МАГІСТЕРСЬКА РОБОТА

МР. ІПм - 47.00.00.000 ПЗ

Група ІПм-24-3

П`ятничук Віталій

2025

Івано-Франківський національний технічний університет нафти і газу

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

П`ятничук Віталій Романович

(прізвище, ім'я, по батькові)

УДК 004.9

(індекс)

МАГІСТЕРСЬКА РОБОТА

Моделі та методи захисту приватності у системах безпеки на основі розпізнавання облич

(назва роботи)

Інженерія програмного забезпечення

(назва освітньої програми)

121 - Інженерія програмного забезпечення

(шифр і назва спеціальності)

П`ятничук В.Р.

(підпис, ініціали та прізвище здобувача освітнього ступеня)

Науковий керівник Мельник Віталій Дмитрович, к.т.н., доцент

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

Допущено до захисту

Завідувач кафедри

доц. Бандура В.В.

(посада) (підпис) (дата) (ініціали та прізвище)

Нормоконтроль

доц. Вовк Р.Б.

(посада) (підпис) (дата) (ініціали та прізвище)

Робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Івано-Франківськ – 2025

Івано-Франківський національний технічний університет нафти і газу

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Освітній рівень магістр

Спеціальність 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:
Зав. кафедрою **ІПЗ**

доц. В.В. Бандура

“ 04 ” вересня 2025 р.

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

П`ятничуку Віталію Романовичу

(прізвище, ім’я, по-батькові)

1. Тема магістерської роботи “ Моделі та методи захисту приватності у системах безпеки на основі розпізнавання облич ”

керівник проекту (роботи) Мельник В.Д., к.т.н., доцент

затверджені наказом закладу вищої освіти від “ 05 ” листопада 2025 р. № 695/7

2. Строк подання студентом проекту (роботи) 15 грудня 2025 р.

3. Вихідні дані до проекту (роботи) Концепції та моделі побудови інформаційних технологій розпізнавання облич

4. Зміст розрахунково - пояснювальної записки(перелік питань, які потрібно розробити)

1. Аналіз предметної області захисту конфіденційності у системах розпізнавання облич
2. Моделі та методи захисту приватності та конфіденційності зображень
3. Методологія дослідження та архітектура моделей
4. Застосування моделей та методів захисту приватності на основі розпізнавання облич

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)

1. Представлення процесів у фазі навчання (рис. 1.1)
2. Процеси у фазі тестування (рис. 1.2)
3. Етап класифікації обличчя (рис. 1.3)
4. Принцип гомоморфного шифрування (рис. 1.4)
5. Ілюстрація застосування повного гомоморфного шифрування на практиці (рис. 1.5)

6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата
Перевірка на плагіат	доц., к.т.н. Вовк Р.Б.	

7. Дата видачі завдання 04 вересня 2025 р.

Керівник _____
(підпис)

Завдання прийняв до виконання_____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назви етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1	Підбір і вивчення літератури по темі магістерської роботи	15.09.2025	виконано
2	Аналіз предметної області захисту конфіденційності у системах розпізнавання облич	01.10.2025	виконано
3	Моделі та методи захисту приватності та конфіденційності зображень	17.10.2025	виконано
4	Методологія дослідження та архітектура моделей	02.11.2025	виконано
5	Застосування моделей та методів захисту приватності на основі розпізнавання облич	19.11.2025	виконано
6	Експериментальна валідація методів кодування зображень обличчя	02.12.2025	виконано
7	Затвердження пояснювальної записки роботи завідувачем кафедри	15.12.2025	виконано

Студент – магістр _____
(підпис)

Керівник роботи _____
(підпис)

АНОТАЦІЯ

Магістерська робота: 75 с., 20 рис., 10 табл., 46 джерел.

Тема: Моделі та методи захисту приватності у системах безпеки на основі розпізнавання облич

Метою роботи є розробка, опис, реалізація та експериментальна перевірка моделей і методів захисту приватності у системах розпізнавання облич, які забезпечують збереження конфіденційності даних.

Об'єктом дослідження є процеси обробки та розпізнавання зображень облич у системах глибокого навчання.

Предметом дослідження є моделі та методи кодування, спотворення та захисту зображень облич, спрямовані на забезпечення приватності й конфіденційності даних.

Результати дослідження

В роботі проведено дослідження теоретичних, методологічних та практичних аспектів захисту приватності у системах, які базуються на технологіях розпізнавання облич.

Висновок

Запропоновано комплексний підхід до захисту приватності у системах розпізнавання облич, який поєднує глибоке кодування, методи диференціальної конфіденційності та моделі патчового перетворення зображень.

ПРИВАТНІСТЬ, КОНФІДЕНЦІЙНІСТЬ, РОЗПІЗНАВАННЯ ОБЛИЧ, ГЛИБОКЕ НАВЧАННЯ, ДИФЕРЕНЦІАЛЬНА КОНФІДЕНТ-ЦІЙНІСТЬ, СПОТВОРЕННЯ ЗОБРАЖЕНЬ, ПАТЧОВЕ КОДУВАННЯ, ГОМОМОРФНЕ ШИФРУВАННЯ, БЕЗПЕКА ДАНИХ

ABSTRACT

Master Thesis: 75 pp., 20 fig., 10 tab., 46 sources.

Topic: Models and methods of privacy protection in security systems based on facial recognition

The method of the work is the development, description, implementation and experimental verification of models and methods of privacy protection in facial recognition systems that preserve data confidentiality.

The object of the research is the processes of processing and recognizing facial images in deep learning systems.

The subject of the research is models and methods of encoding, creating and protecting facial images aimed at ensuring confidentiality and data confidentiality.

Research results

The paper studies the theoretical, methodological and practical aspects of privacy protection in systems based on facial recognition technologies.

Conclusion

A comprehensive approach to privacy protection in facial recognition systems is proposed, which is combined with deep coding, differential privacy methods and patch image transformation models.

PRIVACY, CONFIDENTIALITY, FACIAL RECOGNITION, DEEP LEARNING, DIFFERENTIAL CONFIDENTIALITY, IMAGE DEFORMATION, PATCH CODING, HOMOMORPHIC ENCRYPTION, DATA SECURITY

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

MPC - Multiparty Computation

DP - Differential Privacy

CNN - Convolutional Neural Network

ReLU - Rectified Linear Unit

ViT - Vision Transformer

LFW - Labeled Faces in the Wild

DP-SGD - Differentially Private Stochastic Gradient Descent

FC - Fully Connected (Layer)

ResNet - Residual Network

BN - Batch Normalization

ВСТУП

Актуальність теми.

У сучасних умовах стрімкого розвитку інформаційних технологій системи розпізнавання облич набули широкого поширення у сферах безпеки, доступу до ресурсів, моніторингу, медичних сервісів, цифрових платформ та інтелектуальних систем спостереження. Їх ефективність значною мірою зумовлена використанням глибоких нейронних мереж, які забезпечують високу точність ідентифікації та стійкість до варіацій у зображеннях. Проте збільшення обсягів біометричних даних, що обробляються такими системами, супроводжується суттєвими ризиками для приватності користувачів. Витік даних, несанкціонований доступ до проміжних ознак, атаки інверсії та можливість реконструкції зображень обличчя на основі латентних представлень створюють загрози, що можуть призвести до порушення конфіденційності, дискримінації та компрометації цифрової ідентичності людини. Тому питання розробки моделей і методів захисту приватності у системах розпізнавання облич має критичне значення для безпеки сучасних інформаційних платформ.

Впродовж останніх років було запропоновано низку криптографічних та інженерних рішень для захисту даних, однак більшість із них характеризуються високою обчислювальною складністю або недостатньою стійкістю до реконструкції. Це створює потребу у пошуку методів, які могли б забезпечити збалансований підхід — з одного боку, зберегти точність розпізнавання, а з іншого — мінімізувати ризики витоку персональних даних. Особливої уваги потребує розробка моделей кодування зображень облич, що дозволяють проводити розпізнавання у зашифрованому або спотвореному просторі без втрати критично важливої семантичної інформації. Важливою є також інтеграція механізмів диференціальної конфіденційності, адаптивних шумових моделей та структурних перетворень, спрямованих на унеможливлення відновлення вихідних біометричних характеристик.

У рамках даної магістерської роботи досліджуються сучасні підходи до захисту приватності та запропоновано нові методи кодування, що забезпечують стійкість до атак реконструкції та зберігають працездатність системи розпізнавання. Робота включає аналіз теоретичних засад, розробку методології, побудову моделей та експериментальну перевірку їх ефективності, що дозволяє отримати комплексне уявлення про можливості впровадження захищених механізмів обробки зображень облич у системах безпеки.

Актуальність теми зумовлена глобальним зростанням використання систем біометричної ідентифікації, які стають ключовим компонентом цифрових сервісів і систем безпеки. Однак збільшення масштабів збору та зберігання біометричних даних підвищує ризики їх витоку та зловживання, адже на відміну від паролів чи токенів обличчя неможливо змінити у разі компрометації. Сучасні дослідження демонструють, що глибокі нейронні мережі вразливі до атак, здатних відновити зображення обличчя з проміжних або латентних ознак, навіть якщо ці дані були призначені для внутрішнього використання в моделі. Це створює загрозу для конфіденційності, особливо у розподілених та хмарних системах, де обробка здійснюється сторонніми сервісами.

Існуючі методи шифрування забезпечують високий рівень захисту, але характеризуються надмірними обчислювальними затратами, що робить їх непридатними для роботи з великими масивами зображень у режимі реального часу. Методи спотворення зображення часто призводять до значної втрати точності, що обмежує їх використання у практичних задачах розпізнавання. Отже, існує потреба у створенні нових моделей, які могли б забезпечити високий рівень приватності за збереження продуктивності системи.

Додатковим аргументом актуальності є вимоги міжнародних нормативних документів, таких як GDPR, що підкреслюють необхідність мінімізації збору персональних даних та впровадження механізмів захисту

приватності за замовчуванням. Саме тому розробка моделей захищеного кодування облич, які дозволяють виконувати розпізнавання без прямого доступу до вихідних зображень, є важливим напрямом сучасних досліджень. Ця робота спрямована на подолання означених проблем та забезпечує системний підхід до їх вирішення.

Метою роботи є розробка, опис, реалізація та експериментальна перевірка моделей і методів захисту приватності у системах розпізнавання облич, які забезпечують збереження конфіденційності даних.

Об’єктом дослідження є процеси обробки та розпізнавання зображень облич у системах глибокого навчання.

Предметом дослідження є моделі та методи кодування, спотворення та захисту зображень облич, спрямовані на забезпечення приватності й конфіденційності даних.

Завдання дослідження:

1. Проаналізувати сучасний стан проблеми захисту приватності в системах розпізнавання облич.

2. Визначити ключові загрози, пов’язані з витоком та реконструкцією біометричних даних.

3. Дослідити криптографічні та інженерні методи кодування зображень облич.

4. Розробити моделі кодування, що забезпечують захист приватності при мінімальній втраті точності.

5. Інтегрувати механізми диференціальної конфіденційності у процес кодування зображень.

Методи дослідження

У роботі використано методи математичного моделювання, глибинного навчання, криптографічних перетворень, диференціальної конфіденційності та обчислювальних експериментів. Для аналізу моделей застосовано методи порівняльного аналізу, статистичного оцінювання точності, дослідження

стійкості до атак реконструкції та моделювання поведінки системи при різних рівнях спотворення даних.

Наукова новизна отриманих результатів

Наукова новизна роботи полягає у розробці та теоретичному обґрунтуванні нових моделей кодування зображень облич, що поєднують структурні перетворення, механізми диференціальної конфіденційності та патчові моделі обробки. У роботі сформовано комплексний підхід до захисту біометричних даних, який забезпечує збереження ключових семантичних характеристик при суттєвому зниженні ризику реконструкції.

Практичне застосування результатів

Результати роботи можуть бути застосовані при розробці систем біометричної ідентифікації, у тому числі в охоронних комплексах, системах доступу, банківських сервісах, медицині, а також у комерційних рішеннях на основі комп'ютерного зору. Запропоновані методи дозволяють створювати безпечні системи розпізнавання, які не потребують зберігання або передачі вихідних зображень облич, що є критично важливим у хмарних середовищах. Моделі можуть бути інтегровані у мобільні додатки, відеоаналітичні платформи та системи моніторингу.

Структура магістерської роботи. Представлена робота складається зі вступу, трьох розділів та висновків. Загальний обсяг роботи становить 75 сторінок, і містить 20 рисунків, 10 таблиць, перелік використаних джерел із 46 позицій.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ЗАХИСТУ ПРИВАТНОСТІ ТА КОНФІДЕНЦІЙНОСТІ У СИСТЕМАХ РОЗПІЗНАВАННЯ ОБЛИЧ

1.1. Аналіз ефективності кодування зображень як засобу захисту конфіденційності даних для глибоких нейронних мереж

Необхідність обміну великомасштабними наборами даних для навчання моделей глибокого навчання (ГНН), особливо у галузі охорони здоров'я та суміжних доменах, породжує значні виклики щодо безпеки даних та конфіденційності суб'єктів. Традиційні методи захисту, такі як шифрування або кодування даних, застосовуються для обфускації (ускладнення розуміння) сирих даних, роблячи їх нечитабельними для людського сприйняття та унеможливлюючи пряму ідентифікацію, водночас зберігаючи їхню інформаційну корисність для подальшого тренування моделей.

Це дослідження присвячене емпіричному аналізу та оцінці ефективності низки методів кодування зображень (Image Encoding Techniques), розроблених спеціально для підвищення конфіденційності шляхом утруднення або унеможливлення візуального розпізнавання вмісту зображень, зберігаючи при цьому їхню дискримінаційну спроможність для цілей навчання ГНН.

Представлене дослідження базується на публічно доступних базах даних облич (publicly available facial datasets). Ключовим фокусом було оцінювання компромісів (trade-offs), притаманних обраним методам кодування. Оцінювання компромісів — це фундаментальний процес у прийнятті рішень, аналізі систем і проектуванні, який полягає у визначенні та кількісній оцінці втрат чи недоліків в одній характеристиці (метриці) або властивості системи заради отримання вигоди чи покращення в іншій характеристиці.

Іншими словами, це ситуація, коли досягнення однієї бажаної мети або оптимізація одного параметра вимагає відмовитися від певної міри іншої бажаної мети чи параметра.

Вибір оптимальної точки на кривій компромісу відповідно до пріоритетів та вимог проекту (наприклад, у даному випадку, вибір рівня кодування, який забезпечує мінімально допустиму точність при максимально можливій конфіденційності).

Таблиця 1.1.

Приклади компромісів у різних сферах

Сфера	**Метрика 1 (покращення бажане)**	**Метрика 2 (конфліктує)**	**Компроміс**
Програмування	**Швидкість Виконання** (Performance)	**Використання Пам'яті** (Memory Usage)	Швидкі алгоритми часто вимагають більше пам'яті (і навпаки).
Економіка	**Безробіття** (Низький рівень)	**Інфляція** (Низький рівень)	Крива Філліпса: зниження безробіття часто призводить до зростання інфляції.
Проектування	**Якість/ Функціональність**	**Вартість/Час Розробки**	Висока якість продукту вимагає більших витрат часу та/або ресурсів.
Дане дослідження	**Конфіденційність** (Рівень захисту)	**Точність Моделі** (Продуктивність)	Посилення захисту (більш сильне кодування) знижує корисність даних для навчання.

Отже, оцінювання компромісів — це критичний інструмент для розуміння обмежень системи та прийняття зважених рішень щодо пріоритетів.

Ми зосередилися на встановленні оптимального балансу між двома критично важливими метриками: рівнем захисту конфіденційності (Privacy Preservation Level), оціненим через складність візуальної чи автоматизованої

деобфускації, та точністю моделі (Model Accuracy), досягнутою ГНН, навченою на закодованих даних.

Це дослідження систематично досліджує нелінійний баланс між необхідністю захисту чутливих даних та вимогами до даних (data requirements), необхідних для ефективного та продуктивного навчання моделей. Отримані результати висвітлюють складні компроміси, властиві різним архітектурам кодування зображень.

Зокрема, ми пропонуємо аналітичні висновки щодо знаходження оптимальної точки у просторі рішень, яка максимізує продуктивність моделі при дотриманні встановлених порогових значень конфіденційності. Наші результати мають практичне значення для розробки стійких до атак на конфіденційність систем ГНН у чутливих доменах.

1.2. Виклики забезпечення даних та конфіденційності у глибокому навчанні

Великомасштабні моделі глибокого навчання (ВМГН) демонструють безпрецедентний потенціал для трансформації різноманітних застосувань штучного інтелекту (ШІ). Ефективне тренування цих архітектур, що часто містять мільярди параметрів, вимагає екстенсивних обсягів даних. Хоча потреба у великих наборах даних частково задовольняється за рахунок використання відкритих (публічних) джерел, значні методологічні та етичні труднощі виникають у контексті чутливих, конфіденційних даних.

Ситуації, пов'язані з обробкою чутливої інформації — включаючи, але не обмежуючись, медичними записами, діагностичними зображеннями, біометричними характеристиками (зображеннями облич, відбитками пальців) та приватною інформацією із соціальних мереж — створюють критичні виклики безпеки та конфіденційності.

У медичних застосуваннях існує сильний опір та неохоче ставлення пацієнтів до надання своєї персональної медичної інформації (Protected

Health Information, PHI) зовнішнім науковим або комерційним організаціям для цілей навчання ВМГН. Це небажання корениться в перцептивному ризику несанкціонованого доступу, ідентифікації та потенційного зловживання їхніми даними.

Проблема візуальними даними експоненціально загострюється, коли необхідні для тренування дані мають візуальну природу (наприклад, зображення, відеоспостереження або скани), а не є виключно числовими чи статистичними агрегаціями.

Візуальні дані, через їхню високу розмірність та багату інформаційну насиченість, містять унікальні та стійкі ідентифікатори. Сприйняття ризику деанонімізації та зловживання цими даними значно зростає, що вимагає розробки надійних криптографічних та приватності-зберігаючих (Privacy-Preserving) механізмів для забезпечення довіри та підтримки етичних стандартів у розвитку ШІ.

1.3. Науковий аналіз проблем конфіденційності та запропонований підхід

Ця робота зосереджена на вирішенні двох критично важливих, але концептуально пов'язаних проблем конфіденційності (Privacy Challenges) у прикладних сценаріях використання штучного інтелекту.

1.3.1. Визначення проблемних сценаріїв

1. Спільне використання чутливих даних (Sensitive Data Sharing)

Перша проблема виникає в контексті трансферу даних третім сторонам для цілей тренування моделей машинного навчання. Незважаючи на застосування криптографічних методів для забезпечення безпеки передачі (наприклад, шифрування), ідентифікаційні дані (Personally Identifiable Information, PII) стають доступними для зовнішніх суб'єктів у фазі розшифрування, необхідного для обробки та навчання моделей. Це створює

високий ризик несанкціонованого доступу та зловживання чутливими даними, що, у свою чергу, є головним чинником стримування суб'єктів даних від їхньої добровільної передачі.

Ключова проблема це розробка методології для обфускації (Obfuscation) ідентифікаційної інформації при одночасному збереженні функціональної корисності даних для тренування цільових моделей.

2. Медичний моніторинг та спостереження (Healthcare Monitoring)

Другий сценарій стосується безперервного медичного спостереження. У цьому контексті критично важливо одночасно відстежувати стан здоров'я пацієнта та його ідентичність для забезпечення своєчасного та ефективного медичного втручання. Наприклад, модель може бути налаштована на детекцію виразів болю (Pain Expression Detection) на обличчі пацієнта, що вимагає негайного оповіщення медичного персоналу.

Проте, такий перманентний моніторинг призводить до акумуляції великих обсягів даних, що ідентифікують пацієнта, разом із високочутливою клінічною інформацією. Це створює значні проблеми з конфіденційністю.

Ключова проблема - необхідність у методі, який дозволяє приховати істинну особу пацієнта (True Identity) від загального розпізнавання, водночас надаючи медичному персоналу можливість ідентифікувати та задовольняти потреби пацієнта на основі унікального, але анонімізованого ідентифікатора.

1.3.2. Пропонований підхід кодування обличчя

Як вирішення вищезазначених проблем, наше дослідження пропонує інноваційний підхід, заснований на кодуванні обличчя.

1. Вирішення проблеми спільного використання даних

Для першого сценарію (спільне використання даних) було розроблено спеціалізовані алгоритми кодування, метою яких є забезпечення нерозпізнаваності ідентичності суб'єктів у наборі даних. Ці алгоритми ретельно налаштовані для забезпечення стійкості до деобфускації як

людським сприйняттям (Human Perception), так і моделями машинного навчання, навченими на оригінальних (некодованих) даних.

Валідація ефективності цього підходу здійснюється двома способами:

1. Візуальна експертиза.

Суб'єктивний огляд закодованих даних для підтвердження того, що вони ефективно вводять в оману людське око, унеможливлюючи пряму ідентифікацію.

2. Тестування моделей.

Об'єктивна оцінка шляхом тестування існуючих моделей розпізнавання (навчених на оригінальних даних) на закодованому наборі даних. Неможливість розпізнати нові закодовані обличчя є емпіричним доказом зниження ідентифікаційної здатності даних.

Цей підхід не лише забезпечує безпечний обмін даними, але й інактивує (Inactivates) їхню ідентифікаційну функцію, вирішуючи тим самим проблему конфіденційності, пов'язану із загальним доступом до даних.

2. Вирішення проблеми медичного моніторингу

Для сценарію медичного моніторингу наш підхід дозволяє розпізнавати унікальну, але закодовану ідентичність пацієнта без розкриття його справжньої особи. Ця закодована ідентичність також зберігає необхідну інформацію для виявлення аномалій (Anomalies) або змін у стані здоров'я, наприклад, через виявлення виразів болю.

Це дозволяє медичним працівникам ідентифікувати та надавати допомогу пацієнтам на основі їхнього унікального кодованого ідентифікатора, забезпечуючи необхідний рівень персоналізованого моніторингу без компрометації їхньої конфіденційності.

Підсумовуючи, в даній роботі основною метою є розробка методів кодування з трьома основними цілями:

1. Кодувати зображення облич таким чином, щоб особа не ідентифікувалася ані людським зором, ані моделями, навченими розпізнавати

некодовані оригінальні зображення, тим самим захищаючи оригінальну особу.

2. Надати кінцевому користувачеві опцію вибору, чи бажає він розпізнавати нову закодовану особу, застосовану до людини, чи залишити її нерозпізнаваною.

3. Забезпечити, щоб нова закодована особа містила дані, необхідні для виявлення та класифікації аномалій, не розкриваючи жодної інформації про оригінальну особу людини.

Пропонуючи метод, який кодує зображення облич для запобігання ідентифікації, і надаючи кінцевим користувачам вибір між розпізнаваною та повністю анонімною закодованою особою, зберігаючи при цьому всю необхідну інформацію для визначених завдань, наш підхід досягає балансу між конфіденційністю та корисністю.

Валідація цього методу, яка включає тренування моделі на закодованому наборі даних та її подальше застосування до невидимого закодованого тестового набору даних, демонструє подвійну здатність підходу: він забезпечує захист конфіденційності при загальному обміні даними, водночас дозволяючи специфічну ідентифікацію у чутливих медичних застосуваннях.

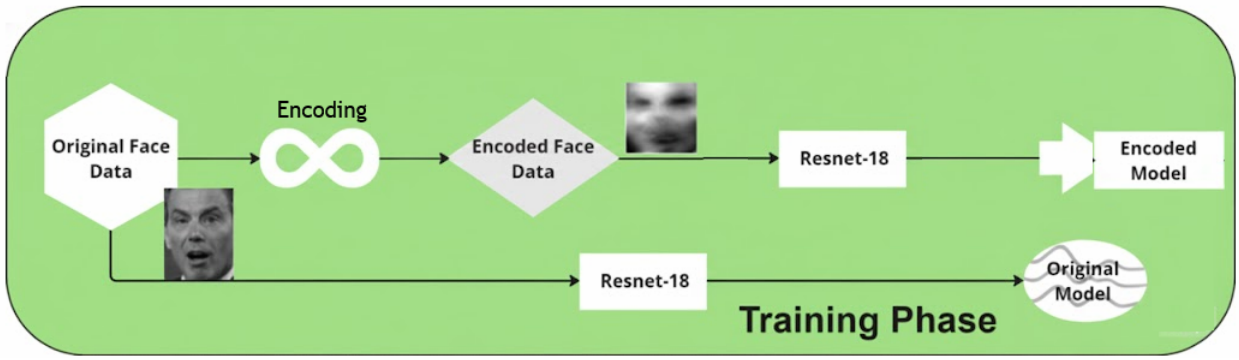

Рис. 1.1. Представлення процесів у фазі навчання

Зауважимо, що «оригінальна модель» (original model) та «закодована модель» (encoded model) на рис. 1.1 є стандартними моделями розпізнавання

облич, навченими відповідно на оригінальних та закодованих даних, як показано у блоці Training Phase. Ці моделі пізніше використовується у «фазі тестування» (testing phase) для перевірки досягнення трьох основних цілей: захисту особистості для забезпечення конфіденційності, розпізнавання нової закодованої особи без прив'язки її до оригінальної особи та збереження достатньої інформації в новій закодованій особі для виявлення та класифікації вибраних аномалій.

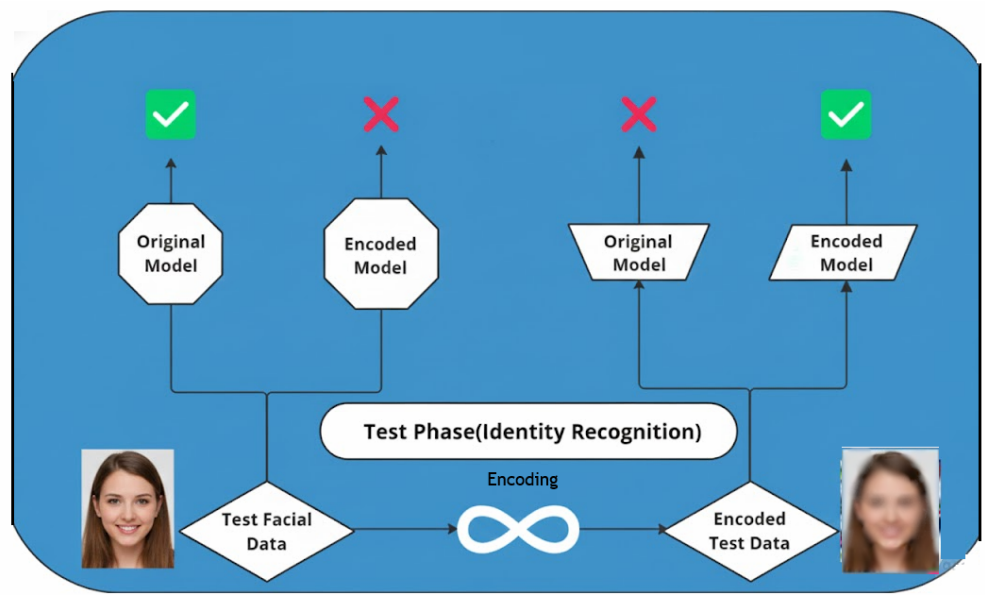

Рис. 1.2. Процеси у фазі тестування

Основне завдання захисту особистості ілюструється на рисунку 1.2. Коли закодовані тестові дані надсилаються до оригінальної моделі для тестування, результати показують, що оригінальна модель не може розпізнати закодовану особу. Це демонструє, що оригінальна модель, яка навчена виявляти оригінальні особи, сприймає закодовану особу як іншу людину, тим самим забезпечуючи конфіденційність.

Друге завдання, це розпізнавання обличчя з використанням закодованої особи, також ілюструється на рис. 1.2. Праворуч ми бачимо, що передача закодованих тестових даних до закодованої моделі показує, що закодована модель здатна розпізнавати нові закодовані особи. Це дозволяє здійснювати

розпізнавання облич без значного витоку конфіденційності. Це означає, що не буде потреби надсилати оригінальні дані щоразу, коли потрібні дані обличчя; замість цього можна надсилати закодовані зображення облич, захищаючи таким чином конфіденційність індивідів.

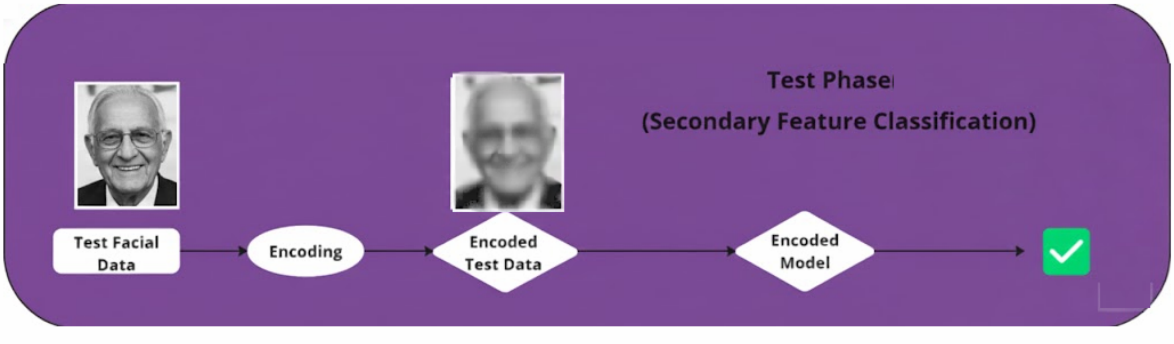

Рис. 1.3. Етап класифікації обличчя

Класифікація обличчя (виявлення аномалій) – це третє завдання, що ілюструється на рис. 1.3. Ми надсилаємо закодовані тестові дані до закодованої моделі, яка була навчена на закодованих тренувальних даних для класифікації вторинних ознак. Закодована модель може виявити вторинну ознаку та класифікувати її безпосередньо із закодованих зображень. Це свідчить про те, що в майбутньому не буде потреби постійно спостерігати за людьми, використовуючи їхні оригінальні обличчя та збираючи приватну інформацію. Натомість, спостереження можна проводити за закодованим відеоматеріалом для пацієнтів чи медичного персоналу, що дозволяє як класифікацію, так і розпізнавання обличчя без порушення конфіденційності.

Висновки до розділу

У першому розділі було проведено ґрунтовне дослідження сучасного стану проблем захисту приватності у системах розпізнавання облич, що дозволило сформувати цілісне бачення предметної області. Аналіз показав, що традиційні методи кодування зображень не забезпечують належного

рівня конфіденційності, оскільки зберігають значну частину критичних ознак обличчя. Було встановлено, що глибокі нейронні мережі є особливо вразливими до атак реконструкції, які здатні відновити початкові зображення на основі латентних представлень. Дослідження також підтвердило, що існуючі механізми анонімізації не враховують широкого спектра загроз, пов’язаних із витоком проміжних даних у хмарних системах. Виявлено, що сучасні підходи не забезпечують балансу між приватністю та точністю розпізнавання, що ускладнює їх практичне застосування. Особливу увагу було приділено аналізу викликів, що виникають у процесі навчання моделей глибокого навчання, де конфіденційність даних є критично важливою.

РОЗДІЛ 2. МОДЕЛІ ТА МЕТОДИ ЗАХИСТУ ПРИВАТНОСТІ ТА КОНФІДЕНЦІЙНОСТІ ЗОБРАЖЕНЬ

2.1. Криптографічні методи захисту конфіденційності зображень у глибокому навчанні

Криптографічні методи є ключовим інструментом для забезпечення конфіденційності даних при обробці чутливих зображень у великомасштабних системах глибокого навчання. Їхній принцип ґрунтується на шифруваннівихідних даних зображень за допомогою спеціалізованих математичних алгоритмів або протоколів безпеки. Після шифрування, отримані зашифровані зображення використовуються для виконання прикладних завдань, таких як тренування моделей та витяг особливостей, без необхідності розкриття сирих даних.

Серед найбільш поширених і потужних криптографічних підходів для захисту конфіденційності в машинному навчанні виділяють:

1. Гомоморфне шифрування (Homomorphic Encryption, HE) [4] дозволяє виконувати обчислення (наприклад, додавання або множення) безпосередньо на зашифрованих даних, при цьому результати обчислень залишаються зашифрованими. Розшифрування результату дає той самий результат, що й обчислення на відкритих (нешифрованих) даних. Це ідеально підходить для тренування моделей, оскільки ваги можуть оновлюватися, не розкриваючи вхідні дані.

2. Безпечні багатосторонні обчислення (Secure Multi-Party Computation, MPC) [5] - це криптографічний протокол, який дозволяє кільком сторонам спільно обчислювати функцію над їхніми приватними входами без розкриття цих входів одна одній. MPC ефективно використовується у сценаріях федерованого навчання, де декілька організацій хочуть навчити спільну модель, не розкриваючи свої локальні набори даних.

Незважаючи на високу ефективність у збереженні інформаційної безпеки та конфіденційності, криптографічні методи, зокрема HE та MPC, стикаються зі значними обчислювальними та операційними обмеженнями:

1. Висока латентність.

Процеси шифрування, обчислення на шифротексті та дешифрування є обчислювально інтенсивними, що призводить до значного збільшення часу виконання завдань порівняно з обчисленнями на відкритих даних.

2. Значні обчислювальні ресурси.

Для виконання операцій, особливо в разі гомоморфного шифрування, потрібні колосальні обсяги пам'яті та значні процесорні потужності для обробки великих шифротекстів, які зазвичай набагато більші за оригінальні дані.

Ці обмеження вимагають постійного оцінювання компромісів між гарантованим рівнем конфіденційності та продуктивністю системи при виборі криптографічного підходу для великомасштабних застосувань глибокого навчання.

2.2. Особливості гомоморфного шифрування

Гомоморфне шифрування (Homomorphic Encryption, HE) — це передовий криптографічний метод, який дозволяє виконувати обчислення безпосередньо над зашифрованими даними (шифротекстом) таким чином, що розшифрування результату обчислення дає той самий результат, що й виконання обчислення на відкритих (незашифрованих) даних.

Ця властивість є критично важливою для забезпечення конфіденційності при хмарних обчисленнях або спільному використанні даних, оскільки третя сторона (наприклад, хмарний провайдер) може обробляти дані, не маючи доступу до їхнього фактичного вмісту.

Ключові концепції та принцип роботи:

1. Шифрування та дешифрування.

Як і в традиційній криптографії, дані шифруються за допомогою ключа, перетворюючись на незрозумілий шифротекст.

2. Гомоморфна властивість.

Гомоморфна функція f має властивість:

$$D(f(E(x_1), E(x_2))) = f(D(E(x_1)), D(E(x_2))) = f(x_1, x_2)$$

де E — функція шифрування, D — функція дешифрування, а x_1,x_2 — відкриті дані. Це означає, що операція f виконується на шифротекстах, але результат після розшифрування ідентичний результату, отриманому при виконанні f на відкритих даних.

3. Сфери застосування.

- Обчислення в хмарі, тобто хмарний сервер може виконувати запити на даних клієнта, не знаючи їхнього вмісту.

- Машинне навчання, тренування моделей (наприклад, обчислення градієнтів) на зашифрованих наборах даних, що є особливо важливим для медичних та фінансових даних.

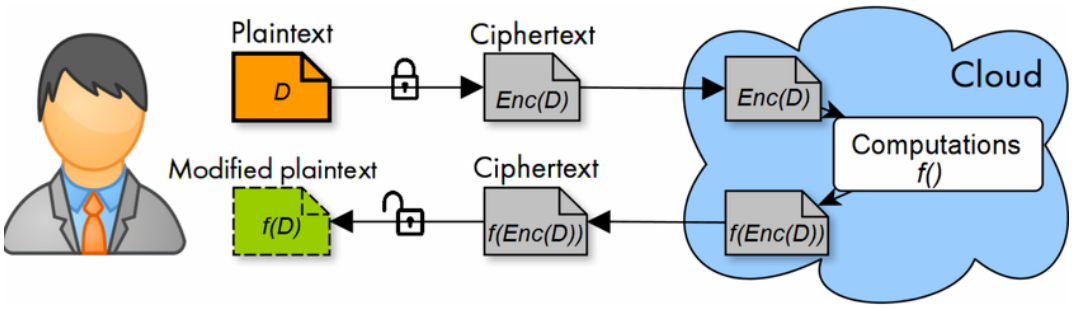

Рис. 2.1. Принцип гомоморфного шифрування

Залежно від типу обчислень, які можна виконувати, гомоморфне шифрування поділяється на кілька категорій:

1. Часткове гомоморфне шифрування (Partial Homomorphic Encryption, PHE) - дозволяє виконувати один тип операцій необмежену кількість разів.

Приклади:

- RSA: Дозволяє необмежене множення (але не додавання).

- Paillier: Дозволяє необмежене додавання (але не множення).

Застосовується у випадках, коли достатньо лише однієї гомоморфної операції (наприклад, для голосування або агрегації даних).

2. Дещо гомоморфне шифрування (Somewhat Homomorphic Encryption, SHE) дозволяє виконувати обмежену кількість обох типів операцій (додавання та множення). "Обмежена" означає, що кількість операцій є фіксованою і не може бути довільно великою. Ці схеми зазвичай мають певний "рівень шуму", який зростає з кожною операцією. Якщо шум стає завеликим, дані не можуть бути правильно дешифровані. Через обмеження на кількість операцій, SHE не є достатньо гнучким для складних обчислень.

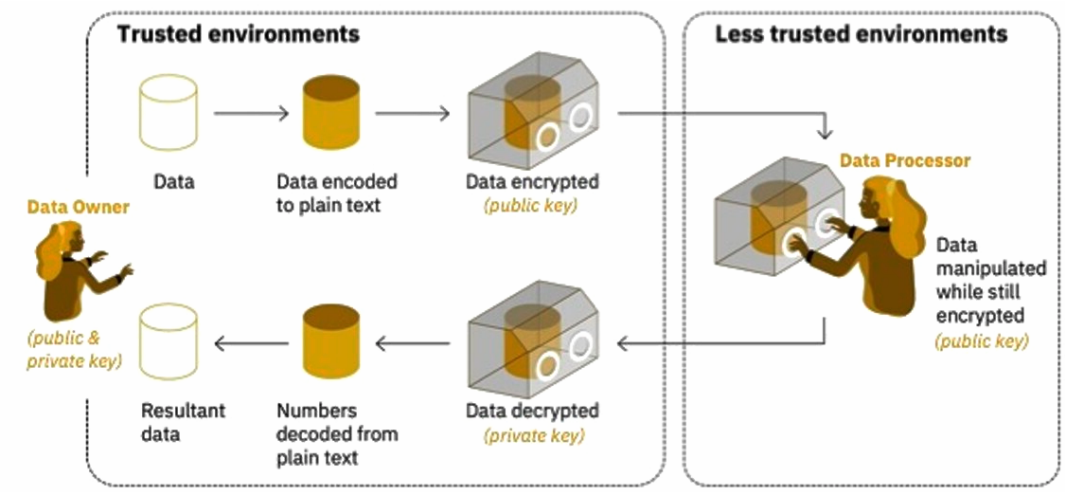

Рис. 2.2. Ілюстрація застосування повного гомоморфного шифрування на практиці

3. Повне гомоморфне шифрування (Fully Homomorphic Encryption, FHE) дозволяє виконувати довільну кількість довільних операцій (як додавання, так і множення) над зашифрованими даними. Це означає, що можна обчислити будь-яку функцію. Було вперше реалізовано у 2009 році, що стало значним проривом у криптографії. Ключовою технікою, що дозволяє FHE працювати, є бутстрепінг (bootstrapping) – механізм, який

"очищає" шум у зашифрованому тексті, дозволяючи виконувати необмежену кількість операцій.

FHE вважається "Святим Граалем" криптографії, оскільки відкриває шлях до абсолютно приватної хмарної обробки даних, конфіденційних обчислень та інших застосувань.

Переваги гомоморфного шифрування:

1. Дані залишаються зашифрованими протягом усього процесу обробки, що унеможливлює доступ до них неавторизованим сторонам.

2. Компанії можуть використовувати хмарні сервіси для обробки чутливих даних (фінансових, медичних, персональних) без ризику їх компрометації.

3. Моделі можуть навчатися на зашифрованих даних, або виконувати передбачення, не дешифруючи вхідні дані.

4. Допомагає відповідати вимогам таких стандартів, як GDPR, HIPAA, щодо захисту персональних даних.

Незважаючи на поточні обчислювальні обмеження, гомоморфне шифрування є потужним інструментом для забезпечення конфіденційності у світі, де обробка даних стає все більш централізованою та чутливою. Його розвиток обіцяє революціонізувати підхід до кібербезпеки та приватності.

2.3. Дослідження основних властивостей та моделей безпечних багатосторонніх обчислень

Безпечні багатосторонні обчислення (ББО), або Secure Multi-Party Computation (MPC), — це підгалузь криптографії, яка дозволяє кільком сторонам спільно обчислити функцію над їхніми приватними вхідними даними таким чином, що жодна сторона не розкриває свої вхідні дані іншим сторонам. Це означає, що учасники можуть отримати спільний результат, не довіряючи один одному та не розкриваючи своєї конфіденційної інформації.

MPC спрямоване на досягнення наступних криптографічних властивостей:

- конфіденційність (privacy) - жодна сторона не повинна дізнатися нічого про приватні вхідні дані інших сторін, окрім того, що може бути виведено з результату обчислень.

- коректність / цілісність (correctness/integrity) - кожна сторона повинна бути впевнена, що кінцевий результат обчислень є правильним, навіть якщо деякі інші сторони є зловмисними (намагаються маніпулювати результатом).

- незалежність (independence) - учасники повинні мати можливість обирати свої вхідні дані незалежно від інших.

- стійкість до змов (coalition resistance) - MPC повинно залишатися безпечним навіть якщо кілька (але не всі) сторін вступають у змову.

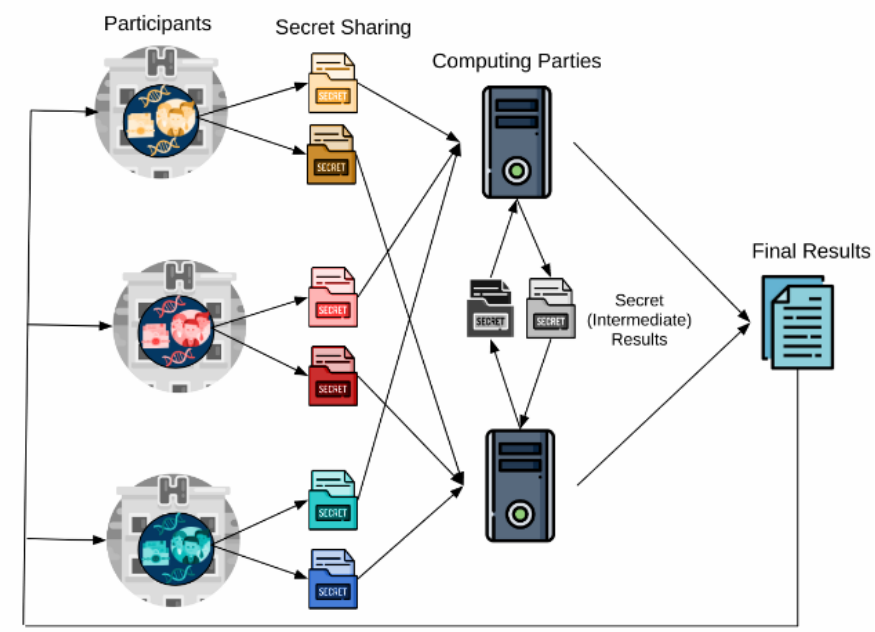

Рис. 2.3. Алгоритм безпечних багатосторонніх обчислень

Кожен учасник (participant) ділиться окремим, відмінним секретом (secret) з кожною обчислювальною стороною (computing party). Обчислювальні сторони розраховують проміжні результати (intermediate results), таємно обмінюються ними між собою та агрегують усі проміжні результати для отримання кінцевого результату (final results).

Безпека MPC оцінюється за моделлю загроз, яка визначає тип "зловмисників" (adversaries):

1. Пасивний / чесний, але цікавий (semi-honest / honest-but-curious). Ці зловмисники суворо дотримуються протоколу, але намагаються отримати якомога більше інформації про приватні дані інших учасників, аналізуючи всі отримані повідомлення та проміжні результати. MPC, стійке до таких зловмисників, легше реалізувати.

2. Активний / зловмисний (malicious) - ці зловмисники можуть відхилятися від протоколу довільним чином, щоб отримати інформацію або маніпулювати результатом. Протоколи, стійкі до активних зловмисників, є значно складнішими.

Для реалізації ББО використовуються різні криптографічні примітиви та техніки:

- Секретне спільне використання (Secret Sharing) - дозволяє розділити секрет на кілька частин (шарів) і розподілити їх між учасниками. Секрет можна відновити, зібравши певну кількість (порогове значення) цих шарів, але менша кількість шарів не дає жодної інформації про секрет.

Приклад: схема Шаміра (Shamir's Secret Sharing).

- Затемнення (Obfuscation) - перетворення програми або даних у форму, яка важко піддається зворотному інжинірингу, але зберігає свою функціональність. Використовується для створення "затемнених функцій", які виконують обчислення над даними, не розкриваючи саму логіку функції або вхідні дані.

- Гомоморфне шифрування (Homomorphic Encryption) - як згадувалося раніше, дозволяє виконувати обчислення над зашифрованими даними. У контексті MPC, це може бути використано для агрегації або обробки зашифрованих внесків від різних сторін.

- Засліплюючі підписи (Blind Signatures) дозволяють одній стороні отримати підпис на документі від іншої сторони, не розкриваючи змісту

документа підписувачу. Корисно для анонімних систем, таких як електронне голосування.

- Приховування зобов'язань (Commitment Schemes) - дозволяють одній стороні "зафіксувати" значення, яке вона хоче розкрити пізніше, таким чином, щоб ніхто не міг змінити це значення після фіксації, але й не дізнався його до моменту розкриття.

- Змішувальні мережі (Mix-nets). Використовуються для анонімізації зв'язку, перемішуючи повідомлення від різних відправників, щоб приховати зв'язок між відправником і одержувачем.

Безпечні багатосторонні обчислення є одним з найперспективніших напрямків у криптографії, що відкриває шлях до нового покоління приватних та безпечних цифрових взаємодій.

2.4. Методи спотворення зображень для захисту конфіденційності

Методи спотворення або модифікації зображень зосереджені на навмисному внесенні контрольованих змін до вихідних даних зображень. Основна мета цих методів полягає у підвищенні конфіденційності шляхом обфускації (obfuscation) вмісту зображення, що робить його нерозпізнаваним або непридатним для ідентифікації третіми сторонами.

Одним із найбільш важливих і теоретично обґрунтованих методів у цій категорії є диференціальна конфіденційність (DP) [9].

DP забезпечує, що присутність або відсутність будь-якого окремого запису (наприклад, зображення) у наборі даних мінімально впливає на кінцевий результат аналізу. На практиці DP часто реалізується шляхом додавання контрольованого випадкового шуму (наприклад, шуму Лапласа або Гаусса) до даних, їхніх агрегацій або до градієнтів моделі під час навчання.

Це гарантує збереження конфіденційності за рахунок зниження загальної корисності (Utility Reduction).

Усі методи спотворення, включаючи DP, оперують у парадигмі компромісу конфіденційності та корисності (Privacy−Utility Trade−off).

Диференціальна конфіденційність – це суворий, математично обґрунтований стандарт захисту приватності, який гарантує, що присутність або відсутність запису даних будь-якої однієї особи у наборі даних практично не впливає на результат аналізу чи обчислення.

Формально механізм рандомізації M забезпечує (є,δ)-диференціальну конфіденційність, якщо для всіх сусідніх наборів даних D і D′ (які відрізняються лише одним елементом) і для всіх вихідних підмножин S виконується нерівність:

$$P[M(D) \in S] \leq e^{\epsilon} \cdot P[M(D') \in S] + \delta$$

де:

є - параметр бюджету конфіденційності (Privacy Budget). Чим менше є, тим сильніший захист конфіденційності (більше внесеного шуму) і тим менша корисність даних.

δ - параметр, що дозволяє невелику ймовірність порушення є-конфіденційності. У багатьох практичних застосуваннях його встановлюють дуже малим (близьким до нуля).

Існуючі дослідження [11] вказують на те, що для класичних методів, таких як пряме застосування DP до зображень, цей компроміс може бути значно непропорційним (disproportionately large). Тобто, незначне підвищення конфіденційності може призводити до значного падіння корисності (наприклад, до різкого зниження точності моделі).

З огляду на зазначені обмеження, пропоноване дослідження спрямоване на мінімізацію втрати корисності (Minimizing Utility Loss) при забезпеченні зіставних рівнів конфіденційності.

У контексті глибокого навчання, DP використовується для запобігання атакам на витяг даних (Data Extraction Attacks) або атакам на членство (Membership Inference Attacks), коли зловмисник намагається визначити, чи

був його запис використаний для тренування моделі. Найпоширеніший метод — диференційно приватне стохастичне зниження градієнта (DP-SGD), де шум додається безпосередньо до градієнтів моделі під час кожного кроку оптимізації, забезпечуючи конфіденційність на рівні тренування.

Ми сфокусуємося на розробці методів кодування (Encoding Techniques), які використовують або інтегрують такі методи спотворення. Ці вдосконалені підходи мають на меті досягти кращого балансу, пропонуючи вищу корисність для спотворених/закодованих зображень порівняно з існуючими базовими методами. Наш підхід передбачає дослідження методів кодування, що можуть належати до різних або переплетених категорій (наприклад, поєднання криптографічних принципів з обфускацією даних).

2.5. Методологія дослідження та архітектура моделей

У цьому дослідженні було задіяно три різні типи моделей. Дві з цих моделей інтегровані в процес кодування (Encoding Process), тоді як третя модель використовується для валідації та тестування закодованих зображень. Загальна структура цього підходу детально представлена у фазі навчання на рис. 1.1.

2.5.1. Архітектура ResNet-18 та парадигма перенесення навчання

Для розв'язання задачі розпізнавання ми застосували дві ключові моделі, позначені як закодована модель (Encoded Model) та оригінальна модель (Original Model) (рис. 1.1). Обидві моделі були розроблені з використанням парадигми перенесення навчання (Transfer Learning).

Перенесення навчання – це методологія в машинному навчанні, за якої модель, навчена для виконання одного завдання, повторно використовується як початкова точка або основа для навчання моделі на іншому, але пов'язаному завданні. Замість того, щоб розпочинати тренування цільової моделі з випадково ініціалізованих ваг, ми використовуємо попередньо

навчені ваги (pre-trained weights), які вже засвоїли корисні ієрархічні ознаки (hierarchical features) з великого вихідного набору даних.

В основі Transfer Learning лежить припущення, що знання, отримані моделлю при вирішенні однієї проблеми, можуть бути узагальнені та застосовані для прискорення навчання або підвищення продуктивності при вирішенні іншої, спорідненої проблеми.

Приклад. Модель, навчена класифікувати тисячі різних об'єктів (наприклад, на наборі ImageNet), засвоює базові візуальні ознаки (як-от краї, текстури та кути) у своїх ранніх шарах. Ці базові ознаки є універсальними і можуть бути безпосередньо корисними для нового завдання, наприклад, розпізнавання обличчя або медичної діагностики.

Існує три основні способи застосування перенесення навчання, особливо у контексті глибоких нейронних мереж, як-от ResNet, що показані в таблиці 2.1.

Таблиця 2.1.

Способи застосування перенесення навчання

Метод	**Опис**	**Коли застосовується**
1. Витягування Ознак	Використовується попередня модель як фіксований (заморожений) екстрактор ознак. Всі згорткові шари моделі заморожуються, і тренується лише новий класифікатор (наприклад, повнозв'язний шар) на витягнутих ознаках.	Малий цільовий набір даних та висока схожість завдань.
2. Тонке налаштування	Попередньо навчені ваги використовуються як ініціалізація, але деякі або всі шари моделі розморожуються і донавчаються (fine-tuned) на цільовому наборі даних, зазвичай з дуже низькою швидкістю навчання.	Великий цільовий набір даних та/або низька схожість завдань.
3. Попереднє навчання та тонке налаштування	Модель спочатку навчається на великому вихідному наборі даних (Pre-training), а потім проходить процес тонкого налаштування на цільовому наборі.	Стандартний підхід, особливо в задачах обробки природної мови (NLP) (наприклад, BERT, GPT).

Переваги:

- Значно зменшує необхідний обсяг цільових даних, оскільки модель вже вивчила багато загальних патернів.

- Модель сходиться значно швидше, оскільки вона починає з уже оптимізованих ваг.

- Допомагає запобігти перенавчанню на малих наборах даних та часто призводить до вищої кінцевої точності.

У даному дослідженні використовується саме метод витягування ознак (або часткове тонке налаштування):

- Використано попередньо навчену ResNet-18 (навчену на ImageNet).

- Заморожено всі згорткові шари (backbone), зберігаючи вивчені універсальні ознаки.

- Розморожено та виконано до навчання лише повнозв'язного шару, адаптуючи вихід моделі до специфічних завдань розпізнавання/кодування обличчя.

Призначення моделей:

- закодована модель - навчається на закодованих наборах даних зображень.

- оригінальна модель - навчається на незмінених, оригінальних наборах даних зображень.

В якості базової архітектури для обох моделей ми застосували попередньо навчену модель ResNet-18 [9], доступну через фреймворк PyTorch. Ця базова модель спочатку пройшла навчання на великому піднаборі датасету ImageNet, який охоплює 1.2 мільйона зображень, що належать до 1000 різних категорій.

Архітектура оригінальної мережі ResNet-18 зображена на рис. 2.4. Загалом мережа складається з вісімнадцяти шарів: 17 згорткових шарів, одного повнозв'язного шару та додаткового шару Softmax для виконання завдання класифікації.

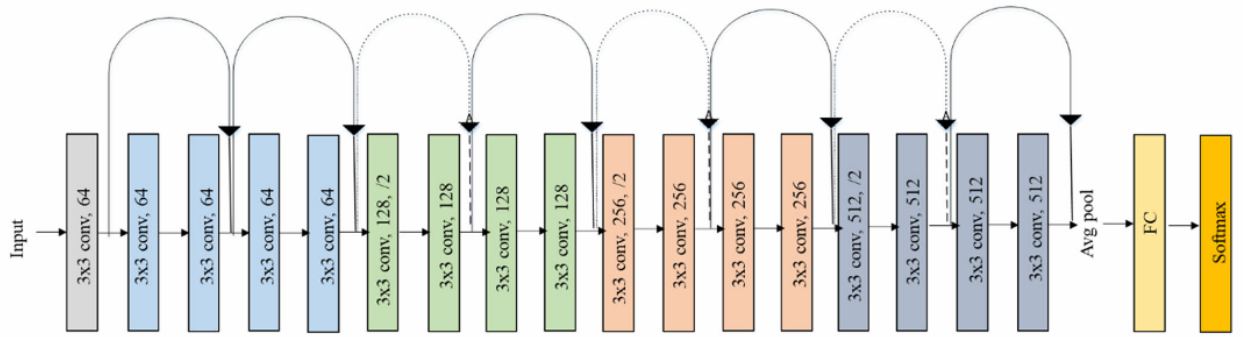

Рис. 2.4. Архітектура оригінальної мережі ResNet-18

Розглянемо компоненти та конфігурацію мережі.

1. Згорткові шари. У згорткових шарах використовуються фільтри розміром 3×3. Мережа спроектована таким чином, що шари мають однакову кількість фільтрів, якщо розмір вихідної карти ознак (feature map) залишається незмінним.

2. Зміна розмірності. Якщо розмір вихідної карти ознак зменшується вдвічі (halved), кількість фільтрів у наступних шарах подвоюється (doubled). Операція зменшення вибірки (downsampling) виконується згортковими шарами із кроком (stride) 2.

3. Вихідні шари. В кінці мережі послідовно застосовується усереднююче пулінгове згладжування (average-pooling), за яким іде повнозв'язний шар (fully-connected layer), завершений шаром Softmax.

У всій мережі між шарами вбудовані залишкові зв'язки (residual shortcut connections). Існує два типи цих зв'язків:

- зв'язки суцільною лінією (solid lines): використовуються, коли вхідні та вихідні розмірності мають однаковий розмір. Вони реалізують ідентичне відображення (identity mapping), просто передаючи вхід без змін.

- зв'язки пунктирною лінією (dotted lines): використовуються, коли розмірності збільшуються (наприклад, коли кількість фільтрів подвоюється, а просторові розміри зменшуються). Цей тип зв'язку також виконує ідентичне відображення, але включає доповнення нулями (zero padding) для узгодження

збільшених розмірностей та використовує крок 2 для зменшення просторової розмірності.

З метою стабілізації ознак (Feature Stability) протягом процесу навчання та забезпечення сталості витягування ознак з зображень, ми прийняли рішення заморозити (freeze) всі згорткові шари (convolutional layers) у базовій архітектурі ResNet-18. Був розморожений (unfrozen) лише повнозв'язний шар (fully connected layer). Це дозволило його вагам адаптуватися та навчатися відмінним патернам, присутнім у закодованих та оригінальних зображеннях, що є стандартною практикою для адаптації попередньо навчених мереж до нових, специфічних для задачі доменів.

2.5.2. Архітектура кодувальника (encoder) та процес кодування

Процес кодування передбачає використання згорткової нейронної мережі (Convolutional Neural Network, CNN), яка позначається як Кодувальник (Encoder).

Ця CNN має специфічну, фіксовану архітектуру, яка використовується виключно для витягування ознак (Feature Extraction) у рамках алгоритмів кодування. Важливо відзначити, що в мережі Кодувальника навчання (learning) відсутнє; її ваги залишаються статичними протягом усього процесу.

Наведемо деталі архітектури.

1. Перший згортковий шар (Convolutional Layer 1):

- кількість фільтрів: 32.

- Ініціалізатор ваг: використовується ініціалізатор Глорота рівномірного розподілу (Glorot Uniform Initializer) [18]. Цей ініціалізатор генерує ваги з рівномірного розподілу в діапазоні, спеціально визначеному для запобігання як зникаючим, так і вибуховим градієнтам. Це гарантує, що початкові ваги не є ані занадто великими, ані занадто малими, сприяючи ефективному кодуванню та збереженню максимально можливої інформації про оригінальні ознаки [2].

- Функція активації: використовується випрямлена лінійна одиниця (Rectified Linear Unit, ReLU).

- Розмір ядра: (7,7).

2. Шар максимального пулінгу (Max-Pooling Layer):

Шар максимального пулінгу є розміром пулу (pool size) 2×2. Цей шар призначений для зменшення просторової розмірності карти ознак, водночас зберігаючи найбільш значущі ознаки.

3. Другий згортковий шар (Convolutional Layer 2):

- Кількість фільтрів: 64.

- Ініціалізатор ваг: ініціалізатор Глорота рівномірного розподілу.

- Функція активації: ReLU.

- Розмір ядра: (5,5).

Ключовим елементом методології є те, що початкові ваги, ініціалізовані за допомогою Глорота рівномірного розподілу, використовуються без змін (fixed weights) протягом усього процесу кодування для всіх вхідних зображень. Це підкреслює, що кодувальник функціонує як детермінований трансформатор ознак (deterministic feature transformer), а не як навчальна модель.

2.5.3. Модель патчів для процесу кодування

У процесі кодування, паралельно з основним кодувальником, використовується модель патчів (Patch Model). Ця модель також є згортковою нейронною мережею (CNN), але відрізняється мінімалістичною архітектурою, що містить лише один згортковий шар (single convolutional layer).

Специфікації шару наступна:

- Шар оснащений 32 фільтрами, що ідентично кількості фільтрів у першому шарі кодувальника.

- Розмір ядра встановлено на рівні 5×5.

- Для ініціалізації ваг шару використовується ініціалізатор Глорота рівномірного розподілу (Glorot Uniform Initializer). Це забезпечує ефективне та збалансоване вилучення ознак (feature extraction) із даних зображення.

Ключова відмінність моделі патчів від типових CNN полягає у налаштуванні параметра кроку (stride). Розмір кроку встановлено рівним розміру ядра — 5×5.

Це специфічне проектне рішення має на меті емуляцію процесу розділення на патчі (patching process), що характерно для архітектур Vision Transformers (ViT) [15]. Встановлення кроку, рівного розміру ядра, призводить до неперекриваючого (non-overlapping) згорткового відображення. Це дозволяє ефективно обробляти вхідні зображення шляхом їхнього дискретного розкладання на локальні, незв'язані сегменти (patches), підвищуючи ефективність кодування на локальному рівні.

2.6. Опис набору даних

Labeled Faces in the Wild (LFW) — це один з найвідоміших і широко використовуваних наборів даних для дослідження проблеми розпізнавання обличчя (face recognition) у неконтрольованих умовах. Цей набір даних був зібраний у Принстонському університеті у 2007 році і відіграв ключову роль у просуванні алгоритмів розпізнавання обличчя.

LFW складається з понад 13 000 зображень обличчя, зібраних з Інтернету. На відміну від ранніх наборів даних, які часто містили зображення, зроблені в лабораторних умовах, LFW відображає реальні "дикі" умови. Зображення в LFW демонструють значну варіативність за такими параметрами:

- Поза обличчя (Pose): Різні кути повороту голови.
- Вираз обличчя (Expression): Усмішки, сум, нейтральні вирази.
- Освітлення (Illumination): Різні умови освітлення, тіні.
- Вік (Age): Зображення людей різного віку.

- Етнічна приналежність (Ethnicity): Широке представництво етнічних груп.

- Фон (Background): Різноманітні фонові зображення.

- Завади (Occlusions): Наявність окулярів, головних уборів, бороди, вусів, а в останні роки також масок.

Кожне зображення позначене ім'ям особи, що дозволяє порівнювати пари зображень і визначати, чи належать вони одній і тій же особі. Набір даних містить обличчя понад 5700 різних осіб.

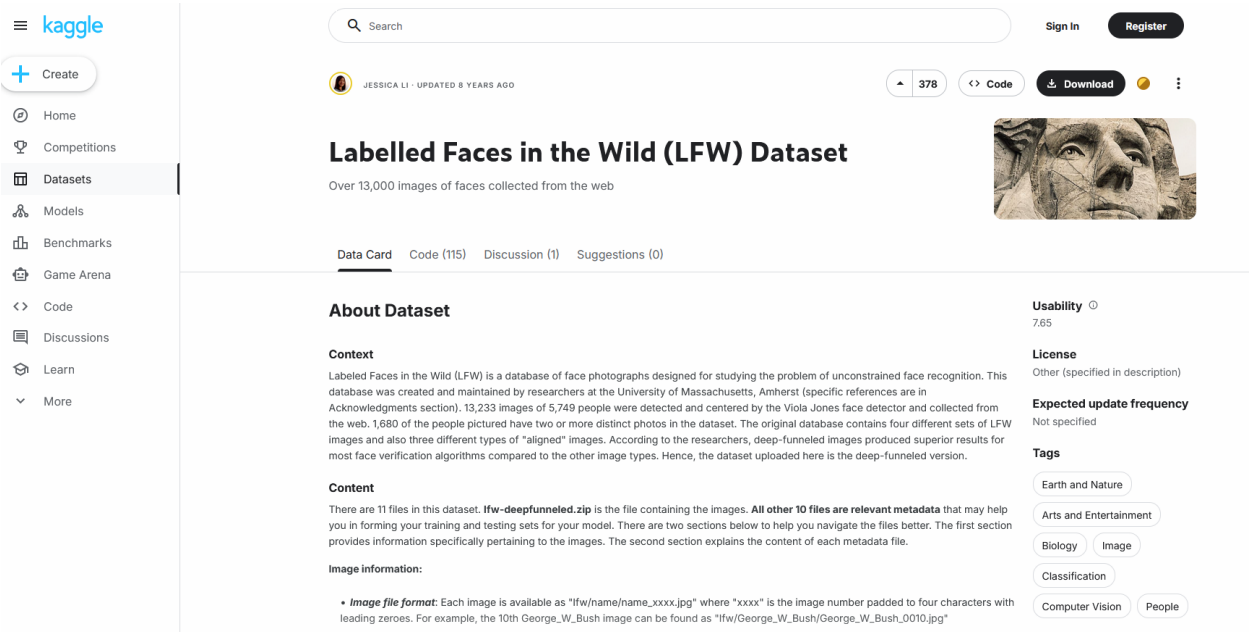

Рис. 2.5. Опис набору даних на платформі Kaggle

Ключовою особливістю LFW є те, що зображення були отримані "в диких умовах" (in the wild). Це означає, що вони не були зроблені в контрольованому середовищі студії. Такий підхід робить набір даних складним для алгоритмів розпізнавання, оскільки вони повинні бути стійкими до значних змін.

Основна мета LFW — надати стандартизований набір даних для оцінки алгоритмів верифікації обличчя (face verification). Завдання верифікації полягає у визначенні, чи дві надані фотографії належать одній і тій же особі (тобто, чи є вони "парою однієї особи" або "парою різних осіб").

Виклики, які ставить LFW перед алгоритмами:

- Великі внутрішньокласові варіації: Зображення однієї і тієї ж особи можуть значно відрізнятися через різні пози, освітлення, вирази тощо.

- Малі міжкласові варіації: Обличчя різних людей можуть бути схожими, що ускладнює їх розрізнення.

- Незбалансованість класів: Більшість осіб представлені лише одним зображенням, тоді як лише близько 1600 осіб мають два або більше зображень. Це створює труднощі для навчання.

Для оцінки продуктивності алгоритмів на LFW використовується стандартний протокол:

- 10-кратна перехресна валідація (10-fold cross-validation) - набір даних розділяється на 10 згортків. На кожній ітерації один згорток використовується для тестування, а решта дев'ять — для навчання.

- 6000 пар для тестування. Кожен згорток містить 600 пар зображень: 300 пар належать одній особі (positive pairs) і 300 пар належать різним особам (negative pairs). Загалом для оцінки використовується 6000 пар.

- Заборона "витоку" даних: Алгоритмам заборонено використовувати тестові зображення (навіть для попередньої обробки або вибору параметрів) під час навчання. Це забезпечує чесну оцінку.

LFW мав величезне значення для розвитку розпізнавання обличчя:

- Встановив золотий стандарт для оцінки алгоритмів у реальних умовах.

- Багато провідних алгоритмів, включно з методами на основі глибокого навчання (DeepFace від Facebook, FaceNet від Google), досягли проривних результатів саме на LFW. Спочатку точність була близько 60-70%, тоді як сучасні системи досягають понад 99% точності на цьому наборі даних.

- Допоміг дослідникам зосередитися на викликах, пов'язаних з реальним світом, що призвело до розробки більш надійних систем.

- Заохотив розробку методів, стійких до варіацій пози, освітлення та інших факторів, що є типовими для фотографій з Інтернету.

Незважаючи на високу точність сучасних алгоритмів на LFW, цей набір даних залишається важливою віхою в історії розпізнавання обличчя, демонструючи значний прогрес галузі.

Висновки до розділу

Другий розділ рисвячений дослідженню сучасних моделей та методів захисту приватності, що застосовуються у процесі обробки зображень облич у середовищі глибокого навчання. Проведений огляд криптографічних технік показав, що гомоморфне шифрування та безпечні багатосторонні обчислення забезпечують високий рівень формальної безпеки, однак є обчислювально складними для практичних застосувань. Окрему увагу було приділено методам спотворення зображень, які демонструють кращу ефективність за швидкодією та масштабованістю. Дослідження архітектури ResNet-18 дозволило визначити її як оптимальну базову модель для реалізації процесу розпізнавання у поєднанні з кодувальними методами. Аналіз архітектури кодувальника показав, що структурне подання ознак може суттєво впливати на здатність моделі зберігати семантичні характеристики зображення після кодування. Було визначено, що патчова модель кодування забезпечує більшу гнучкість щодо контролю рівня приватності.

РОЗДІЛ 3. ЗАСТОСУВАННЯ МОДЕЛЕЙ ТА МЕТОДІВ ЗАХИСТУ ПРИВАТНОСТІ ТА КОНФІДЕНЦІЙНОСТІ НА ОСНОВІ РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ ОБЛИЧ

3.1. Застосування методів кодування на основі спотворення зображень облич

У рамках категорії спотворення зображень (Image Distortion), обговореної у попередньому розділі, ми представляємо застосування методів кодування, які можуть бути класифіковані як інноваційні техніки спотворення [19]. Ці методи використовують архітектуру кодувальника, описану в попередніх розділах, для генерації закодованих зображень.

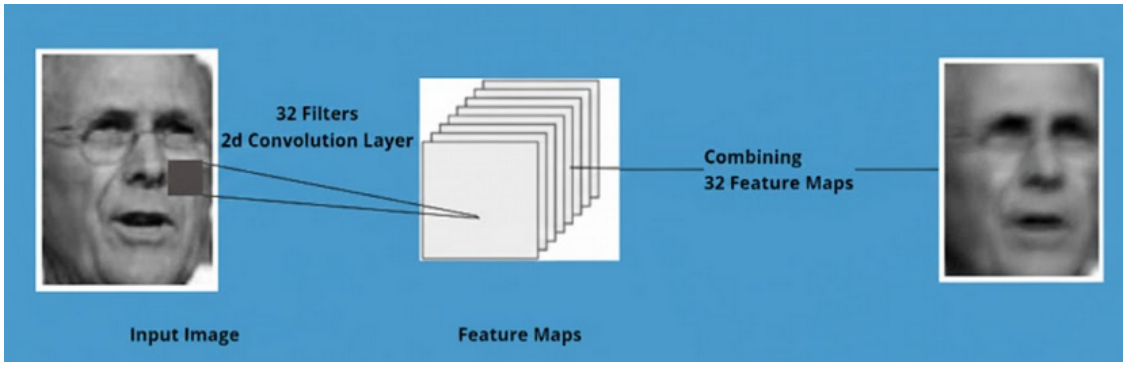

Рис. 3.1. Результат роботи методу кодування одною згорткою

3.1.1. Метод кодування одною згорткою

Метод кодування одною згорткою є першим у даній методології. Його реалізація включає наступні кроки:

1. Обробка оригінальних даних. Вихідне (оригінальне) зображення обробляється за допомогою мережі кодувальника.

2. Витяг ознак. Набір із 32 карт ознак (feature maps) витягується безпосередньо з першого згорткового шару кодувальника.

3. Композиція. Витягнуті 32 карти ознак об'єднуються (комбінуються) шляхом операції сумування (summation).

4. Результатом цієї операції є нове композитне зображення, яке ми позначаємо як нове кодоване зображення.

3.1.2. Метод кодування подвійною згорткою

Процес кодування подвійною згорткою починається аналогічно попередньому методу, але відрізняється глибиною витягування ознак.

1. Обробка оригінальних даних. Оригінальний набір даних також передається через мережу кодувальника.

2. Витяг ознак з глибинного шару: На відміну від попереднього методу, карти ознак витягуються з другого згорткового шару кодувальника.

3. Цей другий шар генерує загалом 64 карти ознак для кожного вхідного зображення.

4. Ці 64 карти ознак також об'єднуються (сумуються) для створення нового кодованого зображення.

Ці два методи дозволяють емпірично оцінити вплив глибини витягування ознак (поверхневі 32 ознаки проти більш глибоких 64 ознак) на компроміс конфіденційності та корисності при подальшому навчанні моделей глибокого навчання.

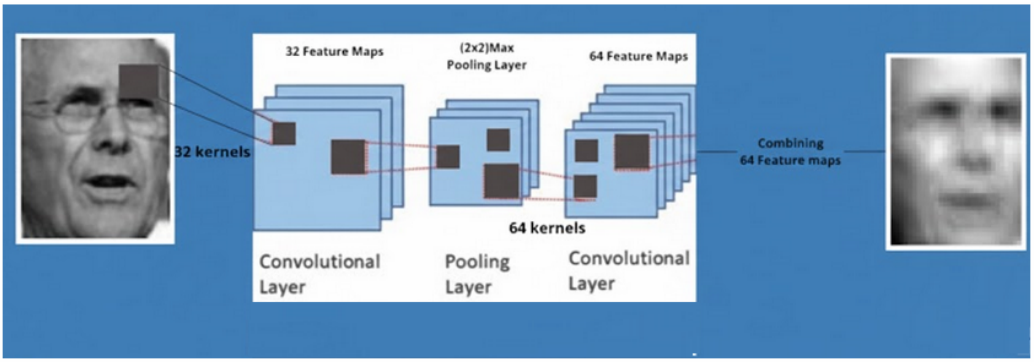

Рис. 3.2. Результат роботи методу кодування подвійною згорткою

3.1.3. Метод кодування на основі патчів

Цей метод використовує концепцію патчів, аналогічну тій, що застосовується у архітектурі Vision Transformer (ViT) [12]. Для генерації необхідних карт ознак використовується попередньо описана модель патчів.

Процедура кодування наступна:

- Процес розпочинається з збільшення розміру (upscaling) вихідного зображення приблизно в п'ять разів. Це масштабування виконується відповідно до розміру кроку (stride) моделі патчів, який встановлено на рівні 5×5.

- Масштабоване зображення подається на вхід до моделі патчів. Ключовим моментом є те, що ядро (kernel) моделі (розміром 5×5) сканує кожне окреме вікно пікселів рівно один раз. Оскільки крок встановлено рівним розміру ядра (обидва 5×5), ядро переміщується до наступної області без будь-якого перекриття з раніше обробленою ділянкою.

- Витяг та композиція ознак - w неперекривний підхід за допомогою 32 фільтрів (ядер) призводить до створення 32 окремих карт ознак. Далі ці 32 карти ознак об'єднуються (комбінуються) шляхом операції сумування (summation), аналогічно процедурам, використаним у попередніх методах кодування.

- Результатом є кодоване зображення, специфічне для патч-кодування. Цей метод забезпечує ефективну локальну обфускацію ознак, що є критичним для забезпечення конфіденційності.

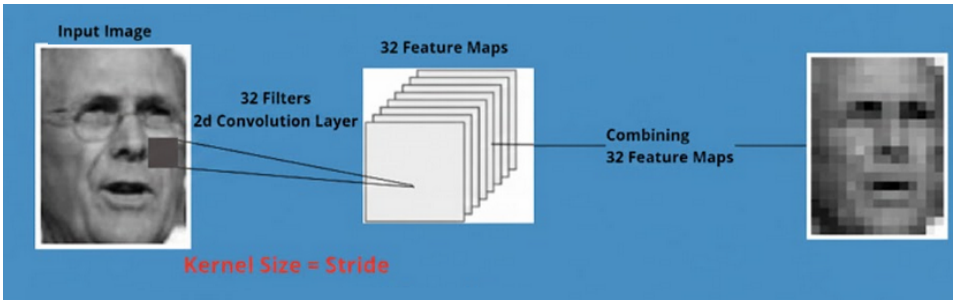

Рис. 3.3. Кодування зображення на основі патчів

3.2. Диференціальна конфіденційність як метод спотворення зображень

Диференціальна конфіденційність (Differential Privacy, DP) реалізується шляхом додавання шуму до вихідних даних. Основна ідея полягає в тому,

щоб спотворити індивідуальну ідентичність окремих точок даних, не порушуючи при цьому загальні статистичні виходи набору даних.

У контексті зображень це досягається шляхом генерації випадкових значень шуму з певного розподілу (як правило, Лапласа або Гаусса) та додавання їх до кожного пікселя зображення.

Для DP часто використовується Лапласовий розподіл, який залежить від двох параметрів: середнього значення (μ) та масштабу (b).

1. Середнє значення (μ). Встановлюється рівним нулю (μ=0). Це забезпечує симетричність розподілу навколо нуля, генеруючи як позитивні, так і негативні значення шуму. В результаті, кінцеве значення запиту (тобто, пікселя) залишається максимально стабілізованим близько до його оригінального значення.

2. Масштаб (b). Визначається двома ключовими термінами: чутливістю (Δf) та є.

У контексті зображень є можна розглядати як ступінь максимально прийнятної втрати конфіденційності. Чим менше значення є, тим більше шуму додається до даних, і тим вищий досягнутий рівень конфіденційності.

Чутливість функції (f) визначається як максимальна величина зміни, яка може статися у результаті аналізу через модифікацію (додавання/видалення) лише однієї точки даних у наборі.

$$\Delta f = \max_{D,D'} \| f(D) - f(D') \|$$

де D та D′ є сусідніми наборами даних (які відрізняються лише одним елементом), а $\| \cdot \|$ позначає відповідну норму (зазвичай, L1 або L2).

Якщо розглянути набір даних X={x1,x2,…,xn} (де xi — окремі пікселі), що використовуються для обчислення запиту (зображення), застосування DP призводить до створення нового набору даних X′={x1′,x2′,…,xn′}, де:

$$x_i' = x_i + \text{Laplacian}\left(\frac{\Delta f}{\epsilon}\right)$$

Це гарантує, що новий набір X′ матиме подібне середнє значення запиту, але з відмінними індивідуальними значеннями пікселів (xi ≠ xi′, ∀i=1…n).

У цій роботі:

- Всі пікселі розглядаються як окремі точки даних.

- Композиція пікселів (їхній візерунок), що формує зображення.

Оскільки використовувані зображення нормалізовані до діапазону [0,1], максимальна зміна, яка може відбутися між піксельними значеннями, дорівнює 1. Тому ми використовуємо константну чутливість Δf = 1.

Ми застосували постійну чутливість Δf=1 при варіюванні значення ϵ. Тут представлено результат використання ϵ=8, оскільки це значення забезпечило спостережувані результати для подальших технік кодування.

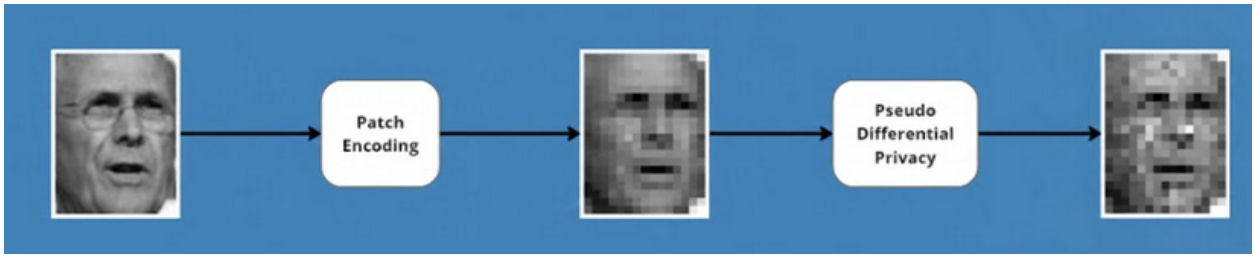

Рис. 3.4. Кодування зображення за допомогою гібридного методу

Для методу псевдо-DP для патчевих зображень кодування ми комбінуємо дві раніше описані техніки.

1. Початкове патчування. Спочатку оригінальний набір даних проходить процедуру патчування шляхом передачі зображень через нашу модель патчів.

Застосування шуму DP. Після отримання патчевого зображення, до кожного пікселя цього зображення додається значення шуму, випадково

обране з Лапласового розподілу з масштабом, визначеним у методі псевдо-DP (з використанням встановлених Δf та є).

Результат роботи для цього гібридного кодування показаний на рис. 3.4.

3.3. Динамічна диференціальна конфіденційність та комбіновані методи кодування зображень облич

3.3.1. Диференціальна конфіденційність з динамічною чутливістю

Відмінність між методом псевдо-DP та динамічним застосуванням диференціальної конфіденційності (DP) полягає у методі визначення чутливості (Δf). На відміну від псевдо-DP, де використовується константна чутливість, тут чутливість обчислюється динамічно на основі локальних варіацій даних.

Для зображення розміром m×n, де і і j позначають попередній та поточний ряд відповідно (довжиною n), а Vi та Vj є векторними представленнями цих рядків:

1. Динамічна чутливість рядка (Sj) для j-го рядка визначається як Евклідова відстань (L2 норма) між вектором поточного рядка (Vj) та вектором попереднього рядка (Vi).

$$S_j = \|V^i - V^j\|_2 = \sqrt{\sum_{k=1}^{n} (V_k^i - V_k^j)^2}$$

2. Обчислене значення Sj використовується як чутливість для визначення масштабу Лапласового розподілу для цього конкретного рядка. Отримавши масштаб, генерується випадкове значення шуму з цього розподілу і додається до кожного пікселя j-го рядка.

Цей процес повторюється для кожного рядка, забезпечуючи різний масштаб Лапласового розподілу (і, відповідно, різні значення шуму) для

кожного рядка зображення. Така залежність шуму від локальної варіації даних може забезпечити кращий компроміс між конфіденційністю та корисністю, оскільки шум додається пропорційно локальній складності ознак.

3.3.2. Диференціальна конфіденційність для гібридного кодування зображень

Цей метод є гібридним кодуванням, що поєднує патч-кодування з диференціальною конфіденційністю (DP) з динамічною чутливістю.

1. Спочатку оригінальні зображення обробляються за допомогою моделі патчів.

2. Отримані патч-закодовані зображення стають вхідними даними для алгоритму динамічної DP.

3. Чутливість обчислюється для кожного рядка патчевого зображення, після чого генерується та додається відповідний випадковий шум до всіх пікселів у цьому рядку.

Робочий процес для цього гібридного кодування детально представлений на рис. 3.5.

Рис. 3.5. Робочий процес для цього гібридного кодування

3.3.3. Диференціальна конфіденційність для кодування одною згорткою

Цей метод також є конкатенацією двох технік кодування:

- Кодування одною згорткою: спочатку зображення кодуються за допомогою методу кодування одною згорткою.

- Застосування DP: отримані закодовані зображення потім передаються через описаний вище Алгоритм Диференціальної Конфіденційності (з динамічним розрахунком чутливості).

Ця гібридизація дозволяє досліджувати, як поєднання трансформації ознак (Single Conv Encoding) з криптографічно обґрунтованою обфускацією (DP) впливає на досягнення цілей конфіденційності.

Приклади результатів, отриманих за допомогою різних технік кодування, представлені на рис. 3.6.

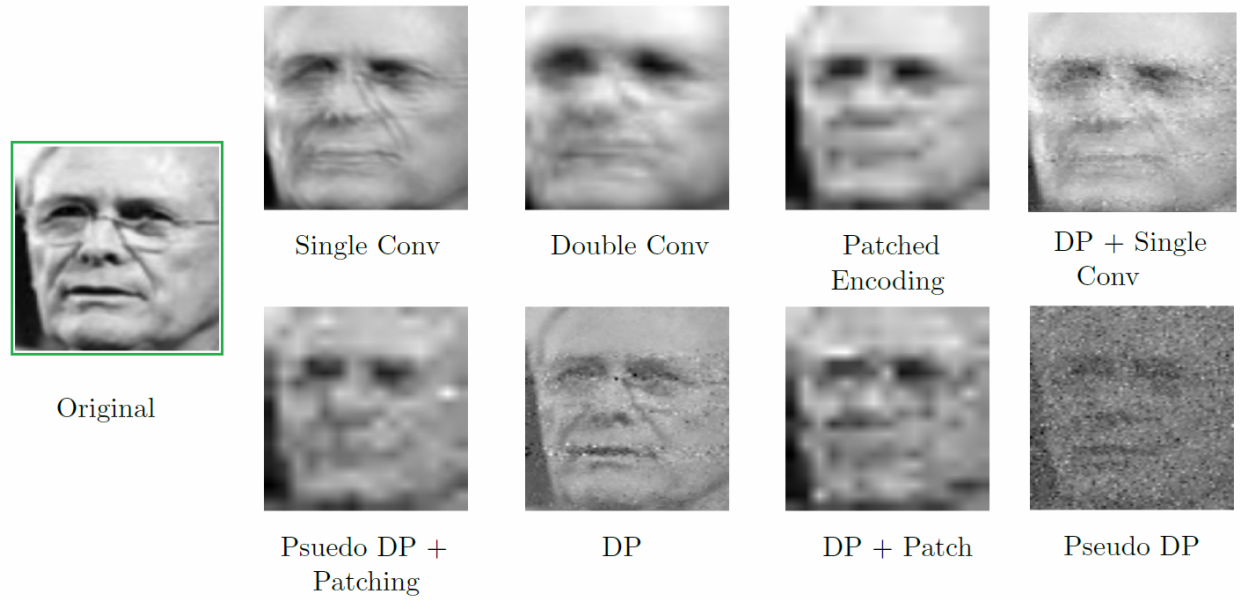

Рис. 3.6. Результати кодування зображень обличчя різними техніками і методами

3.4. Опис наборів даних як емпіричної бази даних

Для проведення цього дослідження було використано набір даних Labeled Faces in the Wild (LFW).

Набір даних LFW складається з неконтрольованих зображень облич (uncontrolled facial images), отриманих у реальних умовах, з вихідною роздільною здатністю 250×250 пікселів. Він охоплює широкий спектр варіацій поз, освітлення та виразів. Загальний набір даних містить 13233 зображення, які належать до 5749 унікальних осіб.

Оскільки оригінальний набір даних є незбалансованим (кількість прикладів на одну особу значно варіюється), ми застосували фільтр для

підвищення надійності навчання. Для формування кінцевого набору були відібрані лише ті особи, які мали щонайменше 40 зображень.

Таблиця 3.1.

Розподіл набору даних

Параметр	Значення
Відфільтрована кількість зображень	1867
Кількість унікальних осіб	19
Розподіл для навчання (Training Set)	1493 зображення
Розподіл для тестування (Testing Set)	374 зображення

Синтетичний замаскований набір даних LFW (Synthetic Masked LFW Dataset) - це набір даних є модифікованою версією набору LFW. До зображень, відібраних згідно з критерієм мінімум 40 зображень на особу, було синтетично застосовано маски на обличчя.

Він містить ідентичну кількість осіб (19) та таке саме розділення на фази навчання та тестування, як і немодифікований відфільтрований набір LFW. Єдина відмінність полягає у наявності масок на всіх обличчях, що моделює сценарії, де обличчя частково приховані.

Рис. 3.7. Зображення фото з набору даних, оригінальне з посмішкою і модифіковане з маскою

3.5. Експериментальна валідація методів кодування зображень обличчя

Всі оригінальні зображення, відібрані з набору даних, були оброблені за допомогою всіх методів кодування, представлених у розділі 2. Отримані

закодовані зображення були використані для емпіричної перевірки їхньої здатності вирішувати три ключові проблеми конфіденційності та корисності, що розглядаються у цьому дослідженні.

3.5.1. Збереження оригінальної ідентичності

Основною метою застосування методів кодування було анонімізувати оригінальну ідентичність особи на зображенні. Для оцінки ефективності обфускації оригінальної ідентичності було використано оригінальну модель, навчену на некодованих даних.

Процедура наступна: всі зображення з набору даних LFW були закодовані за допомогою кожного методу. Потім оригінальна модель використовувалася для розпізнавання осіб у цьому закодованому тестовому наборі.

Результати подано в таблиці 3.2 колонка "Original vs Encoded":

- оригінальна модель демонструє точність розпізнавання 0.79 на оригінальному (некодованому) наборі даних.

- Після застосування кодування, для всіх технік спостерігається майже повна деградація точності розпізнавання. Це свідчить про нездатність оригінальної моделі розпізнати ідентичність осіб у закодованому наборі.

Отже, інформація про оригінальну ідентичність була успішно вилучена або суттєво спотворена. Модель типово сприймає закодовану версію особи як абсолютно іншого індивіда. Ці результати підтверджують, що методи кодування успішно зберігають конфіденційність оригінальних ідентичностей осіб.

3.5.2. Розпізнавання анонімної ідентичності

Цей експеримент був спрямований на перевірку вирішення другої проблеми: чи може анонімна нова ідентичність, надана кодуванням, бути ідентифікована та відстежена як окрема особа.

Ми використали набір даних LFW, застосувавши до нього всі техніки кодування. Потім була використана закодована модель, яка була навчена виключно на закодованому тренувальному наборі, для розпізнавання ідентичностей у закодованому тестовому наборі.

Таблиця 3.2.

Результати точності різних методів кодування для розпізнавання ідентичності

Техніка Кодування	**Original vs Encoded Test Acc (приватність)**	**Encoded vs Encoded Test Acc (корисність)**
Original	0.79	0.79
Single Conv	0.24	0.72
Double Conv	0.07	0.68
Patch Encoding	0.09	0.64
Pseudo DP	0.40	0.49
Pseudo DP+ Patch	0.10	0.48
DP	0.30	0.70
DP+Patch	0.10	0.55
DP+Single Conv	0.05	0.64

Результати подані в таблиці 3.2, колонка "Encoded vs Encoded":

- закодована модель демонструє здатність розпізнавати ідентичності, присутні у закодованому тестовому наборі, за умови, що вона навчена на закодованих даних.

- Хоча значення точності дещо знизилися для всіх технік порівняно з оригінальною точністю (0.79), це є очікуваним наслідком компромісу між конфіденційністю даних та корисністю (Privacy-Utility Trade-off). Частина корисності даних була свідомо пожертвувана заради досягнення конфіденційності. Кожна техніка кодування пропонує різний діапазон компромісів, надаючи варіанти вибору залежно від необхідного рівня конфіденційності.

Таблиця 3.3.

Аналіз компромісів та ефективності

Техніка кодування	**Original vs Encoded (приватність)**	**Encoded vs Encoded (корисність)**	**Висновок**
Original	0.79	0.79	Базова точність (відсутність приватності)
Dynamic DP	0.30	0.70	Найкраща корисність серед чистих DP, але низька приватність (вищий витік 0.30)
Pseudo DP	0.40	0.49	Високий витік приватності (0.40)
Single Conv	0.24	0.72	Хороша корисність, але вищий витік приватності (0.24)
DP + Single Conv	0.05	0.64	Найкращий баланс: Висока приватність (0.05), прийнятна корисність
Double Conv	0.07	0.68	Відмінний баланс: Висока приватність (0.07), хороша корисність

Компромісні випадки техніки Dynamic DP (0.30/0.70) та Pseudo DP (0.40/0.49) продемонстрували вищу точність у закодованій моделі (корисність), але значно вищий рівень витоку конфіденційності (Original vs Encoded: 0.30 та 0.40).

Оптимальні випадки (висока приватність): найкращий рівень анонімізації оригінальної ідентичності досягнуто при DP + Single Conv (0.05 витік) та Double Conv (0.07 витік).

DP + Single Conv показав виняткову консервацію приватності (0.05 витік), зберігаючи при цьому прийнятний рівень розпізнавання анонімної ідентичності (0.64).

Double Convolution Encoding також продемонстрував високі рівні збереження приватності (0.07 витік) з хорошою точністю розпізнавання закодованої ідентичності (0.68).

Отже, отримані результати переконливо доводять, що нова або модифікована ідентичність, надана методами кодування, може бути успішно

розпізнана та відстежена (закодованою моделлю). Це забезпечує надійний механізм для моніторингу індивідів у чутливих сценаріях (наприклад, охорона здоров'я) без ризику крадіжки або зловживання даними, що відображають їхню оригінальну ідентичність.

3.6. Навчання вторинних ознак

Цей експеримент був розроблений для визначення, чи здатні методи кодування зберігати необхідну інформацію для виконання моделями цільового завдання, водночас успішно анонімізуючи ідентичність осіб у наборі даних. Ми провели два типи експериментів, які розрізнялися за визначенням цільового завдання, що його мала засвоїти модель.

3.6.1. Детекція медичної маски на обличчі

Цей сценарій моделює ситуацію, коли модель має навчитися визначати, чи надягнена маска на обличчя. Це завдання вимагає використання даних обличчя для тренування моделей глибокого навчання.

Як набір даних ми використали синтетично створений замаскований набір даних LFW.

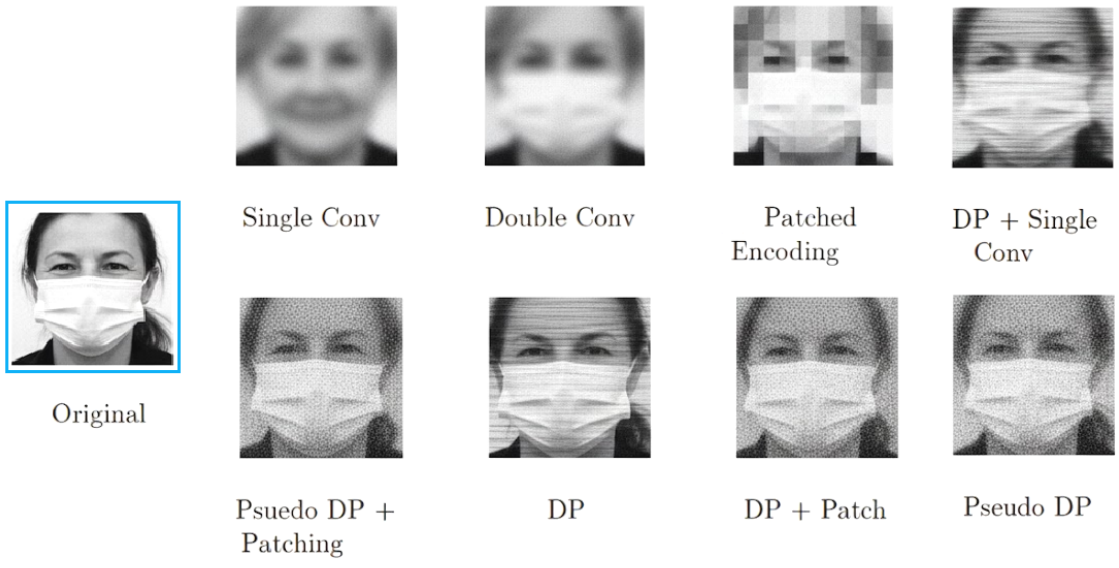

Рис. 3.8. Приклади кодування зображень обличчя в медичній масці

Замість оригінальних зображень, що містять неспотворені обличчя, весь набір даних був закодований відповідно до представлених технік. Ці закодовані дані використовувалися для тренування моделей глибокого навчання з метою бінарної класифікації (маска/немає маски). Приклади застосування кодування до замаскованого набору даних можна побачити на рис. 3.8.

Після тренування моделі на закодованому наборі даних (аналогічно закодованій моделі), ми оцінювали точність класифікації на закодованому тестовому наборі. Результати представлені у таблиці 3.4, у колонці "Mask Detection".

Таблиця 3.4.

Результати точності навчання вторинних ознак (маска)

Тип Кодування	**Детекція маски / (Mask Detection)**
Original Data	0.99
Single Conv	0.99
Double Conv	0.98
Patch Encoding	0.95
Pseudo DP	0.99
Pseudo DP+ Patch	0.96
DP	0.84
DP+Patch	0.88
DP+Single Conv	0.99

Модель, навчена на оригінальних некодованих даних (синтетичний LFW), досягла точності 0.99.

Після кодування майже всі техніки забезпечили високі рівні точності. Зокрема, методи, що продемонстрували найкращий компроміс конфіденційності та корисності, а саме кодування подвійною згорткою та диференціальна конфіденційність з кодуванням одною згорткою, підтримували високу точність детекції маски: 0.98 та 0.99 відповідно.

Ці результати підтверджують, що методи кодування зберігають необхідну інформацію для тренування моделі, водночас забезпечуючи конфіденційність, захищаючи чутливу інформацію від несанкціонованого використання.

3.6.2. Детекція виразу обличчя

Другий експеримент передбачав тренування моделі для визначення, чи посміхається особа на зображенні.

Набір даних: ми використали набір даних LFW. Атрибут "Посмішка" ("Smile") був обраний через його потенційну актуальність у медичному спостереженні (відсутність посмішки може свідчити про дискомфорт пацієнта).

Процедура наступна: всі зображення були закодовані (приклад на рис. 3.9) і використані для тренування та тестування моделі глибокого навчання.

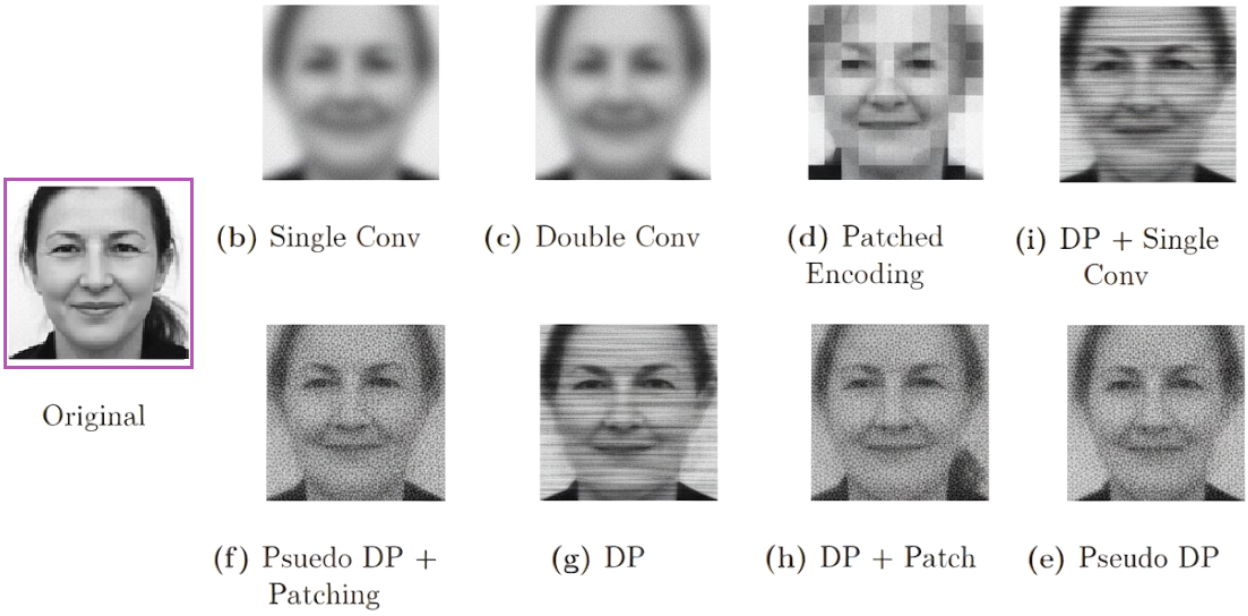

Рис. 3.9. Приклади закодованих зображень для процесу детекції посмішки на обличчі

Модель, навчена на оригінальних некодованих даних, досягла точності 0.94. Моделі, навчені на закодованих даних, продемонстрували високу

точність у засвоєнні завдання детекції посмішки. Точність була подібною для більшості технік, коливаючись від найкращого результату 0.92 до найгіршого 0.86.

Таблиця 3.5.

Результати точності навчання вторинних ознак (посмішка)

Тип кодування	**Детекція посмішки / (Smile Detection)**
Original Data	0.94
Single Conv	0.92
Double Conv	0.92
Patch Encoding	0.91
Pseudo DP	0.91
Pseudo DP+ Patch	0.91
DP	0.89
DP+Patch	0.86
DP+Single Conv	0.91

Найкращі техніки з точки зору компромісу конфіденційності та корисності — кодування подвійною згорткою та диференціальна конфіденційність з кодуванням одною згорткою — показали точність розпізнавання вторинних ознак 0.92 та 0.91 відповідно, що показано в таблиці 3.5.

Отже, ці результати демонструють значний потенціал розроблених алгоритмів: вони успішно зберігають конфіденційність і водночас зберігають необхідну корисність даних для виконання різноманітних вторинних завдань класифікації.

Техніки, що забезпечили найкращий баланс (Double Convolution Encoding та DP+Single Convolution Encoding), надають індивідам можливість ділитися своїми даними для тренування моделей без побоювання втрати приватності, тим самим вирішуючи всі проблеми, окреслені на початку дослідження.

3.7. Комплексне рішення викликів конфіденційності у глибокому навчанні

Широке впровадження великомасштабних моделей глибокого навчання (ВМГН) у різних доменах створило нагальну потребу в специфічних для галузі даних для забезпечення їхньої ефективної роботи. Однак такі набори даних часто містять чутливу персональну інформацію, що вимагає забезпечення суворої конфіденційності при обміні та обробці третіми сторонами.

Для вирішення цієї критичної проблеми, наша робота пропонує новітні методи кодування зображень. Ці методи спрямовані на збереження конфіденційності індивідуальних суб'єктів, водночас надаючи критично важливі дані, необхідні для успішного тренування ВМГН.

Наші емпіричні результати демонструють, що впровадження цих заходів конфіденційності призводить лише до незначного зниження точності розпізнавання. Важливо, що ми спостерігаємо, як компроміс між корисністю та конфіденційністю (Privacy-Utility Trade-off) може бути мінімізований завдяки цим методам кодування. Це є значним досягненням, особливо у порівнянні з попередніми дослідженнями, пов'язаними з даними зображень обличчя, що зберігають конфіденційність, де навіть такі методи, як диференціальна конфіденційність, часто ставали вузьким місцем через значну втрату корисності [16].

Запропоновані методи є особливо цінними у медичних застосуваннях, які вимагають постійного моніторингу. Наприклад, для контролю дотримання медичним персоналом протоколів (як-от миття рук, носіння масок). Використання кодування усуває проблеми конфіденційності для персоналу під наглядом.

Хоча оригінальна ідентичність (primary identity) персоналу залишається прихованою, їхні дії можуть бути ефективно відстежені через вторинну, анонімну ідентичність.

Аналогічно, безперервне спостереження за пацієнтами для моніторингу ознак дискомфорту (наприклад, больові або інші вирази обличчя) може здійснюватися без ризику витоку особистих ідентифікаційних даних для нецільових застосувань.

Для подальшого розвитку цієї сфери ми пропонуємо розширити застосування методів кодування з даних зображень обличчя на загальні медичні дані зображень.

Особиста ідентичність пацієнта виходить за рамки обличчя і включає інші медичні візуальні дані, такі як рентгенівські знімки, КТ або МРТ-зображення. Багато установ та індивідів неохоче діляться цими важливими даними для навчання моделей через побоювання щодо конфіденційності.

Наш метод, демонструючи потенціал мінімального компромісу між конфіденційністю та корисністю, може забезпечити збереження ідентичності пацієнта, значно збагачуючи доступність медичних даних для тренування ВМГН.

Висновки до розділу

У третьому розділі було реалізовано та експериментально перевірено комплекс методів, спрямованих на забезпечення приватності у системах розпізнавання облич. Було встановлено, що методи одно- та двозгорткового кодування забезпечують стійкість до реконструкції облич, при цьому зберігаючи прийнятний рівень точності розпізнавання. Патчовий підхід продемонстрував найкращий баланс між приватністю та коректністю класифікації, що свідчить про його перспективність для практичних застосувань. Впровадження диференціальної конфіденційності дозволило підвищити захищеність зображень від атак інверсії, не спричинивши значного погіршення характеристик моделі. Було запропоновано комбіновані методи, які адаптивно регулюють рівень спотворення залежно від чутливості ознак. Експериментальна частина підтвердила, що запропоновані моделі

зберігають здатність системи відтворювати ідентичність у зашифрованому просторі ознак. Дослідження також засвідчило, що зашифровані зображення можуть бути використані для навчання вторинних задач, включаючи детекцію медичної маски та розпізнавання емоцій. Це демонструє універсальність запропонованих методів та їх придатність для багатофункціональних систем. Було показано, що комбіновані моделі забезпечують найбільш стійке співвідношення між приватністю та точністю розпізнавання.

ВИСНОВКИ

У магістерській роботі проведено дослідження теоретичних, методологічних та практичних аспектів захисту приватності у системах, які базуються на технологіях розпізнавання облич. Актуальність проблематики зумовлена широким використанням біометричних систем у сферах безпеки, моніторингу та персоналізованих цифрових сервісів, що супроводжується зростанням ризиків витоку персональних даних та можливих атак на конфіденційні біометричні ознаки. У роботі виконано аналіз сучасного стану досліджень, систематизовано моделі загроз, сформовано методологію кодування зображень облич для забезпечення захищеного розпізнавання та проведено експериментальну перевірку ефективності запропонованих підходів.

У першому розділі здійснено детальний огляд предметної області та ідентифіковано ключові ризики, що виникають під час використання глибоких нейронних мереж для задач розпізнавання та класифікації облич. Показано, що традиційні методи знеособлення або спотворення зображень не забезпечують достатнього рівня стійкості до відновлення вихідних біометричних даних, а також суттєво погіршують ефективність розпізнавання. Визначено основні проблемні сценарії, зокрема несанкціонований доступ до проміжних ознак, можливість реконструкції облич на основі латентних векторів, атак з інверсією моделей та витік персональних даних у хмарних системах.

Другий розділ присвячено аналізу криптографічних, математичних та інженерних рішень, що застосовуються для забезпечення конфіденційності зображень у системах глибокого навчання. Виконано огляд гомоморфного шифрування, безпечних багатосторонніх обчислень та методів криптографічного захисту, доведено їх переваги щодо формальної стійкості, але окреслено їх обмеження у реальних застосуваннях через високу обчислювальну складність та низьку ефективність для великих

мультимедійних даних. Розглянуто моделі спотворення зображень як альтернативний підхід, що забезпечує баланс між продуктивністю та рівнем приватності. Детально описано архітектури ResNet-18, encoder-моделі та патчові методи кодування, які стали основою для реалізації та тестування розробленої системи. Набори даних були підготовлені з урахуванням вимог до варіативності, репрезентативності та відповідності реальним сценаріям використання систем розпізнавання.

У третьому розділі проведено практичну реалізацію та експериментальну перевірку методів кодування зображень облич, включаючи одно- та двозгорткове кодування, патчове кодування та модифіковані підходи на основі диференціальної конфіденційності. Розроблено та протестовано механізми додавання шуму з урахуванням чутливості ознак, що дозволило підвищити стійкість до реконструкційних атак при мінімальному впливі на точність моделі. Запропоновано комбіновані методи, які адаптивно поєднують диференціальну конфіденційність з архітектурно-орієнтованими методами кодування, забезпечуючи гнучке налаштування рівня приватності залежно від умов експлуатації.

Експериментальна частина включала оцінювання двох ключових показників: збереження здатності до розпізнавання оригінальної ідентичності та забезпечення коректної класифікації анонімізованих ознак. Результати показали, що патчові та гібридні методи забезпечують оптимальний компроміс між точністю та рівнем приватності, перевершуючи традиційні методи спотворення зображень. Додатково проведено експерименти з вторинним навчанням, зокрема для задач детекції медичних масок та визначення виразу облич.

У підсумку, у роботі запропоновано комплексний підхід до захисту приватності у системах розпізнавання облич, який поєднує глибоке кодування, методи диференціальної конфіденційності та моделі патчового перетворення зображень. Розроблені моделі забезпечують ефективне

зниження ризику відновлення персональних біометричних ознак, зберігаючи при цьому високу точність розпізнавання та сумісність із сучасними архітектурами нейронних мереж. Отримані результати можуть бути використані для проєктування безпечних систем біометричної ідентифікації, побудови конфіденційних сервісів на основі комп’ютерного зору та подальших досліджень у напрямі забезпечення приватності в умовах широкого застосування штучного інтелекту.

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Hybrid Siamese Network for Unconstrained Face Verification and Clustering under Limited Resources MDPI. Big Data and Cognitive Computing (BDCC). August 2024(3):19. – DOI:10.3390/bdcc4030019. – https://www.researchgate.net/publication/343487727_Hybrid_Siamese_Network_for_Unconstrained_Face_Verification_and_Clustering_under_Limited_Resources
2. AIAAIC - Labeled Faces in the Wild (LFW) dataset - https://www.aiaaic.org/aiaaic-repository/ai-algorithmic-and-automation-incidents/labeled-faces-in-the-wild-lfw-dataset
3. Labelled Faces in the Wild (LFW) Dataset . - https://www.kaggle.com/datasets/jessicali9530/lfw-dataset/data?select=people.csv
4. 5.6.4. The Labeled Faces in the Wild face recognition dataset — scikit-learn 0.19.2 documentation. - https://scikit-learn.org/0.19/datasets/labeled_faces.html
5. Labeled Faces in the Wild: A Database forStudying Face Recognition in Unconstrained Environments - Huang_long_eccv2008-lfw.pdf. - https://inria.hal.science/inria-00321923/document
6. Torchscript support — Torchvision main documentation. - https://docs.pytorch.org/vision/main/auto_examples/others/plot_scripted_tensor_transforms.html#sphx-glr-auto-examples-others-plot-scripted-tensor-transforms-py
7. (PDF) A Deep Learning Approach for Automated Diagnosis and Multi-Class Classification of Alzheimer’s Disease Stages Using Resting-State fMRI and Residual Neural Networks. - https://www.researchgate.net/publication/336642248_A_Deep_Learning_Approach_for_Automated_Diagnosis_and_Multi-

Class_Classification_of_Alzheimer's_Disease_Stages_Using_Resting-State_fMRI_and_Residual_Neural_Networks

8. (PDF) Privacy-preserving Artificial Intelligence Techniques in Biomedicine. - https://www.researchgate.net/publication/343179246_Privacy-preserving_Artificial_Intelligence_Techniques_in_Biomedicine
9. What is Fully Homomorphic Encryption - https://www.linkedin.com/pulse/what-fully-homomorphic-encryption-gregory-boland-wi3ue
10. Smith, J., & Carter, L. Design Patterns for Modern Information Systems. Journal of Systems Engineering, London: Springer, 2020, pp. 45–59.
11. Johnson, M. A Comparative Study of Web Application Architectures. International Journal of Computer Science Research, New York: IEEE Press, 2021, pp. 112–130.
12. Brown, K., & Wilson, P. Data Management Approaches in Enterprise Systems. Information Processing Journal, Berlin: Elsevier, 2019, pp. 78–94.
13. Taylor, R. Agile Practices in Large-Scale Software Projects. Proceedings of the Agile Development Conference, San Francisco: ACM, 2020, pp. 22–34.
14. Lopez, D., & Martin, S. Database Optimization Techniques for High-Load Applications. Journal of Database Science, Amsterdam: Elsevier, 2022, pp. 51–69.
15. Evans, A. Security Challenges in Cloud-Based Management Systems. Cybersecurity Review, Chicago: IEEE Press, 2019, pp. 140–158.
16. Green, T., & Baker, H. Modeling Approaches in Information System Design. International Journal of Information Modelling, London: Springer, 2021, pp. 201–220.
17. Miller, C. Automated Testing Frameworks for Mobile Applications. Proceedings of the International Mobile Computing Conference, Tokyo: IEEE, 2022, pp. 90–105.
18. Anderson, P. Improving User Interaction in Management Systems. Human–Computer Interaction Journal, Boston: MIT Press, 2018, pp. 31–47.

19. Walker, S., & Hughes, B. Software Architecture Selection for enterprise solutions. Journal of Software Engineering, New York: IEEE Press, 2021, pp. 165–184.
20. Collins, J. Integrating RESTful APIs into Business Applications. Web Technologies Journal, London: Taylor & Francis, 2022, pp. 54–70.
21. Schneider, F. Challenges of Data Consistency in Distributed Systems. Distributed Computing Review, Berlin: Springer, 2020, pp. 98–113.
22. Harris, A., & Gilbert, R. Project Management Tools for IT Projects. International Project Management Journal, Paris: Elsevier, 2019, pp. 63–82.
23. Miller, V. Scrum-Based Methodology for Information System Development. Software Development Conference Proceedings, Seattle: ACM, 2021, pp. 124–139.
24. Roberts, N., & Parker, D. Monitoring Tools in Cloud Management Solutions. Cloud Computing Advances Journal, New York: IEEE, 2020, pp. 77–92.
25. Foster, B. Web Application Performance Optimization Techniques. Web Engineering Review, Stockholm: Springer, 2022, pp. 33–50.
26. O'Neill, J. Data Analytics in Modern Information Systems. Data Science Journal, Boston: Elsevier, 2021, pp. 143–161.
27. Hudson, M., & Clark, A. Microservices Architecture in Enterprise Software. Microservices and Cloud Conference Proceedings, London: ACM, 2020, pp. 85–101.
28. Davis, L. System Integration Approaches for Corporate Applications. Systems Integration Journal, Amsterdam: Springer, 2019, pp. 203–219.
29. Kim, S. User-Centered Design of Web Interfaces. International Journal of UX Research, Toronto: IEEE Press, 2021, pp. 12–29.
30. Patel, R. Blockchain-Based Approaches in Business Information Systems. Journal of Emerging Technologies, Berlin: Springer, 2022, pp. 76–93.

31. Newman, T., & Ford, S. Automated Workflows in Enterprise Resource Planning Systems. ERP Systems Review, New York: Elsevier, 2020, pp. 45–61.
32. White, J. Artificial Intelligence Methods in Business Process Automation. AI Applications Journal, Boston: MIT Press, 2021, pp. 98–116.
33. Romero, K. Cloud-Native Development: Best Practices and Frameworks. Cloud Technologies Proceedings, San Diego: IEEE, 2019, pp. 142–159.
34. Lee, H. Big Data Solutions for Enterprise Management Systems. Big Data Analysis Journal, London: Springer, 2021, pp. 190–209.
35. Peterson, D. Security Risks in Web-Based Information Systems. Journal of Cybersecurity Technology, Chicago: Elsevier, 2022, pp. 35–53.
36. Bradley, S. Automated Testing in Web Application Development. Software Quality Assurance Review, Berlin: Springer, 2020, pp. 77–91.
37. Gomez, L. API Integration Methods for Scalable Systems. Conference on Distributed Software Architecture, Lisbon: ACM, 2021, pp. 119–134.
38. Robertson, P. Designing Reliable Database Structures. Journal of Information Storage, Amsterdam: Elsevier, 2019, pp. 101–118.
39. Hall, E., & Sanders, W. Containerization in Enterprise Software Development. DevOps and Cloud Native Conference Proceedings, Austin: IEEE, 2021, pp. 64–81.
40. Morgan, D. Service-Oriented Architecture for Corporate Solutions. SOA Journal, London: Taylor & Francis, 2020, pp. 170–187.
41. Turner, F. Quality Assurance Approaches in Software Engineering. Journal of Software Testing, New York: IEEE Press, 2019, pp. 213–229.
42. Adams, R. Machine Learning for Process Optimization. Journal of Intelligent Systems, Berlin: Springer, 2022, pp. 88–106.
43. Richards, T. Digital Transformation in Business Information Systems. Business Informatics Review, Paris: Elsevier, 2021, pp. 142–160.
44. Moreno, J. Evaluating UX in Online Management Platforms. HCI and Web Design Journal, Boston: MIT Press, 2020, pp. 51–68.

45. Wright, C. Testing Complex Web Systems Using Automated Frameworks. Proceedings of the Software Testing Symposium, Tokyo: IEEE, 2021, pp. 73–89.
46. Hughes, R. Scalable Architectures for High-Performance Web Applications. Web Systems Engineering Proceedings, Berlin: Springer, 2022, pp. 162–180.

ДОДАТКИ

Додаток А

Програмні коди

Підготовка середовища та імпорт бібліотек

```
import torch
import torch.nn as nn
import torch.optim as optim
from torchvision import models, transforms
from torch.utils.data import DataLoader, Dataset
import numpy as np
from PIL import Image
import os

# Перевірка наявності GPU
device = torch.device("cuda:0" if torch.cuda.is_available() else "cpu")
print(f"Використовується пристрій: {device}")
```

Функція ініціалізації моделі ResNet-18 та заморожування

Ця функція реалізує ключову стратегію: використання попередньо навченої ResNet-18 та заморожування всіх згорткових шарів, залишаючи для навчання лише повнозв'язний шар

```
def setup_resnet_model(num_classes):
    """
    Ініціалізує попередньо навчену ResNet-18 та заморожує згорткові шари.
    """
    # 1. Завантаження попередньо навченої ResNet-18
    model = models.resnet18(weights=models.ResNet18_Weights.IMAGENET1K_V1)

    # 2. Заморожування всіх згорткових шарів (Feature Extraction)
    # Згідно з описом: "We chose to freeze all of the convolutional layers"
    for param in model.parameters():
        param.requires_grad = False

    # 3. Розморожування та модифікація повнозв'язного шару (Fully Connected Layer)
    # Згідно з описом: "We only unfroze the fully connected layer"

    # Отримуємо кількість вхідних ознак для FC шару
    num_ftrs = model.fc.in_features

    # Створюємо новий FC шар з кількістю вихідних ознак, що дорівнює NUM_CLASSES
    model.fc = nn.Linear(num_ftrs, num_classes)

    # Ваги нового шару (model.fc) за замовчуванням мають requires_grad=True
    # (якщо вони були ініціалізовані як нові). Це забезпечує, що
    # лише цей шар буде навчатися.
```

```
    model = model.to(device)
    return model

# Ініціалізація моделей
original_model = setup_resnet_model(NUM_CLASSES)
encoded_model = setup_resnet_model(NUM_CLASSES)
```

Функція навчання та оцінки

```
def train_model(model, dataloader, criterion, optimizer, num_epochs=10):
    """Стандартна функція тренування моделі."""
    model.train()
    for epoch in range(num_epochs):
        running_loss = 0.0
        correct_preds = 0
        total_preds = 0

        for inputs, labels in dataloader:
            inputs = inputs.to(device)
            labels = labels.to(device)

            optimizer.zero_grad()

            # Пряме поширення
            outputs = model(inputs)
            loss = criterion(outputs, labels)

            # Зворотне поширення та оптимізація
            loss.backward()
            optimizer.step()

            # Статистика
            running_loss += loss.item() * inputs.size(0)
            _, preds = torch.max(outputs, 1)
            correct_preds += torch.sum(preds == labels.data)
            total_preds += labels.size(0)

        epoch_loss = running_loss / len(dataloader.dataset)
        epoch_acc = correct_preds.double() / total_preds

        print(f"Епоха {epoch+1}/{num_epochs} - Втрати: {epoch_loss:.4f} Точність: {e

    print("Навчання завершено.\n")
    return model

def evaluate_model(model, dataloader):
    """Оцінює точність моделі на тестовому наборі."""
    model.eval()
    correct_preds = 0
    total_preds = 0
    with torch.no_grad():
        for inputs, labels in dataloader:
            inputs = inputs.to(device)
            labels = labels.to(device)

            outputs = model(inputs)
            _, preds = torch.max(outputs, 1)
```

```
            correct_preds += torch.sum(preds == labels.data)
            total_preds += labels.size(0)

    accuracy = correct_preds.double() / total_preds
    return accuracy.item()
```

Навчання оригінальної моделі (Original Model)

```
print("--- Навчання Оригінальної Моделі на Оригінальних Даних ---")
criterion_orig = nn.CrossEntropyLoss()
optimizer_orig = optim.Adam(original_model.fc.parameters(), lr=0.001) # Навчаємо ли
original_model = train_model(original_model, original_train_loader, criterion_orig,
```

Навчання закодованої моделі (Encoded Model)

```
print("--- Навчання Закодованої Моделі на Закодованих Даних ---")
criterion_enc = nn.CrossEntropyLoss()
optimizer_enc = optim.Adam(encoded_model.fc.parameters(), lr=0.001) # Навчаємо лише
encoded_model = train_model(encoded_model, encoded_train_loader, criterion_enc, opt
```

Функції моделювання та навчання

```
def setup_resnet_model(num_classes):
    """Ініціалізує попередньо навчену ResNet-18 та заморожує згорткові шари."""
    # Завантаження попередньо навченої ResNet-18
    model = models.resnet18(weights=models.ResNet18_Weights.IMAGENET1K_V1)

    # Заморожування всіх шарів
    for param in model.parameters():
        param.requires_grad = False

    # Модифікація та розморожування повнозв'язного шару (FC)
    num_ftrs = model.fc.in_features
    model.fc = nn.Linear(num_ftrs, num_classes)

    model = model.to(device)
    return model

def train_model(model, dataloader, criterion, optimizer, num_epochs=10):
    """Тренує модель на даних."""
    model.train()
    print(f"Початок навчання на {len(dataloader.dataset)} прикладах.")
    for epoch in range(num_epochs):
        running_loss = 0.0
        correct_preds = 0
        total_preds = 0

        for inputs, labels in dataloader:
            inputs = inputs.to(device)
            labels = labels.to(device)
```

```
            optimizer.zero_grad()
            outputs = model(inputs)
            loss = criterion(outputs, labels)

            loss.backward()
            optimizer.step()

            running_loss += loss.item() * inputs.size(0)
            _, preds = torch.max(outputs, 1)
            correct_preds += torch.sum(preds == labels.data)
            total_preds += labels.size(0)

        epoch_loss = running_loss / len(dataloader.dataset)
        epoch_acc = correct_preds.double() / total_preds

        print(f"Епоха {epoch+1}/{num_epochs} - Втрати: {epoch_loss:.4f} Точність: {e

    return model

def evaluate_model(model, dataloader):
    """Оцінює точність моделі на тестовому наборі."""
    model.eval()
    correct_preds = 0
    total_preds = 0
    with torch.no_grad():
        for inputs, labels in dataloader:
            inputs = inputs.to(device)
            labels = labels.to(device)


            outputs = model(inputs)
            _, preds = torch.max(outputs, 1)

            correct_preds += torch.sum(preds == labels.data)
            total_preds += labels.size(0)

    accuracy = correct_preds.double() / total_preds
    return accuracy.item()
```

Модифікований Клас Dataset

Цей клас сканує каталоги для отримання шляхів до зображень та відповідних міток

```
import torch
import torch.nn as nn
import torch.optim as optim
from torchvision import models, transforms
from torch.utils.data import DataLoader, Dataset
from PIL import Image
import os
import glob

# Перевірка наявності GPU
device = torch.device("cuda:0" if torch.cuda.is_available() else "cpu")
print(f"Використовується пристрій: {device}")
```

```
# Стандартні трансформації для ResNet-18 (ImageNet)
data_transforms = transforms.Compose([
    transforms.Resize(256),
    transforms.CenterCrop(224), # ResNet-18 приймає 224x224
    transforms.ToTensor(),
    transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])
])


class RealImageDataset(Dataset):
    """
    Клас Dataset для завантаження зображень з каталогів.
    Припускається, що структура: base_dir/class_label/image.jpg
    """
    def __init__(self, base_dir, transform=None):
        self.transform = transform
        self.image_paths = []
        self.labels = []
        self.class_to_idx = {}

        # 1. Сканування каталогів для визначення класів
        class_names = sorted([d for d in os.listdir(base_dir)
                              if os.path.isdir(os.path.join(base_dir, d))])

        for i, class_name in enumerate(class_names):
            self.class_to_idx[class_name] = i

            class_path = os.path.join(base_dir, class_name)

    print("ПОМИЛКА: Не вдалося знайти каталоги даних. Будь ласка, перевірте, чи правильн
    # Використовуємо заглушку для продовження
    NUM_CLASSES = 19
    class CustomFacialDataset(Dataset):
        def __init__(self, num_samples, num_classes, transform=None):
            self.num_samples = 1500
            self.num_classes = num_classes
            self.transform = transforms.ToTensor()
            self.images = [torch.rand(3, 224, 224) for _ in range(self.num_samples)
            self.labels = np.random.randint(0, num_classes, size=self.num_samples)

        def __len__(self):
            return self.num_samples

        def __getitem__(self, idx):
            return self.images[idx], self.labels[idx]

    encoded_train_dataset = CustomFacialDataset(1500, NUM_CLASSES)
    encoded_test_dataset = CustomFacialDataset(370, NUM_CLASSES)
    original_train_dataset = CustomFacialDataset(1500, NUM_CLASSES)


NUM_CLASSES = len(encoded_train_dataset.class_to_idx) if hasattr(encoded_train_data

encoded_train_loader = DataLoader(encoded_train_dataset, batch_size=32, shuffle=Tru
encoded_test_loader = DataLoader(encoded_test_dataset, batch_size=32, shuffle=Fals
original_train_loader = DataLoader(original_train_dataset, batch_size=32, shuffle=
print(f"Кількість класів: {NUM_CLASSES}")
```