=8):
        inputs = Input(shape=shape)
        x = Conv2D(32, (4,10), padding="same")(inputs)
        x = BatchNormalization()(x)
        x = Activation("relu")(x)
        x = MaxPool2D()(x)
        x = Dropout(rate=0.2)(x)

        x = Conv2D(32, (4,10), padding="same")(x)
        x = BatchNormalization()(x)
        x = Activation("relu")(x)
        x = MaxPool2D()(x)
        x = Dropout(rate=0.2)(x)

        x = Conv2D(32, (4,10), padding="same")(x)
        x = BatchNormalization()(x)
        x = Activation("relu")(x)
        x = MaxPool2D()(x)
        x = Dropout(rate=0.2)(x)

        x = Conv2D(32, (4,10), padding="same")(x)
        x = BatchNormalization()(x)
        x = Activation("relu")(x)
```

```
        x = MaxPool2D()(x)
        x = Dropout(rate=0.2)(x)

        x = Flatten()(x)
        x = Dense(64, activation='relu')(x)
        x = Dense(256, activation='relu')(x)

        emotion_output = Dense(n_emotions, activation='softmax',
name='emotion_output')(x)

        model = Model(inputs, emotion_output)
        return model
    def get_callbacks(name_model):
        callbacks = [
            EarlyStopping(monitor="val_loss", mode="min",
patience=20),
            ModelCheckpoint(name_model, monitor='val_loss',
verbose=1, save_best_only=True)
        ]
        return callbacks
    model_melspec = get_2d_melspec_model((60, 94, 1)) # Make sure
this model has only one output layer for emotion classification
    model_melspec.compile(optimizer=Adam(),
loss="categorical_crossentropy", metrics=['accuracy'])
    history_melspec = model_melspec.fit(x_train_melspec,
e_y_train, validation_data=(x_val_melspec, e_y_val),
callbacks=get_callbacks('best_melspec.h5'), epochs=100,
batch_size=32)
    # history_melspec = model_melspec.fit(x_test_melspec,
e_y_test, validation_data=(x_val_melspec, e_y_val),
callbacks=get_callbacks('best_melspec.h5'), epochs=10,
batch_size=32)
    plt.plot(history_melspec.history['accuracy'])
    plt.plot(history_melspec.history['val_accuracy'])
    plt.legend(['accuracy', 'val_accuracy'])
    plt.show()

    # Evaluate the model on the test data
    test_loss, test_accuracy =
model_melspec.evaluate(x_test_melspec, e_y_test)

    print(f"Test Loss: {test_loss}")
    print(f"Test Accuracy: {test_accuracy}")
```

```
    model_melspec_aug = get_2d_melspec_model((60, 94, 1)) # Make
sure this model has only one output layer for emotion
classification
    model_melspec_aug.compile(optimizer=Adam(),
loss="categorical_crossentropy", metrics=['accuracy'])
    history_melspec_aug =
model_melspec_aug.fit(x_train_aug_melspec, e_y_train_aug,
validation_data=(x_val_melspec, e_y_val),
callbacks=get_callbacks('best_melspec.h5'), epochs=100,
batch_size=32)
    # history_melspec = model_melspec_aug.fit(x_test_melspec,
e_y_test, validation_data=(x_val_melspec, e_y_val),
callbacks=get_callbacks('best_melspec.h5'), epochs=10,
batch_size=32)

    plt.plot(history_melspec_aug.history['accuracy'])
    plt.plot(history_melspec_aug.history['val_accuracy'])
    plt.legend(['accuracy', 'val_accuracy'])
    plt.show()

    # Evaluate the model on the test data
    test_loss, test_accuracy =
model_melspec_aug.evaluate(x_test_melspec, e_y_test)
    print(f"Test Loss: {test_loss}")
    print(f"Test Accuracy: {test_accuracy}")
    from sklearn.metrics import classification_report,
confusion_matrix

    e_y_test = np.argmax(e_y_test, axis=1)

    def get_prediction(model, x):
        y_pred_e = model.predict(x)
        y_pred_e = np.argmax(y_pred_e, axis=1)
        return y_pred_e

    def display_results(y_pred_e, y_true_e, df=test_dataset):
        e_conf_matrix = confusion_matrix(y_true_e, y_pred_e)
        e_df = pd.DataFrame(e_conf_matrix,
index=list(df['Emotion'].unique()),
columns=list(df['Emotion'].unique()))
        print(classification_report(y_true_e, y_pred_e,
target_names=list(df['Emotion'].unique())))
        sns.heatmap(e_df, annot=True, fmt='g')
        plt.show()
```

```
y_pred_e = get_prediction(model_melspec, x_test_melspec)
display_results(y_pred_e, e_y_test)
y_pred_e = get_prediction(model_melspec_aug, x_test_melspec)
display_results(y_pred_e, e_y_test)
```

АРКУШ ОЦІНЮВАННЯ

курсової роботи студента

_____

(ПІБ студента)

_____

(назва роботи)

_____

Науковий керівник

_____

(науковий ступінь, вчене звання, посада, ПІБ)

Критерії оцінювання	Оцінка	Примітки
1. Актуальність дослідження		від 3 до 5 балів
2. Самостійність дослідження, повнота проведеного аналізу		від 3 до 5 балів
3. Реалізація мети дослідження; відповідність висновків обраній темі		від 3 до 5 балів
4. Структура і логічність композиції роботи		від 3 до 5 балів
5. Обізнаність із класичними і новітніми науковими джерелами та володіння навичками реферативного опрацювання		від 3 до 5 балів
6. Відповідність обсягу роботи встановленим вимогам		від 3 до 5 балів
7. Відповідність вимогам наукового стилю; дотримання вимог щодо оформлення роботи відповідно до чинних стандартів		від 3 до 5 балів
8. Ступінь виконання завдань керівника та дотримання термінів		від 3 до 5 балів

виконання роботи		
9. Структура і обсяг презентації роботи на захисті		від 6 до 10 балів
10. Мовна компетенція студента		від 6 до 10 балів
11. Повнота відповідей на поставлені запитання		від 6 до 10 балів
12. Вміння захищати власну думку		від 6 до 10 балів
13. Ступінь володіння матеріалом		від 6 до 10 балів
14. Культура захисту		від 6 до 10 балів
Загальна кількість балів		**максимум 100 балів**

Науковий керівник _____
(підпис)

Дата _____

Члени комісії:

_____
(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)

_____
(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)

_____
(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)