

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»  
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ  
Кафедра інформаційної безпеки**

До захисту допущено  
Завідувач кафедри

\_\_\_\_\_ Дмитро ЛАНДЕ  
(підпис)

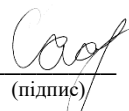
«\_\_\_\_\_» \_\_\_\_\_ 2024 р.

**Магістерська дисертація  
на здобуття ступеня магістра  
за освітньо-професійною програмою «Системи технічного захисту  
інформації»  
спеціальності 125 «Кібербезпека»**

на тему: Підвищення стійкості мультимодальних систем розпізнавання облич до атак  
підміни даних

Виконав: здобувач вищої освіти **II** курсу, групи ФЕ-21мп  
(шифр групи)

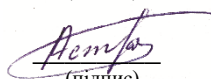
Журавльов Олександр Володимирович  
(прізвище, ім'я, по батькові)

  
(підпис)

Керівник: доцент к.т.н. Дмитро Олександрович Прогонов  
(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

\_\_\_\_\_  
(підпис)

Рецензент: доцент к.т.н. Астраханцев Андрій Анатолійович  
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище, ім'я, по батькові)

  
(підпис)

Засвідчую, що у цій дипломній роботі немає  
запозичень з праць інших авторів без  
відповідних посилань.  
Здобувач вищої освіти \_\_\_\_\_

  
(підпис)

Київ – 2024 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**  
**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ**  
**імені ІГОРЯ СІКОРСЬКОГО»**  
**НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ**  
Кафедра інформаційної безпеки

Рівень вищої освіти – другий (магістерський)  
Спеціальність – 125 «Кібербезпека»  
Освітньо-професійна програма «Системи технічного захисту інформації»

ЗАТВЕРДЖУЮ  
Завідувач кафедри

\_\_\_\_\_ Дмитро ЛАНДЕ  
(підпис)

«\_\_\_» \_\_\_\_\_ 2024 р.

**ЗАВДАННЯ**  
**на дипломну роботу здобувачу вищої освіти**

Журавльов Олександр Володимирович

(прізвище, ім'я, по батькові)

1. Тема роботи: Підвищення стійкості мультимодальних систем розпізнавання облич до атак підміни даних, керівник роботи доцент к.т.н. Дмитро Олександрович Прогонов

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від « 10 » 11 2023 р. № 5236-С

2. Термін подання здобувачем вищої освіти роботи «    » \_\_\_\_\_ 2024 р.

3. Вихідні дані до роботи: методи біометричної аутентифікації користувачів на мобільних пристроях, міжнародні стандарти в галузі біометричної аутентифікації користувачів.

4. Зміст роботи: проведено огляд методів автентифікації, запропоновано модель штучної нейронної мережі для виявлення атак підміни, проведено порівняльний аналіз ефективності сучасних методів та запропонованої моделі, підготовлено матеріали з розробки стартапу.


5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо): презентація (14 слайдів), 29 таблиць, 19 рисунків

6. Дата видачі завдання: 10.09.23 \_\_\_\_\_

## Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів дипломної роботи	Примітка
1	Узгодження організаційних питань з науковим керівником	01.09.23-21.09.23	Виконано
2	Узгодження теми роботи та початок роботи над першим розділом	21.09.23-26.10.23	Виконано
3	Пошук та обробка набору даних для налаштування штучної нейронної мережі	26.10.23-5.11.23	Виконано
4	Створення та налаштування та тестування штучної нейронної мережі	5.11.23-27.11.23	Виконано
5	Підготовка другого розділу роботи	27.11.23-30.11.23	Виконано
6	Порівняння з вже наявними методами	30.11.23-7.12.23	Виконано
7	Підготовка третього розділу роботи	7.12.23-10.12.23	Виконано
8	Розробка стартапу	10.12.23-18.12.23	Виконано
9	Оформлення дипломної роботи	18.12.23-23.12.23	Виконано
10	Підготовка презентації до захисту	25.12.23-03.01.24	Виконано
11	Захист магістерської дисертації	18.01.24-19.01.24	Виконано

Здобувач вищої освіти

  
 (підпис)

Олександр Журавльов  
 (Власне ім'я, ПРІЗВИЩЕ)

Керівник роботи

 \_\_\_\_\_  
 (підпис)

Дмитро Прогонов  
 (Власне ім'я, ПРІЗВИЩЕ)

## РЕФЕРАТ

Біометрична аутентифікація на сьогодні є дуже популярною. Одним із видів біометричної автентифікації є автентифікація по обличчю. Проте такі системи є вразливими до атак підміни. Для вирішення даного обмеження було запропоновано модель штучної нейронної мережі що встановлює відповідність між обличчям і голосом задля виявлення атак підміни.

По результатам порівняння вже існуючих методів та запропонованого методу стало зрозуміло, що метод працює, але потребує вдосконалення і подальших досліджень задля підвищення точності.

Обсяг роботи 118 сторінок, робота містить 19 ілюстрації, 29 таблиць, 7 додатків, 61 джерело літератури.

Метою роботи є підвищення стійкості систем розпізнавання облич до атак підміни обличчя, заснованих на використанні масок.

Об'єктом дослідження є системи протидії атакам підміни обличчя, заснованих на використанні масок.

Предметом дослідження є методи підвищення стійкості систем біометричної автентифікації, а саме методи порівняння зображення обличчя і запис голосу людини з метою встановлення відповідності між ними.

АВТЕНТИФІКАЦІЯ ЗА ОБЛИЧЧЯМ, АТАКИ ПІДМІНИ ДАНИХ, РОЗПІЗНАВАННЯ ОБЛИЧ, ШТУЧНА НЕЙРОННА МЕРЕЖА, СПІВСТАВЛЕННЯ ГОЛОСУ І ОБЛИЧЧЯ.

## **ABSTRACT**

Biometric authentication is very popular today. One type of biometric authentication is face authentication. However, such systems are vulnerable to spoofing attacks. To resolve this limitation, we propose an artificial neural network model that establishes a correspondence between face and voice to detect spoofing attacks.

Based on the results of comparing existing methods and the proposed method, it became clear that the method works, but needs to be improved and further researched to improve accuracy.

The volume of the work is 118 pages, the work contains 19 illustrations, 29 tables, 7 appendices, 61 sources.

The purpose of the work is to increase the resistance of face recognition systems to face substitution attacks based on the use of masks.

The object of research is systems for counteracting face substitution attacks based on the use of masks.

The subject of the study is methods of increasing the resistance of biometric authentication systems, namely methods of comparing a face image and a human voice recording in order to establish a match between them.

FACE AUTHENTICATION, DATA SPOOFING ATTACKS, FACE RECOGNITION, ARTIFICIAL NEURAL NETWORK, VOICE AND FACE MATCHING.

## ЗМІСТ

СПИСОК ТЕРМІНІВ І СКОРОЧЕНЬ .....	8
ВСТУП.....	9
1 ОГЛЯД МЕТОДІВ АУТЕНТИФІКАЦІЇ КОРИСТУВАЧІВ .....	11
1.1 Методи біометричної аутентифікації користувачів .....	11
1.1.1 Методи аутентифікації за відбитками пальців .....	12
1.1.2 Методи аутентифікації за райдужною оболонкою очей .....	14
1.1.3 Методи аутентифікації за зображенням обличчя .....	16
1.1.4 Методи аутентифікації за мовними сигналами .....	22
1.2 Стандарти аутентифікації .....	24
1.3 Методи покращення точності роботи систем аутентифікації .....	27
1.4 Стійкість систем розпізнавання обличчя .....	30
Висновки до розділу 1 .....	33
2 МУЛЬТИМОДАЛЬНА МОДЕЛЬ ВИЯВЛЕННЯ АТАК ПІДМІНИ ОБЛИЧЧЯ.....	34
2.1 Аналіз взаємозалежності зовнішності людини і голосу .....	35
2.2 Набір даних для налаштування .....	36
2.3 Принцип роботи запропонованої моделі штучної нейронної мережі. ....	38
2.4 Процес налаштування запропонованої моделі штучної нейронної мережі	39
Висновки до розділу 2 .....	45
3 ПОРІВНЯННЯ СУЧАСНИХ МЕТОДІВ ТА ЗАПРОПОНОВАНОЇ МОДЕЛІ ШТУЧНОЇ НЕЙРОННОЇ МЕРЕЖІ.....	46
3.1 Результати тестування .....	46
3.2 Порівняння сучасних методів та запропонованої моделі штучної нейронної мережі .....	47
Висновки до розділу 3.....	50
4 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ.....	52
4.1 Опис ідеї проекту .....	52
4.2 Технологічний аудит ідеї проекту.....	55
4.3 Аналіз ринкових можливостей запуску стартап-проекту .....	57
4.5 Розроблення маркетингової програми стартап-проекту .....	70
Висновки до розділу 4.....	74

ВИСНОВКИ .....	75
ПЕРЕЛІК ПОСИЛАНЬ .....	77
ДОДАТОК А. ПРОГРАМНА РЕАЛІЗАЦІЯ ОБРІЗАННЯ ВІДЕО .....	84
ДОДАТОК Б. ПРОГРАМНА РЕАЛІЗАЦІЯ ВИОКРЕМЛЕННЯ ЗОБРАЖЕННЯ ОБЛИЧЧЯ З ВІДЕО .....	87
ДОДАТОК В. ПРОГРАМНА РЕАЛІЗАЦІЯ СТВОРЕННЯ СПЕКТРОГРАМ ...	90
ДОДАТОК Г. ПРОГРАМНА РЕАЛІЗАЦІЯ VGG-FACE.....	93
ДОДАТОК Ґ. ПРОГРАМНА РЕАЛІЗАЦІЯ НАЛАШТУВАННЯ МУЛЬТИМОДАЛЬНОЇ ЧАСТИНИ .....	98
ДОДАТОК Д. ПРОГРАМНА РЕАЛІЗАЦІЯ НАЛАШТУВАННЯ МОДЕЛІ БІНАРНОЇ КЛАСИФІКАЦІЇ .....	104
ДОДАТОК Е. ПРОГРАМНА РЕАЛІЗАЦІЯ ОДНОЧАСНОГО НАЛАШТУВАННЯ МУЛЬТИМОДАЛЬНОЇ ЧАСТИНИ ТА ЧАСТИНИ ЩО ВИКОНУЄ БІНАРНУ КЛАСИФІКАЦІЮ .....	111

## СПИСОК ТЕРМІНІВ І СКОРОЧЕНЬ

APCER - Attack Presentation Classification Error Rate (рівень помилок класифікації атак представлення)

AP- Average Pooling (усереднювальне агрегування)

BN - Batch Normalization (нормалізація партії)

BPCER - Bona Fide Presentation Classification Error Rate (частота помилок класифікації добросовісної презентації)

CNN - Convolutional Neural Network (згорткова нейронна мережа)

CL - Convolutional Layer(згортковий шар)

ELBP - Entropy based Local Binary Pattern (локальний бінарний патерн на основі ентропії)

FAR - False Acceptance Rate (рівень помилкових пропусків)

FRR - False Rejection Rate (рівень помилкових відхилень)

IAR - Imposter Acceptance Rate (рівень пропусків самозванців)

IoT - Internet of Things (інтернет речей)

HTER - Half-Total Error Rate (половина повної помилки)

LBP - Local Binary Pattern (локальний двійковий патерн)

MP - Max Pooling (максимізаційне агрегування)

MTCNN - Multi-Task Cascaded Convolutional Neural Network  
(багатозадачна каскадна згорткова нейронна мережа)

NBP - Neighborhood Binary Pattern (бінарний патерн на основі сусідніх елементів)

PGD - Projected Gradient Descent (прогнозований градієнтний спуск)

ReLU - Rectified Linear Unit (зрізаний лінійний вузол)

SAR - Spoof Acceptance Rate (рівень приймання підробок)

ДП “УкрНДНЦ” - Державне підприємство «Український науково-дослідний і навчальний центр проблем стандартизації, сертифікації та якості»

ПШ - Повнозв'язний шар



## ВСТУП

Наразі одним із ключових елементів кібербезпеки є аутентифікація користувачів. Вона дозволяє встановити належність користувачеві інформації в системі [1]. Це допомагає запобігти несанкціонованому доступу до конфіденційної інформації або зловживань привілеями.

Сьогодні біометрична аутентифікація є одним із найпопулярніших методів аутентифікації [19]. Одним з видів біометричної автентифікації є автентифікація по зображенню обличчя. Проте такі системи вразливі до атак підміни оличчя.

Метою роботи є підвищення стійкості систем розпізнавання облич до атак підміни обличчя, заснованих на використанні масок. Для цього необхідно виконати наступні завдання:

- Провести огляд методів біометричної аутентифікації користувачів
- Запропонувати і реалізувати модель штучної нейронної мережі, що здатна виявляти атаки підміни обличчя
- Порівняти запропонований метод із вже існуючими

Об'єктом дослідження є системи протидії атакам підміни обличчя, заснованих на використанні масок.

Предметом дослідження є методи підвищення стійкості систем біометричної автентифікації, а саме методи порівняння зображення обличчя і запис голосу людини з метою встановлення відповідності між ними.

Елемент наукової новизни роботи полягає у використанні мультимодальних даних, а саме залежності між голосом і обличчям користувача, для виявлення атак підміни при автентифікації.

Практичне значення роботи полягає в розробці прототипу запропонованої системи. Матеріали даної роботи можуть стати у нагоді для майбутніх досліджень виявлення атак підміни за допомогою штучних нейронних мереж. Запропонована система після доопрацювання може бути інтегрована в мобільні

пристрої для або інші пристрої, при користуванні якими користувачі потребують автентифікації за обличчям.

За матеріалами роботи було підготовлено та опубліковано тези доповіді на IV Всеукраїнській Студентській Науковій Конференції «НАУКОВИЙ ПРОСТІР: АНАЛІЗ, СУЧАСНИЙ СТАН ТРЕНДИ ТА ПЕРСПЕКТИВИ» [61].

# 1 ОГЛЯД МЕТОДІВ АУТЕНТИФІКАЦІЇ КОРИСТУВАЧІВ

## 1.1 Методи біометричної аутентифікації користувачів

Останні кілька десятиліть людство активно прагне автоматизувати максимальну кількість процесів. Слід зазначити, що на фоні пандемії COVID-19, зокрема ізоляції та віддаленої роботи, цей процес ще більш прискорився. Водночас важливо враховувати й негативні аспекти цифровізації, такі як зниження приватності, збільшення випадків шахрайства через інтернет, маніпуляція і дезінформація. Це обумовлює актуальність та важливість надійного захисту конфіденційних даних від зловмисників.

Одним з важливих елементів кібербезпеки є аутентифікація користувачів, яка має на меті встановлення належності користувачеві інформації в системі [1]. Це дозволяє уникнути несанкціонованого доступу до конфіденційної інформації або зловживання привілеями. Процес аутентифікації може включати введення логіна та пароля, сканування відбитків пальців чи сітківки ока, розпізнавання обличчя тощо [35].

На сьогодні методи біометричної аутентифікації набули великої популярності [19]. Біометрична аутентифікація заснована на виявленні та аналізі унікальних фізичних або поведінкових характеристик особи для підтвердження її ідентичності. Вагомою перевагою біометричних методів аутентифікації у порівнянні з паролями є відсутність необхідності запам'ятовувати та вводити складні комбінації символів. За результатами дослідження міжнародної корпорації Visa в 2017 році встановлено, що 70 % опитаних зазначили зручність використання методів біометричної аутентифікації у порівнянні з введенням паролів [2]. Відмітимо, що наразі запропоновано значну кількість методів аутентифікації користувачів за біометричними даними, наприклад за геометрією вен на руці [55], відбитком пальця [20], сітківкою ока [56], обличчям, райдужною оболонкою ока [4] тощо. Тому становить інтерес огляд найбільш популярних

біометричних методів аутентифікації, які базуються на скануванні відбитків пальців, розпізнавання обличчя, розпізнавання голосу та райдужної оболонки ока.

### **1.1.1 Методи аутентифікації за відбитками пальців**

Аутентифікація користувачів за відбитками пальців заснована на встановленні та використанні унікальних характеристик пальцевих відбитків для перевірки ідентичності особи. Цей метод аутентифікації ґрунтується на тому що:

- не існує двох осіб, які мають ідентичний набір пальцевих візерунків — кожен пальцевий відбиток відрізняється від інших за рядом характеристик, таких як петлі, пелюстки та дуги [20]. Ця унікальність дозволяє системі аутентифікації встановлювати однозначну ідентичність людини, порівнюючи її пальцевий відбиток зі збереженими у базі даних;

- пальцеві візерунки залишаються практично незмінними протягом усього життя особи — природні фактори, такі як старіння шкіри або незначні травми, не мають суттєвого впливу на структуру та форму пальцевих відбитків.

Наразі широко застосовують наступні типи сканерів відбитків пальців [19]:

- оптичні сканери — використовують видиме світло для отримання зображення відбитка пальця, розміщеного на скляній пластині. Ці сканери досить точні та недорогі, проте є чутливими до бруду, подряпин, мокрих (спітнілих) пальців;

- ємнісні сканери — використовують ємнісний потенціал людської шкіри проти повітряних зазорів між виступами пальця для формування відображення відбитка. Такі сканери дозволяють використовувати жести задля керування приладом в якому вони встановлені. Наприклад користувач може провести пальцем вниз аби опустити панель сповіщень на смартфоні. Недоліком даного типу сканерів можливо є неможливість їх встановлення під екран смартфонів;

- теплові сканери — можуть формувати схему відбитка пальця за результатами дослідження різниць температур між частинами поверхні пальця. Такий тип сканера потребує лінійного руху пальця по датчику [57];
- ультразвукові давачі — використовують звукові імпульси як свого роду «сонар», що зчитує поверхню відбитка пальця.

Після сканування пальця з використанням наведених типів давачів, отриманий відбиток оброблюється для отримання його характеристик. В процесі аутентифікації проводиться оцінка ступеня подібності отриманого зразка відбитка пальця і зразка збереженої (типової) копії зареєстрованого користувача. Якщо дана оцінка перевищує заданий поріг, то користувач вважається автентифікованим.

Важливо відзначити, що в процесі порівняння відсканованого зразка зі збереженою копією забезпечується певний рівень стійкості щодо спотворень отриманих біометричних даних. Дані спотворення обумовлені впливом подряпин поверхні сканера, наявністю бруду чи вологи на поверхні пальця / сканера. Внаслідок цього зміни відбитка пальця можуть бути настільки суттєвими, що це призведе до помилкового спрацювання. Тому для порівняння якості роботи сучасні системи біометричної аутентифікації зазвичай користуються наступними типами помилок [58] : хибне спрацювання (FRR), пропуск цілі (FAR), IAR, SAR, BPCER та APCER.

На сьогодні, ультразвукові сканери відбитків пальців набули широкої популярності в смартфонах завдяки своїй компактності та відносно невеликій вартості (від 2 до 20 доларів) [21]. Проте, у пристроях, призначених спеціально для сканування відбитків пальців, частіше за все використовують оптичні сканери. Крім того, сканери відбитків пальців часто інтегруються з іншими типами методів аутентифікації, наприклад сканери MIFARE смарт-картках або камери Futronic FS26, Aratek BM7500 та Aratek BM5510 [36, 37, 38]. Такі пристрої зазвичай застосовуються у складі систем контролю та керування

доступом підприємств, зокрема для створення ідентифікаційних карток громадян.

### 1.1.2 Методи аутентифікації за райдужною оболонкою очей

В процесі аутентифікації з застосуванням райдужної оболонки ока відбувається сканування і вимірювання унікальних візерунків на поверхні райдужної оболонки ока. Для підвищення точності визначення даних візерунків біометричні сканери освітлюють райдужну оболонку інфрачервоним світлом.

Основні етапи отримання даних з використанням сканерів розпізнавання райдужної оболонки ока наведені на рис. 1.1.

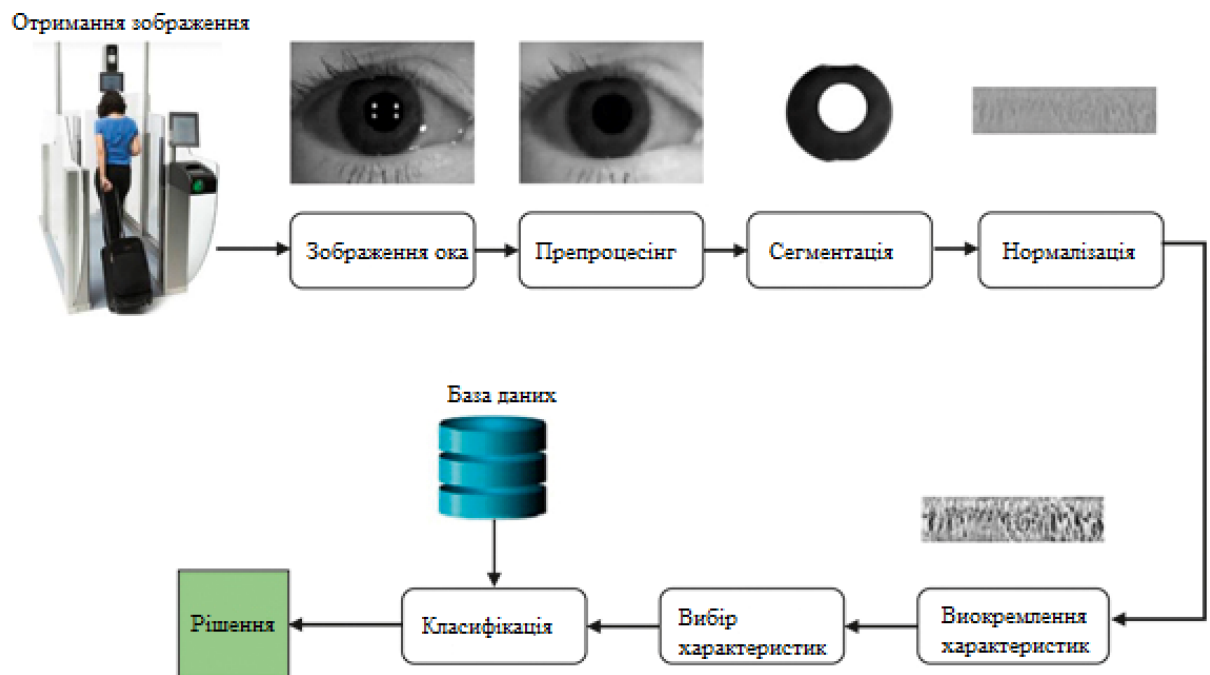


Рисунок 1.1 – Алгоритм аутентифікації користувача за райдужною оболонкою ока [4]

На першому етапі відбувається підготовка отриманого зображення для подальшої обробки (рис. 1.1). Зокрема проходить вилучення даних, що

відносяться до вії, повіки та дзеркальні відображення навколишнього середовища на отриманому зображенні ока, які зазвичай захиляють частини райдужної оболонки. Кінцевим результатом сегментації є набір пікселів, що містить лише райдужну оболонку. Після цього проводиться аналіз малюнка ліній та кольорів ока, за результатами якого формується відповідний бітовий малюнок. Отримані дані порівнюються з шаблонами, що зберігаються в базі даних, для перевірки (збіг шаблону один до одного) або ідентифікації (порівняння шаблону один до багатьох) [3].

Відмітимо, що основним етапом даного алгоритму є виокремлення характеристик. Це можна зробити багатьма методами, наприклад LBP(Local Binary Pattern), NBP(Neighborhood Binary Pattern), ELBP (Entropy based Local Binary Pattern) тощо. В цьому огляді буде розглянуто тільки один з них, а саме Neighborhood-based Binary Pattern (NBP), який є стійким до поворотів вікна і має кращу точність ніж класичний LBP [16].

Основні етапи роботи методу NBP наведені на рис. 1.2. Алгоритм починається з верхньої лівої частини вікна і проходить по годинниковій стрілці. Якщо значення сусідньої комірки у вікні (по годинниковій стрілці) більше то комірку заповнюють 0 і 1 якщо його рівень яскравості більше ніж у наступної комірки. Тобто якщо перша комірка має рівень яскравості 4, а друга 6, то двійковий код першої комірки дорівнює 0 якщо третя комірка має значення 7, а четверта значення 9 то двійковий код для третьої комірки дорівнює 1 і так далі. Після отримання двійкового коду з 8 цифр цей код перетворюється у десяткове число 26. Це число вважається значенням центрального пікселя у вікні. Таким чином утворюється зображення яке характеризує райдужну оболонку ока.

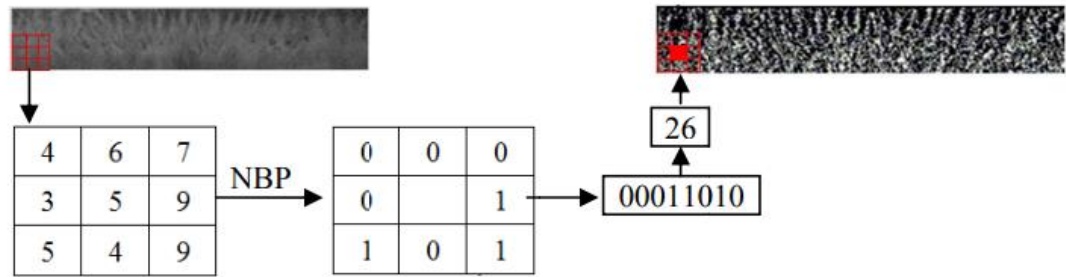


Рисунок 1.2 – Основні етапи роботи методу NBP для отримання характеристик райдужної оболонки ока (за матеріалами [16])

На сьогодні сканування райдужної оболонки ока використовується в системах контролю доступу, наприклад СМІ-Tech EF45, СМІTech ЕМА-30 та IriShield-USB МК 2120U. Проте через свою високу ціну даний метод аутентифікації не популярний для мобільних пристроїв.

### 1.1.3 Методи аутентифікації за зображенням обличчя

Аутентифікація користувачів за допомогою зображення обличчя в загальному вигляді передбачає три основних етапи: виявлення обличчя, розпізнавання обличчя та порівняння облич з зображеннями у базі даних користувачів [22]. Наразі, дані методи широко використовують машинне навчання.

Автентифікація користувачів за зображенням обличчя проводиться в кілька етапів. На першому етапі відбувається/проводиться визначення та виділення саме тієї області, що містить обличчя на вхідному зображенні. Один з найпростіших і найшвидших варіантів реалізації даної процедури заснований на використанні каскадів Хаара [5]. Цей метод не потребує великих обчислювальних ресурсів, що робить його популярним рішенням для мобільних пристроїв із невеликою обчислювальною потужністю.

Метод на основі каскадів Хаара (рис. 1.3) використовує функцію Хаара та рухоме (ковзаюче) вікно. Обчислюються характеристики для кожного



положення вікна та класифікуються як позитивні чи негативні. Позитивні якщо в області вікна має бути обличчя і негативні якщо обличчя немає [5].

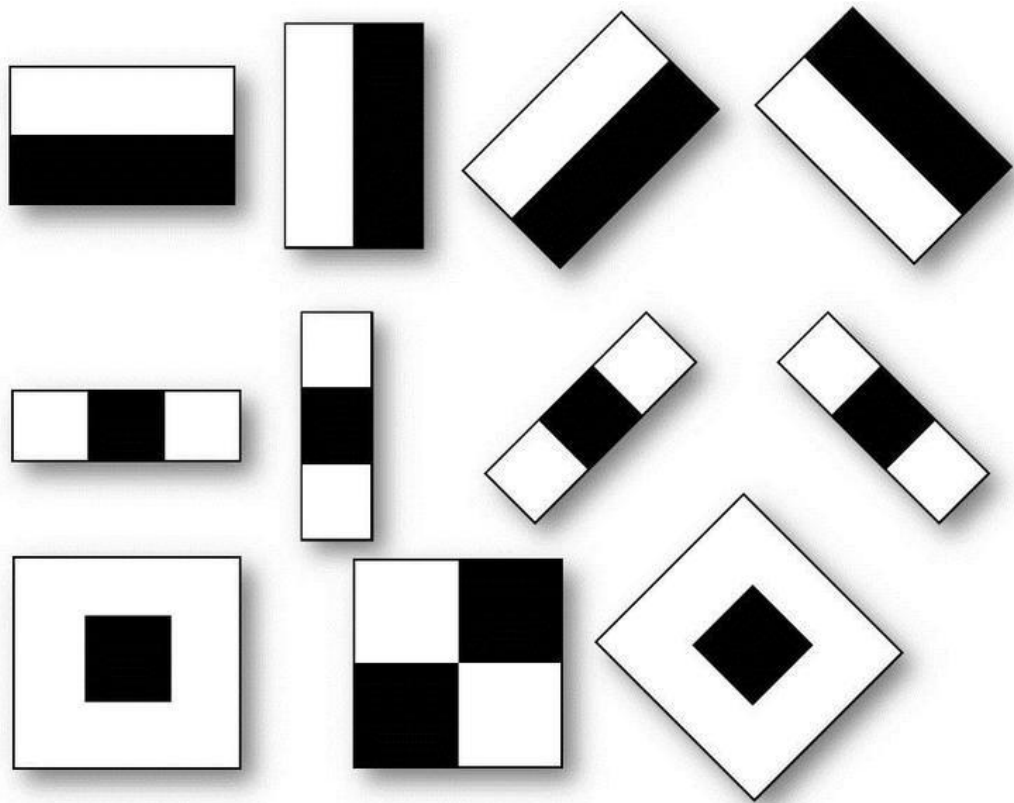


Рисунок 1.3 – Приклади функцій Хаара [5]

Функції Хаара використовуються для знаходження більш світлих і темних областей на зображенні, характерних саме для людських облич. Наприклад, в середньому яскравість щік або чола буде більшою ніж у очей, а брови будуть темнішими за ніс [5]. За результатом роботи алгоритму буде виявлено ділянку зображення де присутнє тільки обличчя.

Слід зазначити, що ціною за швидкість роботи методу на основі каскадів Хаара є його відносно низька точність. Тому наразі запропоновані більш надійні системи виявлення облич на основі багат шарових нейроподібних мереж, такі як MTCNN (Multi-Task Cascaded Convolutional Neural Network) та SCRFD (Sample and Computation Redistribution for Efficient Face Detection).

Другий етап аутентифікації користувачів за результатами розпізнавання обличчя, включає процес ідентифікації особливостей обличчя, які можуть слугувати унікальними характеристиками для кожного користувача. На цьому етапі застосовують штучну нейронну мережу, яка репрезентує обличчя на зображенні як вектор вкладень. Для вирішення задачі репрезентації обличчя людини як вектору наразі часто користуються сіамськими нейронними мережами (Siamese Network) з триплетною функцією втрат (triplet loss) [7]. Слід відзначити що збільшення кількості шарів в штучній нейронній мережі призводить до збільшення кількості параметрів, що в свою чергу впливає на кількість обчислень які треба здійснити для налаштування.

Сіамська мережа не класифікує зображення на певні категорії або мітки, а лише визначає відстань між будь-якими двома заданими зображеннями [7]. На рис. 1.4 наведено схему сіамської нейронної мережі з використанням триплетної помилки.

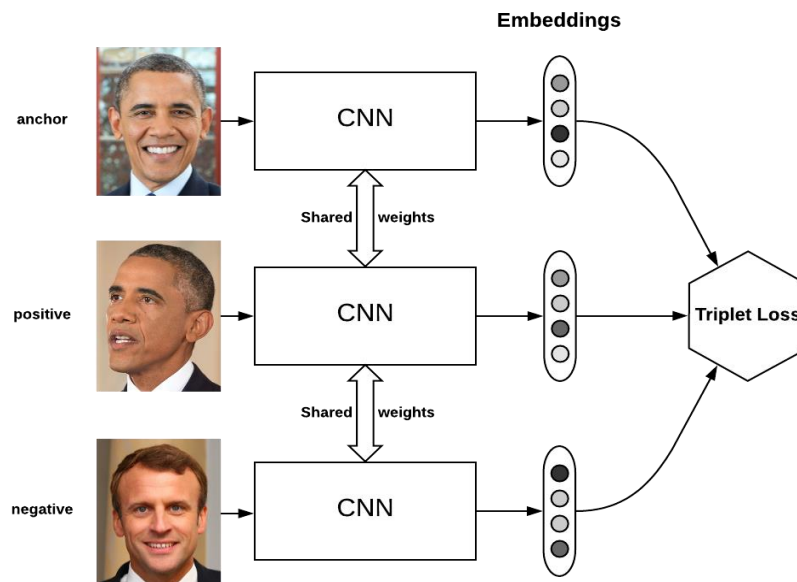


Рисунок 1.4 – Схема роботи сіамських нейромереж, де CNN (Convolutional Neural Network) – це згорткова нейронна мережа, embeddings – вектор вкладень, який репрезентує обличчя у векторному просторі [6]

Якщо зображення мають однакову мітку, то мережа повинна бути налаштована таким чином, щоб вона генерувала меншу відстань між векторами що були отримані з цих зображень. В протилежному випадку (якщо зображення належать до різних міток), то відстань між векторами повинна бути більшою.

Формула для розрахунку Triplet Loss:

$$L = \sum_i^N \left[ \|f(x_i^a) - f(x_i^p)\|_2 - \|f(x_i^a) - f(x_i^n)\|_2 + \alpha \right], \quad (1)$$

де  $f(x)$  – приймає  $x$  в якості вхідного зображення і репрезентує його в вигляді вектора вкладень (embedding); нижній індекс  $a$  – вказує на те що зображення є якорем,  $p$  – вказує на те що зображення є позитивним,  $n$  – вказує на те що зображення є негативним;  $\alpha$  – відстань, яка дотримується між додатними і негативними парами.

Графічне відображення роботи триплетної функції втрат наведено на рисунку 1.5.

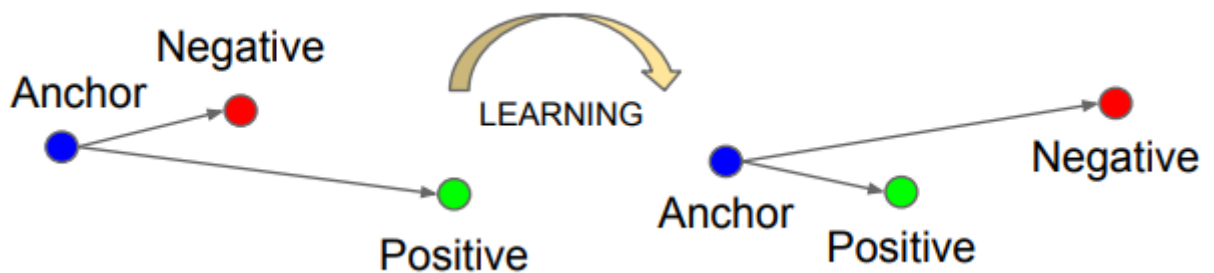


Рисунок 1.5 – Принцип роботи Triplet Loss [7]

Принцип роботи Triplet Loss полягає в [7]:

1.Вибирається трійка зображень :

- якір (Anchor): це обличчя, для якого ми хочемо покращити представлення, на рис. 1.4 це перше зображення Обами;

- позитивний приклад (Positive): це обличчя схоже на обличчя з якоря, і ми хочемо зблизити їхні представлення, на рис. 1.4 це друге зображення Обама;
- негативний приклад (Negative): це обличчя не схоже на обличчя з якоря, і ми хочемо збільшити відстань між ними, на рис. 1.4 це зображення Макрона.

2. Після походження цих трьох зображень через сіамську нейромережу отримуються 3 вектори вкладень (embedding) і розраховується відстань між ними (зазвичай це Евклідова відстань).

3. Формується вираз, який старається мінімізувати відстань між якорем і позитивним прикладом і одночасно максимізувати відстань між якорем і негативним прикладом.

Триплетна функція втрат, яка добре показує себе в задачах вибору найкращого представлення (embedding) для об'єкта в просторі векторів [6].

Третій етап аутентифікації за обличчям — порівняння обличчя людини з тими що присутні в базі даних користувачів. При перевищенні ступеня заданого порогового значення подібності векторів, аутентифікація проходить успішно. Для порівняння використовується вектор вкладень отриманий на попередньому етапі. Найбільш розповсюдженими методами порівняння двох векторів вкладень є евклідова та косинусна відстань між елементами векторного простору [23, 24].

Евклідова дистанція між двома векторами  $x$  та  $y$  може бути розрахована за наступною формулою.

$$\|x - y\|_2 = \sqrt{\sum_{i=1}^N (x_i - y_i)^2} = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_N - y_N)^2} \quad (2)$$

В свою чергу косинусна подібність для векторів  $x$  та  $y$  обчислюється за формулою 3.

$$\cos \cos (\theta) = \frac{x \cdot y}{\|x\| \cdot \|y\|} \quad (3)$$

Графічне представлення евклідової відстані та косинусної відстані для векторів **A** і **B** показано на рис. 1.6.

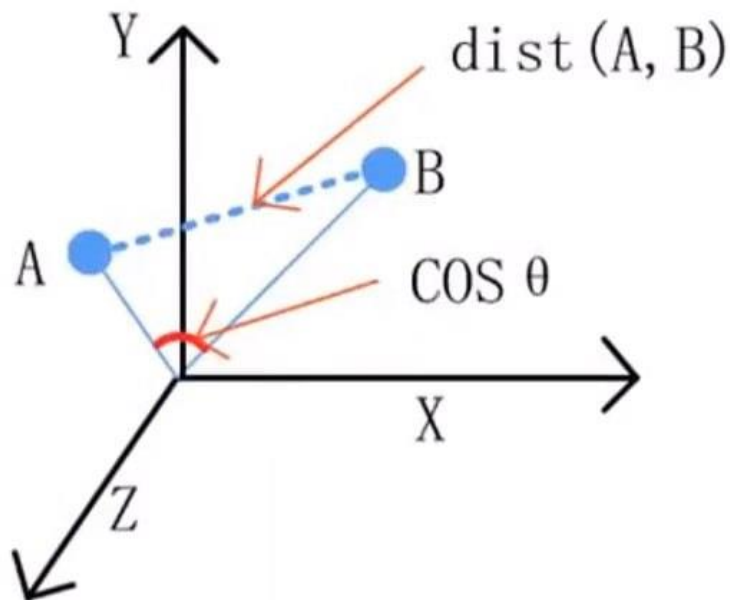


Рисунок 1.6 – Графічне представлення евклідової відстані та косинусної подібності між векторами **A** і **B** [24]

Прикладом реалізації сіамських нейронних мереж можуть бути VGG-face [52], FaceNet [7], ResNet [59] та інші. Наразі системи розпізнавання облич широко застосовуються для виявлення правопорушників в публічних місцях та громадському транспорті, також не менш популярним є застосування розпізнавання обличчя для аутентифікації в мобільних пристроях на кшталт смартфонів.

### 1.1.4 Методи аутентифікації за мовними сигналами

Аутентифікація користувачів за голосом зазвичай проходить в чотири кроки [25]:

1. Збір голосових даних: Спершу система повинна зібрати голосові дані користувача. Це може включати в себе запис різних фраз або речень, щоб отримати достатню кількість голосового матеріалу для подальшої обробки.

2. Отримання (екстракція) голосових характеристик: Після збору голосових

даних система обчислює голосові характеристики, такі як частота, інтонація, ритм, спектрограма та інші. Набір цих параметрів може змінюватися в залежності від реалізації системи.

3. Створення голосового шаблону: Голосові характеристики обробляються та використовуються для створення унікального голосового шаблону для користувача.

4. Аутентифікація: При спробі входу користувача система збирає його голосовий зразок і порівнює його з збереженим голосовим шаблоном. Якщо характеристики голосу відповідають шаблону з певною визначеною точністю, автентифікація успішна і користувач отримує доступ.

Для виокремлення голосових характеристик можна використовувати різні методи, наприклад віконне перетворення Фур'є [17]. Такий спосіб є більш ефективним, ніж звичайне перетворення Фур'є або використання необробленого сигналу в хвильовій формі, бо дозволяє відобразити зміну спектра протягом часу. Віконне перетворення Фур'є полягає в поділі звукового сигналу на короткі відрізки (вікна) і обчислення перетворення Фур'є для кожного з них.

$$F(\tau, \omega) = \int_{-\infty}^{\infty} x(t) \omega(t - \tau) e^{-i\omega t} dt \quad (4)$$

де  $x(t)$  - це сигнал, який потрібно перетворити;  $\omega(t)$  - функція вікна.

Основні етапи роботи віконного перетворення Фур'є наведені на рис. 1.7.

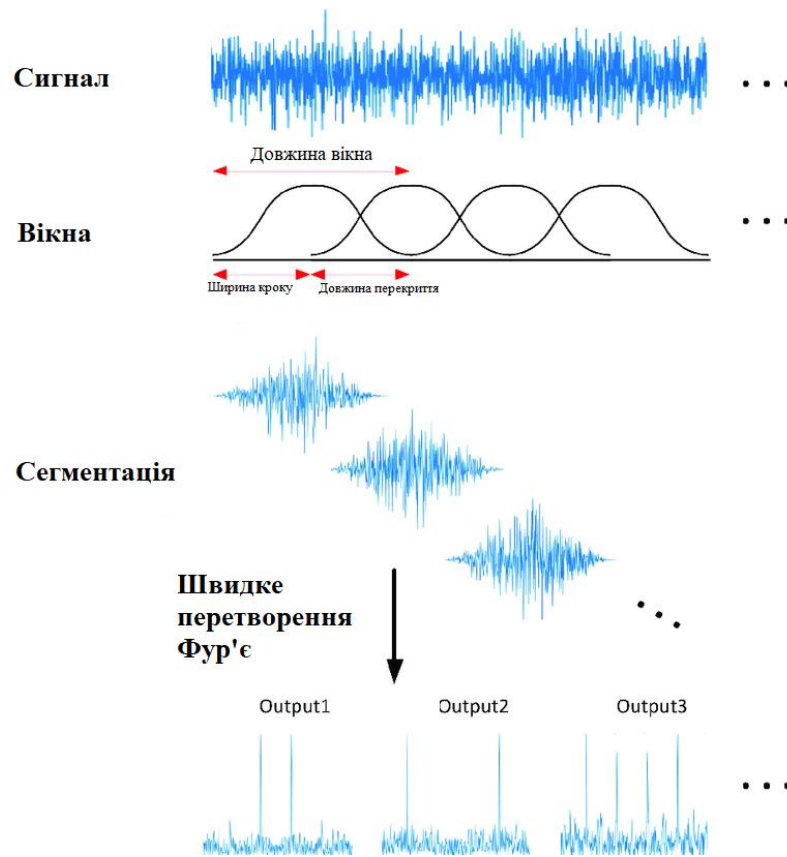


Рисунок 1.7 – Алгоритм роботи віконного перетворення Фур'є [17]

Потім зазвичай будують спектри, що змінюються за функцію часу, відому як спектрограма рис. 1.8.

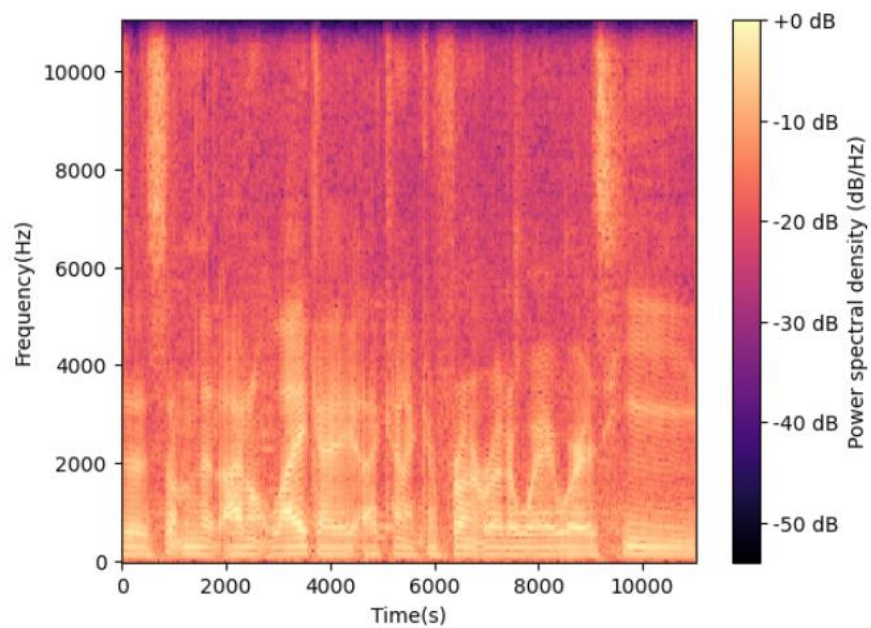


Рис. 1.8 – Приклад спектрограми мовного сигналу людини

В залежності від змісту аудіозапису з голосом голосова аутентифікація поділяється на три типи:

- текстонезалежна – цей підхід вирізняється тим що для проходження аутентифікації системі не важливо, що саме говорить користувач. Проте цей метод потребує щонайменше 6-8 секунди аудіозапису, що може бути незручним для користувачів;
- текстозалежна з статичним паролем – підтвердження особистості відбувається за допомогою парольної фрази, яку користувачі створюють під час реєстрації. Парольна фраза повинна тривати принаймні 2-3 секунди;
- текстозалежна з динамічним паролем – працює так само як і попередній метод за тим винятком, що від користувача кожен раз вимагається прочитати випадково згенероване число чи речення.

Наразі системи аутентифікації голосу зазвичай використовують в кол-центрах, пристроях IoT (Internet of Things) та для віртуальних асистентів.

До переваг даного типу біометричної автентифікації відносяться зручність для користувачів, безконтактність, а отже, менш інвазивний та більш гігієнічний. До обмежень відноситься понижена точність у порівнянні з іншими біометричними методами, низька стійкість до атак підміни (англ. spoofing) [26] та суттєвий вплив фонових шумів.

## **1.2 Стандарти аутентифікації**

Згідно з наказом ДП “УкрНДНЦ” «Про прийняття нормативних документів України, гармонізованих з міжнародними та європейськими нормативними документами» [9] ДСТУ щодо аутентифікації в Україні повністю спираються на стандарти групи ISO/IEC 9798. Основними з них є ДСТУ ISO/IEC 9798- 1 :2002 (Інформаційні технології. Методи захисту. Автентифікація



суб'єктів. Частина 1. Загальні положення), ДСТУ ISO/IEC 9798-3:2002 (Інформаційні технології. Методи захисту. Автентифікація об'єктів. Частина 3. Механізми, що ґрунтуються на цифровому підписі). На жаль, у відкритому доступі знаходиться тільки інформативна частина цих документів. Тож для більшого розуміння ситуації із стандартами аутентифікації, розглянемо стандарти щодо роботи систем аутентифікації на мобільних пристроях, зокрема запропоновані для операційних систем Apple і Android. В Apple розробили низку пропрієтарних методів аутентифікації, такі як FaceID чи TouchID, проте політика компанії не передбачає публікації матеріалів про принципи за якими були розроблені дані методи. На противагу цьому, стандарти щодо роботи методів аутентифікації для операційної системи Android [10,11] є відкритими. Згідно з «Вимірювання безпеки біометричного розблокування» Android розділяє три рівні аутентифікації :

- основний рівень – заснований на факторі знань, наприклад, PIN-код, шаблон (графічний ключ) або пароль, що гарантують теоретично найвищу потенційну надійність системи аутентифікації. На практиці це потребує використання складних (довгих) паролів;
- вторинний рівень – заснований на використанні біометричних даних користувачів, наприклад відбитку пальця чи розпізнаванні обличчя. Як вже було встановлено в першому розділі роботи, біометрія пропонує більш зручні для користувачів, але менш надійні методи аутентифікації;
- третинний рівень – заснований на навколишньому середовищі. Це може бути використання фізичного токена, наприклад, через довірені пристрої Smart Lock [27], які дозволяють розблокування телефону при поєднанні з Bluetooth-пристроєм, включеним до списку дозволених пристроїв. Або це може бути пов'язано з фізичним довкіллям пристрою, наприклад, з використанням функції «Trusted Places» Smart Lock [27], яка дозволяє розблокувати телефон, коли він знаходиться в безпечному місці.

Графічне представлення трьох рівнів аутентифікації Android наведено на рис. 1.9.

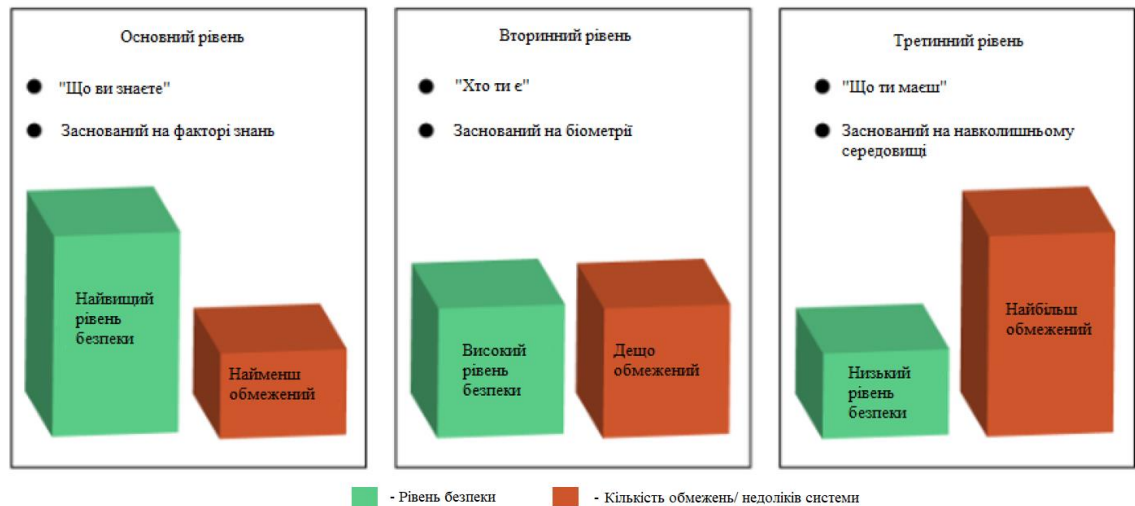


Рисунок 1.9 – Рівні аутентифікації Android [10]

Зосередимось на другому рівні, який пропонує компроміс між рівнем безпеки і зручності. Згідно з стандартами для операційної систем Android, наразі використовується 3 метрики для вимірювання рівня безпечності :

- Spoof Acceptance Rate (SAR): ймовірності того, що система аутентифікації на основі біометричних даних прийме підроблений зразок і пропустить зловмисника. Наприклад, фотографію чи аудіозапис голосу, які були зроблені заздалегідь сприйме як справжню людину;
- Imposter Acceptance Rate (IAR): ймовірності того, що система аутентифікації на основі біометричних даних приймає вхідні дані (фотографію, відбиток пальця, аудіозапис голосу), які призначені для імітації конкретного власника пристрою за справжні;
- False Acceptance Rate (FAR): показник того, як часто система аутентифікації на основі біометричних даних помилково сприймає невідомих персон як власника пристрою.

Спираючись на ці метрики, в стандарті від Android визначено три класи безпеки таблиця 1.1.

Таблиця 1.1 – Класифікація рівнів безпеки згідно з стандартом Android

Клас	Метрики	Процедура обробки біометричних даних
Клас 3 (Сильний)	SAR: 0-7% FAR: 1/50 тисяч FRR: 10%	Безпечна
Клас 2 (Слабкий)	SAR: 7-20% FAR: 1/50 тисяч FRR: 10%	Безпечна
Клас 1 (Зручний)	SAR: >20% FAR: 1/50 тисяч FRR: 10%	Небезпечна/Безпечна

Класи біометричної безпеки призначаються на основі безпечності процедури обробки біометричних даних в пристрої аутентифікації та трьох метрик – FAR, IAR та SAR. У випадках, коли атака самозванця (Imposter) не існує, розглядають лише FAR і SAR. При цьому в стандарті зазначається, що силіконові або керамічні маски обличчя не розглядаються під час розрахунків значень SAR та IAR для аутентифікації за зображенням обличчя.

### 1.3 Методи покращення точності роботи систем аутентифікації

Сучасні підходи до біометричної аутентифікації обмежені використанням лише одного типу біометричних даних, тобто вони мають одномодальний характер. Це може приводити до того що в деяких випадках вони через особливості своєї конструкції можуть функціонувати неефективно або ставати вразливими до певних видів атак.

Розглянемо деякі з проблем з якими може зіткнутися користувач при використанні біометричної аутентифікації [28, 29]:

- відбитки пальців: серед недоліків можна відзначити можливість втрати доступу користувачем після травм пальців. До того, система є вразливою до підробок через те, що користувачі залишають свої відбитки пальців в різних місцях, практично залишаючи свої "ключі" до системи повсюди. Отримавши відбиток пальця зломисники можуть виробити копію [8] і пройти аутентифікацію;
- райдужна оболонка очей: серед недоліків цієї системи можна відзначити високу вартість впровадження та необхідність якісного освітлення для її функціонування. Крім того, існує ризик втрати доступу у випадку втрати чи травми очей;
- розпізнавання обличчя: до недоліків цієї системи можна виділити зниження ефективності в умовах зміни освітлення, різних виразів обличчя та наявності лицьових аксесуарів, таких як окуляри, маски і татуювання. Крім того, розпізнавання обличчя може бути неефективним у випадку, якщо зломисники використовують спеціальні маски або діпфейки;
- розпізнавання голосу: серед недоліків можна відзначити низьку стійкість до спуфінгу (підміни) та чутливість до шуму в навколишньому середовищі, яка може заважати процесу аутентифікації.

Відзначимо що кожен метод біометричної аутентифікації має власні обмеження та переваги. Особливу увагу варто приділити розпізнаванню по обличчю і голосу, адже практичне застосування таких систем не потребує використання додаткових датчиків в мобільних пристроях. Отже, з концентруємо увагу на методах, що здатні подолати або пом'якшити приведені недоліки.

Якщо не враховувати методи покращення точності аутентифікації спрямовані на збільшення шарів штучної нейронної мережі, балансування набору даних для навчання, нормалізацію даних тощо то лишається не так багато підходів щодо підвищення точності розпізнавання облич. Звісно, можна покращити роботу системи застосовуючи кілька додаткових видів датчиків, так

як це зроблено в FaceID від Apple [18]. Зокрема, в даній системі використовується одразу 3 датчики:

1. лазерний точковий проектор, який проектує сітку маленьких інфрачервоних точок на обличчя користувача;
2. прожектор, який освітлює обличчя інфрачервоним світлом;
3. інфрачервона камера, яка робить інфрачервоне зображення користувача та зчитує отриманий результат.

В результаті роботи системи FaceIT отримується тривимірна модель обличчя користувача, що суттєво знижує ефективність атак, заснованих на використанні фото чи відеозапису людини. Такий метод, звісно, підвищує безпеку системи аутентифікації, проте потребує встановлення додаткових датчиків, що збільшує ціну пристрою.

Також існує метод підвищення надійності, коли під час автентифікації на екрані пристрою з'являються «підказки», наприклад повернути голову, кліпнути декілька разів очима або посміхнутися [30]. Такий метод дозволяє переконатись, що для проходження автентифікації не використовується попередньо записаний зразок. Проте такий підхід може займати від 10 до 15 секунд, що може виявитись незручним для користувачів, які потребують частої автентифікації. Схожий метод є і для розпізнавання голосу [31], коли для аутентифікації користувачі повинні прочитати «підказку» (число чи речення). Це дозволяє впевнитись, що голос не є записаним, проте зважаючи на останні досягнення таких штучних нейронних мереж як PaddleSpeech [32] та Bark-with-voice-clone [33], зловмисники можуть синтезувати мовний сигнал, якщо в них є зразок голосу користувача.

Альтернативним підходом до підвищення стійкості системи до атак підміни, це використання мультимодальності. Даний підхід заснований на аналізі взаємозв'язку між окремими типами біометричних даних (модальностей), наприклад між голосом і зовнішнім виглядом людини [12,13,14]. Відмітимо, що в літературі обмежені відомості щодо використання мультимодальних систем

для вирішення задач автентифікації та кібербезпеки. Проте за результатами даних досліджень встановлено, що існує певна взаємозалежність між особливостями обличчя (наприклад, формою носа, підборіддя, щоками тощо) та голосом. Отже, володіючи інформацією про голос людини, можна робити певні припущення щодо її зовнішності. Тому становить інтерес практичне застосування даного підходу для покращення стійкості системи розпізнавання обличчя за допомогою додаткової інформації, яку можна отримати з голосу. Звісно, такий підхід не може повністю вирішити проблему атак підміни, але він може значно ускладнити завдання для злоумисників, оскільки для атаки їм доведеться підробити одразу дві модальності.

#### 1.4 Стійкість систем розпізнавання обличчя

Стійкість систем автентифікації за допомогою розпізнавання обличчя безпосередньо залежить від точності роботи штучної нейронної мережі у виявленні та ідентифікації обличчя [39]. Як приклад можливо навести вплив різних видів завад на функціонування штучних нейронних мереж [15]. В даній роботі як завади використали окуляри, наліаки, маски та заваду на основі прогнозованого градієнтного спуску (PGD attack) [34]. Графічне відображення цих типів завад наведено на рис. 1.10.

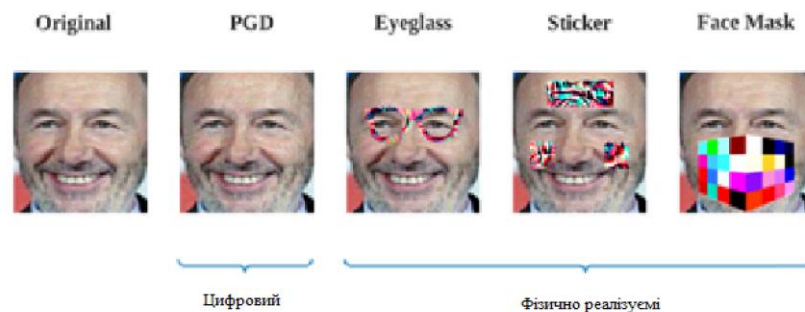


Рисунок 1.10 – Види завад [15]

Якщо завади в вигляді окулярів, наліпок та масок є інтуїтивно зрозумілими то атака PGD потребує пояснення. Атака прогнозованого градієнтного спуску — це вид атаки на моделі штучних нейронних мереж, який передбачає що зломисник має копію штучної нейронної мережі яку атакує, і може обчислити її градієнти. Атака на основі PGD намагається створити спотворення, яке максимізує втрати моделі на конкретному входному значенні, при цьому зберігаючи розмір спотворення меншим, ніж вказане значення [34]. Наглядний результат роботи такого методу атаки наведено на рис. 1.11, де верхній ряд це оригінальні зображення, а нижній ряд це зображення спотворене з використанням атаки PGD.

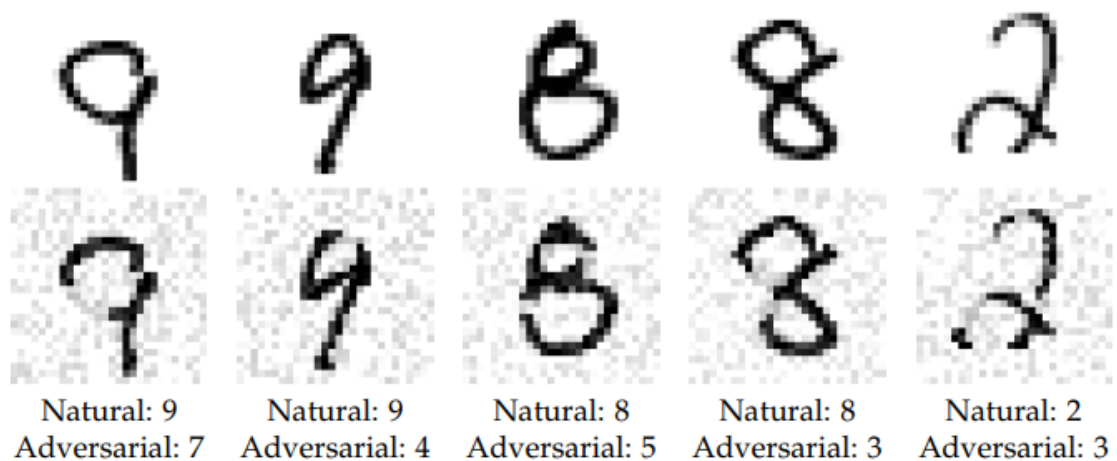


Рисунок 1.11 – Приклад спотворених зображень [34]

Для тестування стійкості були використані сучасні штучні неймережі VGGFace, FaceNet, ArcFace18, ArcFace50 та Arcface101. В таблиці 1.2 [15] наведені результати для вибірки зі 100 класів і 181 зображення.

Таблиця 1.2 – Вірогідності невдалого виявлення обличчя при використанні завад з рис. 1.10 та набору даних що містить 181 зображення для 100 класів.

Тип штучної нейронної мережі	Тип завади	Ймовірність невдалого розпізнавання обличчя
VGGFace	PGD	0,15
	Окуляри	0,23
	Наліпки	0,09
	Маска	0,56
FaceNet	PGD	0,21
	Окуляри	0,62
	Наліпки	0,61
	Маска	0,91
ArcFace18	PGD	0,08
	Окуляри	0,08
	Наліпки	0,00
	Маска	0,67
ArcFace50	PGD	0,05
	Окуляри	0,07
	Наліпки	0,00
	Маска	0,72
Arcface101	PGD	0,03
	Окуляри	0,02
	Наліпки	0,00
	Маска	0,67

За результатами аналізу даних з таблиці 1.2 можна зробити висновки, що збільшення кількості шарів (відповідно, складності) штучної нейромережі дозволяє нівелювати вплив завад. Зокрема при використанні маски як завади ймовірність що штучна нейронна мережа не зможе вірно розпізнати людину



зменшується від 0.91 для нейронної мережі FaceNet (22 шари) до 0.67 для мережі Arcface101 (101 шар). Це є очікуваним результатом, оскільки маска закриває найбільшу частину обличчя у порівнянні з окулярами та наліпками. Винятком з цього є VGGFace, яка дозволяє розпізнати лице з маскою з ймовірністю 0.56, незважаючи на те, що вона має лише 22 шари.

## **Висновки до розділу 1**

За результатом проведеного огляду біометричних систем аутентифікації встановлено обмеження їх практичного застосування. Зокрема вони вразливі до атак підміни (спуфінгу), і їхня точність може залежати від умов навколишнього середовища (наприклад, освітлення, шуму, бруду та температури). Для підвищення стійкості до спуфінгу розглянуто кілька варіантів, в тому числі й використання мультимодальних систем, які поєднують дві пов'язані модальності. Відмітимо відсутність літератури в якій досліджується використання залежності обличчя людини та її голосу з метою виявлення атак підміни обличчя. Тож становить інтерес проведення дослідження такого методу виявлення атак підміни.

Метою роботи є підвищення стійкості систем розпізнавання облич до атак підміни обличчя, заснованих на використанні масок. Отже, для досягнення поставленої мети необхідно надалі:

- Запропонувати і реалізувати модель штучної нейронної мережі, що здатна виявляти атаки підміни обличчя
- Порівняти запропонований метод із вже існуючими

## 2 МУЛЬТИМОДАЛЬНА МОДЕЛЬ ВИЯВЛЕННЯ АТАК ПІДМІНИ ОБЛИЧЧЯ

Як вже було показано в першому розділі роботи, системи розпізнавання обличчя, є вразливими до атак підміни. Тож для протидії цьому виду атак застосовують різні моделі класифікаторів, які визначають чи є обличчя на зображенні справжнім.

За результатами першого розділу встановлено, що наявні системи розпізнавання облич є вразливими до атак підміни, зокрема атак здійснених за допомогою спеціальних масок чи діпфейків, приклади яких зображені на рис. 2.1 та 2.2 відповідно.

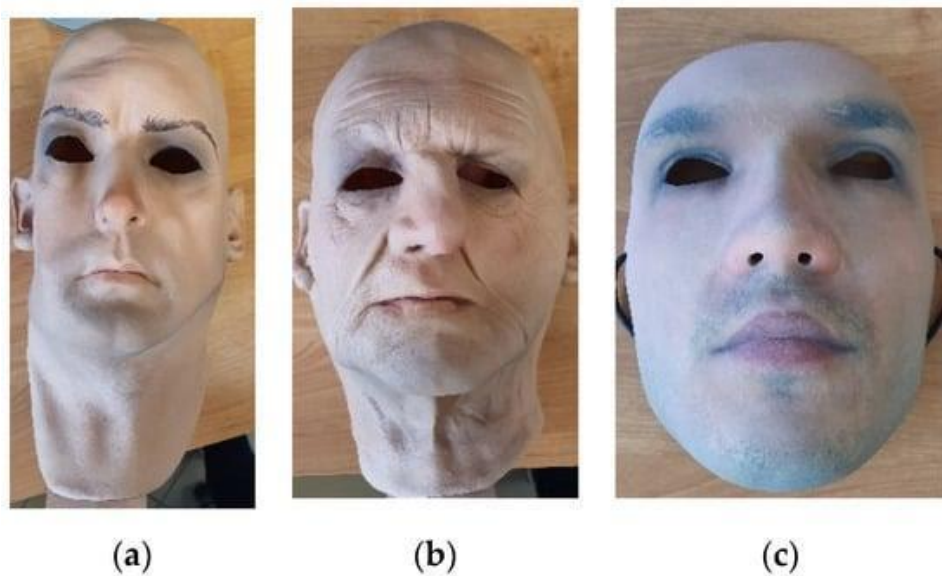


Рисунок 2.1 – Зображення латексних масок(a, b) та маски роздрукованої на 3D-принтері(c) [42]



Рисунок 2.2 – Зображення справжнього обличчя та обличчя підробленого за допомогою deepfacelab [43]

Для подолання даних обмежень запропоновано мультимодальний метод виявлення атак підміни обличчя. Особливістю методу полягає в тому, що не було досліджень які б використовували залежність між голосом і обличчям, для виявлення атак підміни при автентифікації.

Головна ідея запропонованої моделі виявлення атак підміни полягає в співставленні характеристик обличчя з голосом. Наприклад, якщо на зображенні представлено жіноче фото, а голос є чоловічий, то система має виявити спробу проведення атаки підміни. Тобто модель штучної нейронної мережі має перевіряти чи відповідає обличчя до голосу.

## 2.1 Аналіз взаємозалежності зовнішності людини і голосу

Для розуміння принципу роботи запропонованої мультимодальної моделі виявлення атак підміни необхідно розглянути як залежить зовнішність і голос людини. Усна мова є результатом залучення голосового апарату людини, який в свою чергу складається з трьох підсистем [40]:

- Система створення тиску, складається з діафрагми, легень, грудних і спинних м'язів. Ця підсистема забезпечує і регулює повітряний тиск, викликаючи вібрацію голосових зв'язок.

- Вібраційна система, складається з гортані та голосових зв'язок. Голосові зв'язки вібрують, створюючи звукові хвилі певної частоти. Ця підсистема впливає на висоту голосу.

- Резонуюча система(голосовий тракт), включає в себе горло, губи, ротову та носову порожнину.

За результатами дослідження [40,41] встановлено кореляцію між голосовим трактом і зовнішністю, а також між голосовим трактом і звуком голосу людини. Поєднання даних методів дозволяє оцінити ступінь залежності між зовнішністю людини за її голосом. Проте, в цих дослідженнях використовують знімки магнітно-резонансної томографії верхньої частини людини, які наразі неможливо отримати використовуючи мобільні пристрої. Тому становить інтерес результати дослідження [12-14], які намагаються встановити залежність безпосередньо між голосом людини і виглядом її обличчя. За результатами дослідження Speech2Face [12] показано що найбільший ступінь кореляції досягається між такими зовнішніми ознаками як ширина і висота носа, висота бокової поверхні верхньої губи та висота вермільйону губ. Використання ансамблю штучних нейронних мереж, кожна з яких розпізнає одну окрему частину обличчя потребує виконання великої кількості обчислень. Враховуючи наші обмежені обчислювальні ресурси, в даній роботі буде використано спрощений варіант моделі виявлення атак підміни. Спрощення полягає в використанні VGG-Face і зображень обличчя загалом, а не окремо носа і губ.

## **2.2 Набір даних для налаштування**

Було використано аудіовізуальний набір даних VoxCeleb2 [54] який поділяється на вибірку для налаштування та вибірку для перевірки. Даний набір даних поділений на дві частини перша містить в собі 1,092,009 відеозаписів для 5,994 людини, в свою чергу друга вибірка містить 36237 відеозаписів для 118 людей. Для налаштування було використано 947,387 відеозаписів для 5,994

людей. Всі відеозаписи були обрізані до трьох секунд (код наведено в додатку А). З кожного відеозапису було взято перший кадр з другої секунди. В якості детектора облич була використана SCRFD-10GF від Insightface [44,45] (додаток Б).

Також з кожного відео було виокремлено аудіозапис. Потім за допомогою віконного перетворення Фур'є було створено спектрограму для кожного аудіозапису. При створенні спектрограми були використані такі параметри:

- частота дискретизації аудіо: 16 кГц;
- кількість каналів звуку: 1;
- кількість вікон: 512;
- довжина вікна: 25 мс;
- довжина кроку: 10 мс.

Після створення спектрограм (див. додаток В), реальна і уявна частини окремо були стиснуті степеневим законом  $\text{sgn}(S) * S^{0.3}$  [12]. В результаті чого було отримано спектрограми розмірністю 257x301 (рис. 2.3)

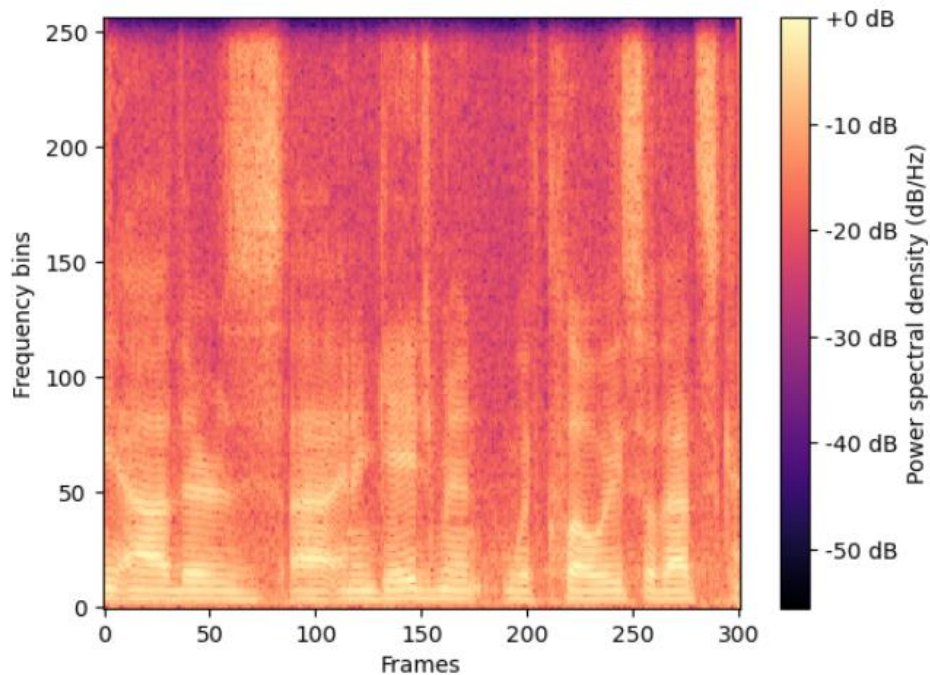


Рисунок 2.3 – Приклад обробленої спектрограми, яка в надалі буде використана для налаштування

## 2.3 Принцип роботи запропонованої моделі штучної нейронної мережі.

Запропонована модель штучної нейронної мережі має на меті визначити чи здійснюється, особою яка хоче пройти аутентифікацію, атака підміни чи ні. Для цього модель намагається встановити чи відповідає обличчя голосу. У випадку якщо голос не відповідає обличчю мережа має видавати “0”, що означає що зломисники намагаються здійснити атаку підміни за допомогою спеціальних масок чи діпфейків, та “1” у протилежному випадку.

Модель складається з трьох частин :

- Попередньо навчена VGG-Face – ця штучна нейронна мережа репрезентує зображення обличчя, як вектор (див. додаток Г).
- Мультиmodalна штучна нейронна мережа, яка на вхід приймає спектрограму голосу.
- Порівняння двох векторів з виходів першої і другої частини. По суті це є штучна нейронна мережа яка вирішує задачу бінарної класифікації (чи є атака підміни, чи ні).

Схему роботи запропонованої моделі зображено на рис. 2.4

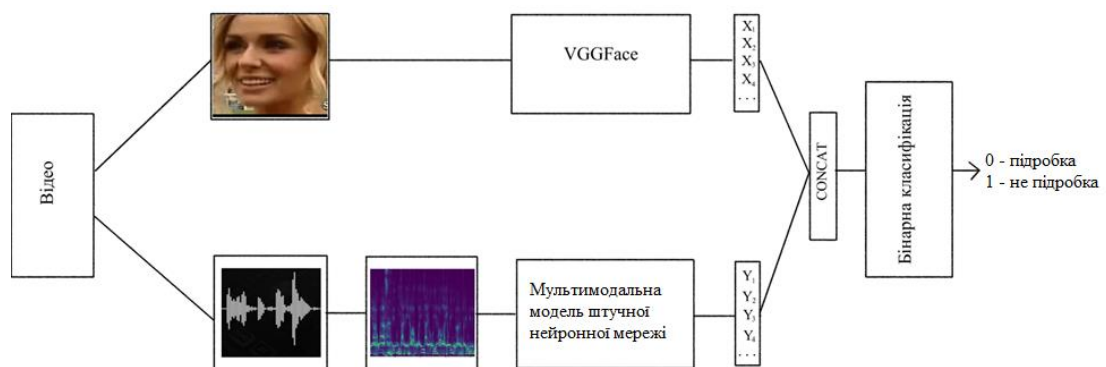


Рисунок 2.4 – Схеми роботи запропонованої моделі штучної нейронної мережі, що використовується для виявлення атаки підміни.

На вхід запропонованої моделі подається трьох секундне відео, на якому знаходиться людина що розмовляє, з якого виокремлюється аудіозапис та кадр що містить зображення обличчя людини. Зображення проходить через VGG-Face і перетворюється на вектор довжиною 4096. В свою чергу з аудіозапису створюється спектрограма з параметрами описаними в розділі 2.2, яка проходить через мультимодальну модель штучної нейронної мережі і також перетворюється у вектор довжиною 4096. Отримані два вектори об'єднуються в один вектор довжиною 8172, який подається на штучну нейронну мережу бінарної класифікації яка вирішує чи є обличчя відповідним до голосу. Якщо ні, то система має видавати 0, якщо так, то 1.

#### **2.4 Процес налаштування запропонованої моделі штучної нейронної мережі**

В якості оптимізатора для налаштування було використано ADAM [47] з темпом навчання 0.0001,  $\beta=0.9$ ,  $\epsilon=10^{-7}$ . Для реалізації усіх трьох частин запропонованої моделі було використано Tensorflow.

Було реалізовано два варіанти моделі штучної нейронної мережі, що описана в пункті 2.2. В першому варіанті частина що виконує бінарну класифікацію проходила налаштування окремо від частини що приймає на свій вхід спектрограму. В другому варіанті вони проходили налаштування одночасно. Розглянемо реалізацію обидва варіанти реалізації запропонованої моделі.

Спершу розглянемо модель штучної нейронної мережі, де кожна частина проходить налаштування окремо (див. додаток Г та Д). Оскільки VGG-Face (рис. 2.5) була використана у вже навченому вигляді і протягом всіх етапів є “замороженою” то з деталями її архітектури та процесом налаштування слід окремо ознайомитись [46].

layer type name	0 input	1 conv conv1_1	2 relu relu1_1	3 conv conv1_2	4 relu relu1_2	5 mpool pool1	6 conv conv2_1	7 relu relu2_1	8 conv conv2_2	9 relu relu2_2	10 mpool pool2	11 conv conv3_1	12 relu relu3_1	13 conv conv3_2	14 relu relu3_2	15 conv conv3_3	16 relu relu3_3	17 mpool pool3	18 conv conv4_1
support	–	3	1	3	1	2	3	1	3	1	2	3	1	3	1	3	1	2	3
filt dim	–	3	–	64	–	–	64	–	128	–	–	128	–	256	–	256	–	–	256
num filts	–	64	–	64	–	–	128	–	128	–	–	256	–	256	–	256	–	–	512
stride	–	1	1	1	1	2	1	1	1	1	2	1	1	1	1	1	1	2	1
pad	–	1	0	1	0	0	1	0	1	0	0	1	0	1	0	1	0	0	1

layer type name	19 relu4_1	20 conv conv4_2	21 relu relu4_2	22 conv conv4_3	23 relu relu4_3	24 mpool pool4	25 conv conv5_1	26 relu relu5_1	27 conv conv5_2	28 relu relu5_2	29 conv conv5_3	30 relu relu5_3	31 mpool pool5	32 conv conv6	33 relu relu6	34 conv conv7	35 relu relu7	36 conv conv8	37 softmax prob
support	1	3	1	3	1	2	3	1	3	1	3	1	2	7	1	1	1	1	1
filt dim	–	512	–	512	–	–	512	–	512	–	512	–	–	512	–	4096	–	4096	–
num filts	–	512	–	512	–	–	512	–	512	–	512	–	–	4096	–	4096	–	2622	–
stride	1	1	1	1	1	2	1	1	1	1	1	1	2	1	1	1	1	1	1
pad	0	1	0	1	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0

Рисунок 2.5 – Конфігурація шарів штучної нейронної мережі VGG-Face [52]

Схема, що зображує налаштування мультимодальної частини, наведена на рис. 2.6. Спершу береться відео та виокремлюється кадр з обличчям і аудіозапис. Потім з аудіозапису створюють спектрограму, як це описано в розділі 2.2. Після цього зображення обличчя подається на вхід попередньо навченої VGG-Face, яка видає на виході вектор довжиною 4096 елементів. В свою чергу мультимодальна мережа намагається передбачити на виході вектор створений VGG-Face. При цьому використовується функція втрат що описується формулою 5. Дана функція є спрощеною версією функції запропонованої в роботі Speech2Face [12].

Розмір однієї партії даних для налаштування мультимодальної моделі штучної нейронної мережі дорівнює трьом.

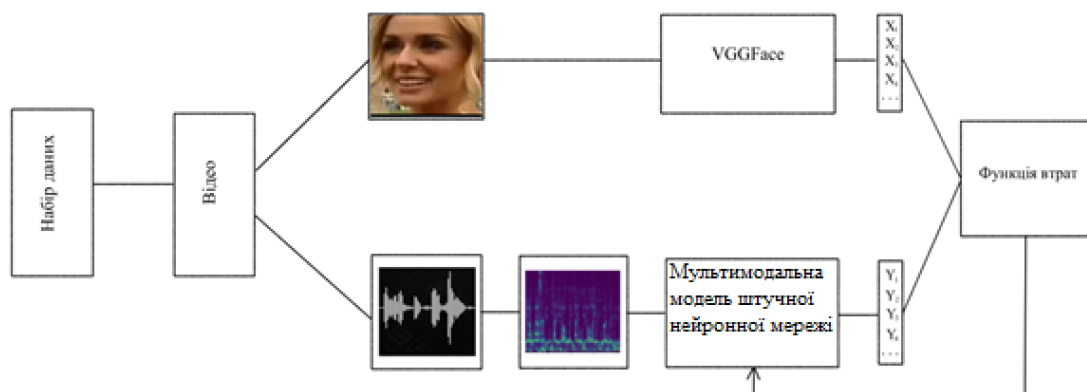


Рисунок 2.6 – Схема налаштування мультимодальної частини штучної нейронної мережі



$$Loss = \left\| \frac{y_{true}}{\|y_{true}\|} - \frac{y_{pred}}{\|y_{pred}\|} \right\|_2^2 + \sum_i^N p_i(y_{true}) \log(p_i(y_{pred})) \tag{5}$$

де  $y_{true}$  — вектор значень отриманий на виході з VGG-Face,  $y_{pred}$  — вектор значень отриманий на виході з мультимодальної частини мережі,  $p_i(y) = \frac{\exp(a_i/2)}{\sum_j \exp(\frac{a_j}{2})}$ .

Архітектура мультимодальної частини моделі штучної нейронної мережі наведена в таблиці 2.1.

Таблиця 2.1 – Архітектура мультимодальної штучної нейронної мережі

Шари	Канали	Крок зсуву	Розмір фільтра
Вхід	2	-	-
CL ReLU BN	32	1	4
CL ReLU BN	64	1	4
CL ReLU BN	64	1	4
MP		2x1	2x1

Продовження таблиці 2.1 – Архітектура мультимодальної штучної нейронної мережі

CL ReLU BN	64	1	4
MP		2x1	2x1
CL ReLU BN	64	1	4
MP		2x1	2x1
CL ReLU BN	128	1	4
MP		2x1	2x1
CL ReLU BN	256	2	4
CL	256	2	4
AP		1	2x62
ППШ	4096	1	1

Після п'яти епох налаштування мультимодальної моделі, йде налаштування моделі, що виконує бінарну класифікацію. Оскільки не існує набору даних який містив би одразу зображення людей в масках і їх голоси, то було взято справжні обличчя людей і справжні голоси, після чого їх було

перемішано і сформовано пари обличчя-голос. Такий підхід ґрунтується на припущенні що зломисники здатні створити маски чи дідфейки що абсолютно точно можуть повторити риси обличчя людини маючи її фотографії.

Для налаштування такої мережі було сформовано 20,915,520 пар голосів і облич, 50% з яких це ті пари в яких обличчя дійсно відповідає голосу, а 50% це такі пари де голос не сходиться з обличчям. На вхід мережі бінарної класифікації поступає вектор довжиною 8,192, отриманий шляхом об'єднання векторів VGG-Face і мультимодальної штучної нейронної мережі. На виході від такої мережі очікується 1 якщо голос і обличчя сходяться, і 0 коли ні. Схема налаштування штучної нейронної мережі бінарної класифікації зображено на рис. 2.7. Для налаштування було використано функцію втрат бінарної крос-ентропії. Процес налаштування тривав п'ять епох. Також в таблиці 2.2 зображено архітектуру цієї моделі штучної нейронної мережі.

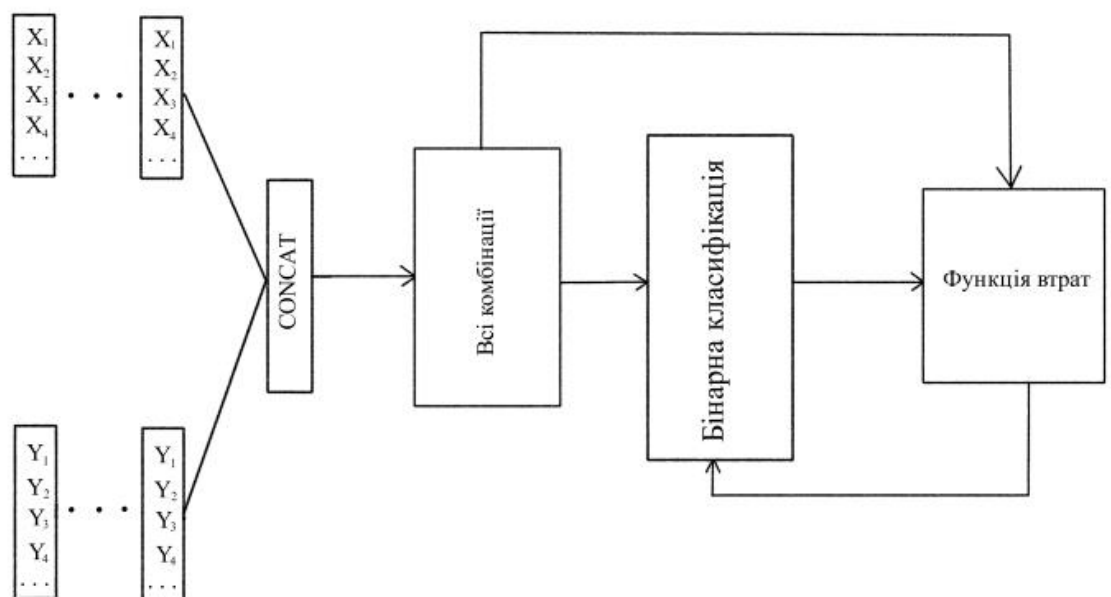


Рисунок 2.7 – Схема налаштування штучної нейронної мережі що виконує бінарну класифікацію

Таблиця 2.2 – Архітектура штучної нейронної мережі що виконує бінарну класифікацію

Шари	ПШ	ПШ	ПШ
Канали	128	16	1
Крок зсуву	1	1	1
Розмір фільтра	1x1	1x1	1x1

Тепер розглянемо варіант моделі штучної нейромережі коли налаштування в обох частинах проходить одночасно (див. додаток Е). При такому підході використовуються ті ж самі параметри шарів штучної нейронної мережі. Єдина відмінність полягає в тому, що шари мультимодальної частини “розморожені” і налаштування разом з частиною що виконує бінарну класифікацію. При цьому використовується функція втрат бінарної крос-ентропії. Графічне зображення процесу налаштування наведено на рис 2.8.

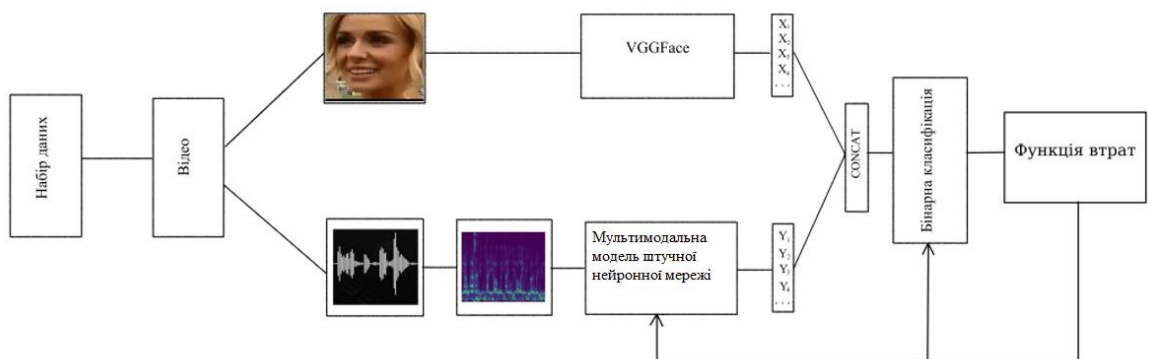


Рисунок 2.8 – Схема налаштування одразу мультимодальної частини та частини що виконує бінарну класифікацію

## **Висновки до розділу 2**

В даному розділі було запропоновано модель штучної нейронної мережі для боротьби з різновидом атак підміни, коли зловмисники використовують різноманітні маски, при розпізнаванні обличчя. Також було описано принцип роботи та архітектуру такої штучної нейронної мережі. Було створено два варіанти реалізації описаної штучної нейронної мережі: в першому всі частини налаштовувались окремо, поки параметри інших частин були заморожені; в другому варіанті параметри заморожені тільки у VGG-Face.

## З ПОРІВНЯННЯ СУЧАСНИХ МЕТОДІВ ТА ЗАПРОПОНОВАНОЇ МОДЕЛІ ШТУЧНОЇ НЕЙРОННОЇ МЕРЕЖІ

### 3.1 Результати тестування

Для тестування системи було сформовано дві тестові вибірки. Одна з яких містить 4,911 відео для 118 людей з тестової вибірки VoxCeleb2. На кожну людину припадає від 6 до 72 відеофайлів. В цій вибірці знаходяться люди яких не було в вибірці для налаштування. Друга тестова вибірка містить 144,622 відео для 5,994 людей, також з VoxCeleb2. Тут на кожну людину припадає від 6 до 91 відеофайлу. В цій тестовій вибірці знаходяться нові відео для людей які були в вибірці для налаштування.

Обидві тестові вибірки були оброблені тим же чином що описаний в розділі 2.2. Після чого, завдяки ним, було протестовано обидві моделі штучної нейронної мережі з пункту 2.4. Результати представлені у таблицях 3.1 та 3.2. Де наведені показники точності та відсоток FAR та FRR серед тих випадків коли модель штучної нейронної мережі не впоралась з поставленою задачею.

Таблиця 3.1 – Результати тестування моделі з окремо навченими частинами

Вибірки даних	Точність	FAR, %	FRR, %
Вибірка 1(118 людей)	0,502	99,8	0,2
Вибірка 2(5994 людей)	0,993	92,8	7.2

Таблиця 3.2 – Результати тестування моделі штучної нейронної мережі коли налаштування в обох частинах проходить одночасно

Вибірки даних	Точність	FAR, %	FRR, %
Вибірка 1(118 людей)	0,738	94,5	5,5
Вибірка 2(5994 людей)	0,995	85,1	14,9

Як видно по даних наведених в таблицях 2.3 та 2.4 система видає доволі високу точність на вибірці в якій знаходяться нові відео людей які вже відомі для системи, проте точність для вибірки в якій знаходяться нові відео для нових людей є доволі низькою. У випадку коли використовувалась модель з окремо навченими частинами система і вибірка з 118 людей, можна сказати що система взагалі не набула жодних узагальнюючих здібностей і просто завжди на виході видавала 1, тим самим пропускаючи будь-кого. Проте при використанні другої моделі штучної нейронної мережі і тієї ж вибірки вже з'явилася певна здатність до узагальнення, тож надалі є сенс розглядати тільки другу модель. Видно що в усіх випадках показник FAR є значно вищими ніж FRR, що може свідчити про те що система буде пропускати багато порушників.

### **3.2 Порівняння сучасних методів та запропонованої моделі штучної нейронної мережі**

Виявлення атак підміни обличчя за допомогою масок є більш складною задачею ніж коли використовуються роздруковані фотографії чи демонстрація відеозапису. Тож не коректно порівнювати запропонований вище метод з такими методами як виявлення моргання очей чи муару, що призначені для виявлення іншого різновиду атак підміни обличчя. Тож будуть розглядатися лише такі

методи які направлені саме на боротьбу з різновидом атак підміни з використанням масок як наведено в таблиці 3.3.

Таблиця 3.3 – Короткий опис переваг і недоліків різних методів виявлення атак підміни при розпізнаванні обличчя

Метод	Переваги	Недоліки	Точність	Посилання
1.Застосування мультиспектрального короткохвильового інфрачервоного зображення	Хороша стійкість; хороша можливість узагальнення	Потрібні спеціальні освітлювальні прилади	FRR < 5.00%, FAR < 7.00%	[48]
2.Аналіз текстури зображення	Простота реалізації; висока точність	Низька стійкість; висока залежність від роздільної здатності зображення	HTER(Half-Total Error Rate) = 0.03%	[49]
3.Аналіз форми обличчя	Простота реалізації; висока точність	Висока обчислювальна вартість; чутливість до високоякісних масок	HTER(Half-Total Error Rate) = 0.91%	[50]



Продовження таблиці 3.3 – Короткий опис переваг і недоліків різних методів виявлення атак підміни при розпізнаванні обличчя

4. Використання теплових зображень	Складність підробки теплових знімків; хороша здатність узагальнення	Висока обчислювальна вартість; необхідна наявність спеціальних пристроїв	APCER(Attack Presentation Classification Error Rate) = 3.5%	[51]
5. Запропонований в цій роботі метод	Нема потреби в наявності спеціальних пристроїв	Висока обчислювальна вартість; Високий FAR; Тривалість	FAR=0.43% (вибірка 1) FAR=24.8% (вибірка 2) FAR=0.07% (вибірка 1) FRR=1.4%(вибірка 2)	-

При застосуванні мультиспектрального короткохвильового інфрачервоного методу можна визначити з якого матеріалу складаються об'єкти, в тому числі і визначити чи справжня шкіра у суб'єкта, чи це маска. Фізіологічні властивості шкіри, які в значній мірі змінюються від віку, статі чи типу шкіри людини не впливають на її властивості в інфрачервоному спектрі [53]. Тож цей метод полягає в комбінуванні виявлення області шкіри на зображеннях зроблених в інфрачервоному діапазоні та розпізнаванні обличчя на зображенні зробленому в діапазоні видимого світла.

При застосуванні методу що аналізує текстуру обличчя, необхідно мати зображення високої роздільної здатності, бо цей метод заснований на аналізі змін

мікро текстур та кольору в області очей та носа. Різкі зміни кольору обличчя в цих областях можуть свідчити про наявність маски на людині.

Для методу заснованому на аналізі форми обличчя необхідно окрім фотографії людини мати ще і зображення глибини що, власне, відображає відстань від об'єктів до камери. Метод полягає в тому аби встановити відповідність між цими зображеннями. Недоліком такого підходу є те що чим якісніше маска, тим більша потрібна висока роздільна здатність зображення глибини.

Інший метод застосовує зображення зняті за допомогою тепловізору. Ці зображення аналізуються за допомогою методів машинного навчання аби виявити ділянки обличчя які мають аномально низьку температуру, як для людського тіла, що може свідчити про застосування масок.

Як вже було сказано в розділі 2, запропонований метод ґрунтується на встановленні відповідності між голосом та обличчям людини. Цей процес відбувається за допомогою комбінації трьох частин/модулів. Основною перевагою такого підходу є те що зникає необхідність в використанні доволі специфічного обладнання та датчиків, бо для описаного методу необхідні лише звичайна камера та мікрофон. Також є можливість для налаштування використовувати звичайні відео, при цьому практично відсутні обмеження по розміру набору даних, в той час, як інші дослідження використовували набори даних обмежені кількістю масок. До недоліків слід віднести низьку точність та високу кількість обчислень для налаштування.

### **Висновки до розділу 3**

Перш за все слід зазначити, що порівняння різних методів описаних в цьому пункті ускладнюються тим що в раніше опублікованих дослідженнях застосовувались, хоч і не якісні, проте справжні маски. А запропонований метод ґрунтується на припущенні що зловмисники використовують високоякісні маски

які неможливо візуальним шляхом відрізнити від справжнього обличчя, тож для налаштування і тестування були використані справжні обличчя без масок.

Після цього було проведено тестування з застосуванням двох вибірок: перша вибірка містила нові зразки для нових людей, яких система ще не бачила; друга вибірка містила нові зразки для тих людей на яких проводилось налаштування. В результаті тестування на першій вибірці можна сказати що система набула здатності до узагальнення лише в випадку коли налаштування усіх частин (окрім VGG-Face) проходило одночасно. В результаті тестування на другій вибірці можна зробити висновок, що, незважаючи на те що вона значно більше першої, дані в ній не достатньо відрізняється від даних використаних для налаштування. Також За результатами що наведені в таблицях 3.1 і 3.2 зрозуміло що запропонована модель штучної нейронної мережі в будь-якому випадку має високий рівень FAR, що є поганим в випадку коли безпеки є основним критерієм.

Оскільки отримана точність запропонованого методу боротьби з атаками підміни обличчя значно поступається вже відомим методом можна зробити висновок що запропонована система більше обмежень. Проте слід зазначити що якби була технічна можливість провести дослідження з більшою кількістю людей з моделлю штучної нейронної мережі що має більше параметрів то існує потенційна можливість досягти більшої точності. Крім того, зазначимо що використовувався спрощений варіант з використанням зображень всього обличчя, а не тільки носа і губ. Очевидно що колір волосся, очей, зачіска тощо ніяк не впливають на формування звуку голосу людини, тож вони лише є зайвими даними які ускладнюють налаштування моделі штучної нейронної мережі. Також враховуючи те що в наборі даних VoxCeleb2 представлені люди різної статі, національності, вікових групи та різні різновиди мов, що записані на різні типи мікрофонів то 5994 людини це все ще мала вибірка.

## **4 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ**

Обов'язковим розділом магістерської дисертації має бути розділ присвячений розробці стартап-проекту. В цьому розділі необхідно розглянути маркетингові аспекти створення стартапу, а саме: відібрати ідею для стартапу, створити концепцію продукту, визначити перспективи реалізації на ринку і розробити маркетингову стратегію.

Першим кроком є обрання назви проекту. Було вирішено дати цьому стартап проекту назву “TFF” (True False Face).

### **4.1 Опис ідеї проекту**

В цьому підпункті було зібрано інформацію з минулих розділів і визначено ідею стартапу, напрямок використання та вигоду для користувачів у порівнянні з раніше описаними методами виявлення атак підміни (данні занесено в табл. 4.1).

Таблиця 4.1 – Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
<p>Суть проекту полягає в створенні програмного забезпечення що визначає відповідність голосу та зображення людини для виявлення атак підміни, що здійснюються за допомогою масок (див. розділ 2).</p>	<p>Використання в мобільних пристроях при біометричній аутентифікації користувачів</p>	<p>Потенційно висока стійкість до атак підміни з використанням масок. Зручність використання аутентифікації по зображенню обличчя. Немає необхідності в додаткових давачах (інфрачервоні прожектори, тепловізори, датчики глибини зображення тощо). Для запропонованого методу необхідними є лише мікрофон та камера, що наразі наявні майже в усіх смартфонах.</p>

Наступним кроком є проведення аналізу потенційних техніко-економічних переваг ідеї у порівнянні з конкурентами. Слід зазначити що чіткого переліку техніко-економічних властивостей та характеристик проектів-конкурентів (табл. 4.2) не існує, адже, по характеристиках системи виявлення атак підміни, зловмисники можуть визначити слабкі місця і прогалини в безпеці пристрою. Тож наразі точно оцінити ті чи інші характеристики проектів-конкурентів не можливо.

Таблиця 4.2 – Характеристики ідеї проекту

Техніко-економічні і характеристики ідеї	Товари/концепції (потенційні) конкурентів				W	N	S
	Мій проект	Конкурент 1 – Використання теплових зображень [51]	Конкурент 2 – Аналіз текстури зображення [49]	Конкурент 3 – Аналіз форми обличчя [50]			
Тривалість автентифікації	Через необхідність робити запис голосу, тривалість значно збільшується	Наразі компактні тепловізори які здатні фізично вміститися в мобільний пристрій, мають доволі низьку кадрову частоту у порівнянні зі звичайними камерами.	Для даного методу використовують звичайні фотографії	Тривалість менша ніж у мого проекту, але більше ніж у другого конкурента	+		

Продовження таблиці 4.2 – Характеристики ідеї проекту

Вартість	Немає необхід ності в викорис танні додатко вих давачів	Вартість тепловізорів	Для даного підходу необхідно ю є камера з високою роздільно ю здатністю	Для даного методу необхідною є камера глибини			+
Точність	Як вже було показан о в розділах 2 та 3 точність є недоліко м такої системи	Має найнижчу заявлену точність серед конкурентів	Має найвищу заявлену точність серед конкурентів	Має помірну заявлену точність серед конкурентів		+	

#### 4.2 Технологічний аудит ідеї проекту

Для того аби визначити здійсненність стартап проекту з технічної точки зору було проведено аудит технологій, за допомогою яких можна реалізувати стартап проект. Результати аудиту занесені в таблицю 4.3.

Таблиця 4.3 – Технологічна здійсненність ідеї проекту

№	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
	Використання в мобільних пристроях при біометричній аутентифікації користувачів	Tensorflow, Pytorch, Insightface, OpenCV, Pandas, Cudann та інші (див. додатки А-Е); Набори даних VoxCeleb2, VoxCeleb1, AWSpeech.	Необхідно вдосконалити запропоновану систему для досягнення більшої точності	Весь програмний код доступний, проте доступ до обчислювальних можливостей та даних для налаштування системи обмежений
Обрана технологія реалізації ідеї проекту: для програмного втілення проекту було обрано Tensorflow, Insightface OpenCV, Pandas та інші (див. додатки А-Е), для налаштування моделі штучної нейронної мережі було обрано VoxCeleb2 та відеокарти від Nvidia.				

Отже, бачимо що всі необхідні технології доступні, проте деякі з них доступні лише в обмеженому масштабі. В такому випадку для збільшення масштабу необхідні фінансові вкладення.



### 4.3 Аналіз ринкових можливостей запуску стартап-проекту

Дослідження ринку — це процес збору і аналізу інформації про ринок, який дозволяє підприємству отримати уявлення про його стан, перспективи розвитку та можливості для власної діяльності. Одним із напрямків комплексного дослідження ринку є аналіз ринкових можливостей (таблиця 4.4). Ці можливості можуть бути пов'язані з попитом на товар або послугу, поведінкою споживачів, конкурентним середовищем або іншими факторами. На основі інформації, отриманої в результаті аналізу ринкових можливостей, підприємство розробляє стратегію і тактику маркетингу. Ця стратегія визначає загальні цілі і напрями маркетингової діяльності підприємства, а тактика - конкретні заходи, які будуть реалізовані для досягнення цих цілей.

Таблиця 4.4 – Попередня характеристика потенційного ринку стартап-проекту

Показники стану ринку (найменування)	Характеристика
Кількість головних гравців, од	3
Загальний обсяг продаж, грн/ум.од	Інформація відсутня в джерелах з відкритим доступом
Динаміка ринку (якісна оцінка)	Зростає
Наявність обмежень для входу (вказати характер обмежень)	Необхідно доопрацювати модель штучної нейронної мережі для чого потрібні фінансові вкладення
Специфічні вимоги до стандартизації та сертифікації	Наведені в розділі 1.2
Середня норма рентабельності в галузі (або по ринку), %	14.2% [60]

На основі кількості основних гравців, зростаючої динаміки ринку, невеликої кількості конкурентів та середньої норми рентабельності можна зробити висновок, що наразі ринок для входження стартап-продукту є привабливим.

Надалі слід встановити основні групи клієнтів та їх властивості (таблиця 4.5) для створення приблизного переліку вимог до продукту стартап проекту.

Таблиця 4.5 – Характеристика потенційних клієнтів стартап-проекту

№	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Необхідність підвищеного рівня стійкості до атак підміни при розпізнаванні облич	Виробники мобільних пристроїв	Зацікавлені у здешевленні виробництва мобільних пристроїв	Надійна система виявлення атак підміни за мінімальні фінансові вкладення

Продовження таблиці 4.5 – Характеристика потенційних клієнтів стартап-проекту

2	Необхідність наявності швидкого та зручного методу виявлення атак підміни обличчя	Користувачі мобільних пристроїв, банки, фінансові установи, підприємства та інші організації, які потребують надійного способу захисту своїх систем від несанкціонованого доступу	Зацікавлені у зручному і надійному способі проходження автентифікації	Зручний метод проходження автентифікації
---	---	---	---	--

Зваживши характеристики потенційних клієнтів, ми переходимо до аналізу ринкового оточення, тобто складаємо таблиці що містять фактори загроз та фактори можливостей (табл. 4.6 та 4.7 відповідно).

Таблиця 4.6 – Фактори загроз

№	Фактор	Зміст загрози	Можлива реакція компанії
1	Недостатній досвід і знання	Необхідність значного досвіду та знань у галузі машинного навчання, штучного інтелекту та кібербезпеки.	Зібрати команду з досвідченими фахівцями у галузі
2	Недостатній досвід у сфері підприємництва	Розробка проекту також вимагає певного досвіду у сфері бізнесу.	Навчитися основам бізнесу, таким як маркетинг, продажі та управління
3	Нестабільні економічні та політичні умови	Політичні та економічні умови в Україні можуть впливати на успіх стартапу.	Розробка стратегії виходу на ринок, яка враховує політичні та економічні умови в Україні
4	Конкуренція	Можливо через високу конкуренцію на ринку, буде важко досягти власної ідентичності	Доопрацювання та покращення характеристик системи

Таблиця 4.7 – Фактори можливостей

№	Фактор	Зміст можливості	Можлива реакція компанії
1	Інвестиційний потенціал	Зацікавлення інвесторів у запропонованому програмному забезпеченні	Виконання вимог від інвесторів
2	Розвиток і покращення стартапу	Стартап має потенціал з підвищення точності і швидкості	Розвивати знання про нові технології в галузі штучного інтелекту
3	Інтеграція програмного продукту у інші галузі	Можливість знаходження нових сфер використання запропонованого програмного продукту	Розвивати знання про нові методи застосування штучного інтелекту в світі

Далі ми здійснюємо аналіз пропозицій: визначаємо основні особливості конкуренції на ринку (табл. 4.8).

Таблиця 4.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства
1. Тип конкуренції: чиста	Всі конкуренти мають рівні можливості розвитку	Виявлення недоліків конкурентів і їх усунення у власному продукті

Продовження таблиці 4.8 – Ступеневий аналіз конкуренції на ринку

2. За рівнем конкурентної боротьби : міжнародна	Продукти конкурентів доступні по всьому світу	Охопити спочатку ринок в одній країні, а вже потім вести роботу по всьому світу
3. За галузевою ознакою: внутрішньогалузева	Галузь біометричної автентифікації	Проведення спроб впровадження продукту в інші галузі
4. Конкуренція за видами товарів: - товарно-видова	У всіх конкурентів різні види товарів які виконують одну і ту ж функцію.	Розширення функціонала програмного продукту
5. За характером конкурентних переваг: цінова	Товар має цінову конкурентну перевагу	Зробити систему зрозумілою для користувачів аби більш наочно продемонструвати свої переваги
6. За інтенсивністю: не марочна	Ринок має низьку різноманітність	Зробити систему зрозумілою для користувачів аби більш наочно продемонструвати свої переваги

Для більш детального аналізу конкуренції застосовують модель М. Портера що передбачає п'ять сил, які впливають на конкуренцію (табл. 4.9). Використовуючи цю модель визначимо, чи можливо працювати на ринку,

ураховуючи конкурентну ситуацію, а також ключові переваги, які повинен мати проект, щоб успішно конкурувати на ньому.

Таблиця 4.9 – Аналіз конкуренції в галузі за М. Портером

	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
Складові аналізу	Компанії які просувають інші методи біометричної автентифікації	Компанії які використовують інші методи виявлення атак підміни	Відсутні	Усі користувачі мобільних пристроїв що використовують автентифікацію за обличчям	Інші методи виявлення атак підміни обличчя
Висновки:	Системи не є досконалими. Це створює можливість і для нових компаній.	Вхід на ринок можливий, але вимагає значних фінансових ресурсів для вдосконалення існуючої системи	В цій галузі розробники і постачальники часто є одним і тим же	Клієнти можуть лише частково впливати на умови роботи ринку	Жодних обмежень не має якщо вдасться доопрацювати систему до прийнятної точності

За результатом аналізу таблиці 4.9 можна зробити висновок що стартап конкурентоспроможний. Також основі даних з цієї таблиці та таблиць 4.2 і 4.5-4.7 було визначено і обґрунтовано перелік факторів конкурентоспроможності стартап-проекту (табл. 4.10).

Таблиця 4.10 – Обґрунтування факторів конкурентоспроможності

№	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Зручність для користувачів	Користувачі не повинні запам'ятовувати і вводити складні паролі
2	Стійкість до якості масок	Особливістю системи є те що вона гіпотетично повинна працювати однаково з усіма видами масок
3	Немає необхідності в специфічних пристроях	Для роботи системи необхідними є лише мікрофон і камера, що присутні в усіх сучасних смартфонах

За урахуванням факторів конкурентоспроможності, що наведенні у таблиці 4.10, ми проведемо аналіз сильних та слабких сторін стартап-проекту (таблиця 4.11).

Таблиця 4.11 – Порівняльний аналіз сильних і слабких сторін стартап проекту

Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з [49]						
		-3	-2	-1	0	+1	+2	+3



Продовження таблиці 4.11 – Порівняльний аналіз сильних і слабких сторін стартап проекту

Зручність для користувачів	10							+
Стійкість до якості масок	20	+						
Немає необхідності в специфічних пристроях	17				+			

Завершальним кроком у ринковому аналізі можливостей для впровадження проекту є створення SWOT-аналізу матриці, що враховує сильні сторони (Strengths) та слабкі сторони (Weaknesses), загрози (Threats) та можливості (Opportunities) (таблиця 4.12). Вона базується на ринкових загрозах та можливостях, виділених з аналізу факторів, а також на сильних і слабких сторонах (див. таблицю 4.11). Перелік ринкових загроз та можливостей очевидно формується на основі аналізу факторів, що становлять загрози та можливості в маркетинговому середовищі.

Таблиця 4.12 – SWOT- аналіз стартап-проекту

Сильні сторони: немає необхідності в специфічних пристроях, стійкість до якості масок.	Слабкі сторони: тривалість автентифікації, точність.
Можливості: розвиток і покращення системи, інвестиційний потенціал.	Загрози: недостатній досвід і знання, недостатній досвід у сфері підприємництва, нестабільні економічні та політичні умови.

Використовуючи SWOT – аналіз стартап-проекту (табл. 4.12) встановлюються альтернативи ринкової поведінки (табл. 4.13). Для цих альтернатив проводиться оцінка ймовірності отримання необхідних ресурсів, а також кількості часу необхідного для реалізації стартапу.

Таблиця 4.13 – Альтернативи ринкового впровадження

№	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Шукати інвесторів для доопрацювання проекту	0,7	1-3 роки
2	Кредитна позика для реалізації проекту	0,9	2 роки
3	Накопичення коштів для доопрацювання власними силами	0,99	3 роки і більше

Проаналізувавши данні з таблиці 4.13 було обрано останню альтернативу, яка передбачає накопичення коштів для доопрацювання власними силами, такий варіант дозволяє зробити проект незалежним від інвесторів чи фінансових установ, а також є найбільш ймовірним.

#### 4.4 Розроблення ринкової стратегії проекту

Для розробки ринкової стратегії стартапу спершу необхідно визначити стратегію охоплення ринку, що включає в себе опис цільових груп потенційних споживачів (табл. 4.14).

Таблиця 4.14 – Вибір цільових груп потенційного користувача

№	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Покупці консюмеристських мобільних пристроїв	середня	середня	середня	низька
2	Корпоративний сегмент	висока	висока	середня	середня
Які цільові групи обрано: обидві групи					

Оскільки всі описані групи зацікавлені в створюваному продукті, було обрано їх усі як цільову аудиторію, що означає що буде використовуватись стратегія масового маркетингу.

Для подальшої роботи з цими групами цільової аудиторії необхідно створити базову стратегію розвитку (табл. 4.15) та стратегію конкурентної поведінки (табл. 4.16).

Таблиця 4.15 – Визначення базової стратегії розвитку

№	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Накопичення коштів для доопрацювання власними силами	Стратегія полягає в поступовому збільшенні долі ринку що використовує запропоновану систему	Високий рівень стійкості до масок різної якості, відсутність необхідності встановлення додаткових давачів у мобільні пристрої	Стратегія спеціалізації

Таблиця 4.16 – Визначення базової стратегії конкурентної поведінки

Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
Ні, вже існують проекти які вирішують ту ж задачу, але іншим способом	Забиратиме існуючих	Ні не буде. Запропонована система має інший принцип роботи, тому це не можливо з технічної точки зору.	Стратегія виклику лідера

На основі аналізу вимог споживачів з обраних сегментів (табл. 4.5), а також з урахуванням обраних стратегій розвитку (табл. 4.15) та конкурентної поведінки (табл. 4.16), було сформовано ринкову позицію торговельної марки/проекту (табл. 4.17).

Таблиця 4.17 – Визначення стратегії позиціонування

№	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1	Надійний, швидкий та зручний спосіб виявлення атак підміни обличчя	Стратегія спеціалізації	Стартап проект пропонує рішення за якого зникає необхідність використання специфічних пристроїв для виявлення атак підміни обличчя, також такий метод є стійким до масок різної якості.	Надійний, заощадливий та зручний метод виявлення атак підміни обличчя

Отже, враховуючи, що проект націлений на широке коло споживачів, для нього обрано стратегію масового маркетингу. Як базову стратегію розвитку

обрано спеціалізацію, яка дозволяє створити продукт, що відповідає потребам споживачів. За стратегію конкурентної поведінки взято виклик лідеру, що передбачає завоювання позиції лідера на ринку згодом.

#### **4.5 Розроблення маркетингової програми стартап-проекту**

Першим кроком у розробці маркетингової стратегії є формування маркетингової концепції товару, який отримає споживач.

Маркетингова концепція товару – це система поглядів, що визначає, яким повинен бути товар, щоб задовольняти потреби споживачів і забезпечувати успіх компанії на ринку. Для формування маркетингової концепції товару необхідно провести аналіз конкурентоспроможності товару. Аналіз конкурентоспроможності товару дозволяє компанії визначити, які характеристики товару необхідно вдосконалити, щоб він був більш конкурентоспроможним. Для цього в таблиці 4.18 було підсумовано попередні результати і визначено ключові переваги потенційного товару.

Таблиця 4.18 – Визначення ключових переваг концепції потенційного товару

№	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Проведення автентифікації з надійним, швидким та зручним способом виявлення атак підміни обличчя	Простий та стійкий спосіб виявлення атак підміни обличчя	Стартап проект пропонує рішення за якого зникає необхідність використання специфічних пристроїв для виявлення атак підміни обличчя, також такий метод є стійким до масок різної якості. Також існує перспектива значного підвищення точності.

Надалі розробляється трирівнева маркетингова модель товару, яка дозволяє визначити, яким повинен бути товар, щоб задовольняти потреби споживачів і забезпечувати успіх компанії на ринку:

- Перший рівень моделі – ідея продукту та/або послуги. На цьому рівні необхідно визначити, що саме буде пропонувати компанія споживачам.
- Другий рівень моделі – фізичні складові товару. На цьому рівні необхідно визначити, якими будуть характеристики товару.
- Третій рівень моделі – особливості процесу надання товару. На цьому рівні необхідно визначити, як товар буде надаватися споживачам.

Таблиця 4.19 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Система виявлення атак підміни обличчя		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Тривалість автентифікації	Нм	Тх
	2.Стійкість до атак з застосуванням масок різної якості	М	Тх
	3. Немає необхідності в додаткових пристроях	Нм	Вр
	Якість: стандарти наведені в розділі 1.2.		
	Марка: @sasha@, “TFF”		
III. Товар із підкріпленням	До продажу: інформування про переваги продукту і гарантія якості		
	Після продажу: підтримка та регулярні оновлення		
За рахунок чого потенційний товар буде захищено від копіювання: патент на промисловий зразок			

Для захисту потенційного товару від копіювання буде використовуватись нормативно-правова база, передбачена патентом на промисловий зразок.

Важливим кроком розроблення маркетингової програми є визначення цінової межі на товар (табл. 4.20). Проте як було вже сказано раніше в розділі 4.1, офіційної інформації про товари-конкуренти і товари-аналоги не має.



Таблиця 4.20 – Визначення меж встановлення ціни

Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
Невідомо	Невідомо	Середній рівень доходів і вище	0-200000 грн

Проаналізувавши межі встановлення цін на товар було визначено параметри системи збуту і занесено результати в таблицю 4.21.

Таблиця 4.21 – Формування системи збуту

Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
Не має	Постачальник повинен провести тестування товару перед збутом	1-3	Постачати програмне забезпечення одразу до розробників мобільних пристроїв та їх операційних систем

Маркетингова програма складається з кількох складових, фінальною з яких є розробка концепції маркетингових комунікацій (дивитись табл. 4.22). Ця

концепція спирається на раніше обрану основу для позиціонування та визначену специфіку поведінки клієнтів.

Таблиця 4.22 – Концепція маркетингових комунікацій

Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
Не має	Будь-які засоби зв'язку від вербального до електронного листування	Стійкість, зручність, швидкість	Показати переваги продукту над іншими конкурентами	Приклад виявлення атаки підміни обличчя за допомогою запропонованого методу

#### Висновки до розділу 4

У результаті розгляду маркетингових аспектів стартап-проекту було визначено що комерціалізація проекту є можливою. Зважаючи на групи потенційних покупців є перспективи впровадження запропонованої системи до мобільних пристроїв, проте перешкодою для цього є необхідність фінансових вкладень для доопрацювання системи, з метою підвищення точності.

В якості альтернативи впровадження було б доцільно обрати накопичення коштів власними силами. Подальша імплементація проекту є доцільною лише у випадку проведення додаткових досліджень на більшому розмірі даних.

## ВИСНОВКИ

В роботі запропоновано новий метод виявлення атак підміни, з застосуванням лицьових масок, та отримано наступні результати:

- Проведено огляд методів біометричної аутентифікації користувачів. За результатом огляду було встановлено, що системи розпізнавання обличчя є одними з найбільш доступних на мобільних пристроях і вже мають розроблену систему стандартів. Проте системи розпізнавання обличчя мають певні обмеження практичного застосування, одним з яких є їх вразливість до атак підміни.
- Для подолання даного обмеження в роботі створено та протестовано дві моделі штучної нейронної мережі, які є втіленням запропонованого методу. Необхідно зазначити відсутність спеціалізованого набору даних який містив би говорючих людей в реалістичних масках тож було використано VoxCeleb2. Тестування показало що лише модель яка була навчена з розмороженими шарами усіх частин(крім VGG-Face) набула певної здатності до узагальнення. Також слід зазначити що за результатами тестування, видно що модель штучної нейронної мережі має точність лише 0.74 та показник FAR значно вищий за FRR, що погано з точки зору забезпечення максимальної безпеки.
- По результатах порівняння запропонованої моделі штучної нейронної мережі з аналогами стало зрозуміло що запропонована система не має обмеження по точності, принаймні в такому вигляді в якому вона знаходиться зараз.

Також було виконано перший етап по розробці стартап-проекту, а саме розглянуто маркетингові аспекти його втілення.

Елемент наукової новизни запропонованого методу полягає у використанні залежності між зовнішністю людини та її голосом для виявлення невідповідностей між ними.

Практична значимість отриманих результатів полягає в розробці прототипу системи виявлення атак підміни обличчя, що здійснюються за допомогою масок. Гіпотетично, збільшення кількості людей в наборі даних, збільшення кількості шарів і параметрів, ускладнення архітектури моделі та уніфікація аудіо даних може призвести до підвищення точності. Проте для перевірки даної гіпотези головною перешкодою є обмеженість наявних обчислювальних ресурсів. Матеріали даної роботи можуть стати у нагоді для майбутніх досліджень виявлення атак підміни за допомогою штучних нейронних мереж. Запропонована система після доопрацювання може бути інтегрована в мобільні пристрої для або інші пристрої, при користуванні якими користувачі потребують автентифікації за обличчям.

За матеріалами роботи було підготовлено та опубліковано тези доповіді на тему «Метрики оцінювання систем розпізнавання обличчя» на IV Всеукраїнській Студентській Науковій Конференції «НАУКОВИЙ ПРОСТІР: АНАЛІЗ, СУЧАСНИЙ СТАН ТРЕНДИ ТА ПЕРСПЕКТИВИ»[61].

## ПЕРЕЛІК ПОСИЛАНЬ

1. Постанова Кабінету Міністрів України. Про затвердження Правил забезпечення захисту інформації в інформаційних, телекомунікаційних та інформаційно-телекомунікаційних системах [Електронний ресурс] / Постанова Каб. Міністрів України. – 2006. – Режим доступу до ресурсу: <https://www.kmu.gov.ua/npas/32791685>.
2. Goodbye, passwords. Hello, biometrics. [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: URL: <https://usa.visa.com/content/dam/VCOM/global/visa-everywhere/documents/visa-biometrics-payments-study.pdf>.
3. Daugman J. How iris recognition works // IEEE transactions on circuits and systems for video technology. – 2004. – №14. – С. 21–30.
4. Malgheet J. R. Iris Recognition Development Techniques: A Comprehensive Review / J. R. Malgheet, N. B. Manshor, L. S. Affendey // Hindawi. – 2021. – №2021. – С. 1–32.
5. Viola P. Rapid object detection using a boosted cascade of simple features / P. Viola, M. Jones // Computer Society Conference on Computer Vision and Pattern Recognition. – 2001. – С. 1–9.
6. Moindrot O. Triplet loss in TensorFlow [Електронний ресурс] / O. Moindrot – Режим доступу до ресурсу: <https://github.com/omoindrot/tensorflow-triplet-loss>.
7. Kalenichenko D. FaceNet: a unified embedding for face recognition and clustering. 2015 / D. Kalenichenko, F. Schroff, Philbin J. // CVPR – 2015..
8. Schultz C. W. Fabrication of 3D fingerprint phantoms via unconventional polycarbonate molding. Scientific reports / C. W.Schultz, J. X.Wong, H. Yu. – 2018. – №8.
9. Про затвердження плану робіт з розроблення національних стандартів, гармонізованих з міжнародними та європейськими стандартами, у

сфері підтвердження відповідності (сертифікації) промислової продукції на 2004-2011 роки [Електронний ресурс]. – 2004. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/123-2004-p#Text>.

10. H. Chen. The tiered authentication model [Електронний ресурс] / H. Chen. – 2020. – Режим доступу до ресурсу: <https://android-developers.googleblog.com/2020/09/lockscreen-and-authentication.html?m=1..>

11. Measuring Biometric Unlock Security [Електронний ресурс] – Режим доступу до ресурсу: <https://source.android.com/docs/security/features/biometric/measure>.

12. Oh T. Speech2Face: Learning the Face Behind a Voice / T. Oh, T. Dekel, C. Kim. // CVPR. – 2019.

13. M. Kamachi. Putting the face to the voice: Matching identity across modality / M. Kamachi, H. Hill. // Current Biology. – 2003. – №13. – С. 1709–1714.

14. J. Smith. Matching novel face and voice identity using static and dynamic facial images / H. M. J. Smith. // Attention, Perception, & Psychophysics. – 2016. – №78. – С. 868–879.

15. L. Tong. A Fine-grained Robustness Evaluation Framework for Face Recognition Systems / L. Tong. // CVPR. – 2021. – №2021.

16. I. Hamouchene. Aouat S. A New Texture Analysis Approach for Iris Recognition / I. Hamouchene // AASRI Procedia. – 2014. – №9. – С. 2–7.

17. E. Sejdić. Time–frequency feature representation using energy concentration: An overview of recent advances / E. Sejdić, I. Djurović, J. Jiang. // Digital Signal Processing. – 2009. – №19. – С. 153–183.

18. About Face ID advanced technology [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://support.apple.com/en-us/102381>.

19. The Business Research Company. Biometrics Global Market Report 2023. 200 p.

20. Cole S. A. Is Fingerprint Identification Valid? Rhetorics of Reliability in Fingerprint Proponents' Discourse / Cole S. A. // Law Policy. – 2006. – №28. – С. 109–135.
21. Optical Fingerprint Sensor Market, Global Outlook And Forecast 2023-2029. – 2023. – С.76.
22. Ion M. Face Recognition Algorithms [Електронний ресурс] / Ion M // Inicio - UPV/EHU. – 2010. – Режим доступа до ресурсу: <https://www.ehu.eus/ccwintco/uploads/e/eb/PFC-IonMarques.pdf>.
23. Alobed M. A Comparative Analysis of Euclidean, Jaccard and Cosine Similarity Measure and Arabic Wordnet for Automated Arabic Essay Scoring / M. Alobed, A. Altrad, Z. Bakar. // Fifth International Conference on Information Retrieval and Knowledge Management. – 2021.
24. Wang L. An Opportunistic Routing for Data Forwarding Based on Vehicle Mobility Association in Vehicular Ad Hoc Networks. Information / L. Wang, Z. Chen, J. Wu. – 2017. – №8. – С. 18.
25. Z. Meng. Active voice authentication / Z. Meng, M. Altaf. // Digital Signal Processing. – 2020. – №101. – С. 37.
26. J. Zhou. 26. Voice spoofing countermeasure for voice replay attacks using deep learning / J. Zhou. // Journal of Cloud Computing. – 2022. – №11.
27. Що таке Smart Lock та як ним користуватися [Електронний ресурс] // Samsung ua – Режим доступа до ресурсу: <https://www.samsung.com/ua/support/mobile-devices/what-is-smart-lock-and-how-do-i-use-it/>.
28. Q. Xiao. Security issues in biometric authentication / Qinghan Xiao. // Man and Cybernetics (SMC) Information Assurance Workshop. – 2005.
29. A. Jain. An Introduction to Biometric Recognition / A. Jain, A. Ross // IEEE Transactions on Circuits and Systems for Video Technology. – 2004. – №14. – С. 4–20.

30. Aleluya E. R. Faceture ID: face and hand gesture multi-factor authentication using deep learning / E. R. Aleluya, C. T. Vicente. // *Procedia Computer Science*. – 2018. – №135. – С. 147–154.
31. Gupta S. Text Dependent Voice Based Biometric Authentication System Using Spectrum Analysis and Image Acquisition / S. Gupta, S. Chatterjee. – Берлін: *Advances in Intelligent and Soft Computing*, 2018.
32. H. Zhang. PaddleSpeech: An Easy-to-Use All-in-One Speech Toolkit [Електронний ресурс] / H. Zhang // *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. – 2022. – Режим доступу до ресурсу: <https://doi.org/10.18653/v1/2022.naacl-demo.12>.
33. Schumacher D. Enhancing Suno's Bark Text-to-Speech Model: Addressing Limitations Through Meta's EnCodec and Pre-Trained Hubert [Електронний ресурс] / D. Schumacher, Jr F. LaBounty. // *Search eLibrary* – Режим доступу до ресурсу: [https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=4443815&download=yes](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4443815&download=yes).
34. A. Madry. Towards Deep Learning Models Resistant to Adversarial Attacks / A. Madry. // *Department of Electrical Engineering and Computer Science*. – 2018. – С. 15–23.
35. S. Zulkarnain. A Review on Authentication Methods / S. Zulkarnain. // *Australian Journal of Basic and Applied Sciences*. – 2013. – №7. – С. 95–107.
36. FS26 USB2.0 Fingerprint Mifare Card Reader [Електронний ресурс] // *Futronic Technology* – Режим доступу до ресурсу: [https://www.futronic-tech.com/pro-detail.php?pro\\_id=1545](https://www.futronic-tech.com/pro-detail.php?pro_id=1545).
37. BM7550 | Multi-factor Biometric Handheld Tablet - ARATEK [Електронний ресурс] // *ARATEK* – Режим доступу до ресурсу: <https://www.aratek.co/product/biometric-terminal-bm7550-tablet>.
38. BM5510 | Biometric Mobile Handheld Terminal - ARATEK [Електронний ресурс] // *ARATEK The World Leader in Biometrics Technology* –



Режим доступа до ресурсу: <https://www.aratek.co/product/biometric-terminal-bm5510>.

39. R. Singh et al. On the Robustness of Face Recognition Algorithms Against Attacks and Bias / R. Singh et al. // Proceedings of the AAAI Conference on Artificial Intelligence. – 2020. – №34. – С. 13583–13589.

40. Voice Anatomy and Physiology [Электронный ресурс] // THE VOICE FOUNDATION – Режим доступа до ресурсу: <https://voicefoundation.org/health-science/voice-disorders/anatomy-physiology-of-voice-production/>.

41. V. Ribeiro. Towards the Prediction of the Vocal Tract Shape from the Sequence of Phonemes to be Articulated [Электронный ресурс] / V. Ribeiro // Interspeech 2021 – Режим доступа до ресурсу: <https://doi.org/10.21437/interspeech.2021-184>.

42. V. Ribeiro. Towards the Prediction of the Vocal Tract Shape from the Sequence of Phonemes to be Articulated [Электронный ресурс] / V. Ribeiro // Interspeech 2021 – Режим доступа до ресурсу: <https://doi.org/10.21437/interspeech.2021-184>.

43. V. Ribeiro. Towards the Prediction of the Vocal Tract Shape from the Sequence of Phonemes to be Articulated [Электронный ресурс] / V. Ribeiro // Interspeech 2021 – Режим доступа до ресурсу: <https://doi.org/10.21437/interspeech.2021-184>.

44. InsightFace Python Library [Электронный ресурс] // GitHub – Режим доступа до ресурсу: <https://github.com/deepinsight/insightface/tree/master/python-package>.

45. J. Guo. Sample and Computation Redistribution for Efficient Face Detection / J. Guo. // Imperial College. – 2021. – С. 10.

46. Parkhi O. VGG Face Descriptor [Электронный ресурс] / O. Parkhi, A. Vedaldi, A. Zisserman // Information Engineering – Режим доступа до ресурсу: [https://www.robots.ox.ac.uk/~vgg/software/vgg\\_face/](https://www.robots.ox.ac.uk/~vgg/software/vgg_face/).

47. P. Kingma D. Adam: a method for stochastic optimization / P. Kingma D, Lei Ba J // ICLR. – 2015..
48. Steiner H. Reliable face anti-spoofing using multispectral SWIR imaging / Steiner H, Jung N, Kolb A. // International Conference on Biometrics. – 2016.
49. Raghavendra R. Novel presentation attack detection algorithm for face recognition system: Application to 3D face mask attack [Электронный ресурс] / R. Raghavendra, C. Busch // International Conference on Image Processing. – 2014. – Режим доступа до ресурсу: <https://doi.org/10.1109/icip.2014.7025064..>
50. Hamdan B. The detection of spoofing by 3D mask in a 2D identity recognition system / Hamdan B. // Egyptian Informatics Journal. – 2018. – С. 75–82.
51. Kowalski M. A Study on Presentation Attack Detection in Thermal Infrared / Kowalski M. // Institute of Optoelectronics, Military University of Technology. – 2020. – №14. – С. 18.
52. Parkhi O. M. Deep Face Recognition [Электронный ресурс] / O. Parkhi, A. Vedaldi, A. Zisserman // British Machine Vision Conference. – 2015. – Режим доступа до ресурсу: <https://doi.org/10.5244/c.29.41>.
53. Spectral reflectance of human skin in the region 0.7-2.6m / J. Jacquez, J. Huss, W. McKeehan та ін.]. // Applied Physiology. – 1955. – №8. – С. 297.
54. Chung J. S. VoxCeleb2: Deep Speaker Recognition. Interspeech 2018 [Электронный ресурс] / J. S. Chung, A. Nagrani, A. Zisserman // ISCA. – 2018. – Режим доступа до ресурсу: <https://doi.org/10.21437/interspeech.2018-1929>.
55. Nayar G. Partial palm vein based biometric authentication [Электронный ресурс] / G. Nayar, T. Thomas // Journal of Information Security and Applications. – 2023. – Режим доступа до ресурсу: <https://doi.org/10.1016/j.jisa.2022.103390>.
56. B. Mazumdar. Retina based biometric authentication system: a review [Электронный ресурс] / B. Mazumdar // International Journal of Advanced Research in Computer Science. – 2018. – Режим доступа до ресурсу: <https://doi.org/10.26483/ijarcs.v9i1.5322..>

57. Fahad E. FingerPrint Scanner or Biometric Scanner and their types [Електронний ресурс] / Fahad E. // Electronic Clinic – Режим доступу до ресурсу: <https://www.electronicclinic.com/fingerprint-scanner-or-biometric-scanner-and-their-types/>.
58. Khandelwal C. Review Paper on Applications of Principal Component Analysis in Multimodal Biometrics System / C. Khandelwal, R. Maheshwari // Procedia Computer Science. – 2021. – С. 482–487.
59. Peng S. More trainable inception-ResNet for face recognition / S. Peng, H. Huang // Neurocomputing. – 2020. – №411. – С. 9–19.
60. Multi-factor Authentication Market Size, Share and Trends Analysis Report By Model [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://www.grandviewresearch.com/industry-analysis/multi-factor-authentication-market>.
61. Журавльов О. Метрики оцінювання систем розпізнавання облич / Журавльов О. // Всеукраїнська студентська наукова конференція «Науковий простір: аналіз, сучасний стан, тренди та перспективи». – 2023. – №4. – С. 411–414.

## ДОДАТОК А

### ПРОГРАМНА РЕАЛІЗАЦІЯ ОБРІЗАННЯ ВІДЕО

```
1.  import glob
2.  import numpy as np
3.  from moviepy.editor import VideoFileClip, concatenate_videoclips
4.  import os
5.  import cv2
6.  import datetime
7.  from pathlib import Path
8.
9.
10. videos_names=sorted(glob.glob(f'./mp4/**/*.mp4'))
11. videos_names=np.array(videos_names)
12. len(videos_names)
13.
14. ytid=sorted(glob.glob(f'./yt/**/*.mp4'))
15. ytid=np.array(ytid)
16. len(ytid)
17.
18. for i in range(len(ytid)):
19.     path = ytid[i].split(sep='/')
20.     Path(f'./ready/video/{path[2]}/{path[3]}').mkdir(parents=True,
exist_ok=True)
21.     Path(f'./ready/audio/{path[2]}/{path[3]}').mkdir(parents=True,
exist_ok=True)
22.
23. target_duration=3
24. counter=0
```

```
25.
26.     for name in videos_names:
27.         try:
28.             data = cv2.VideoCapture(name)
29.             frames = data.get(cv2.CAP_PROP_FRAME_COUNT)
30.             fps = data.get(cv2.CAP_PROP_FPS)
31.             seconds = frames / fps
32.             if seconds < target_duration: continue
33.
34.             video = VideoFileClip(name)
35.             current_duration = seconds
36.             print(current_duration)
37.
38.             if current_duration > target_duration:
39.
40.                 video = video.subclip(0, target_duration)
41.
42.             elif current_duration == 3.00: pass
43.
44.             print(video.duration)
45.
46.             path = name.split(sep='/')
47.
48.             output_path = os.path.basename(name).split(sep='.')[0]
49.             video.write_videofile(f"./ready/video/{path[2]}/{path[3]}/
{output_path}.mp4")
50.
51.
52.     #AUDIO
```

```
53.  
54.     audio = video.audio  
55.     audio.write_audiofile(f"./ready/audio/{path[2]}/{path[3]}/  
{output_path}.mp3")  
56.  
57.     os.remove(name)  
58.     except:pass
```

**ДОДАТОК Б**  
**ПРОГРАМНА РЕАЛІЗАЦІЯ ВИОКРЕМЛЕННЯ ЗОБРАЖЕННЯ**  
**ОБЛИЧЧЯ З ВІДЕО**

```
1.  import numpy as np
2.  import os
3.  import glob
4.  import cv2
5.  import matplotlib.pyplot as plt
6.  import PIL
7.  import insightface
8.  from insightface.app import FaceAnalysis
9.  from insightface.data import get_image as ins_get_image
10. from pathlib import Path
11. import pandas as pd
12. import pickle
13. import imageio
14.
15. def delete_files_in_directory(directory):
16.     for filename in os.listdir(directory):
17.         file_path = os.path.join(directory, filename)
18.         if os.path.isfile(file_path):
19.             os.remove(file_path)
20.
21. def extract_frames(video_path, output_folder):
22.     cap = cv2.VideoCapture(video_path)
23.     frame_rate = cap.get(cv2.CAP_PROP_FPS)
24.     frame_count = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))
25.
```

```

26.     for i in range(0, 3):
27.         frame_index = int(i * frame_rate)
28.         cap.set(cv2.CAP_PROP_POS_FRAMES, frame_index)
29.         success, frame = cap.read()
30.
31.         if success:
32.             frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
33.             frame_path = f"{output_folder}/frame_{i}.jpg"
34.             imageio.imwrite(frame_path, frame_rgb)
35.         cap.release()
36.
37.
38.     output_folder = "./extracted frames"
39.
40.     def create_directories():
41.         ytid=sorted(glob.glob(f'./ready/video/*/'))
42.         ytid=np.array(ytid)
43.         for i in range(len(ytid)):
44.             path = ytid[i].split(sep='/')
45.             Path(f'./faces/{path[3]}/{path[4]}').mkdir(parents=True,
exist_ok=True)
46.
47.     create_directories()
48.
49.     video_names=sorted(glob.glob(f'./ready/video/*/'))
50.     len(video_names)
51.
52.     for name in video_names:
53.         try:

```



```

54.
55.     extract_frames(name, output_folder)
56.
57.     for j in range(3):
58.         try:
59.             frame_path= f'./extracted frames/frame_{j}.jpg'
60.             source=cv2.imread(frame_path)
61.             path = name.split(sep='/')
62.
63.             output_path = os.path.basename(name).split(sep='.')[0]
64.
cv2.imwrite(f'./faces/{path[3]}/{path[4]}/{output_path}.png',source[:160,50:180,])
65.         break
66.
67.     except Exception as error:
68.         print("An error occurred:", error)
69.         delete_files_in_directory('./extracted frames/')
70.
71.
72.     except Exception as error:
73.         print("An error occurred:", error)

```

## ДОДАТОК В

### ПРОГРАМНА РЕАЛІЗАЦІЯ СТВОРЕННЯ СПЕКТРОГРАМ

```
1.  import librosa
2.  import pandas as pd
3.  import numpy as np
4.  import matplotlib.pyplot as plt
5.  import pickle
6.  import os
7.  import tensorflow as tf
8.
9.  import glob
10. from pathlib import Path
11.
12. audio_pathes=sorted(glob.glob(f'./ready/audio/*/.*'))
13. len (audio_pathes)
14.
15. embeddings_pathes=sorted(glob.glob( f'./vggface embeddings/*/.*'))
16. len (embeddings_pathes)
17.
18. def power_law_compression(data):
19.     return np.sign(data)*np.power(np.abs(data),0.3)
20.
21. def create_directories():
22.     ytid=sorted(glob.glob(f'./ready/audio/*/.*'))
23.     ytid=np.array(ytid)
24.     for i in range(len(ytid)):
25.         path = ytid[i].split(sep='/')
```

```

26.      Path(f'./spectrograms/{path[3]}/{path[4]}').mkdir(parents=True,
exist_ok=True)
27.
28.  create_directories()
29.
30.  for name in audio_paths:
31.      try:
32.          ytID=os.path.basename(name).split(sep='.')[0]
33.          path=name.split(sep='/')
34.          if f'./vggface embeddings/{path[3]}/{path[4]}/{ytID}.pickle' in
embeddings_paths:
35.
36.              audio, sr = librosa.load(name, duration = 3 ,mono =
True,sr=16000)
37.
38.              stft = librosa.stft(audio,n_fft = 512, hop_length = 160,win_length
= 400)
39.
40.              stft_normalized = librosa.util.normalize(stft)
41.              magnitude = np.abs(stft_normalized)
42.              phase = np.angle(stft)
43.
44.              magnitude=power_law_compression(magnitude)
45.              phase=power_law_compression(phase)
46.              stft= tf.stack([magnitude,phase],2)
47.
48.              print(stft.shape)
49.
50.              pickle_file = f'./spectrograms/{path[3]}/{path[4]}/{ytID}.pickle'

```

```
51.  
52.  
53.         with open(pickle_file, 'wb') as f:  
54.             pickle.dump(stft, f)  
55.  
56.         f.close()  
57.  
58.  
59.         print(f'Spectrogram for {ytID} was sucesfully created ')  
60.         else:pass  
61.         except Exception as error:  
62.             print("An error occurred:", error)
```

## ДОДАТОК Г

### ПРОГРАМНА РЕАЛІЗАЦІЯ VGG-FACE

```

1.  import tensorflow as tf
2.  import glob
3.  import itertools
4.  import pandas as pd
5.  import os
6.  from tensorflow.keras.models import Model, Sequential
7.  from tensorflow.keras.layers import Input, Conv2D, ZeroPadding2D,
MaxPool2D, Flatten, Dense, Dropout, Activation
8.  from PIL import Image
9.  import numpy as np
10. from numpy.linalg import norm
11. import pickle
12. from pathlib import Path
13. from tensorflow.keras.utils import load_img, save_img, img_to_array
14. from tensorflow.keras.applications.imagenet_utils import
preprocess_input
15. from tensorflow.keras.preprocessing import image
16. import matplotlib.pyplot as plt
17.
18. def preprocess_image(image_path):
19.     img=load_img(image_path, target_size=(224,224))
20.     img=img_to_array(img)
21.     img=np.expand_dims(img,axis=0)
22.     ing=preprocess_input(img)
23.     return img
24.

```

```

25.  def findEuclideanDistance(source_representation, test_representation):
26.      euclidean_distance = source_representation - test_representation
27.      euclidean_distance = np.sum(np.multiply(euclidean_distance,
euclidean_distance))
28.      euclidean_distance = np.sqrt(euclidean_distance)
29.      return euclidean_distance
30.
31.  def findCosineSimilarity(A, B):
32.      A=A[0][0][0]
33.      B=B[0][0][0]
34.      return np.dot(A,B)/(norm(A)*norm(B))
35.
36.  model = Sequential()
37.  model.add(ZeroPadding2D((1,1),input_shape=(224,224, 3)))
38.  model.add(Conv2D(64, (3, 3), activation='relu'))
39.  model.add(ZeroPadding2D((1,1)))
40.  model.add(Conv2D(64, (3, 3), activation='relu'))
41.  model.add(MaxPool2D((2,2), strides=(2,2)))
42.
43.  model.add(ZeroPadding2D((1,1)))
44.  model.add(Conv2D(128, (3, 3), activation='relu'))
45.  model.add(ZeroPadding2D((1,1)))
46.  model.add(Conv2D(128, (3, 3), activation='relu'))
47.  model.add(MaxPool2D((2,2), strides=(2,2)))
48.
49.  model.add(ZeroPadding2D((1,1)))
50.  model.add(Conv2D(256, (3, 3), activation='relu'))
51.  model.add(ZeroPadding2D((1,1)))
52.  model.add(Conv2D(256, (3, 3), activation='relu'))

```

```
53. model.add(ZeroPadding2D((1,1)))
54. model.add(Conv2D(256, (3, 3), activation='relu'))
55. model.add(MaxPool2D((2,2), strides=(2,2)))
56.
57. model.add(ZeroPadding2D((1,1)))
58. model.add(Conv2D(512, (3, 3), activation='relu'))
59. model.add(ZeroPadding2D((1,1)))
60. model.add(Conv2D(512, (3, 3), activation='relu'))
61. model.add(ZeroPadding2D((1,1)))
62. model.add(Conv2D(512, (3, 3), activation='relu'))
63. model.add(MaxPool2D((2,2), strides=(2,2)))
64.
65. model.add(ZeroPadding2D((1,1)))
66. model.add(Conv2D(512, (3, 3), activation='relu'))
67. model.add(ZeroPadding2D((1,1)))
68. model.add(Conv2D(512, (3, 3), activation='relu'))
69. model.add(ZeroPadding2D((1,1)))
70. model.add(Conv2D(512, (3, 3), activation='relu'))
71. model.add(MaxPool2D((2,2), strides=(2,2)))
72.
73. model.add(Conv2D(4096, (7, 7), activation='relu'))
74. model.add(Dropout(0.5))
75. model.add(Conv2D(4096, (1, 1), activation='relu'))
76. model.add(Dropout(0.5))
77. model.add(Conv2D(2622, (1, 1)))
78. model.add(Flatten())
79. model.add(Activation('softmax'))
80.
81. from keras.models import model_from_json
```

```

82.  model.load_weights('vgg_face_weights.h5')
83.
84.  vgg_face_descriptor = Model(inputs=model.layers[0].input,
outputs=model.layers[-5].output)
85.
86.  def create_directories():
87.      ytid=sorted(glob.glob(f'./ready/audio/*/'))
88.      ytid=np.array(ytid)
89.      for i in range(len(ytid)):
90.          path = ytid[i].split(sep='/')
91.          Path(f'./vggface
embeddings/{path[3]}/{path[4]}').mkdir(parents=True, exist_ok=True)
92.
93.  create_directories()
94.
95.  face_pathes=sorted(glob.glob('./faces/*/'))
96.  len(face_pathes)
97.
98.  img1_representation =
vgg_face_descriptor.predict(preprocess_image(face_pathes[0]))
99.  img2_representation =
vgg_face_descriptor.predict(preprocess_image(face_pathes[10]))
100. print(findEuclideanDistance(img1_representation,img2_representation))
101.
102. for face_path in face_pathes:
103.     output_path = os.path.basename(face_path).split(sep='.')[0]
104.     path = face_path.split(sep='/')
105.     pickle_file = f'./vggface
embeddings/{path[2]}/{path[3]}/{output_path}.pickle'

```



```
106.  
107.     with open(pickle_file, 'wb') as f:  
108.  
pickle.dump(vgg_face_descriptor.predict(preprocess_image(face_path)), f)  
109.     f.close()
```

# ДОДАТОК Г

## ПРОГРАМНА РЕАЛІЗАЦІЯ НАЛАШТУВАННЯ МУЛЬТИМОДАЛЬНОЇ ЧАСТИНИ

```

1.  import numpy as np
2.  import pandas as pd
3.  import cv2
4.  from keras.callbacks import LearningRateScheduler
5.  import os
6.  import pickle
7.  import tensorflow as tf
8.  import glob
9.  import matplotlib.pyplot as plt
10. from keras.layers import Dense, Flatten, Input, Convolution2D,
Dropout, Activation, BatchNormalization, Conv2D,
MaxPooling2D, AveragePooling2D, ReLU
11. from keras.models import Sequential, Model
12. from keras.regularizers import l2
13. from keras.models import load_model
14. from tensorflow.keras.utils import Sequence
15. from keras.optimizers import Adam
16. from keras import losses
17.
18. y_names_train=sorted(glob.glob(f'./vggface embeddings/*/*/*'))
19. y_names_train=np.array(y_names_train)
20. len(y_names_train)
21.
22. x_names_train=sorted(glob.glob('./spectrograms/*/*/*'))
23. x_names_train=np.array(x_names_train)

```

```

24. len(x_names_train)
25.
26. for name in y_names_train:
27.     if os.path.getsize(name)== 0:
28.         print(name)
29.
30. for name in x_names_train:
31.     if os.path.getsize(name)== 0:
32.         print(name)
33.
34. def total_loss(y_true, y_pred):
35.     first_part=tf.norm(tf.math.l2_normalize(y_true)-
tf.math.l2_normalize(y_pred),ord=2)**2
36.
37.
38.     a=tf.exp(y_true/tf.constant(2.0, dtype=tf.float32)) /
tf.reduce_sum(tf.exp(y_true/tf.constant(2.0, dtype=tf.float32)), keepdims=True)
39.     b=tf.exp(y_pred/tf.constant(2.0, dtype=tf.float32)) /
tf.reduce_sum(tf.exp(y_pred/tf.constant(2.0, dtype=tf.float32)), keepdims=True)
40.
41.     Ldistill=tf.keras.losses.CategoricalCrossentropy()(a,b)
42.     return tf.add(first_part,Ldistill)
43.
44. class DataGenerator_with_load(Sequence):
45.     def __init__(self, x_set, y_set, batch_size):
46.         self.x, self.y = x_set, y_set
47.         self.batch_size = batch_size
48.
49.         self.indices = np.arange(self.x.shape[0])

```

```

50.
51.     def __len__(self):
52.         return int(np.ceil(len(self.x) / float(self.batch_size)))
53.
54.     def __getitem__(self, idx):
55.         inds = self.indices[idx * self.batch_size:(idx + 1) * self.batch_size]
56.         embeddings=[]
57.         for y in self.y[inds]:#idx * self.batch_size:(idx + 1) *
self.batch_size]:
58.             with open(y,'rb') as f:
59.                 embedding = pickle.load(f)
60.
61.                 embeddings.append(embedding[0][0][0])
62.             f.close()
63.
64.         embeddings=np.array(embeddings)
65.
66.         spectrograms=[]
67.         for x in self.x[inds]:
68.             with open(x,'rb') as f:
69.                 spectrograms.append(pickle.load(f))
70.             f.close()
71.         spectrograms=np.array(spectrograms)
72.
73.         batch_x = spectrograms
74.         batch_y = embeddings
75.
76.
77.         return batch_x, batch_y

```

```
78.
79.     def on_epoch_end(self):
80.         np.random.shuffle(self.indices)
81.
82.
83.     train_gen = DataGenerator_with_load(x_names_train, y_names_train, 3)
84.
85.     model = Sequential()
86.     model.add(Conv2D(32, kernel_size = 4, strides=1 ,input_shape = (301,
257,1)))
87.     model.add(ReLU())
88.     model.add(BatchNormalization())
89.
90.     model.add(Conv2D(32, kernel_size = 4, strides=1,padding='same'))
91.     model.add(ReLU())
92.     model.add(BatchNormalization())
93.
94.     model.add(Conv2D(64, kernel_size = 4, strides=1,padding='same'))
95.     model.add(ReLU())
96.     model.add(BatchNormalization())
97.
98.     model.add(MaxPooling2D(pool_size = (2,1),strides = (2,1)))
99.
100.    model.add(Conv2D(64, kernel_size = 4, strides=1,padding='same'))
101.    model.add(ReLU())
102.    model.add(BatchNormalization())
103.
104.    model.add(MaxPooling2D(pool_size = (2,1),strides = (2,1)))
105.
```

```
106. model.add(Conv2D(64, kernel_size = 3, strides=1,padding='same'))
107. model.add(ReLU())
108. model.add(BatchNormalization())
109.
110. model.add(MaxPooling2D(pool_size = (2,1),strides = (2,1)))
111.
112. model.add(Conv2D(128, kernel_size = 4, strides=1,padding='same'))
113. model.add(ReLU())
114. model.add(BatchNormalization())
115.
116. model.add(MaxPooling2D(pool_size = (2,1),strides = (2,1)))
117.
118. model.add(Conv2D(256, kernel_size = 4, strides=1,padding='same'))
119. model.add(ReLU())
120. model.add(BatchNormalization())
121.
122. model.add(MaxPooling2D(pool_size = (2,1),strides = (2,1)))
123.
124. model.add(Conv2D(256, kernel_size = 4, strides=2,padding='same'))
125. model.add(ReLU())
126. model.add(BatchNormalization())
127.
128. model.add(Conv2D(256, kernel_size = 4, strides=2))
129.
130. model.add(AveragePooling2D(pool_size=(2,62),
strides=1,padding='valid'))
131. model.add(ReLU())
132. model.add(BatchNormalization())
133. model.add(Flatten())
```

```
134.  
135.     bias_regularizer=l2(0.01)))  
136. model.add(Dense(4096,kernel_regularizer=l2(0.01) ))  
137.  
138.  
139. adam=tf.keras.optimizers.Adam(learning_rate=0.001,beta_1=0.5,epsilon  
=1e-04)#learning_rate = 1e-3)  
140.  
141. model.compile(optimizer=adam, loss=total_loss, metrics=[cosin_sim])  
142.  
143. model.summary()  
144.  
145. epochs = 5  
146. sv=tf.keras.callbacks.ModelCheckpoint(  
147.     filepath='./tmp/checkpoint',  
148.     save_weights_only=True,  
149.     monitor='val_loss',  
150.     mode='min',  
151.     verbose=1,  
152.     save_best_only=True)  
153.  
154. initial_history=model.fit(train_gen,  
155.                             epochs=epochs,  
156.                             callbacks=[sv],  
157.                             validation_data=test_gen,  
158.                             verbose=1)  
159.  
160. model.save(f'mm')
```

## ДОДАТОК Д

### ПРОГРАМНА РЕАЛІЗАЦІЯ НАЛАШТУВАННЯ МОДЕЛІ БІНАРНОЇ КЛАСИФІКАЦІЇ

```

1.  import numpy as np
2.  import pandas as pd
3.  import cv2
4.  import random
5.  from pathlib import Path
6.  from keras.callbacks import LearningRateScheduler
7.  import os
8.  import shutil
9.  import seaborn as sn
10. import pickle
11. import itertools
12. import tensorflow as tf
13. from keras.models import load_model
14. import glob
15. import matplotlib.pyplot as plt
16. from keras.layers import Dense, Flatten, Input, ZeroPadding2D, Conv1D,
Convolution2D, Dropout, Activation, BatchNormalization, Conv2D, MaxPool2D,
MaxPooling2D, AveragePooling2D, ReLU
17. from keras.models import Sequential, Model
18. from keras.regularizers import l2
19. from keras.models import load_model
20. from tensorflow.keras.utils import Sequence
21.
22.
23.  directory_pathes=sorted(glob.glob(f'./predicted_embeddings/*/'))

```



```
24.
25.     for directory_path in directory_pathes:
26.
27.         path=directory_path.split(sep='/')
28.         src=directory_path
29.         dst= f'./classification_data/all_mymodel/{path[3]}'
30.
31.         shutil.copytree(src, dst)
32.
33.         files_in_dst=sorted(glob.glob(f'{dst}/*'))
34.
35.         for file_name in files_in_dst[:10]:
36.             os.remove(file_name)
37.
38.     directory_pathes=sorted(glob.glob(f'../VGGFace/vggface
embeddings/*/*'))
39.
40.     person_pathes=sorted(glob.glob(f'./test vggface embeddings/*'))
41.
42.     all_first_vggface_embeddings=[]
43.
44.     for person in person_pathes:
45.         videos_pathes=glob.glob(f'{person}/*')
46.
47.         for video in videos_pathes:
48.             try:
49.                 first_vggface_embeddings=glob.glob(f'{video}/*')[0]
50.
```

```

51.
all_first_vggface_embeddings.append(first_vggface_embeddings)
52.     except:pass
53.
54.     len(all_first_vggface_embeddings)
55.
56.     vgg_emb_names=sorted(all_first_vggface_embeddings)
57.     vgg_emb_names_test=np.array(vgg_emb_names)
58.     len(vgg_emb_names_test)
59.
60.
61.     for directory_path in directory_pathes:
62.
63.         path=directory_path.split(sep='/')
64.         src=directory_path
65.         dst= f'./classification_data/all_vggface/{path[4]}'
66.
67.         shutil.copytree(src, dst)
68.
69.         files_in_dst=sorted(glob.glob(f'{dst}/*'))
70.
71.         for file_name in files_in_dst[:10]:
72.             os.remove(file_name)
73.
74.     person_pathes=sorted(glob.glob(f'./vggface embeddings/*'))
75.
76.     all_first_vggface_embeddings=[]
77.
78.     for person in person_pathes:

```

```

79.     videos_paths=glob.glob(f'{person}/*')
80.
81.     for video in videos_paths:
82.         try:
83.             first_vggface_embeddings=glob.glob(f'{video}/*')[0]
84.
85.
all_first_vggface_embeddings.append(first_vggface_embeddings)
86.         except:pass
87.
88.     len(all_first_vggface_embeddings)
89.
90.     vgg_emb_names=sorted(all_first_vggface_embeddings)
91.     vgg_emb_names_train=np.array(vgg_emb_names)
92.     len(vgg_emb_names_train)
93.
94.
95.     multimodal_emb_names=sorted(glob.glob(f'./classification_data/all_my
model_train/*/*/*))
96.     multimodal_emb_names_train=np.array(multimodal_emb_names)
97.     len(multimodal_emb_names_train)
98.
99.     class DataGenerator(Sequence):
100.         def __init__(self, x_set, y_set, batch_size):
101.             self.x, self.y = x_set, y_set
102.             self.batch_size = batch_size
103.             self.combinations=itertools.product(self.x,self.y)
104.             self.indices = np.arange(len(self.x))
105.

```

```

106.     def __len__(self):
107.         return int((np.ceil(len(self.x) / float(self.batch_size))))
108.
109.     def __getitem__(self, idx):
110.
111.         batch_y=[]
112.         batch_x=[]
113.
114.         for i in range(self.batch_size):
115.
116.             pair=next(self.combinations)
117.             path1=pair[0].split(sep='/')
118.             path2=pair[1].split(sep='/')
119.
120.             if path1[2]==path2[3]:
121.
122.                 batch_y.append(1)
123.
124.                 with open(pair[0],'rb') as f:
125.                     vgg_emb = pickle.load(f)
126.                     vgg_emb=vgg_emb[0][0][0]
127.                 f.close()
128.
129.                 with open(pair[1],'rb') as f:
130.                     pred_emb = pickle.load(f)
131.                     pred_emb= pred_emb
132.                 f.close()
133.                 sample=tf.concat([vgg_emb, pred_emb],0)
134.                 batch_x.append(sample.numpy())

```

```

135.
136.         else:
137.             batch_y.append(0)
138.             with open(pair[0], 'rb') as f:
139.                 vgg_emb = pickle.load(f)
140.                 vgg_emb = vgg_emb[0][0][0]
141.             f.close()
142.
143.             with open(pair[1], 'rb') as f:
144.                 pred_emb = pickle.load(f)
145.                 pred_emb = pred_emb
146.             f.close()
147.
148.             sample = tf.concat([vgg_emb, pred_emb], 0)
149.             batch_x.append(sample.numpy())
150.
151.         batch_x = np.array(batch_x)
152.         batch_y = np.array(batch_y)
153.         return batch_x, batch_y
154.
155.     def on_epoch_end(self):
156.         np.random.shuffle(self.indices)
157.
158.     train_gen =
DataGenerator(vgg_emb_names_train, multimodal_emb_names_train, 32)
159.     test_gen =
DataGenerator(vgg_emb_names_test, multimodal_emb_names_test, 32)
160.
161.     model = Sequential()

```

```
162.  
163. model.add(Dense(128,input_dim=4096*2, activation='relu'))  
164. model.add(Dense(16, activation='relu'))  
165. model.add(Dense(1, activation='sigmoid'))  
166.  
167. adam=tf.keras.optimizers.Adam(learning_rate=0.001)  
168. model.compile(optimizer=adam, loss='binary_crossentropy',  
metrics='accuracy')  
169.  
170. model.build()  
171. model.summary()  
172.  
173. model.fit(train_gen,validation_data=test_gen,epochs=5)
```

## ДОДАТОК Е

### ПРОГРАМНА РЕАЛІЗАЦІЯ ОДНОЧАСНОГО НАЛАШТУВАННЯ МУЛЬТИМОДАЛЬНОЇ ЧАСТИНИ ТА ЧАСТИНИ ЩО ВИКОНУЄ БІНАРНУ КЛАСИФІКАЦІЮ

```

1.  import numpy as np
2.  import pandas as pd
3.  import cv2
4.  import random
5.  from pathlib import Path
6.  from keras.callbacks import LearningRateScheduler
7.  import os
8.  import shutil
9.  import seaborn as sn
10. import pickle
11. import itertools
12. import tensorflow as tf
13. from keras.models import load_model
14. import glob
15. import matplotlib.pyplot as plt
16. from keras.layers import Dense, concatenate,
    Flatten, Input, ZeroPadding2D, Conv1D, Convolution2D, Dropout, Activation,
    BatchNormalization, Conv2D, MaxPool2D,
    MaxPooling2D, AveragePooling2D, ReLU
17. from keras.models import Sequential, Model
18. from keras.regularizers import l2
19. from keras.models import load_model
20. from tensorflow.keras.utils import Sequence
21.

```

```

22.  emb_names_train=sorted(glob.glob(f'./vggface embeddings/**/*.*/**/*'))
23.  emb_names_train=np.array(emb_names_train)
24.  len(emb_names_train)
25.
26.  spec_names_train=sorted(glob.glob('./spectrograms/**/*.*/**/*'))
27.  spec_names_train=np.array(spec_names_train)
28.  len(spec_names_train)
29.
30.  zipped = list(zip(spec_names_train,emb_names_train))
31.  random.shuffle(zipped)
32.  spec_names_train, emb_names_train = zip(*zipped)
33.  del zipped
34.
35.  spec_names_train=np.array(spec_names_train)
36.
37.  emb_names_train=np.array(emb_names_train)
38.
39.
40.  class DataGenerator_with_load(Sequence):
41.      def __init__(self, x_set, y_set, batch_size):
42.
43.          self.x, self.y = x_set, y_set
44.          self.combinations=itertools.product(self.x,self.y)
45.          self.batch_size = batch_size
46.          self.pconter=0
47.          self.nconter=0
48.
49.          self.indices = np.arange(self.x.shape[0])
50.

```



```

51.     def __len__(self):
52.         return int(np.ceil(len(self.x) / float(self.batch_size)))
53.
54.     def __getitem__(self, idx):
55.
56.         spectrograms=[]
57.         vgg_embeddings=[]
58.         labels=[]
59.
60.         while len(labels)<self.batch_size:
61.             pair=next(self.combinations)
62.             spectrogram_path=pair[0].split(sep='/')
63.             vgg_emb_path=pair[1].split(sep='/')
64.
65.             if spectrogram_path[2]==vgg_emb_path[2]:
66.
67.
68.                 labels.append(1)
69.
70.                 with open(pair[0],'rb') as f:
71.                     vgg_embeddings.append(pickle.load(f)[0][0][0])
72.                 f.close()
73.
74.                 with open(pair[1],'rb') as f:
75.                     spectrograms.append(pickle.load(f))
76.                 f.close()
77.             else:
78.                 if labels.count(1)<labels.count(0):continue
79.                 labels.append(0)

```

```

80.         with open(pair[0], 'rb') as f:
81.             vgg_embeddings.append(pickle.load(f)[0][0][0])
82.         f.close()
83.
84.         with open(pair[1], 'rb') as f:
85.             spectrograms.append(pickle.load(f))
86.         f.close()
87.
88.
89.         vgg_embeddings=np.array(vgg_embeddings)
90.         spectrograms=np.array(spectrograms)
91.         labels=np.array(labels)
92.
93.         batch_x = [spectrograms, vgg_embeddings]
94.         batch_y = labels
95.
96.         return batch_x, batch_y
97.
98.     def on_epoch_end(self):
99.         zipped = list(zip(self.x, self.y))
100.        random.shuffle(zipped)
101.        self.x, self.y = zip(*zipped)
102.        self.combinations=itertools.product(self.x,self.y)
103.
104.
105.    train_gen = DataGenerator_with_load(emb_names_train,
spec_names_train, 8)
106.    test_gen = DataGenerator_with_load(emb_names_test, spec_names_test,
8)

```

```
107.
108. train_gen[0]
109.
110. #MY MODEL
111.
112. input_spec = Input(shape=(301, 257,1))
113. x = Conv2D(32, kernel_size = 4, padding='same')(input_spec)
114. x = ReLU()(x)
115. x = BatchNormalization()(x)
116.
117.
118. x = Conv2D(32, kernel_size = 4, strides=1, padding='same')(x)
119. x = ReLU()(x)
120. x = BatchNormalization()(x)
121.
122.
123. x = Conv2D(64, kernel_size = 4, strides=1, padding='same')(x)
124. x = ReLU()(x)
125. x = BatchNormalization()(x)
126.
127. x = MaxPooling2D(pool_size = (2,1),strides = (2,1))(x)
128.
129.
130. x = Conv2D(64, kernel_size = 4, strides=1, padding='same')(x)
131. x = ReLU()(x)
132. x = BatchNormalization()(x)
133.
134. x = MaxPooling2D(pool_size = (2,1),strides = (2,1))(x)
135.
```

```
136.
137. x = Conv2D(64, kernel_size = 3, strides=1, padding='same')(x)
138. x = ReLU()(x)
139. x = BatchNormalization()(x)
140.
141. x = MaxPooling2D(pool_size = (2,1),strides = (2,1))(x)
142.
143.
144. x = Conv2D(128, kernel_size = 4, strides=1, padding='same')(x)
145. x = ReLU()(x)
146. x = BatchNormalization()(x)
147.
148. x = MaxPooling2D(pool_size = (2,1),strides = (2,1))(x)
149.
150.
151. x = Conv2D(256, kernel_size = 4, strides=1, padding='same')(x)
152. x = ReLU()(x)
153. x = BatchNormalization()(x)
154.
155. x = MaxPooling2D(pool_size = (2,1),strides = (2,1))(x)
156.
157.
158. x = Conv2D(256, kernel_size = 4, strides=2, padding='same')(x)
159. x = ReLU()(x)
160. x = BatchNormalization()(x)
161.
162.
163. x = Conv2D(256, kernel_size = 4, strides=2, padding='same')(x)
164.
```

```

165.
166. x = MaxPooling2D(pool_size = (3,65),strides = 1)(x)
167. x = ReLU()(x)
168.
169. x = Flatten()(x)
170.
171.
172. x = Dense(4096,kernel_regularizer=l2(0.01),
bias_regularizer=l2(0.01))(x)
173.
174. x=Model(inputs=input_spec, outputs=x)
175.
176.
177.
178. # VGGFace stream
179. input_emb=Input(shape=(4096,))
180. y = Model(inputs=input_emb,outputs=input_emb)
181.
182. #Clasification
183. combined= concatenate([x.output,y.output])
184. z = Dense(512,input_dim=4096*2, activation='relu')(combined)
185. z = Dense(256, activation='relu')(z)
186. z = Dense(32, activation='relu')(z)
187. z = Dense(1, activation='sigmoid')(z)
188.
189.
190. model= Model(inputs=[x.input,y.input], outputs=z)
191. model.summary()
192. adam=tf.keras.optimizers.Adam(learning_rate=0.001)

```

```
193. model.compile(optimizer=adam, loss='binary_crossentropy',
metrics='accuracy')
194.
195. sv=tf.keras.callbacks.ModelCheckpoint(
196.     filepath='./emb_classification_with_mm2',
197.     save_weights_only=True,
198.     verbose=1,
199.     save_best_only=True)
200.
201. model.fit(train_gen,validation_data=test_gen,epochs=5, callbacks=[sv])
```