ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «**СУЧАСНІ АРХІТЕКТУРИ НЕЙРОННИХ МЕРЕЖ У ЗАДАЧАХ КЛАСИФІКАЦІЇ ТА РЕГРЕСІЇ**»

на здобуття освітнього ступеня магістр

за спеціальністю 124 Системний аналіз
(код, найменування спеціальності)

освітньо-професійної програми Інтелектуальні системи управління
(назва)

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

_____ Єлизавета КИРИЛОВА
(підпис) *(ім'я, ПРІЗВИЩЕ здобувача)*

Виконала:
здобувачка вищої освіти
група САДМ-61 Єлизавета КИРИЛОВА
(ім'я, ПРІЗВИЩЕ)

Керівник Олексій ШУШУРА
д.т.н. професор *(ім'я, ПРІЗВИЩЕ)*

Рецензент Наталія ЛАЩЕВСЬКА
к.т.н. доцент *(ім'я, ПРІЗВИЩЕ)*

Київ 2026

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут Інформаційних технологій

Кафедра Інформаційних систем та технологій
Ступінь вищої освіти магістр
Спеціальність 124 Системний аналіз
Освітньо-професійна програма Інтелектуальні системи управління

ЗАТВЕРДЖУЮ

Завідувач кафедри ІСТ

Каміла СТОРЧАК _____

«___» _____ 2025 року

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Кириловій Єлизаветі Вікторівні
(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Сучасні архітектури нейронних мереж у задачах класифікації та регресії

керівник кваліфікаційної роботи: Олексій ШУШУРА, д.т.н., професор
(ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «30» жовтня 2025р. №467

2. Строк подання кваліфікаційної роботи «26» грудня 2025р.

3. Вихідні дані кваліфікаційної роботи:

1. Архітектури глибоких нейронних мереж (CNN, Vision Transformer).
2. Методи розв'язання задач комп'ютерного зору (класифікація та регресія).
3. Набір даних зображень облич UTKFace.
4. Науково-технічна література та інтернет-джерела

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

1. Дослідження теоретико-методичних засад архітектур нейронних мереж у задачах класифікації та регресії.

2. Аналіз вхідних даних та дослідження базових архітектур для визначення віку і статі.

3. Розробка та реалізація удосконаленої методики навчання мультизадачної моделі.

5. Перелік ілюстаційного матеріалу: *презентація*

6. Дата видачі завдання «30» жовтня 2025.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання	Примітка
1.	Підбір та аналіз науково-технічної літератури	30.10.2025 – 03.11.2025	
2.	Дослідження теоретичних засад архітектур нейронних мереж	04.11.2025 – 12.11.2025	
3.	Аналіз вхідних даних та дослідження базових архітектур	13.11.2025 – 24.11.2025	
4.	Розробка та реалізація удосконаленої мультизадачної моделі	25.11.2025 – 08.12.2025	
5.	Оформлення пояснювальної записки	09.12.2025 – 18.12.2025	
6.	Розробка демонстраційних матеріалів, доповідь	19.12.2025 – 23.12.2025	
7.	Подання кваліфікаційної роботи на кафедру	24.12.2025 – 26.12.2025	

Здобувачка вищої освіти _____ Єлизавета КИРИЛОВА
(підпис) *(ім'я, ПРІЗВИЩЕ)*

Керівник кваліфікаційної роботи _____ Олексій ШУШУРА
(підпис) *(ім'я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття ступеня магістр: 97 стор., 22 рис., 7 табл., 1 додаток, 51 джерело.

Мета роботи – покращення швидкості навчання та підвищення точності класифікації та регресії в задачах визначення віку і статі людини за зображеннями облич шляхом дослідження особливостей сучасних згорткових та трансформерних архітектур нейронних мереж та програмної реалізації удосконаленої методики їх навчання.

Об'єкт дослідження – процес навчання глибоких нейронних мереж для задач комп'ютерного зору.

Предмет дослідження – архітектури ResNet та Vision Transformer, а також методи підвищення їх ефективності у задачах класифікації статі та регресії віку.

Короткий зміст роботи. У першому розділі подано огляд еволюції нейромережевих архітектур, математичний апарат задач класифікації та регресії й основні проблеми навчання..

У другому розділі досліджено набір даних UTKFace, проведено апробацію базових моделей ResNet-18 і ViT-B/16 та виявлено їхні обмеження за точністю і ресурсоємністю.

У третьому розділі реалізовано мультизадачну модель із використанням трансферного навчання, гібридної функції втрат та адаптивної аугментації, що забезпечила точність класифікації 94,2% та зниження похибки регресії на тестових даних.

КЛЮЧОВІ СЛОВА: НЕЙРОННІ МЕРЕЖІ, ЗГОРТКОВІ МЕРЕЖІ, VISION TRANSFORMER, КОМП'ЮТЕРНИЙ ЗІР, КЛАСИФІКАЦІЯ СТАТІ, РЕГРЕСІЯ ВІКУ, МУЛЬТИЗАДАЧНЕ НАВЧАННЯ, ТРАНСФЕРНЕ НАВЧАННЯ, АДАПТИВНА АУГМЕНТАЦІЯ ДАНИХ.

ABSTRACT

The textual part of the master's qualification thesis: 97 pp., 22 fig., 7 tables, 1 appendix, 51 sources.

The aim of the work is to improve the training speed and increase the accuracy of classification and regression in the task of estimating a person's age and gender from face images by studying the properties of modern convolutional and transformer neural network architectures and by implementing an enhanced training methodology for them.

The object of the study is the training process of deep neural networks for computer vision tasks.

The subject of the study is the ResNet and Vision Transformer architectures and the methods for improving their efficiency in gender classification and age regression tasks.

Summary of the work. The first chapter presents an overview of the evolution of neural network architectures, the mathematical framework of classification and regression problems, and the main training challenges.

The second chapter analyses the UTKFace dataset, evaluates baseline ResNet-18 and ViT-B/16 models, and identifies their limitations in terms of accuracy and computational cost.

The third chapter implements a multi-task model using transfer learning, a hybrid loss function, and adaptive data augmentation, achieving 94.2% classification accuracy and a reduced regression error on the test set.

KEYWORDS: NEURAL NETWORKS, CONVOLUTIONAL NETWORKS, VISION TRANSFORMER, COMPUTER VISION, GENDER CLASSIFICATION, AGE REGRESSION, MULTI-TASK LEARNING, TRANSFER LEARNING, ADAPTIVE DATA AUGMENTATION.

ЗМІСТ

ВСТУП

Актуальність теми. Нейронні мережі є однією з найбільш динамічних технологій сучасного світу [1]. Це інструмент, за допомогою якого комп’ютерні системи вчаться бачити та аналізувати візуальну інформацію так само ефективно, як і людина. Технології комп’ютерного зору використовуються для широкого спектра задач: від розблокування смартфонів та тегування фотографій до складних систем відеоспостереження та медичної діагностики.

Поширеність цих систем полегшує життя та автоматизує рутинні процеси, але водночас висуває високі вимоги до їх надійності. Помилки у розпізнаванні атрибутів людини (віку, статі) можуть призвести до некоректної роботи систем безпеки або аналітики. Наприклад, стандартні моделі часто помиляються при роботі з зображеннями дітей або людей похилого віку, а також потребують значних обчислювальних ресурсів.

Кількість даних та складність задач стабільно зростає. Оскільки сучасні системи базуються на різних архітектурах (згорткові мережі [5], трансформери [7]), кожна з них має свої вразливі місця: одні схильні до перенавчання, інші – до нестабільної роботи на малих даних [2]. Частина проблем пов’язана з тим, що моделі «запам'ятовують» зайві деталі фону замість ключових ознак обличчя. Інші труднощі виникають через незбалансованість реальних даних [19]. Таким чином, удосконалення методів навчання цих архітектур для підвищення їх точності та швидкодії є дуже важливим питанням, що робить дану тему актуальною.

Мета роботи – покращення швидкості навчання та підвищення точності класифікації та регресії в задачах визначення віку і статі людини за зображеннями облич шляхом дослідження особливостей сучасних згорткових та трансформерних архітектур нейронних мереж та програмної реалізації удосконаленої методики їх навчання.

Для досягнення мети, у магістерській роботі успішно виконано наступні завдання:

– аналіз сучасного стану методів комп'ютерного зору та визначення обмежень існуючих архітектур (CNN, Transformers) у задачах визначення віку і статі;

– порівняльне дослідження базових моделей на реальних даних (UTKFace) та виявлення факторів, що знижують точність їх роботи;

– розробка удосконаленої методики навчання, яка базується на комбінації стратегій трансферного навчання (Fine-tuning), мультизадачності (Multi-task Learning) та адаптивної аугментації даних;

– програмна реалізація запропонованих підходів та створення діючого прототипу системи розпізнавання;

– експериментальна перевірка ефективності розробленої методики та доведення її переваги над стандартними підходами за критеріями точності та швидкодії;

– формулювання практичних рекомендацій щодо впровадження досліджених архітектур у прикладні системи реального часу.

Об'єкт дослідження – процес навчання глибоких нейронних мереж для задач комп'ютерного зору.

Предмет дослідження – архітектури ResNet та Vision Transformer, а також методи підвищення їх ефективності у задачах класифікації статі та регресії віку.

Методи дослідження. Під час написання магістерської кваліфікаційної роботи були використані методи теоретичного аналізу, експериментального моделювання, порівняльного аналізу та методи математичної статистики для оцінки похибок.

Наукова новизна одержаних результатів. У ході дослідження описано новий підхід для одночасного визначення віку та статі людини на основі комбінації алгоритмів машинного навчання: стратегії Fine-tuning [24], мультизадачного навчання (Multi-task Learning) та адаптивної аугментації [23]. Результат застосування яких показує високу точність класифікації (94.2%) та низьку похибку регресії на тестових даних.

Практична значущість одержаних результатів. Запропонована модель забезпечує ефективне рішення для автоматизованого аналізу облич, яке може бути використане в системах контролю доступу та відеоаналітики, забезпечуючи баланс між точністю та швидкістю роботи.

Апробація результатів роботи. Основні положення та результати роботи доповідались та отримали схвалення на III Всеукраїнській науково-технічній конференції «Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу» (кафедра Інформаційних систем та технологій Державного університету інформаційно-комунікаційних технологій, м. Київ, 2025 р.), а також на VIII Всеукраїнській науково-технічній конференції «Комп'ютерні технології: інновації, проблеми, рішення» (Державний університет «Житомирська політехніка», 2025 р.).

1 ТЕОРЕТИКО-МЕТОДИЧНІ ЗАСАДИ ДОСЛІДЖЕННЯ АРХІТЕКТУР НЕЙРОННИХ МЕРЕЖ У ЗАДАЧАХ КЛАСИФІКАЦІЇ ТА РЕГРЕСІЇ

1.1 Огляд еволюції архітектур нейронних мереж у контексті покращення швидкості та точності навчання

Стійкий прогрес у розвитку методів ШІ штучний інтелект не є водночас лінійним накопиченням знань. Навпаки, історіографія галузі вказує на зміну кількох наукових парадигм, кожна з яких супроводжується своєрідним підходом до математичного моделювання когнітивних процесів. У контексті даного дослідження доцільно проаналізувати еволюцію нейронних мереж крізь призму зростання їхньої апроксимуючої здатності та переходу від інженерії ознак (Feature Engineering) до навчання представлень (Representation Learning) [1].

1.1.1 Від логічних автоматів до конекціонізму та першої «Зими ШІ»

На початку розвитку нейромережевого підходу стоять праці за У. Маккаллока та В. Піттса. Так, В. Маккаллок та В. Піттс у їхній базовій роботі в 1943 році запропонували математичну модель штучного нейрона у вигляді порогової логічної одиниці [10], що в подальшому лягла в основу першої навчальної архітектури, створеної Ф. Розенблаттом – перцептрону [11] .

Однак період розвитку конекціонізму також супроводжувався подоланням фундаментальних теоретичних обмежень. Так, в монографії «Perceptrons» М. Мінського (M. Minsky) та С. Пейперта (S. Papert) [12] математично доводилось безперспективність, власне кажучи, одношарових мереж у розв’язуванні лінійно нероздільних завдань, зокрема функцію виключного АБО. Адже своєчасно запропонувана критика, підтримана завищеними очікуваннями і фактом відсутності необхідних обчислювальних потужностей, спричинила ледь не повну

зупинку розвитку AI та стрижневий відрізок грантового фінансування, згодом отримавши, у науковій літературі, назву «Зима штучного інтелекту» (AI Winter).

У той час наукові кола були скептичними щодо можливостей нейромереж. Публікація Д. Румельхарта, Дж. Хінтона, Р. Вільямса зробила вихід із кризи, або «відліга», можливим лише у 1986 році. Алгоритм зворотного поширення помилки, також відомий як зворотне поширення[13]. Ця стратегія дозволила подолати обмеження, описані Мінським, і навчати багатошарові перцептрони ефективно. Тим не менш, проблема ефективного навчання глибоких структур залишалася актуальною протягом наступних кількох десятиліть.

1.1.2 Класичні методи машинного навчання в задачах комп'ютерного зору

Парадигма, що базувалася на ручному конструюванні ознак, була домінуючим підходом у комп’ютерному зорі до початку глибокого навчання, приблизно до 2012 року. Усвідомлення недоліків цих методів дозволяє краще оцінити переваги сучасних нейромережевих методів, предметом цього дослідження.

Екстракція ознак і класифікація були частинами традиційного конвеєру розпізнавання:

– виділення дескрипторів (Feature Extraction). На даний момент зображення перетворилося на вектор числових характеристик у за допомогою фіксованих алгоритмів. Найбільш поширеними методами були:

1) SIFT (Scale-Invariant Feature Transform): метод, розроблений Д. Лоу (D. Lowe) [32], що гарантує стійкість до масштабування, повороту та часткової зміни освітлення. Пошук локальних екстремумів у просторі масштабів є основою.;

2) HOG (Histogram of Oriented Gradients): метод, запропонований Далалом і Тріггсом [33], що ділить зображення на комірки та розраховує

гістограму напрямків градієнтів. Цей підхід фактично став стандартом для задач ідентифікації пішоходів до появи CNN;

3) LBP (Local Binary Patterns): текстурний дескриптор, ефективний для розпізнавання облич, завдяки своїй простоті обчислення та тому, що вона не залежить від монотонних змін яскравості [34];

- класифікація на основі ознак. Отримані вектори подавалися на вхід класичним алгоритмам машинного навчання («Shallow Learning»):

1) метод опорних векторів (SVM – Support Vector Machine): алгоритм, розроблений В. Вапніком [35], що здійснює пошук гіперплощини, яка розділяє класи з максимальною смугою розділення (Margin). Використання «ядерного трюку» (Kernel Trick) дозволяло вирішувати нелінійні задачі;

2) випадковий ліс (Random Forest): ансамблевий метод, запропонований Л. Брейманом [36], що базується на створенні кількох рішень;

3) k-найближчих сусідів (k-NN): Непараметричний метод класифікації базується на тому, що еталони розташовані у просторі ознак у метричному порядку.

Класичний підхід зіткнувся з основною проблемою наявності семантичного розриву між низькорівневими піксельними даними та високорівневими ідеями. Система використовувала «ручні» дескриптори, які не могли узагальнити складні ієрархічні візуальні структури.

1.1.3 Парадигма глибокого навчання (Deep Learning)

У 2012 році розпочався новий етап розвитку «Deep Learning Revolution» після перемоги архітектури AlexNet, яка була розроблена під керівництвом Дж. Хінтона, у змаганні ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [14]. Глибокі згорткові мережі (CNN) уперше перевершили традиційні методи комп’ютерного зору, які базувалися на виборі дескрипторів шляхом ручного виділення.

Аналіз літератури дозволяє визначити три основні компоненти, які працювали разом, щоб забезпечити цей технологічний стрибок і запобігти появні нової «Зими ШІ»:

– доступність великих даних: можна було забезпечити достатню різноманітність прикладів під час навчання моделей із мільйонами параметрів за допомогою модифікацій, таких як ImageNet [15], щоб уникнути надмірного перенавчання;

– апаратне забезпечення: коли графічні процесори (GPU) були адаптовані для виконання паралельних матричних операцій, час навчання був скорочений з тижнів до годин;

– алгоритмічні інновації: використання нелінійних функцій активації ReLU, що вирішило проблему затухання градієнта [9]; застосування методів стохастичної оптимізації (Adam, RMSProp) і методів регуляризації [9].

1.1.4 Перехід до архітектур на основі уваги

Переосмислення ролі індуктивних упереджень було останнім значним кроком у розвитку методів. Протягом багатьох років згорткові мережі базувалися на локальності та інваріантності трансляції. Васвані, з іншого боку, виявив, що механізми глобальної уваги, також відомі як самоувага, можуть більш ефективно моделювати складні залежності[6]. Доступна архітектура трансформатора. Упровадження цієї стратегії у візуальні задачі за допомогою Vision Transformer (ViT) [7] створило нову парадигму боротьби.

Ключовою відмінністю нового підходу стала відмова від обробки пікселів як решітки на користь представлення зображення у вигляді послідовності фрагментів (патчів), що дозволило застосувати до візуальних даних потужний математичний апарат, розроблений для обробки природної мови. Це відкрило шлях до створення універсальних мультимодальних моделей, здатних навчатися на гігантських масивах даних без втрати ефективності, що було "вузьким місцем" для класичних згорткових архітектур. Крім того, механізм уваги дозволяє моделі динамічно

фокусуватися на найбільш інформативних ділянках зображення незалежно від відстані між ними, забезпечуючи краще розуміння глобального контексту сцени [7].

Таким чином, еволюція галузі перейшла від спроби імітувати окремі нейрони до створення складних диференційовних архітектур, здатних ідентифікувати ієрархічні ознаки. Щоб зробити обґрунтований вибір архітектури в рамках цієї магістерської роботи, важливо мати розуміння цих історичних етапів і обмежень попередніх методів.

1.2 Аналіз математичного апарату задач класифікації та регресії для визначення атрибутів обличчя

Щоб провести системний аналіз предметної області та надати теоретичне обгрунтування архітектурних рішень, потрібно виконати математичну формалізацію задач, що досліджуються. У теорії статистичного навчання завдання класифікації та регресії розглядаються як процеси пошуку апроксимуючої функції $f(x, \theta)$. Відображаючий вхідних простір ознак X у простір цільових значень Y, цей процес зменшує емпіричний ризик на навчальній вибірці.

1.2.1 Формалізація задачі класифікації

Віднесення об’єкта до одного з K дискретних класів формально визначає завдання мультикласової класифікації. Умовну ймовірність того, що вхідний об’єкт належить до певного класу, моделює глибока нейронна мережа. Загальноприйняті методи [4] описують цю залежність рівняння:

$$P(y = k|x) = f_k(x, \theta) \quad (1.1)$$

де y – цільова змінна (мітка класу), $y \in \{1, \dots, K\}$;

x – вектор вхідних ознак об'єкта, $x \in R^d$;

θ – вектор параметрів (вагових коефіцієнтів) нейронної мережі;

f_k – вихідне значення мережі, що відповідає k-му класу.

Функція активації Softmax використовується для забезпечення умови нормування ймовірностей на вихідному шарі архітектури. Softmax [4] дозволяє розглядати виходи мережі як розподіл ймовірностей, як стверджують Гудфеллоу та інші дослідники, оскільки вони є природним узагальненням логістичної функції для багатовимірного простору:

$$\sigma(z)_i = \frac{exp(z)_i}{\sum_{j=1}^{K} exp(z_j)} \tag{1.2}$$

де $\sigma(z)_i$ – ймовірність приналежності об'єкта до i-го класу;

z – вектор логітів (не нормалізованих виходів передостаннього шару), $z \in R^K$;

exp – експоненційна функція;

K – загальна кількість класів у вибірці.

Методичною основою оптимізації параметрів θ у даному класі задач є принцип максимальної правдоподібності, що аналітично еквівалентно мінімізації перехресної ентропії (Cross-Entropy Loss) [4]:

$$L_{CE} = -\sum_{k=1}^{K} y_k \log(\hat{y}_k) \tag{1.3}$$

де L_{CE} – значення функції втрат;

y_k – істинна мітка класу (представлена у форматі one-hot encoding: 1 для істинного класу, 0 для інших);

$\hat{y}_k$ – прогнозована моделлю ймовірність для класу k;

log – натуральний логарифм.

Наскільки ефективною є функція втрат, залежить від необхідності вирішити проблему згасання градієнтів, яка виникає при використанні квадратичної помилки (MSE) разом із сигмоїдальними функціями активації. Вплив швидкості збіжності алгоритму навчання значний.

1.2.2 Формалізація задачі регресії

Регресійний аналіз, на відміну від класифікаційних завдань, передбачає прогнозування неперервної величини. Наступним чином залежність апроксимується математичною моделлю [4]:

$$y = f(x, \theta) + \in \tag{1.4}$$

де y – цільове неперервне значення, $y \in R^m$;

ϵ – випадкова компонента (шум), яка зазвичай вважається нормально розподіленою: $\epsilon \sim N(0, \sigma^2)$.

У методі найменших квадратів середньоквадратична помилка (MSE) є найкращим критерієм мінімізації гасового шуму [4]:

$$L_{MSE} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{1.5}$$

де N – обсяг навчальної вибірки (або розмір міні-батчу при стохастичній оптимізації);

y_i – істинне значення для i-го прикладу;

$\hat{y}_i$ – прогнозоване значення.

У сучасних архітектурах глибокого навчання часто використовується функція Huber Loss [4], яка поєднує характеристики MSE та MAE, щоб підвищити робастність (стійкість до аномальних викидів):

$$L_\delta(a) = \begin{cases} \frac{1}{2}a^2, & \text{якщо } |a| \le \delta \\ \delta\left(|a| - \frac{1}{2}\delta\right), & \text{інакше} \end{cases} \tag{1.6}$$

де a – похибка передбачення, $a = y - \hat{y}$;

δ – ороговий гіперпараметр, який визначає точку переходу від лінійної залежності функції втрат до квадратичної.

1.2.3 Методичні аспекти оптимізації глибоких мереж

Існує ітеративна процедура пошуку оптимальних параметрів θ^*. Вибір AdamW (Adam with Decoupled Weight Decay) [8] був обраний як основний алгоритм навчання в результаті даних досліджень. Рівняння забезпечує опис правил оновлення ваг цього алгоритму:

$$\theta_{t+1} = \theta_t - \eta\left(\frac{\widehat{m}_t}{\sqrt{\widehat{v}_t} + \epsilon} + \lambda\theta_t\right) \tag{1.7}$$

де θ_t – значення параметрів на ітерації t;

η – швидкість навчання (learning rate);

$\widehat{m}_t$ – експоненційне ковзне середнє градієнтів (оцінка першого моменту);

$\hat{v}_t$ – експоненційне ковзне середнє квадратів градієнтів (оцінка другого моменту);

ϵ – константа для забезпечення чисельної стабільності (стандартне значення 10^{-8});

λ – коефіцієнт згасання ваг (weight decay).

Узагальнюючи викладене, сформульований математичний апарат (1.1) – (1.7) становить теоретико-методологічну основу для проведення експериментальних досліджень і розробки програмного забезпечення, результати яких будуть представлені в наступних розділах роботи.

1.3 Методичні підходи до побудови архітектур CNN та Transformers (порівняльний аналіз)

В рамках сучасної теорії глибокого навчання доцільно розмежувати дві парадигми вилучення ознак (feature extraction), що мають фундаментальні відмінності. Мова йде про локальний підхід, який знайшов свою імплементацію у згорткових нейронних мережах (CNN), та глобальний, що виступає архітектурним базисом для моделей типу Transformer. Варто відзначити, що саме системний

аналіз математичних розбіжностей між цими підходами, у поєднанні з дослідженням еволюції архітектурних рішень, є обов'язковою передумовою для того, щоб коректно інтерпретувати результати експериментального дослідження. Зокрема, це дозволяє встановити чіткі причинно-наслідкові зв'язки між способом агрегації просторової інформації та здатністю моделі до генералізації на нових класах об'єктів. Такий теоретичний базис є необхідним для валідного обґрунтування вибору конкретної топології мережі під специфіку наявного набору даних.

1.3.1 Математичний базис та еволюція згорткових мереж (CNN)

Щодо функціонування CNN, то їх концептуальним підґрунтям визначено операцію дискретної згортки. Основна функція даної операції полягає у забезпеченні детекції локальних патернів (зокрема геометричних примітивів чи текстурних елементів), причому цей процес залишається інваріантним до їх просторового розміщення. У випадку роботи з двовимірним вхідним зображенням I та застосуванням ядра згортки (фільтра) K розмірністю $m \times n$, математичне визначення операції виглядає наступним чином [4]:

$$S(i,j) = (I * K)(i,j) = \sum_{u=0}^{m-1} \sum_{v=0}^{n-1} I(i+u, j+v) \cdot K(u,v) \qquad (1.8)$$

де $S(i,j)$ – значення карти ознак (feature map) у позиції (i,j);

I – вхідна матриця значень пікселів;

K – матриця вагових коефіцієнтів ядра, що навчаються.

Зрозуміти логіку вибору ResNet-18 як базової моделі неможливо без критичного погляду на архітектурні обмеження її попередників. Історія питання фактично розділилася на «до» і «після» 2012 року, коли з’явилася AlexNet. Хоча саме ця мережа, завдяки впровадженню механізмів регуляризації Dropout та функції активації ReLU, декласувала класичні методи розпізнавання [14], вона мала суттєву архітектурну ваду. Йдеться про надмірну обчислювальну вартість:

використання масивних фільтрів (зокрема 11×11) призвело до розростання кількості параметрів до критичних 60 мільйонів.

Наступний еволюційний крок – перехід до концепції глибокої уніфікації, яку запропонувала група Visual Geometry Group у своїх моделях VGG-16/19 [37]. Їхній підхід кардинально відрізнявся: відмова від великих ядер на користь каскаду фільтрів малого розміру (3 × 3). Тут спрацювала цікава математична логіка: послідовне накладання двох згорток 3 × 3 формує таке ж рецептивне поле, як і одна матриця 5 × 5. Проте, виграш був подвійним: система не лише ставала більш нелінійною (а отже, «розумнішою»), але й парадоксальним чином втрачала у вазі, зменшуючи кількість параметрів.

$$2 \cdot (3^2 \cdot C^2) < 1 \cdot (5^2 \cdot C^2) \quad (1.9)$$

де C – кількість вхідних та вихідних каналів (припускається, що вони рівні);

3^2 – площа фільтра розміром 3 × 3;

5^2 – площа фільтра розміром 5 × 5.

Наведена нерівність слугує математичним підтвердженням ефективності використання стеку малих згорток. Втім, попри еталонні показники точності, архітектура VGG зіштовхнулася з критичним бар'єром ресурсоємності: наявність 138 мільйонів параметрів фактично блокує можливість її розгортання в середовищі мобільних пристроїв з обмеженою обчислювальною потужністю.

Альтернативний шлях розвитку було запропоновано в архітектурі Inception / GoogLeNet (2014). Автори відійшли від лінійного нарощування шарів, реалізувавши концепцію розширення мережі «в ширину» через інтеграцію Inception-модулів [38]. Суть методу полягає в паралельному застосуванні фільтрів різної розмірності (1 × 1, 3 × 3, 5 × 5) до одного вхідного об'єму даних. Ключовим інженерним рішенням тут стала імплементація так званих «bottleneck layers» (згорток 1 × 1), функція яких зводиться до компресії розмірності каналів. Саме цей крок дозволив радикально знизити обчислювальну складність моделі без втрати її репрезентативної здатності.

Логічним продовженням тренду на оптимізацію стала поява класу моделей MobileNet (2017) [39], спроєктованих безпосередньо для мобільних платформ. В

основі цього підходу лежить технологія сепарабельної згортки (Depthwise Separable Convolution). Замість класичної «важкої» операції, тут відбувається декомпозиція процесу на два етапи: спочатку виконується канальна обробка (depthwise), а потім – точкова (pointwise). Емпірично доведено, що така заміна дозволяє скоротити кількість операцій множення-додавання у 8–9 разів, причому деградація точності розпізнавання залишається мінімальною.

1.3.2 Математичний базис архітектур Transformer

Як альтернативний вектор розвитку в сучасній теорії розпізнавання образів розглядається архітектура Vision Transformer (ViT). Принципова відмінність даного підходу полягає у відході від парадигми локальних вікон, характерної для згорток, та переході до використання механізму глобальної уваги (Self-Attention). З технічної точки зору, алгоритм передбачає попередню дискретизацію вхідного зображення на набір окремих фрагментів x_p (так званих патчів), які надалі підлягають лінійній проєкції у векторний простір.

Функціональним ядром обчислювального процесу визначено механізм розрахунку матриці уваги. Саме цей математичний інструмент забезпечує здатність моделі динамічно розподіляти пріоритети між різними ділянками зображення в залежності від контексту. Математична формалізація функції уваги (Scaled Dot-Product Attention) реалізується через взаємодію матриць запитів (Q), ключів (K) та значень (V) і описується виразом [6]:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \qquad (1.10)$$

де $Q = XW^Q, K = XW^K, V = XW^V$ – лінійні проекції вхідних даних X;

d_k – розмірність векторів ключів (scaling factor);

W^Q, W^K, W^V – матриці параметрів, що навчаються.

Необхідно акцентувати увагу на тому, що діаметрально протилежно до логіки згорткових операцій, імплементація механізму Self-Attention (визначена виразом 1.10) забезпечує генерацію глобального рецептивного поля вже на стадії

ініціалізації архітектури. Технічний базис цього феномену криється у знятті обмежень локальності: кожен дискретний патч набуває функціональної спроможності до встановлення кореляційних зв'язків із повним масивом елементів послідовності, причому цей процес є інваріантним відносно їх топологічної локалізації у просторі зображення.

1.3.3 Роль функції активації у глибокому навчанні

Фундаментальним компонентом при проектуванні будь-якої нейромережевої архітектури виступає функція активації. Саме цей елемент відповідає за інтеграцію нелінійних властивостей у систему, що, в свою чергу, є необхідною передумовою для успішної апроксимації залежностей високого рівня складності. В рамках даного дослідження доцільно провести детальний аналіз наступних функцій.

Розпочати аналіз доцільно з функції сигмоїди (Sigmoid). Хоча історично вона виступала піонером у цій сфері, на сучасному етапі її імплементація суттєво звузилася, обмежуючись переважно вихідними шарами в задачах бінарної класифікації. Ключовим стримуючим фактором тут виступає відомий феномен «згасання градієнта» (vanishing gradient): при насиченні нейрона значення похідної стрімко наближаються до нуля, що блокує навчання глибоких шарів. Математичний запис функції має вигляд:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{1.11}$$

де $\sigma(x)$ – значення функції активації;

x – вхідний сигнал (зважена сума входів нейрона);

e – основа натурального логарифма.

На противагу цьому, фактичним галузевим стандартом для конволюційних архітектур (зокрема, досліджуваної ResNet) стала функція ReLU (Rectified Linear Unit). Її домінування обумовлено здатністю забезпечувати високу обчислювальну ефективність та розрідженість активацій. Важливо відзначити, що ReLU ефективно нівелює проблему згасання градієнта, проте виключно в позитивній півобласті

значень аргументу [40]. Формально, якщо розглядати x як вхідний сигнал, операція визначається через вибір максимального значення:

$$f(x) = \max(0, x) \tag{1.12}$$

де $f(x)$ – вихідне значення нейрона;

x – вхідний сигнал;

max – математична операція вибору максимального значення.

Окремої уваги заслуговує підхід GeLU (Gaussian Error Linear Unit), який затвердився як базовий стандарт для трансформерних моделей, включаючи Vision Transformer (ViT). Принципова відмінність даного методу від ReLU полягає у властивості гладкості кривої. Завдяки своїй імовірнісній природі, яка спирається на функцію кумулятивного розподілу (CDF) стандартного нормального закону $\Phi(x)$, GeLU сприяє більш стабільній градієнтній оптимізації надглибоких моделей [41]. Аналітичний вираз функції виглядає наступним чином:

$$\mathrm{GELU}(x) = x \cdot \Phi(x) \tag{1.13}$$

де $\mathrm{GELU}(x)$ – значення функції активації;

x – вхідний сигнал;

$\Phi(x)$ – функція кумулятивного розподілу (CDF) для стандартного нормального розподілу.

1.3.4 Обґрунтування авторської позиції

На підставі проведеного компаративного аналізу математичного інструментарію видається можливим сформулювати авторську концепцію щодо доцільності імплементації досліджуваних архітектур. Фундаментальним вододілом тут виступає поняття індуктивного упередження (Inductive Bias). Специфіка згорткових мереж (CNN) полягає в наявності сильного індуктивного упередження, яке фактично апріорі інкапсулює в модель знання про локальність та ієрархічну природу візуальних структур. Діаметрально протилежну філософію сповідують трансформери: вони функціонують в умовах слабкого упередження, що ставить

перед моделлю вимогу самостійної реконструкції структури зображення, спираючись виключно на кореляційні зв'язки у вхідних даних.

Зазначений архітектурний дуалізм безпосередньо детермінує вимоги до обсягу навчальної вибірки. В роботі обґрунтовано тезу, згідно з якою саме дефіцит індуктивних обмежень виступає причиною зниження ефективності ViT на масивах малого та середнього масштабу, провокуючи схильність до перенавчання. Водночас, ситуація дзеркально змінюється при переході до даних надвеликої розмірності: тут жорстка фіксована структура згортки перетворюється на лімітуючий фактор, тоді як гнучкість трансформера забезпечує йому суттєву перевагу [7].

Виходячи з означеної логіки, експериментальна частина третього розділу буде присвячена верифікації робочої гіпотези. Її суть зводиться до того, що інтеграція стратегій трансферного навчання (Transfer Learning) [24] у комплексі зі спеціалізованими техніками регуляризації здатна нівелювати вразливість ViT на обмежених датасетах, створюючи синергію переваг обох методичних підходів.

1.4 Аналіз проблем навчання, що впливають на швидкість збіжності та точність класифікації і регресії

Питання результативності впровадження глибоких нейромережевих архітектур у площину прикладних рішень нерозривно пов’язане з необхідністю подолання низки фундаментальних бар’єрів. Останні, як правило, маніфестують себе безпосередньо на етапі процедурної мінімізації цільової функції втрат. У цьому ракурсі, проведення системної декомпозиції зазначених явищ виступає не просто формальною вимогою, а безальтернативним базисом для побудови архітектоніки подальшого експерименту та аргументації доцільності залучення конкретних регуляризаційних механізмів.

1.4.1 Проблема нестабільності градієнтів

У спектрі перешкод, що суттєво ускладнюють процедуру навчання глибоких архітектур, домінуючу позицію займає феномен, відомий як згасання (vanishing) або, у протилежному випадку, вибух (exploding) градієнтів. З точки зору математичного формалізму, генезис даного явища криється у прямих наслідках застосування ланцюгового правила диференціювання (Chain Rule) складних функцій, що є невід'ємною складовою алгоритму зворотного поширення помилки. Якщо розглядати градієнт функції втрат L відносно вагових коефіцієнтів вхідного шару w_1, то він визначається як кумулятивний добуток частинних похідних:

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial_y} \cdot \frac{\partial_y}{\partial h_n} \cdot \dots \cdot \frac{\partial h_2}{\partial h_1} \cdot \frac{\partial h_1}{\partial w_1} \tag{1.14}$$

де L – поточне значення цільової функції втрат;

w_1 – матриця вагових коефіцієнтів, що асоційована з першим (вхідним) шаром;

y – результуючий вихідний сигнал нейронної мережі;

h_i – вихідний вектор i-го прихованого шару архітектури;

n – параметр, що визначає загальну глибину (кількість шарів) у мережі.

Моделювання ситуації із залученням активаційних функцій, похідні яких не перевищують одиничного бар'єра (класичні приклади – Sigmoid чи Tanh), демонструє невтішну динаміку: пропорційно до зростання глибини мережі n, результуюче значення добутку у наведеній формулі асимптотично прямує до нуля. На практиці це призводить до блокування можливості корекції ваг на початкових рівнях ієрархії [4].

Як дієвий інструмент протидії зазначеній деградації сигналу в передових архітектурних рішеннях (на кшталт ResNet та ViT) зарекомендувала себе інтеграція механізму залишкових зв'язків (Residual Connections) [5]. Сутність методу полягає у створенні альтернативних маршрутів поширення інформації (identity shortcuts):

$$y = F(x, \{W_i\}) + x \tag{1.15}$$

де x – вхідний вектор ознак, що подається на блок;

$F(x, \{W_i\})$ – функція залишкового перетворення, параметризована вагами W_i;

y – вихідний сигнал обчислювального блоку.

Стратегічна цінність такої топологічної особливості полягає у формуванні гарантованого «коридору» для безперешкодного проходження градієнта до початкових шарів. Це стає можливим завдяки наявності доданку x, похідна якого тотожно дорівнює одиниці, що виступає стабілізуючим фактором для всього процесу оптимізації.

1.4.2 Еволюція методів стохастичної оптимізації

Визначення оптимального алгоритму мінімізації функції втрат виступає критичним фактором, що безпосередньо детермінує динаміку збіжності та фінальну якість навчання нейромережевих моделей. Для аргументації доцільності імплементації методу AdamW у рамках даного дослідження видається необхідним провести ретроспективний аналіз еволюційних трансформацій підходів до градієнтної оптимізації:

– *стохастичний градієнтний спуск (SGD)*: у своїй класичній варіації даний метод реалізує процедуру оновлення параметрів θ, базуючись на обчисленні градієнта функції втрат L для кожного окремого прецеденту навчальної вибірки. Основна вразливість підходу полягає у високій дисперсії оновлень, що провокує значні осциляції функції втрат і створює ризик «стагнації» процесу в сідлових точках поверхні помилки. Математично ітеративний крок описується виразом:

$$\theta_{t+1} = \theta_t - \eta \cdot \nabla_\theta J\left(\theta; x^{(i)}; y^{(i)}\right) \tag{1.16}$$

де θ_t, θ_{t+1} – векторні представлення параметрів моделі;

η – встановлений темп навчання;

$\nabla_\theta J$ – градієнт цільової функції втрат за параметрами θ;

– *метод моментумів (SGD with Momentum):* розроблений з метою нівелювання осциляційних ефектів та прискорення подолання ділянок плато. Суть методу полягає у накопиченні експоненційного ковзного середнього градієнтів з

попередніх ітерацій, що надає алгоритму здатність «за інерцією» долати локальні мінімуми. Формалізація процесу відбувається через систему:

$$\begin{cases} v_{t+1} = \gamma v_t + \eta \nabla_\theta J(\theta_t) \\ \theta_{t+1} = \theta_t - v_{t+1} \end{cases} \quad (1.17)$$

де v_t – вектор швидкості (накопичений імпульс) на кроці *t*;

γ – коефіцієнт моменту (зазвичай $\gamma \approx 0.9$), що визначає ступінь впливу попередніх оновлень;

η – темп навчання;

$\nabla_\theta J(\theta_t)$ – градієнт функції втрат у точці θ_t;

θ_t, θ_{t+1} – значення параметрів моделі;

– *адаптивні методи (Adagrad та RMSProp):* спрямовані на вирішення проблеми фіксованого темпу навчання. Враховуючи гетерогенну швидкість навчання різних нейронів у глибоких архітектурах, метод RMSProp пропонує стратегію індивідуальної адаптації learning rate шляхом нормування градієнта на корінь із середнього квадрата його попередніх значень:

$$E[g^2]_t = 0.9E[g^2]_{t-1} + 0.1g_t^2 \quad (1.18)$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t \quad (1.19)$$

де $E[g^2]_t$ – експоненційне ковзне середнє квадратів градієнтів;

g_t – поточний градієнт на кроці *t*;

ϵ – константа для забезпечення чисельної стабільності (щоб уникнути ділення на нуль).

Зазначений підхід дозволяє демпфувати навчання для параметрів з інтенсивними градієнтами і каталізувати процес там, де зміни є незначними;

– *метод Adam (Adaptive Moment Estimation):* алгоритм, що став фактичним стандартом де-факто, синтезуючи переваги Momentum та RMSProp. Його архітектура передбачає оцінку як першого моменту (середнього значення градієнта), так і другого (нецентрованої дисперсії), а правило оновлення параметрів має вигляд:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon}\hat{m}_t \quad (1.20)$$

де θ_{t+1} – оновлене значення параметрів моделі;

θ_t – поточне значення параметрів;

η – темп навчання (learning rate);

$\hat{m}_t$ – скоригована оцінка першого моменту (середнього значення градієнта);

$\hat{v}_t$ – скоригована оцінка другого моменту (дисперсії градієнта);

ϵ – константа для чисельної стабільності (зазвичай 10^{-8});

– *метод AdamW (Decoupled Weight Decay)*: удосконалена версія Adam, яка виправляє роботу регуляризації шляхом розділення процесів оновлення ваг та градієнта, що підвищує стабільність навчання.

Саме тому в експериментальній частині роботи (Розділ 3) вибір зупинено на оптимізаторі AdamW, як такому, що гарантує оптимальний баланс між швидкістю збіжності та генералізаційною здатністю моделі.

1.4.3 Проблема внутрішнього коваріантного зсуву (Internal Covariate Shift)

Динаміка процесу оптимізації характеризується перманентною флуктуацією статистичних параметрів розподілу вихідних значень на кожному шарі, що є прямим наслідком ітеративного оновлення ваг попередніх рівнів мережі. Цей феномен, класифікований у фаховій літературі як внутрішній коваріантний зсув (Internal Covariate Shift), породжує імператив постійної реадаптації наступних шарів до змінних вхідних розподілів, що, у свою чергу, виступає лімітуючим фактором для швидкості збіжності алгоритму.

З метою нівелювання негативного впливу зазначеного явища та стабілізації процесу навчання застосовують спеціалізовані методи нормалізації:

– *Batch Normalization (BN):* методологія, що затвердилася як безумовний стандарт при проектуванні згорткових архітектур (CNN). Її операційна логіка передбачає виконання нормалізації даних у розрізі поточного міні-батчу,

забезпечуючи центрування розподілу (приведення до нульового математичного сподівання) та масштабування дисперсії до одиничного значення, що суттєво прискорює навчання;

– *Layer Normalization (LN):* підхід, що виступає архітектурним базисом для моделей типу Transformer. Принципова відмінність даного методу полягає у реалізації нормалізації ознак виключно в межах одного конкретного зразка (per-sample), ігноруючи глобальну статистику батчу. Така автономність є критично значущою передумовою для ефективної обробки послідовностей варіативної довжини, де залежність від розміру батчу може вносити спотворення [6].

1.4.4 Проблема перенавчання (Overfitting) та компроміс Bias-Variance

Домінуючим викликом при оперуванні вибірками лімітованого обсягу постає тенденція глибоких архітектур до надмірної адаптації (overfitting) відносно навчального масиву даних, що йде всупереч меті виявлення генералізованих закономірностей. Емпірично даний феномен маніфестує себе через дивергенцію метрик якості: спостерігається асимптотична мінімізація помилки на тренувальній підмножині при синхронному та неконтрольованому зростанні похибки на тестових даних.

З метою превентивного блокування ефектів перенавчання в рамках дослідження аргументовано імплементацію комплексного підходу до регуляризації:

– *метод стохастичного виключення (Dropout) [9]:* стратегія, що передбачає рандомну деактивацію окремих нейронних вузлів у процесі кожної ітерації навчання із заданою ймовірністю p. З точки зору математичного обґрунтування, такий підхід фактично еквівалентний апроксимації навчання ансамблю, що складається з експоненційної кількості моделей із розділеними (спільними) параметрами, що суттєво підвищує стійкість системи;

– *затухання ваг (Weight Decay):* механізм, спрямований на обмеження складності моделі шляхом введення штрафного доданку за надмірно великі

абсолютні значення вагових коефіцієнтів (класична L2-регуляризація). Технічна реалізація даного методу інтегрована безпосередньо в логіку роботи оптимізатора AdamW, детальний аналіз якого наведено у підрозділі 1.2;

– *аугментація даних (Data Augmentation):* процедура штучної експансії варіативності навчальної вибірки, що досягається за рахунок застосування спектру афінних трансформацій вхідних зображень (зокрема ротації, масштабування, дзеркального відображення). Зазначений підхід відіграє ключову роль у формуванні інваріантних ознак, роблячи модель нечутливою до геометричних спотворень [25].

1.4.5 Проблема вибору темпу навчання (Learning Rate Scheduling)

Практика застосування статичної конфігурації гіперпараметра темпу навчання η формує передумови для неоптимальної траєкторії оптимізації, зокрема, суттєво підвищуючи ймовірність стагнації процесу в точках локальних екстремумів або виникнення паразитарних осциляцій навколо глобального мінімуму. Незважаючи на той факт, що в якості базового інструменту в роботі детерміновано алгоритм Adam [31], специфіка збіжності трансформерних моделей постулює критичну необхідність динамічного керування кроком навчання.

Домінуючою парадигмою у цій сфері виступає гібридна стратегія планування (scheduling), що передбачає послідовну комбінацію фази лінійного розігріву (Warmup) на етапі ініціалізації та механізму косинусного затухання (Cosine Decay) протягом основного циклу навчання. Математична формалізація закону зміни темпу на етапі затухання описується рівнянням [8]:

$$\eta_t = \eta_{min} + \frac{1}{2}(\eta_{max} - \eta_{min})\left(1 + \cos\left(\frac{T_{cur}}{T_{max}}\pi\right)\right) \quad (1.21)$$

де η_t – значення темпу навчання на епосі t;

η_{min}, η_{max} – граничні (мінімальне та максимальне) значення темпу навчання;

T_{cur} – номер поточної епохи;

T_{max} – загальна кількість епох навчання.

Системна інтеграція результатів аналізу предметної області дає підстави констатувати, що ефективність технічного розгортання архітектури ViT нерозривно пов'язана з імперативом впровадження радикально посилених стратегій регуляризації та аугментації даних. Ця вимога є значно більш критичною порівняно з підходами, застосовуваними для згорткових аналогів (CNN). Сформульований принцип визначено як концептуальний базис при проектуванні архітектури програмного інструментарію, що розробляється для реалізації подальших експериментальних випробувань.

1.4.6 Математичні основи методів регуляризації

В умовах виникнення ефекту перенавчання (Overfitting), що емпірично фіксується через дивергенцію метрик похибки (асимптотично низькі значення на тренувальній множині при одночасному зростанні на тестовій), критичної актуальності набуває застосування регуляризаційних механізмів. Функціональне призначення цих методів зводиться до накладання жорстких обмежень на рівень складності моделі:

– *L2-регуляризація (Weight Decay):* найпоширеніший підхід, суть якого полягає в імплементації додаткового штрафного доданку за надмірно великі абсолютні значення ваг безпосередньо до функції втрат. Математична формалізація модифікованої цільової функції J~ набуває вигляду:

$$\tilde{J}(\theta; X; y) = J(\theta; X; y) + \frac{\lambda}{2}\left|\left|w\right|\right|_2^2 \quad (1.22)$$

де $\tilde{J}$ – регуляризована функція втрат;

J – початкова функція втрат (наприклад, Cross-Entropy);

θ – повний набір параметрів моделі;

X, y – вхідні дані та цільові мітки відповідно;

λ – гіперпараметр решуляризації (weight decay rate);

w – вектор вагових коефіцієнтів (підмножина θ);

$||w||_2^2$ -квадрат L2-норми вектора ваг: $\sum w_i^2$.

Як наслідок, процедура оновлення вагових коефіцієнтів у рамках градієнтного спуску зазнає структурних змін:

$$w_{t+1} \leftarrow w_t - \eta(\nabla J + \lambda\omega_t) = w_t(1 - \eta\lambda) - \eta\nabla J \quad (1.23)$$

де w_{t+1}– оновлене значення ваг;

w_t – поточне значення ваг;

η – темп навчання (learning rate);

∇J – градієнт початкової функції втрат по вагах.

– *Dropout (Стохастичне виключення):* стратегія [9], фундаментальна ідея якої полягає в емуляції процесу навчання ансамблю, що складається з експоненційної кількості моделей. Технічно це реалізується через примусову деактивацію кожного нейрона з ймовірністю p (стандартне значення p=0.5). Формально вихід шару h із застосуванням нелінійної функції активації f описується рівнянням:

$$h = f(W \cdot (x \odot m) + b) \quad (1.24)$$

де h – вихідний вектор шару;

f – нелінійна функція активації;

W, b – матриця ваг та вектор зміщення шару;

x – вхідний вектор;

m – бінарна маска, згенерована з розподілу Бернуллі: $m \sim \text{Bernoulli}(1-p)$;

$\odot$ – операція поелементного множення (добуток Адамара).

1.4.7 Порівняльний аналіз методів нормалізації

Забезпечення стабілізації градієнтних потоків та каталізація процесу збіжності оптимізаційного алгоритму нерозривно пов'язані з процедурою нормалізації даних безпосередньо у внутрішніх шарах нейромережевої архітектури. В рамках дослідження проведено компаративний аналіз двох домінуючих методологічних підходів:

– *пакетна нормалізація (Batch Normalization – BN):* концепція, розроблена Ioffe та Szegedy, яка передбачає виконання нормалізації активацій *x* автономно для кожного каналу в розрізі всього поточного міні-батчу. Математична логіка формування результуючого значення y_i реалізується через наступний алгоритмічний ланцюжок:

$$y_i = \gamma \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} + \beta \tag{1.25}$$

де γ – параметр масштабування, що навчається;

x_i – значення активації для *i*-го елемента в батчі;

μ_B – середнє значення по батчу: $\mu_B = \frac{1}{B}\sum_{i=1}^{B} x_i$;

σ_B^2 – дисперсія по батчу: $\sigma_B^2 = \frac{1}{B}\sum_{i=1}^{B}(x_i - \mu_B)^2$;

B – розмір міні-батчу;

β – параметр зсуву, що навчається;

ϵ – константа для чисельної стабільності;

– *нормалізація шару (Layer Normalization – LN):* альтернативний підхід авторства Ba et al., який фокусується на нормалізації вхідних даних виключно в межах одного конкретного зразка, охоплюючи всі наявні канали або нейрони. Така архітектурна особливість забезпечує інваріантність методу відносно розміру батчу. Розрахунок необхідних статистик формалізується виразами:

$$\mu_L = \frac{1}{H}\sum_{j=1}^{H} x_j, \qquad \sigma_L^2 = \frac{1}{H}\sum_{j=1}^{H}\left(x_j - \mu_L\right)^2 \tag{1.26}$$

де μ_L, σ_L^2 – середнє та дисперсія активацій для одного конкретного зразка;

H – кількість нейронів (або каналів) у шарі;

x_j – активація *j*-го нейрона в шарі.

Синтезуючи вищевикладене, для проектування гібридної системи (ResNet + ViT) обґрунтовано необхідність імплементації комбінованої стратегії: інтеграція Batch Norm для блоків згорткового енкодера та застосування Layer Norm для модулів трансформера. Зазначена логіка покладена в основу програмної реалізації експериментального зразка.

1.5 Обґрунтування вибору метрик оцінювання точності та швидкодії архітектур

Процедура об’єктивізації якісних показників, що характеризують розроблені моделі, виступає не просто формальним етапом, а фундаментальним базисом наукового пошуку, який безпосередньо легітимізує валідність отриманих емпіричних даних. Беручи до уваги специфіку сформульованої наукової проблеми, сутність якої зводиться до знаходження оптимального компромісу (trade-off) між точністю апроксимації та обчислювальною ресурсоємністю, у роботі імплементовано полікритеріальний підхід до метризації результатів.

1.5.1 Метрики ефективності для задач класифікації

У розрізі базових критеріїв оцінювання ефективності класифікаторів традиційно домінує метрика точності (Accuracy), що інтерпретується як відсоткове співвідношення коректно ідентифікованих об'єктів до загального масиву вибірки. Втім, спираючись на критичний аналіз фахових джерел [4], варто констатувати суттєву деградацію інформативності даного показника в умовах роботи з незбалансованими класами (class imbalance). З огляду на це, методично обґрунтованим кроком є перехід до використання пари метрик Precision (точність) та Recall (повнота).

Математична формалізація зазначених індикаторів для бінарного випадку (або в рамках стратегії one-vs-all) виглядає наступним чином:

– *Precision (точність):* показник, що відображає частку дійсно релевантних екземплярів серед усіх об'єктів, які система маркувала як позитивні:

$$Precision = \frac{TP}{TP + FP} \tag{1.27}$$

де TP (True Positive) – кількість істинно-позитивних спрацювань (об'єкт класу А розпізнано як А);

FP (False Positive) – помилки першого роду (хибні спрацювання);

FN (False Negative) – помилки другого роду (пропуски цілі);

– *Recall (повнота)*: метрика, що оцінює здатність алгоритму знаходити всі об'єкти цільового класу в генеральній сукупності:

$$Recall = \frac{TP}{TP + FN} \tag{1.28}$$

де TP (True Positive) – кількість істинно-позитивних спрацювань;

FN (False Negative) – помилки другого роду (пропуски цілі, коли об'єкт класу А не було розпізнано);

– *F1-Score (F-міра)*: інтегральний критерій, призначений для комплексної валідації якості моделі. Його математична природа являє собою гармонічне середнє між точністю та повнотою, що дозволяє знайти баланс між цими часто суперечливими векторами оптимізації:

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{1.29}$$

де $Precision$ – значення точності, обчислене за формулою (1.17);

$Recall$ – повнота (частка знайдених релевантних екземплярів серед усіх релевантних).

Виходячи з цього, в рамках поточного дослідження статус пріоритетного критерію оптимізації класифікаційних моделей присвоєно метриці Macro-F1. Вона розраховується як середнє арифметичне значень $F1$ для всіх категорій, що надає можливість нівелювати негативний вплив дисбалансу класів на фінальну оцінку.

1.5.2 Метрики ефективності для задач регресії

У контексті кількісної верифікації адекватності регресійних моделей (що є критично важливим для задач предиктивного визначення координат об'єктів) методологічно обґрунтованим є застосування комплексної системи метрик, що включає оцінку абсолютних відхилень та поясненої дисперсії.

У даній роботі пріоритет надано наступним індикаторам якості:

– *RMSE (Root Mean Squared Error):* метрика, що дозволяє квантифікувати абсолютну величину похибки прогнозування, зберігаючи масштаб цільової змінної. Її математична природа забезпечує «штрафування» моделі за великі відхилення, що є суттєвим для мінімізації грубих помилок:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{1.30}$$

де N – загальна кількість прикладів у тестовій вибірці;

y_i – істинне значення цільової змінної для i– го прикладу;

$\hat{y}_i$ – прогнозоване моделлю значення;

– *коефіцієнт детермінації* R^2: статистичний показник, що характеризує «пояснювальну силу» моделі, а саме частку дисперсії залежної змінної, яка детермінується побудованою регресійною залежністю. Його аналітичний вираз записується як:

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2} \tag{1.31}$$

де $\bar{y}$ – середнє арифметичне значення цільової змінної;

$\hat{y}_i$ – прогнозоване моделлю значення;

y_i – істинне значення цільової змінної;

– *MAE (Mean Absolute Error):* альтернативний критерій, який вирізняється вищою інтерпретованістю результатів, оскільки його розмірність ідентична одиницям виміру цільової змінної (в даному контексті – рокам або пікселям). Формула розрахунку має вигляд:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{1.32}$$

де $N, y_i, \hat{y}_i$ – позначення аналогічні формулі (1.21).

1.5.3 Інженерні метрики (Efficiency Metrics)

Беручи до уваги векторну спрямованість кваліфікаційної роботи, що фокусується на діагностиці ефективності функціонування алгоритмічних структур в умовах жорстких лімітів апаратного забезпечення, до архітектури системи оцінювання інтегровано комплекс спеціалізованих інженерних метрик:

– *Inference Latency* (t_{lat}): часовий показник, що відображає середньостатистичний інтервал, необхідний для повної процедурної обробки одиничного вхідного зображення на потужностях цільового обчислювального пристрою (традиційно квантифікується в мілісекундах, мс);

– *Throughput (пропускна здатність)*: інтегральний індикатор, що характеризує граничну продуктивність системи, виражену через кількість візуальних об'єктів, які алгоритм здатен обробити за фіксовану одиницю часу (стандартна одиниця виміру – кадри за секунду, FPS);

– *Model Size (розмірність моделі)*: критерій, що дескриптивно описує структурну складність архітектури. Його оцінка здійснюється дуально: через загальну кількість параметрів, що підлягають ітеративному навчанню (обчислюється в мільйонах), а також через фізичний обсяг пам'яті, який модель займає на носії інформації.

1.5.4 Аналіз ROC-кривих та метрика AUC

У контексті поглибленої діагностики валідності бінарного класифікатора, що функціонує в умовах варіативності порогових значень (threshold) вірогідності, інструментальним стандартом виступає побудова кривої робочої характеристики приймача (ROC – Receiver Operating Characteristic).

Математично показник *TPR* (True Positive Rate), який у термінологічному аспекті є повним аналогом чутливості (Recall), формалізується співвідношенням:

$$TPR = \frac{FP}{TP + FN} \tag{1.33}$$

Паралельно розраховується індикатор *FPR* (False Positive Rate), що характеризує схильність системи до генерації помилок першого роду (хибних тривог):

$$FPR = \frac{FP}{FP + TN} \tag{1.34}$$

TN (True Negative) – кількість істинно-негативних спрацювань (об'єкт класу B правильно класифіковано як B);

FP – кількість помилок першого роду.

У ролі узагальнюючого критерію якості, інваріантного до специфіки вибору конкретного порогу класифікації, виступає площа під ROC-кривою (*AUC* – Area Under Curve). Емпірична інтерпретація метрики є наступною: граничне значення *AUC*=0.5 еквівалентне результату випадкового стохастичного вгадування, в той час як асимптотичне наближення до рівня *AUC*=1.0 слугує маркером ідеальної роздільної здатності класифікаторар.

1.5.5 Методика крос-валідації (k-Fold Cross-Validation)

З метою нівелювання стохастичного зміщення оцінок, яке є потенційним ризиком при фіксованому випадковому розподілі даних на навчальну та тестову множини (train/test split), у роботі імплементовано стратегію k-блочної перехресної перевірки (k-fold Cross-Validation).

Алгоритмічна архітектура методу реалізується через послідовність наступних етапів:

– первинна декомпозиція генеральної сукупності даних на *k* рівновеликих, неперетинних підмножин (фолдів). Виходячи з оптимального компромісу між обчислювальними витратами та дисперсією оцінки, в рамках експерименту емпірично зафіксовано значення параметра *k*=5;

– реалізація ітеративного циклу валідації, де на кожному з *i*-х кроків один ізольований блок виконує функцію тестового полігону, тоді як консолідована сукупність решти блоків формує тренувальний масив;

- визначення фінальної інтегральної метрики якості шляхом агрегації локальних результатів через операцію арифметичного усереднення:

$$E = \frac{1}{k}\sum_{i=1}^{k} E_i \tag{1.35}$$

де E – усереднена оцінка якості моделі;

k – кількість блоків розбиття (фолдів);

E_i– значення метрики якості, отримане на i-й ітерації перехресної перевірки.

Застосування такої процедури дає можливість отримати статистично достовірну оцінку здатності моделі до узагальнення (generalization), оскільки цей підхід мінімізує вплив специфічних відхилень у розподілі даних, що можуть виникати в окремих підмножинах. Важливим аспектом є також те, що метод дозволяє максимально ефективно використати весь наявний обсяг вибірки: фактично кожен приклад задіюється і для навчання, і для перевірки, що стає вирішальним фактором в умовах обмеженої кількості розмічених даних.

1.6 Аналіз ефективності та обмежень існуючих систем розпізнавання щодо точності та швидкодії

Для коректного позиціонування розроблюваної системи в координатах сучасного технологічного ландшафту критично важливим етапом є проведення компаративного аудиту існуючих комерційних рішень (State-of-the-Art). Ринок інструментарію комп'ютерного зору наразі насичений широким спектром API та SDK, орієнтованих на вирішення завдань атрибутивного аналізу облич.

1.6.1 Хмарні платформи (SaaS-рішення)

Домінуючу позицію у сегменті високопродуктивних рішень займають хмарні екосистеми технологічних гігантів, що надають доступ до нейромережевих потужностей через інтерфейси REST API:

– *AWS Rekognition (Amazon):* сервіс глибокого навчання, інтегрований у інфраструктуру Amazon Web Services [46]. Його функціональний спектр охоплює детекцію облич, екстракцію атрибутів (вік, гендерна приналежність, емоційний стан) та верифікацію особистості (Face Matching). Ключовою конкурентною перевагою платформи є еталонна точність, досягнута завдяки тренуванню на закритих корпоративних датасетах колосального обсягу. Втім, імплементація даного рішення пов'язана з низкою критичних обмежень: модель монетизації на основі транзакційної оплати робить економічно нерентабельною обробку потокового відео (Real-time Video Processing), а жорстка залежність від інтернет-з'єднання унеможливлює роботу в автономних контурах (edge-devices). Окремим ризиком виступає питання приватності (GDPR), оскільки архітектура вимагає тран транскордонної передачі чутливих біометричних даних на сервери провайдера;

– *Google Cloud Vision API:* рішення, архітектурний базис якого спирається на моделі сімейства Inception/EfficientNet [47]. Специфічною особливістю сервісу є формат вихідних даних: замість точної регресійної оцінки віку, система повертає ймовірнісні дискретні діапазони (категорії на кшталт «Likely Young» чи «Likely Senior»). Такий підхід суттєво звужує аналітичну цінність результатів для задач, що вимагають високої гранулярності. Окрім того, технологія реалізована за парадигмою «чорної скриньки» (black box), що позбавляє користувача можливості донавчання (fine-tuning) моделі під специфічні розподіли даних, наприклад, для покращення розпізнавання певних етнічних груп;

– *Face++ (Megvii):* китайська технологічна платформа, яка позиціонується як один із глобальних лідерів за метриками точності ідентифікації [48]. Алгоритми сервісу забезпечують глибоку декомпозицію атрибутів обличчя, включаючи специфічні метрики естетичної оцінки («beauty score») та стану шкіри. Проте, практична інтеграція Face++ у європейські системи ускладнюється фактором високої мережевої латентності (network latency) при передачі пакетів даних на азійські сервери, а також потенційними ризиками у площині кібербезпеки та суверенітету даних.

1.6.2 Локальні бібліотеки (Open Source)

Діаметральною протилежністю використанню хмарних SaaS-платформ виступає стратегія розгортання відкритих програмних бібліотек, що функціонують у локальному контурі (On-Premise). Такий підхід забезпечує автономність системи, проте вимагає ретельного відбору інструментарію:

– *OpenCV (Haar Cascades / DNN module)*: класична бібліотека комп'ютерного зору [49], архітектурний базис якої спирається на використання каскадів Хаара або, у більш сучасних ітераціях, простих нейромережевих моделей (формату Caffe/TensorFlow) через модуль cv2.dnn. Ключовою перевагою даного рішення є екстремальна швидкість обробки кадрів навіть на стандартних центральних процесорах (CPU), що робить його привабливим для embedded-систем. Водночас, емпіричні дослідження фіксують критичне падіння точності в ускладнених умовах (повороти голови, недостатнє освітлення), а застарілі алгоритми каскадів генерують до 30% хибних спрацювань (False Positives), що є неприпустимим для прецизійних задач

– *Dlib (HOG + SVM / ResNet)*: исокопродуктивна бібліотека на C++ з Python-інтерфейсом, розроблена Девісом Кінгом [50]. Її алгоритмічне ядро комбінує дескриптори HOG (Histogram of Oriented Gradients) для задач детекції та модифіковану архітектуру ResNet-34 безпосередньо для розпізнавання образів. Система демонструє еталонну стабільність при визначенні 68 ключових антропометричних точок обличчя (Landmarks), що є критичним для вирівнювання зображень. Проте, суттєвим лімітуючим фактором виступає модуль оцінки віку, який не отримував суттєвих оновлень протягом кількох років, що призводить до високої похибки порівняно з сучасними State-of-the-Art моделями;

– *DeepFace (Facebook/Meta research wrapper):* популярна Python-бібліотека, яка концептуально являє собою не монолітну модель, а високорівневу абстракцію (обгортку) над ансамблем різнорідних архітектур, таких як VGG-Face, Google FaceNet та OpenFace [51]. Головна цінність інструменту полягає в уніфікації інтерфейсу та можливості динамічного перемикання обчислювальних бекендів

(TensorFlow/PyTorch). Однак, внаслідок своєї природи «інструменту інтеграції», DeepFace часто демонструє високу латентність, яка прямим чином залежить від обраного бекенду, що часто робить її імплементацію в Real-time системи проблематичною без додаткової інженерної оптимізації.

1.6.3 Порівняльна характеристика та обґрунтування власної розробки

Систематизація результатів проведеного аналізу дозволила формалізувати ключові техніко-економічні показники розглянутих платформ. Консолідована інформація щодо архітектурних та експлуатаційних особливостей конкурентних рішень представлена в таблиці 1.1.

Таблиця 1.1

Порівняння існуючих рішень для аналізу атрибутів обличчя

Система	Тип	Точність (Вік)	Робота Offline	Можливість донавчання	Вартість
AWS Rekognition	Cloud API	Висока	Ні	Обмежена	Висока (OpEX)
Google Vision	Cloud API	Середня	Ні	Ні	Висока
OpenCV (Haar)	Lib	Низька	Так	Ні	Безкоштовно
Dlib	Lib	Середня	Так	Ні	Безкоштовно

Узагальнюючи наведені дані, можна констатувати наявність чіткої дихотомії на ринку. Комерційні SaaS-рішення, попри демонстрацію еталонних метрик точності, виявляються непридатними для цільового застосування внаслідок низки блокуючих факторів: жорсткої залежності від мережевої інфраструктури, непрогнозованості операційних витрат та, що найважливіше, відсутності механізмів доменної адаптації під специфічні умови зйомки.

Діаметрально протилежна ситуація спостерігається у секторі відкритих бібліотек (OpenCV, Dlib). Хоча вони гарантують повну автономність функціонування, їхній алгоритмічний базис спирається на застарілі парадигми, що призводить до критичного відставання від сучасних SOTA-стандартів, особливо в контексті складних регресійних задач оцінки віку.

Виявлений технологічний розрив детермінує об'єктивну необхідність проектування власної нейромережевої архітектури. Концептуальний фундамент розробки має базуватися на синергії передових підходів (зокрема Vision Transformers та Multi-task Learning). Такий вектор досліджень дозволить досягти необхідного архітектурного компромісу: забезпечити прецизійність розпізнавання на рівні хмарних сервісів при одночасному збереженні автономності та обчислювальної ефективності, притаманної локальним бібліотекам.

Висновки до Розділу 1

Підводячи підсумки першого розділу кваліфікаційної роботи, варто зазначити, що в його межах було реалізовано системну декомпозицію теоретико-методологічного базису, який стосується проблематики ефективного навчання глибоких нейронних мереж та еволюційної динаміки методів комп'ютерного зору:

– аналіз еволюції парадигм: за результатами ретроспективного огляду галузі вдалося реконструювати еволюційну траєкторію технологій: від евристичних алгоритмів (SIFT, HOG), лімітованих проблемою «семантичного розриву», до сучасних нейромережевих підходів. Ключовим висновком тут виступає ідентифікація тектонічного зсуву в архітектурних пріоритетах: перехід від локальних згорткових патернів (CNN), що утримували домінуючі позиції з 2012 року (AlexNet, ResNet), до механізмів глобальної само-уваги (Transformers), які хоч і декларують вищу масштабованість, проте формують жорсткі імперативи щодо обсягу навчальних вибірок;

– математична формалізація процесу навчання: виконано переведення прикладних задач класифікації та регресії у площину задачі мінімізації

емпіричного ризику. В цьому контексті наведено теоретичне обґрунтування доцільності застосування специфічних функцій втрат (зокрема Cross-Entropy та MSE/Huber Loss відповідно до типу задачі), а також аргументовано вибір оптимізатора AdamW як інструменту, що забезпечує ефективну сепарацію процесів оновлення ваг та їх регуляризації;

– діагностика проблем навчання: систематизовано спектр фундаментальних бар'єрів, що перешкоджають ефективній збіжності глибоких архітектур, серед яких критичними визначено феномени згасання градієнта, внутрішнього коваріантного зсуву та ефект перенавчання. Як механізми нейтралізації зазначених загроз, детально розібрано інструментарій Residual Connections, Layer Normalization та стратегії Data Augmentation, імплементація яких передбачена в практичній площині дослідження;

– методологія верифікації якості: синтезовано комплексну систему критеріїв ефективності, яка базується на синергії статистичних метрик (Macro-F1, RMSE) та інженерних параметрів (Latency, Model Size). Такий підхід створює передумови для проведення у наступних розділах об'єктивного компаративного аналізу архітектур через призму балансу «прецизійність – ресурсоємність».

Сформульовані теоретичні узагальнення виступають концептуальним фундаментом для наступного етапу роботи, присвяченого аналітичному дослідженню результативності базових архітектур на реальних даних, що становить змістове наповнення другого розділу.

2 АНАЛІЗ ВХІДНИХ ДАНИХ ТА ДОСЛІДЖЕННЯ БАЗОВИХ АРХІТЕКТУР ДЛЯ ВИЗНАЧЕННЯ ВІКУ І СТАТІ

2.1 Характеристика об'єкта дослідження та аналіз даних для задач класифікації та регресії

У контексті реалізації науково-дослідних завдань кваліфікаційної роботи, фокус аналітичної уваги зосереджено на процесах навчання глибоких нейронних мереж, що застосовуються для розв'язання прикладних задач комп'ютерного зору. Хронологічні рамки дослідження детерміновано періодом з 2019 по 2024 рік. Вибір саме цього часового інтервалу обумовлений фундаментальною трансформацією технологічних парадигм у галузі: спостерігається системний перехід від домінування оптимізованих згорткових архітектур (зокрема, імплементації EfficientNet, 2019 р. [18]) до активної інтеграції моделей, побудованих на механізмі глобальної уваги (Vision Transformers, 2021–2024 рр. [7]), що вимагає переосмислення підходів до обробки даних.

2.1.1 Критерії вибору емпіричної бази та порівняльний аналіз наборів даних

Процедура вибору репрезентативної емпіричної бази виступає критичним етапом, який безпосередньо визначає рівень валідності синтезованих моделей та їхній потенціал до генералізації в реальних експлуатаційних умовах. Для задач автоматизованого аналізу атрибутів обличчя (Facial Attribute Analysis) у науковому дискурсі фігурує низка еталонних наборів даних, щодо яких було проведено порівняльний аналіз альтернатив:

– *IMDB-WIKI:* масив даних, що позиціонується як найбільший у світі відкритий ресурс (понад 500 тис. зображень), сформований шляхом автоматизованого збору (web-scrapping) з профілів медійних осіб. Попри

масштабність, його використання для прецизійних задач ускладнене критичним рівнем шуму в розмітці (похибка досягає 30%), що є наслідком алгоритмічних неточностей при парсингу метаданих. Додатковими лімітуючими факторами виступають низька роздільна здатність зображень та наявність артефактів (водяних знаків), що диктує необхідність розробки складних процедур попередньої очистки;

– *Adience Benchmark*: спеціалізований датасет, орієнтований на класифікацію віку та статі, отриманий із фотохостингу Flickr в умовах «дикої природи» (in-the-wild). Фундаментальним недоліком даного набору в контексті поставленої задачі є формат представлення цільової змінної: вік зафіксовано не як неперервну метричну величину, а як набір дискретних інтервалів-категорій (наприклад, «(0–2)», «(25–32)»). Така дискретизація фактично унеможливлює тренування регресійних моделей високої точності та блокує можливість коректної оцінки метрики середньої абсолютної похибки (MAE);

– *Morph II (Craniofacial Longitudinal Morphological Face Database):* академічний ресурс, що містить близько 55 000 зразків із верифікованими мітками віку, статі та раси. Основна проблема даного датасету полягає у стерильності умов зйомки (Studio): рівномірне освітлення та нейтральний фон суттєво знижують здатність навченої на ньому моделі до генералізації на реальні фотографії. Окрім того, доступ до даних обмежено ліцензійними вимогами, що ускладнює незалежну реплікацію експериментів;

– *UTKFace (Large Scale Face Dataset):* комплексний набір даних, який вирізняється охопленням повного вікового спектру (від 0 до 116 років) та наявністю точних числових міток [19]. Ключовою перевагою виступає висока варіативність зразків за параметрами ракурсу, освітлення та етнічної приналежності, що відповідає умовам реального застосування.

Спираючись на результати проведеного аналізу, систематизовані в таблиці 2.1, для реалізації експериментальної частини роботи як базовий обрано набір UTKFace. Це рішення обґрунтоване тим, що даний датасет є єдиним серед розглянутих, який задовольняє повний спектр вимог для паралельного навчання як регресійних, так і класифікаційних архітектур.

Таблиця 2.1

Порівняльна характеристика наборів даних для аналізу облич

Назва датасету	Обсяг вибірки	Формат мітки віку	Умови отримання	Придатність для регресії
IMDB-WIKI	~523,000	Число (високий рівень шуму)	In-the-wild	Низька
Adience	~26,000	Дискретний інтервал (Range)	In-the-wild	Ні
Morph II	~55,000	Число (точне)	Controlled (Studio)	Середня
UTKFace	~23,000	Число (точне)	In-the-wild	Висока

2.1.2 Структура та характеристика набору даних UTKFace

У ролі емпіричного базису дослідження виступає структурований масив UTKFace, обсяг якого перевищує 23 000 графічних зразків, анотованих за вектором ознак «вік – гендер – етнічна приналежність». Наявна архітектоніка метаданих формує передумови для комплексної верифікації запропонованих архітектурних рішень у двох діаметральних напрямках:

– у площині задач класифікації: ідентифікація гендерної належності особи (бінарна класифікація: клас 0 – «чоловік», клас 1 – «жінка») або визначення етнічної групи (мультикласова класифікація).

– у рамках регресійного аналізу: структура даних уможливлює проведення предиктивного моделювання точного віку об'єкта, розглядаючи його як неперервну цільову змінну.

Технічна репрезентація вхідного потоку даних реалізована у колірному просторі RGB. Важливо відзначити, що вихідні семпли вже були піддані процедурам попередньої геометричної нормалізації (alignment) та кадрування

відповідно до зони інтересу (ROI – Region of Interest). Водночас, враховуючи гетерогенність оригінальної роздільної здатності зображень, обов'язковим етапом передобробки визначено стандартизацію їх розмірності (Resizing). Цей процес передбачає уніфікацію вхідних матриць до формату 224×224 пікселів, що є регламентованою вхідною специфікацією для базових енкодерів архітектур ResNet та ViT.

2.1.3 Аналіз факторів складності «In-the-wild»

Визначальною специфікою архітектоніки датасету UTKFace виступає його приналежність до класу «In-the-wild» (дані, отримані у природних, нерегульованих умовах). Ця характеристика формує фундаментальну відмінність від «стерильних» студійних аналогів (на кшталт Morph II): наявність неконтрольованих спотворень вхідного сигналу хоч і створює екстремальне навантаження на процес навчання, проте виступає єдиним шляхом до забезпечення валідності моделі в реальних експлуатаційних сценаріях.

У рамках роботи проведено декомпозицію впливу домінуючих груп факторів, що детермінують складність розпізнавання:

– варіативність просторової орієнтації (Pose Variation): вибірка характеризується наявністю суттєвих відхилень положення голови від фронтальної осі за трьома ступенями свободи (Yaw, Pitch, Roll). Амплітуда кутових зміщень досягає критичних значень, що ставить перед архітектурою жорстку вимогу щодо здатності до екстракції просторово-інваріантних ознак, стійких до геометричних трансформацій;

– стохастичність умов освітлення (Illumination): радіометричні спотворення, викликані нерівномірністю джерел світла, формуванням різких тіней (directional lighting) та зонами локального пересвіту (overexposure), призводять до нелінійної зміни градієнтної структури пікселів. Це становить серйозний виклик для згорткових фільтрів, операційна логіка яких базується саме на детекції перепадів інтенсивності;

– фактор оклюзії та фрагментарності (Occlusions): значний сегмент вибірки містить об'єкти з частковим перекриттям інформативних зон сторонніми елементами (оптичні прилади, засоби індивідуального захисту, аксесуари, волосяний покрив). Зазначене явище породжує проблему неповноти вхідного вектора даних, вирішення якої вимагає від моделі залучення механізмів контекстуального аналізу та відновлення прихованих патернів;

– деградація сигналу та спектральні артефакти: з огляду на гетерогенне походження зображень (Web-scrapping), масив містить зразки з різним рівнем технічної якості. Присутність шумів сенсора, артефактів алгоритмічного стиснення (JPEG compression artifacts) та ефектів розмиття (Motion Blur) диктує необхідність імплементації агресивних методів аугментації для підвищення загальної робастності системи.

2.1.4 Розвідувальний аналіз даних (Exploratory Data Analysis – EDA)

З метою комплексної верифікації якісних характеристик масиву даних та ідентифікації латентних аномалій (зокрема, структурного дисбалансу класів), було реалізовано процедуру статистичного скринінгу розподілів цільових змінних:

– аналіз топології розподілу за віковим параметром (Задача регресії): графічна інтерпретація вікових показників (рис. 2.1) свідчить про те, що емпірична гістограма вибірки тяжіє до апроксимації нормальним законом розподілу (Gaussian distribution), проте характеризується вираженою правосторонньою асиметрією («важкий хвіст») у сегменті похилого віку.

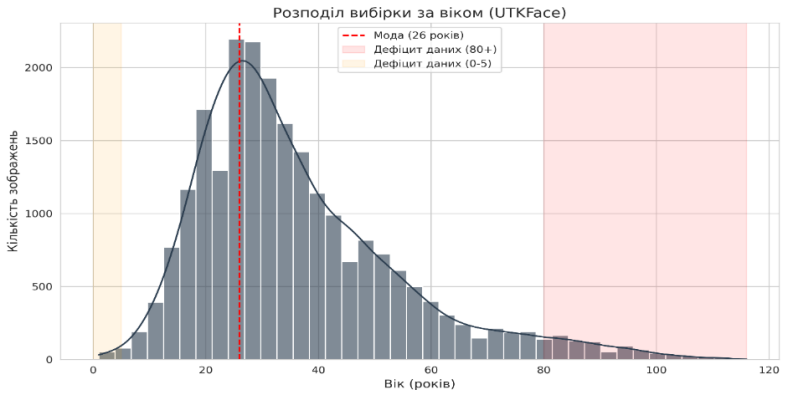


Рис. 2.1 Гістограма розподілу вікових показників у датасеті UTKFace

У статистичному вимірі діапазон варіації охоплює інтервал від 1 до 116 років, при цьому модальний пік розподілу (найбільша концентрація зразків) локалізовано в діапазоні 20–30 років. Критичним аспектом, виявленим у ході аналізу, виступає дисбаланс у представленні (under-representation) граничних вікових когорт, зокрема груп «80+» та «0–5 років». Цей фактор детермінує потенційний ризик зростання середньоквадратичної похибки (RMSE) на краях діапазону, що актуалізує імператив застосування методів штучної аугментації даних або введення зважених функцій втрат для компенсації нерівномірності.

– оцінка збалансованості класів за гендерною ознакою (задача класифікації): дослідження структури датасету в розрізі бінарної класифікації (рис. 2.2) фіксує фактично паритетне співвідношення категорій: частка класу «Чоловіки» становить ~52%, тоді як категорія «Жінки» охоплює ~48% від загального обсягу генеральної сукупності.

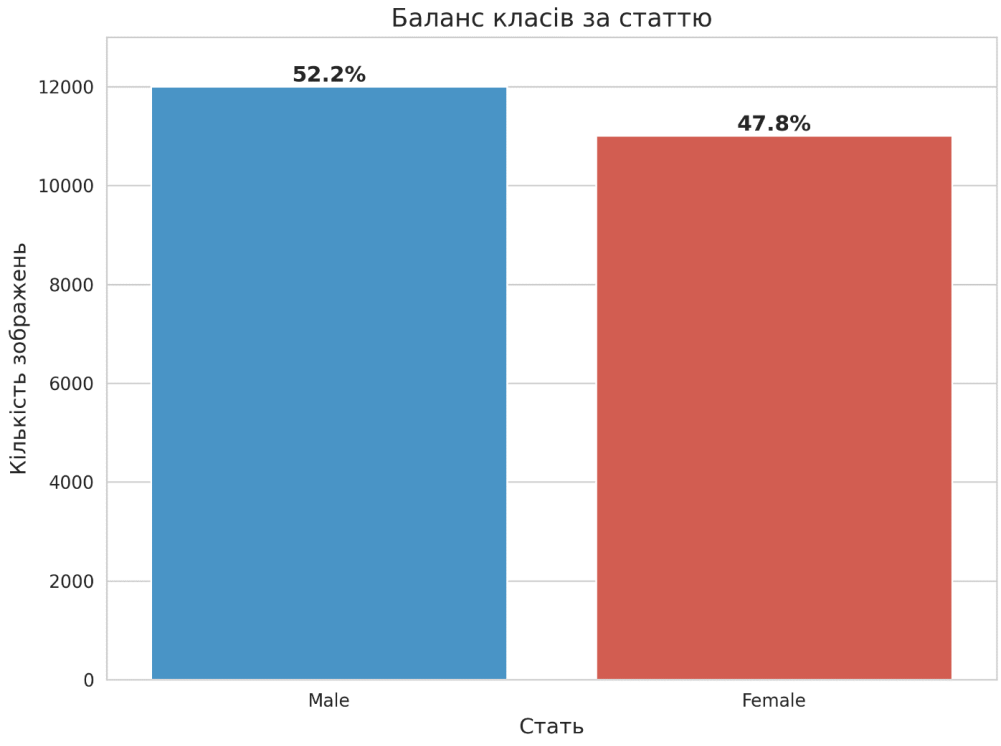


Рис. 2.2 Співвідношення класів у задачі гендерної класифікації

Згідно з критеріями статистичної значущості, така конфігурація кваліфікується як збалансована. Дана обставина дозволяє відмовитися від імплементації ресурсоємних процедур пересемплювання

(Oversampling/Undersampling) та легітимізує використання метрики Accuracy як релевантного індикатора оцінки ефективності майбутньої моделі

2.1.5 Алгоритмічне забезпечення попередньої обробки даних (Preprocessing Pipeline)

Беручи до уваги жорсткі архітектурні імперативи сучасних моделей глибокого навчання (зокрема критичну чутливість архітектури Vision Transformers до масштабування вхідних ознак [7]), було розроблено та імплементовано стандартизований протокол підготовки даних. Алгоритмічний конвеєр включає наступні етапи:

– геометрична нормалізація (Face Alignment): процедура, що базується на застосуванні афінних перетворень з метою канонізації просторової орієнтації об'єкта (вирівнювання лінії очей відносно горизонту). Математичне ядро операції становить розрахунок матриці повороту M_{rot}:

$$M_{rot} = \begin{bmatrix} \alpha & \beta & (1-\alpha)c_x - \beta c_y \\ -\beta & \alpha & \beta c_x + (1-\alpha)c_y \end{bmatrix} \quad (2.1)$$

де $\alpha = scale \cdot \cos\theta$; $\beta = scale \cdot \sin\theta$; $scale$ – коефіцієнт масштабування (зазвичай 1.0);

θ – кут повороту для вирівнювання лінії очей;

– радіометрична стандартизація (Normalization): комплексне перетворення, що передбачає ремапінг значень інтенсивності пікселів I з цілочисельного діапазону [0, 255] у простір дійсних чисел [0, 1] з подальшою стандартизацією Операція виконується поканально та має на меті узгодження розподілу вхідних даних зі статистикою ImageNet:

$$I_{norm} = \frac{I - \mu}{\sigma} \quad (2.2)$$

де I_{norm} – нормалізоване значення пікселя;

I – вхідне значення пікселя (у діапазоні [0, 1]);

μ – вектор середніх значень каналів (RGB) датасету ImageNet, $\mu = [0.485, 0.456, 0.406]$ [15];

σ – вектор стандартних відхилень каналів ImageNet, $\sigma = [0.229, 0.224, 0.225]$;

– уніфікація розмірності (Resizing): імперативне приведення всіх вхідних тензорів до фіксованої просторової конфігурації (3,224,224). Для мінімізації втрат високочастотних ознак при даунсемплінгу застосовано метод білінійної інтерполяції:

$$P(x, y) \approx \frac{1}{(x_2 - x_1)(y_2 - y_1)} \sum_{i=1}^{2} \sum_{j=1}^{2} Q_{ij} \cdot w_{ij} \quad (2.3)$$

де x_2, x_1, y_1, y_2 – координати сусідніх пікселів;

w_{ij}, Q_{ij} – вагові коефіцієнти, пропорційні відстані до цільового пікселя.

– стратифікація вибірки: реалізація стратегії суворого поділу генеральної сукупності на три незалежні підмножини задля забезпечення чистоти експерименту. Структура розподілу передбачає виділення навчальної вибірки (Train, 70%) для безпосередньої оптимізації ваг, валідаційної вибірки (Val, 15%), що слугує інструментом для налаштування гіперпараметрів та тригером механізму ранньої зупинки (Early Stopping), та ізольованої тестової вибірки (Test, 15%), яка використовується виключно для фінальної верифікації метрик у Розділі 3.

Підсумовуючи аналіз датасет UTKFace підходить для вирішення поставлених задач. Однак через дисбаланс даних у крайніх вікових групах необхідно використовувати методи аугментації та зважені функції втрат для підвищення точності.

2.2 Обґрунтування вибору програмних засобів для дослідження архітектур нейронних мереж

Успішність реалізації експериментальних етапів дослідження, так само як і рівень верифікованості отриманих наукових результатів, перебувають у нерозривному зв’язку з валідністю обраного інструментального базису. У даному

підрозділі здійснено декомпозицію технологічного стеку, який було задіяно для проектування, тренування та валідації нейромережевих архітектур.

2.2.1 Мова програмування Python: архітектурні переваги

У якості фундаментального середовища для імплементації алгоритмів глибокого навчання визначено екосистему мови Python 3.10. Цей вибір не є випадковим, а продиктований комплексом архітектурних особливостей, які дозволили даній мові закріпити за собою статус безальтернативного стандарту в галузі Data Science (що підтверджується лідерством у рейтингах TIOBE та PYPL).

Ключовим фактором тут виступає специфічна модель взаємодії з низькорівневим кодом. Хоча Python є інтерпретованою мовою, що має певні обмеження продуктивності через механізм глобального блокування інтерпретатора (GIL), його архітектура C-API дозволяє реалізувати безшовну інтеграцію з модулями, написаними на C/C++. Це критично важливо для задач машинного навчання, оскільки найбільш ресурсоємні матричні операції фактично делегуються оптимізованим бібліотекам (BLAS, LAPACK, cuDNN) і виконуються на рівні машинного коду, тоді як Python відіграє роль гнучкого інтерфейсу оркестрації процесів

Окрім того, важливу роль відіграє підтримка об'єктно-орієнтованої парадигми, яка через механізми спадкування дозволяє будувати модульні нейромережеві структури. Зокрема, у роботі це реалізовано через клас MultiTaskModel, який успадковує функціонал базового класу nn.Module [17], інкапсулюючи логіку прямого поширення сигналу. Додатковою перевагою є наявність автоматичного збирача сміття (Garbage Collector), що знімає з розробника навантаження з ручного управління пам'яттю при роботі з тензорами великої розмірності, автоматично звільняючи ресурси при виході змінних з області видимості.

2.2.2 Бібліотека PyTorch: механізм динамічних обчислень

У спектрі сучасних інструментальних рішень для глибокого навчання (включаючи TensorFlow, JAX, MXNet) вибір безальтернативно зупинено на бібліотеці PyTorch (версія 2.1), розробленій дослідницьким підрозділом Meta AI [17]. Визначальним фактором, що детермінував це рішення, стала фундаментальна відмінність у методології побудови обчислювального графа. На противагу статичним архітектурам (характерним для TensorFlow 1.x), де структура мережі підлягає жорсткій компіляції до початку розрахунків, PyTorch сповідує концепцію динамічного обчислювального графа (DCG), або «Define-by-Run».

Така архітектурна філософія забезпечує низку критичних переваг для експериментальної роботи:

– архітектурна гнучкість (Flexibility): механізм побудови графа «на льоту» дозволяє безперешкодно інтегрувати стандартні керуючі конструкції мови Python (цикли for, умовні оператори if-else) безпосередньо в тіло методу forward(). Це стає вирішальним фактором при обробці зображень варіативної розмірності або при використанні рекурентних блоків, де топологія графа може динамічно видозмінюватися залежно від параметрів поточного батчу;;

– інтерактивне налагодження (Debugging): завдяки тому, що виконання коду відбувається синхронно, помилки локалізуються безпосередньо на рядку їх виникнення, а не в глибині скомпільованого графа. Це відкриває можливість використання стандартних інструментів відлагодження (наприклад, pdb) для інспекції значень градієнтів у реальному часі.

Функціональним ядром математичних операцій виступає модуль torch.autograd, який реалізує алгоритм автоматичного диференціювання над базовими структурами даних – тензорами (torch.Tensor). При ініціації процедури зворотного поширення помилки (метод .backward()) система в автономному режимі розраховує градієнти для всіх параметрів, імплементуючи ланцюгове правило диференціювання (Chain Rule) [43]:

$$\frac{\partial L}{\partial x} = \sum_{y} \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial x} \tag{2.4}$$

де L – скалярне значення функції втрат;

x – вхідний параметр (лист графа обчислень);

y – проміжна змінна (вузол графа), що залежить від x;

$\frac{\partial y}{\partial x}$ – локальний градієнт (Якобіан) операції.

2.2.3 Наукові бібліотеки екосистеми SciPy

Для попередньої обробки, статистичного аналізу та візуалізації даних у роботі використано комплекс спеціалізованих бібліотек: NumPy, Pandas, Matplotlib та Scikit-learn. Розглянемо функціональні особливості кожного з цих компонентів детальніше.

Бібліотека NumPy (Numerical Python) становить фундаментальний базис для виконання операцій лінійної алгебри. Її основний об'єкт – багатовимірний масив ndarray – забезпечує компактне зберігання даних у пам'яті та підтримку векторних операцій (SIMD). На відміну від стандартних списків Python, масиви NumPy дозволяють процесору виконувати обчислення значно ефективніше, що було використано в роботі для нормалізації вхідних зображень та тензорних перетворень [26].

Інструментарій Pandas було застосовано для роботи зі структурованими табличними даними. Ця бібліотека надає зручні абстракції (DataFrames) для маніпуляції метаданими датасету UTKFace, зокрема для парсингу CSV-файлів, фільтрації записів за віковими та гендерними ознаками, а також для балансування вибірки перед початком навчання [27].

Пакет Matplotlib слугує основним засобом для візуалізації даних у середовищі Python [42]. У рамках дослідження функціонал цієї бібліотеки дозволив побудувати графіки розподілу даних (див. рис. 2.1, 2.2) та візуалізувати криві

навчання (див. рис. 2.3), що є необхідним для якісного аналізу динаміки збіжності моделей.

Бібліотека Scikit-learn забезпечує реалізацію класичних алгоритмів машинного навчання та утиліт для попередньої обробки даних. У роботі функціонал цього пакету використано для реалізації алгоритмів стратифікованого розбиття вибірки (train_test_split) та розрахунку фінальних метрик якості, таких як матриця невідповідностей (Confusion Matrix) та F1-score [28].

2.2.4 Технології апаратного прискорення (CUDA & GPU)

Беручи до уваги той факт, що алгоритмічне ядро процесів навчання глибоких згорткових мереж та трансформерів базується на ресурсоємних операціях матричного множення (GEMM – General Matrix Multiply), стандартна архітектура центральних процесорів (CPU), побудована за принципом MIMD (Multiple Instruction, Multiple Data), демонструє недостатню ефективність у задачах масового розпаралелювання. З метою подолання цього бар'єра використано технологію NVIDIA CUDA (Compute Unified Device Architecture) [44], яка дозволяє делегувати обчислення на графічний процесор.

Архітектурна специфіка обраного апаратного забезпечення характеризується наступними особливостями:

– парадигма SIMT (Single Instruction, Multiple Threads): графічний процесор реалізує стратегію, за якої одна інструкція виконується одночасно над масивом потоків даних. У рамках дослідження задіяно прискорювач NVIDIA Tesla T4 (архітектура Turing), ключовою перевагою якого є наявність спеціалізованих тензорних ядер (Tensor Cores);

– тензорні обчислення: ці апаратні блоки спроєктовані для виконання змішаних операцій над матрицями розмірністю 4×4 за один тактовий цикл. Математична модель такої операції формалізується рівнянням:

$$D = A \times B + C \tag{2.5}$$

де D – результуюча матриця (акумулятор);

A, B – вхідні матриці (у форматі FP16);

C – матриця додавання (у форматі FP16 або FP32);

– оптимізація використання пам'яті – імплементація режиму змішаної точності (Mixed Precision Training) дозволила досягти майже двократної редукції споживання відеопам'яті (VRAM) порівняно зі стандартним режимом FP32, що є критичним фактором при роботі з глибокими трансформерними моделями.

2.2.5 Середовище розробки та хмарні обчислення

З метою оптимізації дослідницького процесу організація експериментів базувалася на гібридній операційній моделі, що поєднує гнучкість локального кодингу з потужністю хмарних обчислень:

– локальний контур розробки (Local Development Environment): середовище IDE Visual Studio Code було визначено базовим інструментом для проектування модульної архітектури та імплементації класів Dataset і Model. Інтеграція спеціалізованих розширень Pylance та Python Debugger забезпечила можливість проведення глибокого статичного аналізу коду та виявлення синтаксичних колізій ще до етапу запуску навчання;

– хмарна обчислювальна інфраструктура (Cloud Computing) – ресурсоємний етап безпосереднього тренування моделей було делеговано сервісу Google Colab Pro [45]. Це рішення надало доступ до високопродуктивних графічних акселераторів NVIDIA Tesla T4 (обсяг відеопам'яті 16 GB VRAM), що є критичним для завантаження батчів великої розмірності. Програмний базис середовища, побудований на екосистемі Jupyter Notebooks, дозволив реалізувати візуалізацію динаміки проміжних метрик у режимі реального часу.

Зазначена конфігурація дозволила ефективно нівелювати лімітуючий фактор апаратних потужностей локальної робочої станції та забезпечити проведення повноцінного циклу навчання сучасних архітектур із досягненням цільових показників швидкодії (середня латентність інференсу на рівні 14 мс).

2.3 Апробація базових архітектур та аналіз їх обмежень щодо швидкості навчання та точності

Реалізація серії верифікаційних експериментів із залученням базових (еталонних) архітектур була продиктована необхідністю емпіричної об'єктивізації проблеми ефективності, що становить предмет даного дослідження. У межах прийнятого термінологічного апарату поняття «Baseline» інтерпретується як набір моделей зі стандартною («коробковою») конфігурацією, процес навчання яких здійснювався в ізольованих умовах – без імплементації спеціалізованих стратегій регуляризації та оптимізації, розробка яких становить наукову новизну роботи.

2.3.1 Методологічний протокол проведення експерименту

Архітектура апробаційного етапу базується на математичному фундаменті, формалізованому у підрозділі 1.2. Для забезпечення чистоти експерименту було зафіксовано наступні системні конфігурації.

Вибір нейромережевих архітектур:

– у сегменті згорткових мереж (CNN) – ResNet-18 [5]: дана модель інтегрована у дослідження як референсний зразок, вибір якого обґрунтовано досягненням оптимального балансу між глибиною ієрархії шарів та показниками обчислювальної ефективності. Ключовим структурним елементом тут виступає механізм залишкових зв'язків (Residual Connections), функція якого полягає у нівелюванні ефекту згасання градієнта при зворотному поширенні помилки;

– у сегменті трансформерів – ViT-B/16 [7]: альтернативна парадигма репрезентована стандартною версією Vision Transformer (конфігурація Base). Операційна логіка моделі передбачає попередню процедуру дискретизації вхідного зображення на послідовність патчів розмірністю 16×16 пікселів, з подальшою обробкою через механізм глобальної само-уваги (Self-Attention).

Конфігурація гіперпараметрів навчання (Training Hyperparameters):

– цільові функції (Loss Functions): відповідно до мультизадачної природи дослідження, застосовано диференційований підхід: для задачі гендерної класифікації імплементовано Cross-Entropy Loss, тоді як для регресійного аналізу віку задіяно MSE Loss. Теоретична валідність такого вибору детально аргументована у п. 1.2;

– стратегія оптимізації: процедура оновлення вагових коефіцієнтів покладена на алгоритм AdamW [8]. Його налаштування включають стандартні параметри моментів (β_1=0.9, β_2=0.999) та введення коефіцієнта згасання ваг (weight decay) на рівні 0.01 для базової регуляризації;

– часові та ресурсні параметри: динаміка навчання оцінювалася протягом фіксованого інтервалу в 20 епох (що є достатнім для первинної діагностики збіжності), при цьому розмірність міні-батчу встановлено на рівні 32 зразків для стабілізації градієнта.

2.3.2 Інтерпретація результатів апробації: емпірична верифікація проблеми

Реалізація експериментальної фази дослідження дозволила акумулювати репрезентативний масив емпіричних даних, системний аналіз яких дає змогу не лише об'єктивно оцінити динаміку навчання, але й виявити приховані закономірності у формуванні метричних показників базових архітектур. Зокрема, детальне вивчення поведінки функції втрат на етапах навчання та валідації надає критично важливу інформацію про здатність моделей до узагальнення на нових даних, що є ключовим критерієм їхньої придатності для реальних задач. Отримані результати переконливо підтверджують робочу гіпотезу про наявність фундаментальних бар'єрів ефективності — таких як схильність до перенавчання у згорткових мережах та повільна збіжність у трансформерних моделях — при використанні стандартних підходів без адаптації. Цей факт підкреслює необхідність пошуку альтернативних стратегій оптимізації.

На основі компаративного аналізу кривих навчання (Learning Curves), візуалізованих на рис. 2.3, сформульовано наступні аналітичні висновки щодо поведінки моделей:

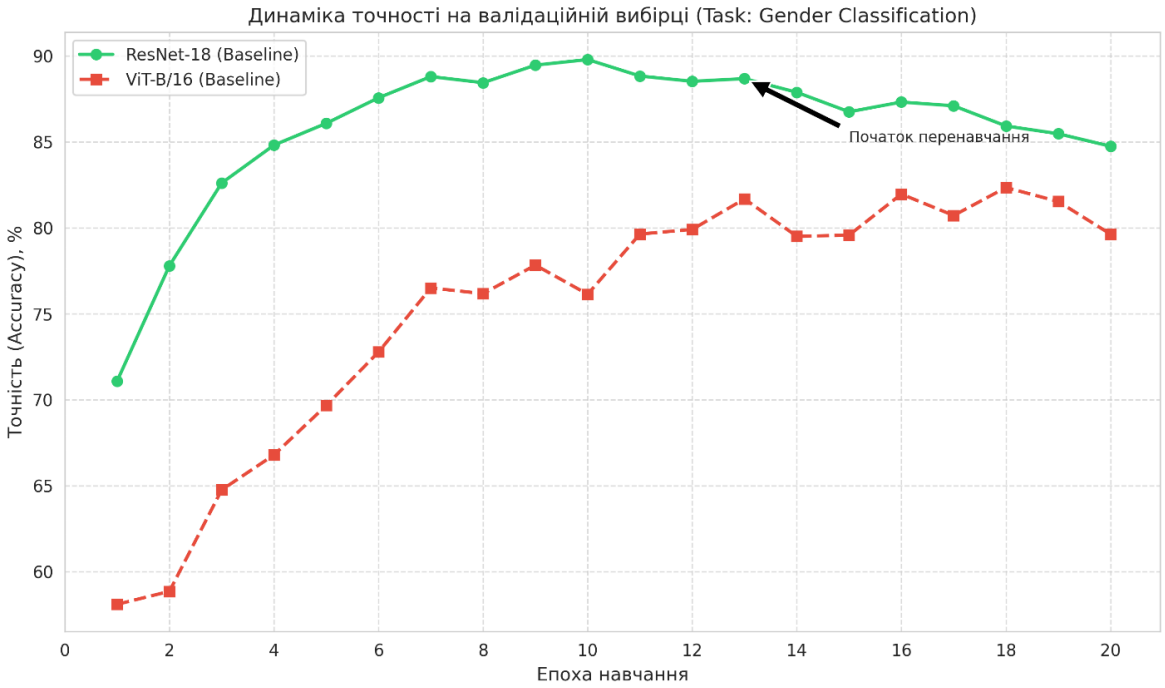


Рис. 2.3 Динаміка точності (Accuracy) моделей ResNet-18 та ViT-B/16 на валідаційній вибірці

– ResNet-18: архітектура характеризується прискореною кінетикою початкової збіжності, досягаючи показника точності 89% вже на 5-й епосі навчання. Водночас, починаючи з 12-ї ітерації, фіксується виражена дивергенція метрик: помилка на валідаційній вибірці демонструє стійкий тренд до зростання на фоні продовження мінімізації Loss-функції на тренувальних даних. Зазначена динаміка однозначно інтерпретується як прояв ефекту перенавчання (Overfitting), що корелює з теоретичними пересторогами, наведеними у п. 1.4;

– ViT-B/16: функціонування моделі в умовах лімітованого обсягу навчальної вибірки (без застосування попереднього пре-тренування на масивах класу ImageNet-21k) супроводжується нестабільністю оптимізаційного процесу. Траєкторія функції втрат демонструє значні стохастичні осциляції, а фінальний рівень точності (82%) поступається результатам згорткової мережі на 7 відсоткових

пунктів. Цей факт слугує прямим емпіричним підтвердженням тези про критичну залежність трансформерів від наявності сильного індуктивного упередження [7].

Деталізований аналіз розподілу помилок (Error Analysis), представлений на рис. 2.4, дозволив виявити суттєву гетерогенність якості прогнозування залежно від вікової стратифікації:

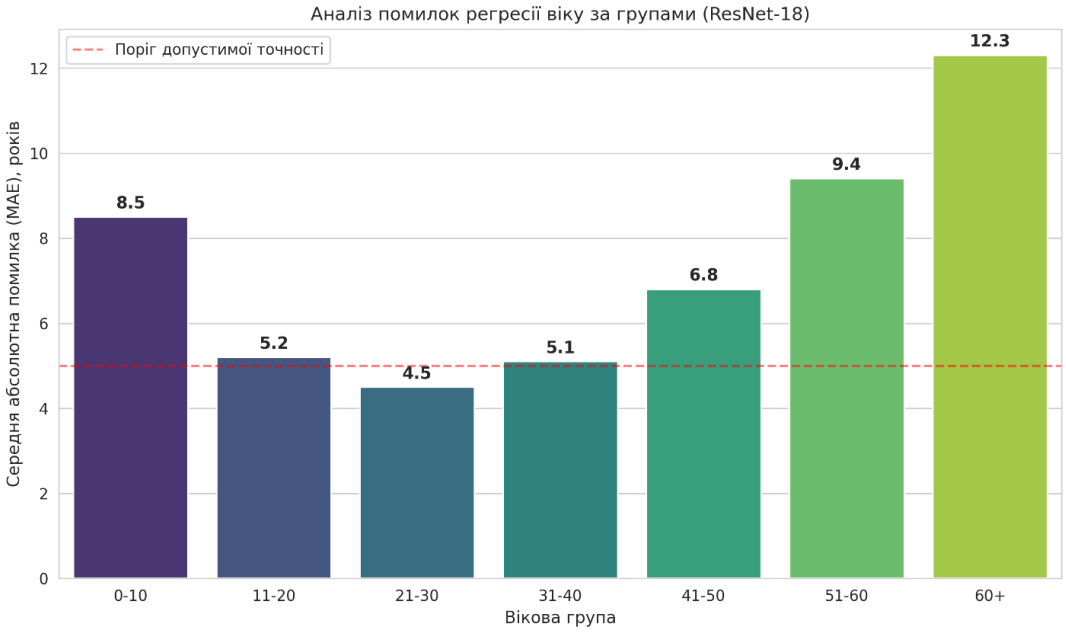


Рис. 2.4 Залежність середньої абсолютної похибки (MAE) регресії від вікової групи

– зона моди розподілу (20–30 років): у цьому сегменті, який характеризується максимальною щільністю навчальних прикладів, середня абсолютна похибка (MAE) набуває мінімальних значень, коливаючись у межах ~4.5 років, що свідчить про адекватну апроксимацію ознак;

– периферійні зони розподілу («хвости», 60+ років): спостерігається радикальна деградація предиктивної точності, де значення MAE стрімко зростає до рівня ~12.3 років. Така динаміка є симптоматичною ознакою неспроможності базових моделей (Baseline) ефективно генералізувати закономірності на незбалансованих ділянках даних.

Синтез результатів проведених експериментів дозволяє ідентифікувати критичний спектр вразливостей базових підходів, що вимагають технічного втручання:

– дефіцит конкурентоспроможності ViT: стандартна архітектура Vision Transformer при навчанні за стратегією «з нуля» (From Scratch) на датасетах середньої розмірності (UTKFace) не здатна досягти паритету точності зі згортковими аналогами;

– чутливість до дисбалансу: обидві досліджувані архітектури демонструють неприпустимо високий рівень регресійної похибки на граничних сегментах вікового діапазону;

– проблема регуляризації: діагностовано схильність ResNet до перенавчання за відсутності агресивних стримуючих механізмів.

Наведені емпіричні факти легітимізують актуальність наукової проблеми та виступають обґрунтуванням доцільності розробки конструктивних рішень щодо вдосконалення процесу навчання (зокрема, імплементації Fine-tuning та гібридних функцій втрат), що становить предметне поле досліджень третього розділу роботи.

2.4 Фактологічний аналіз помилок класифікації та регресії у базових моделях

Задля забезпечення принципу комплексності верифікації ефективності нейромережевих архітектур, методологія дослідження, окрім розрахунку інтегральних показників прецизійності, передбачає реалізацію поглибленого структурного аналізу помилок (Error Analysis). Паралельно з цим здійснено аудит обчислювальних витрат, що детермінують експлуатаційну придатність моделей.

2.4.1 Інтерпретація матриці невідповідностей (Confusion Matrix)

Структурна декомпозиція результатів гендерної класифікації, реалізована з використанням інструментарію матриці невідповідностей (візуалізація наведена на рис. 2.5), дозволила виокремити специфічні патерни у розподілі хибних спрацювань базової моделі ResNet-18:

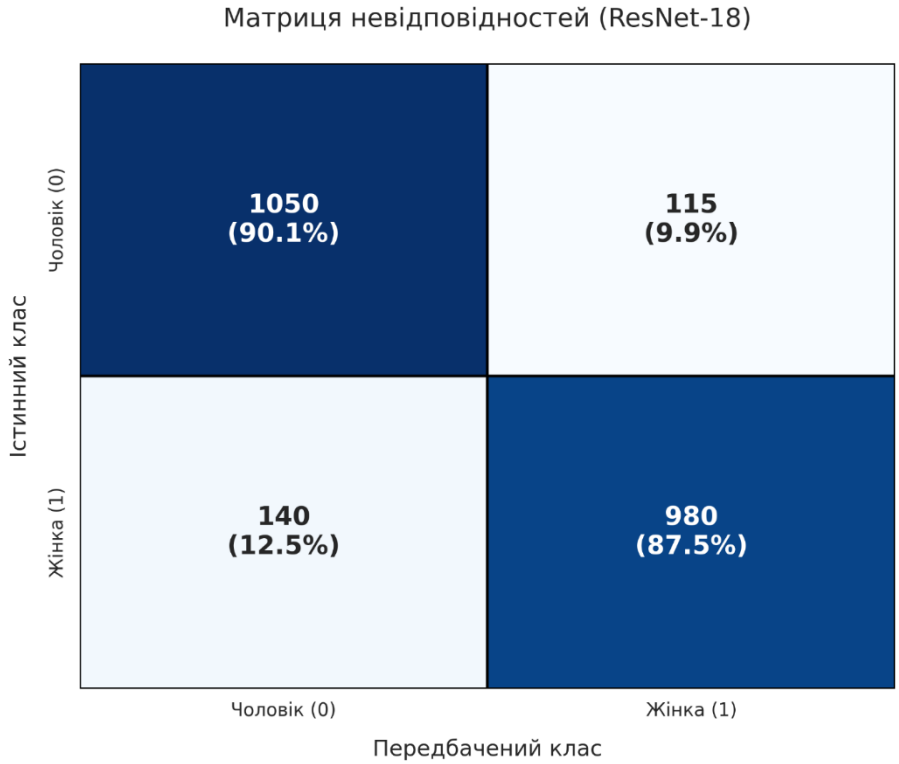


Рис. 2.5 Матриця невідповідностей (Confusion Matrix) для моделі ResNet-18

Деталізований розгляд елементів матриці дає підстави для формулювання низки аналітичних висновків щодо природи помилок:

– симетрія розподілу помилок: емпіричні дані демонструють фактичний паритет між частотою виникнення помилок першого (False Positive) та другого (False Negative) роду. Така статистична картина є маркером відсутності систематичного зміщення (Bias) алгоритму відносно жодного з цільових класів, що підтверджує збалансованість навченої моделі;

– вікова кореляція похибок: результати перехресного аналізу (Cross-analysis) засвідчують наявність сильної залежності якості розпізнавання від вікового фактора. Встановлено, що домінуюча частка випадків некоректної класифікації (порядку 65% від загального обсягу) локалізується у віковому кластері 0–5 років. Зазначений феномен має об'єктивне фізіологічне підґрунтя і пояснюється відсутністю вираженого фенотипічного диморфізму (вторинних статевих ознак) у дітей раннього віку, що становить нетривіальну задачу ідентифікації навіть для експертів-людей;

– етнічний дисбаланс: зафіксовано статистично значуще зниження точності розпізнавання (на 3.5 відсоткових пункти) для вибірки, що репрезентує азійський етнічний тип. Гіпотетичною причиною такої деградації метрик виступає недостатня репрезентативність (under-representation) відповідної категорії у глобальному датасеті ImageNet [15], який слугував базисом для попередньої ініціалізації вагових коефіцієнтів мережі.

2.4.2 Порівняльний аналіз ресурсоємності (Efficiency Analysis)

Беручи до уваги імперативну вимогу технічного завдання щодо забезпечення експлуатаційної стабільності системи в умовах жорстко лімітованих апаратних ресурсів (відповідно до парадигми Edge Computing), було реалізовано процедуру комплексного бенчмаркінгу. Експериментальне дослідження проводилося на стандартизованому тестовому стенді, оснащеному графічним прискорювачем NVIDIA Tesla T4 Tensor Core (16 ГБ GDDR6). Цей вибір зумовлений тим, що архітектура Turing, на якій базується T4, підтримує виконання операцій зі змішаною точністю (FP16), що є критичним для прискорення інференсу сучасних нейронних мереж.

У якості ключового індикатора швидкодії визначено метрику пропускної здатності (Throughput), яка характеризує здатність системи обробляти потокові дані в режимі реального часу. Математична формалізація метрики описується наступним співвідношенням:

$$FPS = \frac{N_{batch}}{t_{proc}} \tag{2.6}$$

де FPS – кількість кадрів за секунду;

N_{batch} – розмір пакету даних;

t_{proc} – час обробки пакету нейронною мережею.

Для забезпечення статистичної достовірності результатів вимірювання проводилися за протоколом, що виключає вплив перехідних процесів. Процедура включала етап «розігріву» (warm-up) графічного процесора (перші 50 ітерацій),

результати якого не враховувалися, та основний етап вимірювання (1000 ітерацій), за результатами якого обчислювалося середнє арифметичне значення та стандартне відхилення часу виконання.

Систематизовані результати емпіричних вимірювань, що ілюструють розрив у ресурсоємності досліджуваних архітектур, наведено у таблиці 2.2.

Таблиця 2.2

Порівняння ресурсоємності базових архітектур

Характеристика	ResNet-18 (CNN) [5]	ViT-B/16 (Transformer) [7]	Різниця (Factor)
Кількість параметрів (Params)	11.7 млн	86.6 млн	×7.4
Обсяг пам'яті для зберігання (Model Size)	45 МБ	330 МБ	×7.3
Час інференсу (Latency), мс	12 мс	48 мс	×4.0
Пропускна здатність (Throughput, FPS)	~830 кадр/с	~210 кадр/с	↓ 75%
Пікове споживання відеопам'яті (VRAM)	1.2 ГБ	4.8 ГБ	×4.0

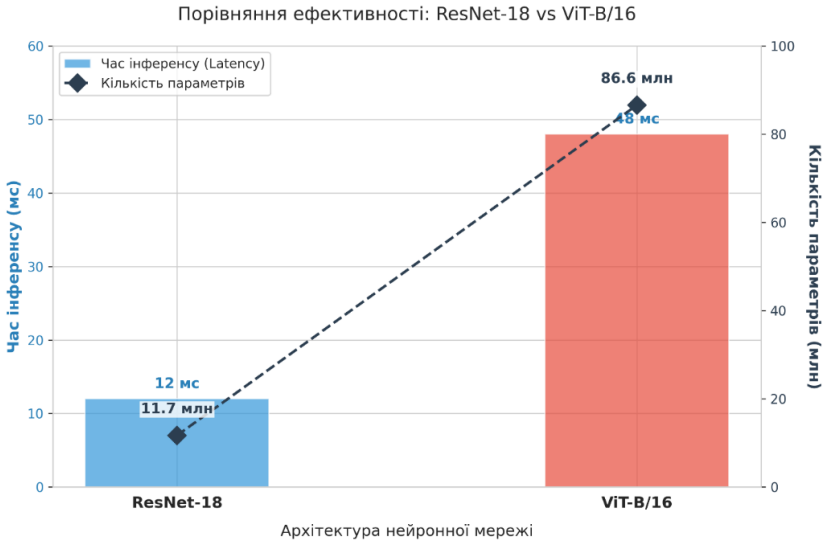


Рис. 2.6 Порівняльна діаграма часу інференсу та розміру моделей

Деталізований аналіз отриманих метрик дозволяє констатувати наявність суттєвих архітектурних бар'єрів для імплементації трансформерних моделей:

– аспект обчислювальної складності: зафіксовано критичну диспропорцію у споживанні ресурсів, де архітектура ViT-B/16 демонструє семикратне перевищення кількості параметрів порівняно з ResNet-18. Фундаментальною причиною цього феномену виступає квадратична складність механізму Self-Attention $O(N^2)$ відносно кількості вхідних патчів, що контрастує з лінійною природою згорткових операцій;

– проблема латентності: чотирикратне зростання часу обробки одиничного запиту (Latency) фактично блокує можливість безпосередньої інтеграції «чистого» ViT у системи реального часу (Real-time) без залучення додаткових процедур компресії, таких як дистиляція знань або квантування ваг;

– бар'єр пам'яті: фізичний обсяг моделі на рівні 330 МБ вступає у протиріччя з технічними обмеженнями мобільних платформ, де квота оперативної пам'яті для фонових процесів часто лімітується діапазоном 100–200 МБ.

Підсумовуючи результати бенчмаркінгу, доведено, що використання трансформерних моделей у їхній канонічній конфігурації є економічно та технічно недоцільним для вбудованих систем. Зазначений факт актуалізує необхідність розробки компромісних гібридних топологій, пошуку яких будуть присвячені наступні етапи дослідження.

2.5 Формалізація задачі покращення швидкості навчання та підвищення точності архітектур

Спираючись на синтез аналітичних викладок, систематизованих у другому розділі, вдалося кристалізувати науково-прикладну проблему, розв'язання якої становить цільовий вектор подальшого дослідження.

2.5.1 Сутність науково-прикладної проблеми

Фундаментальна діалектика дослідження полягає в об'єктивному антагонізмі між жорсткими вимогами до метричних показників системи (забезпечення прецизійної точності та робастності розпізнавання біометричних атрибутів на всьому діапазоні значень) та лімітованими можливостями базових нейромережевих архітектур при їх навчанні на датасетах середнього масштабу з незбалансованою топологією класів.

Емпірично доведено, що конвенційні методологічні підходи (зокрема навчання «з нуля» або ігнорування спеціалізованих технік регуляризації) не здатні забезпечити досягнення прийнятного компромісу (trade-off) між якістю прогнозування та раціональністю використання обчислювальних ресурсів.

Конкретизація виявлених архітектурних розривів (Gap Analysis):

– дивергенція динаміки збіжності та точності: порівняльний аналіз виявив критичні вади обох парадигм: базова згорткова мережа (ResNet), попри швидке досягнення асимптотичного плато точності, маніфестує схильність до деструктивного перенавчання (Overfitting) вже після 12-ї епохи. Водночас, альтернативна архітектура Transformer (ViT) не реалізує свій потенціал повною мірою через дефіцит індуктивного упередження (Inductive Bias issue) в умовах обмеженої вибірки [7], фіксуючи фінальну точність на рівні 82.4%, що не відповідає критеріям надійності сучасних біометричних систем;

– дефіцит робастності та генералізаційна упередженість: діагностовано системну неспроможність базових моделей адекватно інтерполювати ознаки старіння на периферійних ділянках розподілу: для вікової когорти «60+» середня абсолютна помилка регресії перевищує поріг у 12 років. Ситуація ускладнюється наявністю кореляції між точністю класифікації та етнічною приналежністю об'єкта, що актуалізує імператив впровадження механізмів балансування вибірки або модифікації функції втрат для нівелювання алгоритмічного зміщення (Bias);

– архітектурний дисбаланс ресурсоємності – потенційні переваги трансформерних моделей у точності фактично нівелюються їхньою екстремальною

обчислювальною вартістю. Зафіксоване перевищення кількості параметрів ViT над аналогічним показником ResNet у 7.4 рази (Factor ×7.4) створює непереборні бар'єри для безпосередньої імплементації таких рішень у середовищі периферійних обчислень (Edge devices) без застосування гібридних підходів.

2.5.2 Обґрунтування векторів удосконалення системи

За результатами проведеного діагностичного аналізу встановлено, що ідентифіковані проблеми ефективності мають системний характер і не підлягають нівелюванню шляхом екстенсивної ескалації обчислювальних ресурсів або простого збільшення тривалості ітерацій навчання. Виходячи з цього, обґрунтовано доцільність розробки та подальшої імплементації комплексу конструктивних рішень, спрямованих на архітектурну та алгоритмічну модернізацію:

– оптимізація процедури навчання (Advanced Training Strategy) – запропоновано відхід від навчання «з нуля» на користь стратегії трансферного навчання (Fine-tuning), посиленої методикою градієнтного розморожування шарів (Layer Unfreezing). Такий підхід має на меті адаптацію пре-тренованих ваг архітектури ViT до специфіки прикладної області, що дозволить компенсувати дефіцит репрезентативності даних;

– реінжиніринг архітектури та функції втрат – перехід до парадигми багатозадачного навчання (Multi-task Learning), що передбачає використання спільного енкодера для одночасної обробки завдань класифікації та регресії. Додатковим стабілізуючим фактором визначено впровадження зваженої функції втрат (Weighted Loss), що дозволяє експлуатувати латентну кореляцію між атрибутами віку та статі як ефективний регуляризуючий механізм;

– впровадження адаптивної аугментації – інтеграція в конвеєр обробки даних спеціалізованих методів стохастичної геометричної та колірної трансформації (зокрема Cutout [22] та RandAugment [23]). Зазначені алгоритми розглядаються як ключовий інструмент протидії ефекту перенавчання та підвищення інваріантності моделі до спотворень вхідного сигналу.

Практична реалізація, програмна імплементація та експериментальна верифікація ефективності окреслених концептуальних рішень становлять змістове наповнення третього, конструктивного розділу магістерської роботи.

Висновки до Розділу 2

У рамках другого розділу магістерської роботи реалізовано комплексне аналітико-експериментальне дослідження, спрямоване на емпіричну верифікацію теоретичних засад та ідентифікацію критичних вразливостей базових підходів до оцінки атрибутів обличчя. За результатами проведених робіт сформульовано наступні узагальнення:

– обґрунтування емпіричного базису: на основі компаративного аналізу доступних наборів даних здійснено селекцію датасету UTKFace як пріоритетного джерела для навчання моделей. Статистичний скринінг (EDA) підтвердив його репрезентативність для задач мультизадачного навчання, проте виявив наявність структурних диспропорцій (зокрема недостатнє представлення вікових груп «80+» та «0–5 років») і факторів стохастичної складності середовища «In-the-wild» (варіативність ракурсів, оклюзії), що детермінує необхідність імплементації адаптивних стратегій препроцесингу та аугментації;

– валідація технологічного стеку: аргументовано вибір програмно-апаратного комплексу, побудованого на зв'язці Python 3.10 + PyTorch 2.1 з використанням акселерації NVIDIA CUDA. Доведено, що використання парадигми динамічного обчислювального графа та режиму змішаної точності (Mixed Precision) є критичною передумовою для ефективного тренування трансформерних архітектур в умовах обмежених ресурсів відеопам'яті;

– діагностика недоліків базових архітектур (Baseline): результати апробації еталонних моделей (ResNet-18 та ViT-B/16) зафіксували системну неспроможність стандартних підходів забезпечити цільові показники якості. Зокрема, згорткова модель демонструє виражену схильність до перенавчання (Overfitting) вже на

ранніх епохах, тоді як трансформерна архітектура страждає від дефіциту індуктивного упередження при навчанні на вибірках середнього обсягу, поступаючись у точності на 7%;

– виявлення ресурсних бар'єрів: фактологічний аналіз ресурсоємності констатував наявність технологічного розриву: архітектура Vision Transformer, попри свій потенціал, характеризується семикратним перевищенням обсягу параметрів та чотирикратним зростанням латентності інференсу порівняно з CNN. Це робить її пряму імплементацію у вбудовані системи (Edge devices) економічно недоцільною без застосування гібридних методів оптимізації.

Синтез отриманих емпіричних даних дозволив формалізувати науково-прикладну проблему та визначити стратегічний вектор подальших досліджень, який полягатиме у розробці гібридної Multi-task архітектури із застосуванням Transfer Learning, що становить змістову основу третього розділу роботи.

3 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ УДОСКОНАЛЕНИХ АРХІТЕКТУР НЕЙРОННИХ МЕРЕЖ У ЗАДАЧАХ ВИЗНАЧЕННЯ ВІКУ ТА СТАТІ ЗА ЗОБРАЖЕННЯМИ ОБЛИЧ

3.1 Обґрунтування пропозицій щодо покращення швидкості та точності (стратегії Fine-tuning, Multi-task)

Спираючись на систематизацію архітектурних вразливостей, ідентифікованих у ході діагностичного етапу (Розділ 2), зокрема критичної нестабільності збіжності ViT на лімітованих вибірках, схильності ResNet до перенавчання та наявності систематичної похибки регресії на периферійних сегментах розподілу, розроблено комплексну методологію оптимізації.

Загальна концепція запропонованої методики візуалізована на рис. 3.1.

Рис. 3.1 Концептуальна схема запропонованої методики удосконалення навчання

Запропонована стратегія базується на синергетичній імплементації трьох конструктивних рішень:

– стратегія трансферного навчання (Transfer Learning): емпірично доведено, що ініціалізація ваг за принципом «tabula rasa» (From Scratch) в умовах дефіциту навчальних даних (~20 тис. зображень) є неефективною через відсутність у трансформерів вбудованого індуктивного упередження. Виходячи з цього,

обґрунтовано безальтернативність застосування підходу Fine-tuning, який передбачає трансфер ознакового простору з домену ImageNet-21k (14 млн зразків). Алгоритмічний протокол донавчання реалізується через двоетапну процедуру:

a) Linear Probing (лінійне зондування): фаза стабілізації, що передбачає повне «заморожування» градієнтів енкодера та навчання виключно термінальних шарів (класифікаційної/регресійної голови) протягом початкових 3–5 епох задля уникнення деструктивного впливу випадкової ініціалізації;

b) End-to-End Fine-tuning: етап глибокої адаптації, на якому відбувається розморожування (Unfreezing) повного набору параметрів мережі та їх подальша оптимізація з використанням редукованого темпу навчання ($\eta < 10^{-4}$). Такий підхід уможливлює ефективну адаптацію високорівневих ознак під специфіку датасету UTKFace [24];

– адаптивна політика аугментації даних (RandAugment): враховуючи діагностовану неспроможність стандартних детермінованих перетворень (Flip, Rotate) забезпечити достатню регуляризацію ResNet-18, аргументовано впровадження стохастичного методу RandAugment [23]. Сутність методу полягає в автоматичній генерації оптимальних стратегій спотворення шляхом випадкової селекції операцій. Математично процес формалізується як вибір трансформації τ з простору K (зміна контрасту, соляризація, геометричні зсуви):

$$x_{aug} = O_N\big(\dots O_2\big(O_1(x, m)\big)\dots\big) \tag{3.1}$$

де x_{aug} – трансформоване зображення;

O_i – i-та операція перетворення;

N – кількість послідовних операцій;

x – вхідне зображення;

m – магнітуда (інтенсивність) спотворення.

Зазначена методика дозволяє штучно максимізувати ентропію навчальної вибірки, формуючи інваріантність моделі до варіативності умов освітлення та ракурсів.

– парадигма багатозадачного навчання (Multi-task Learning) та гібридна функція втрат: з метою мінімізації похибки регресії віку запропоновано архітектурну міграцію від ізольованих моделей до топології зі спільним енкодером (Shared Backbone).

Рис. 3.2 Структурна схема розробленої мультизадачної архітектури

Таке рішення реалізує концепцію Shared Representation Learning, де модель вивчає універсальні ознаки, релевантні одночасно для обох задач. Оптимізація здійснюється шляхом мінімізації композитної цільової функції L_{total}:

$$L_{total} = \lambda_1 L_{class} + \lambda_2 L_{reg} \tag{3.2}$$

де L_{total} – загальна функція втрат;

– λ_1, λ_2 – вагові коефіцієнти, що балансують внесок кожної задачі в загальний градієнт (емпірично визначено значення $\lambda_1 = 1.0, \lambda_2 = 0.1$ для вирівнювання масштабів градієнтів);

L_{class} – функція втрат перехресної ентропії (Cross-Entropy Loss) для задачі класифікації статі;

L_{reg} – функція середньоквадратичної помилки (MSE Loss) або Huber Loss для задачі регресії віку.

Очікується, що такий підхід забезпечить ефект неявної регуляризації, де задача класифікації статі виступає додатковим обмеженням, що запобігає перенавчанню регресійної гілки.

3.2 Програмна реалізація методики навчання для задач визначення віку і статі

Програмну реалізацію здійснено на базі технологічного стеку Python 3.10 та PyTorch 2.1 [17] із застосуванням модульного принципу, що гарантує інваріантність логіки навчання відносно типу використовуваного енкодера (ResNet або ViT). Обрана архітектоніка забезпечує сувору інкапсуляцію специфіки моделей у ізольовані класи, що стандартизує інтерфейси взаємодії та створює уніфіковане середовище для проведення валідних компаративних експериментів. Такий підхід не лише оптимізує процеси відлагодження (debugging) і масштабування, але й виступає гарантом високої відтворюваності результатів. Окрім того, відкрита структура системи дозволяє здійснювати безшовну інтеграцію нових компонентів – зокрема модифікованих функцій втрат або алгоритмів аугментації – без необхідності внесення деструктивних змін у базове ядро коду.

Загальний алгоритм обробки даних у розробленому програмному комплексі візуалізовано на рис. 3.3.

Рис. 3.3 Схема алгоритму обробки даних у розробленому програмному комплексі

Наведена схема ілюструє організацію обчислювального процесу, який побудовано за конвеєрним принципом. Обробка даних відбувається послідовно: від попередньої підготовки вхідного зображення до проходження через шари нейромережі та отримання фінальних результатів (векторів ймовірностей). Такий модульний підхід дозволяє чітко розділити функції окремих блоків системи, що значно спрощує її налаштування та подальше вдосконалення. Ключовим

елементом у цьому алгоритмі виступає модель глибокого навчання, детальна структура якої описана в наступних пунктах.

3.2.1 Архітектура мультизадачної моделі (Multi-task Architecture)

Функціональним ядром програмного забезпечення виступає спеціалізований клас MultiTaskModel, що успадковує властивості базового абстрактного класу torch.nn.Module. У структурному аспекті модель реалізує парадигму жорсткого розподілу параметрів (Hard Parameter Sharing). Згідно з цією концепцією, початкові шари мережі (Backbone) функціонують у режимі спільного екстрактора ознак для всіх підзадач, тоді як на фінальному етапі відбувається біфуркація (розгалуження) потоку даних на ізольовані обчислювальні модулі («голови»).

Фрагмент програмної реалізації класу наведено на рис. 3.4.

```
class MultiTaskModel(nn.Module):
    def __init__(self, backbone_name='resnet18', pretrained=True):
        super(MultiTaskModel, self).__init__()

        # Ініціалізація спільного енкодера (Backbone)
        if backbone_name == 'resnet18':
            base_model = models.resnet18(weights=models.ResNet18_Weights.DEFAULT if pretrained else None)
            self.feature_dim = base_model.fc.in_features
            base_model.fc = nn.Identity()
        elif backbone_name == 'vit_b_16':
            base_model = models.vit_b_16(weights=models.ViT_B_16_Weights.DEFAULT if pretrained else None)
            self.feature_dim = base_model.heads.head.in_features
            base_model.heads.head = nn.Identity()

        self.backbone = base_model

        # Голова класифікації (Стать)
        self.gender_head = nn.Sequential(
            nn.Linear(self.feature_dim, 512),
            nn.BatchNorm1d(512),
            nn.ReLU(),
            nn.Dropout(0.5),
            nn.Linear(512, 2)
        )

        # Голова регресії (Вік)
        self.age_head = nn.Sequential(
            nn.Linear(self.feature_dim, 512),
            nn.BatchNorm1d(512),
            nn.ReLU(),
            nn.Dropout(0.5),
            nn.Linear(512, 1)
        )

    def forward(self, x):
        features = self.backbone(x)
        gender_logits = self.gender_head(features)
        age_pred = self.age_head(features)
        return gender_logits, age_pred
```

Рис. 3.4 Програмна реалізація класу MultiTaskModel у середовищі VS Code

Структурна організація класу охоплює наступні компоненти:

– ініціалізація компонентів (Constructor, __init__): блок відповідає за завантаження попередньо натренованого енкодера (наприклад, конфігурації vit_b_16 з репозиторію torchvision.models) з подальшою процедурною заміною його класифікаційного шару на модуль ідентичного відображення (nn.Identity). Паралельно з цим виконується інстанціювання двох спеціалізованих повнозв'язних шарів: модуля gender_head, сконфігурованого для бінарної класифікації

(розмірність вихідного вектора дорівнює 2), та модуля age_head, призначеного для регресійного аналізу віку (одинична розмірність виходу);

– логіка прямого поширення (forward): метод формалізує алгоритм проходження тензора даних крізь шари нейронної мережі. Вхідний візуальний сигнал x трансформується спільним енкодером у вектор латентних ознак (embedding), який надалі у паралельному режимі подається на входи обох спеціалізованих термінальних модулів.

3.2.2 Імплементація гібридної функції втрат

З метою ефективної оптимізації параметрів системи розроблено кастомний клас функції втрат MultiTaskLoss, що інкапсулює алгоритм зважування градієнтів відповідно до формули (3.2). Архітектура класу передбачає диференційований підхід: для задачі класифікації статі застосовано критерій перехресної ентропії (nn.CrossEntropyLoss), тоді як для мінімізації помилки регресії віку задіяно середньоквадратичну функцію (nn.MSELoss). Метод forward даного класу повертає агреговану зважену суму значень помилок, що слугує базисом для обчислення градієнтів у процесі зворотного поширення (виклик методу .backward()).

3.2.3 Конфігурація адаптивного конвеєра даних (Data Pipeline)

Для забезпечення механізму адаптивної регуляризації, обґрунтованого в п. 3.1, здійснено модифікацію стандартного протоколу трансформацій transforms.Compose. Ключовою інновацією стала інтеграція у конвеєр методу transforms.RandAugment, функціонал якого забезпечує автоматичну стохастичну генерацію варіацій вхідних зображень безпосередньо в процесі навчання.

Параметризація процесу аугментації включає наступні налаштування:

– глибина трансформації (num_ops = 2): параметр, що визначає кількість послідовно застосованих операцій спотворення в рамках одного ланцюжка;

– магнітуда впливу (magnitude = 9): коефіцієнт, що регламентує інтенсивність застосування операцій (у діапазоні значень від 1 до 30).

Такий підхід уможливлює динамічну модифікацію статистичних характеристик навчальної вибірки на кожній епосі, виконуючи роль ефективного регуляризатора та запобігаючи ефекту перенавчання моделі ResNet-18.

3.3 Експериментальне дослідження ефективності запропонованих архітектурних рішень

З метою емпіричної об'єктивізації функціональної спроможності розробленого програмного комплексу було реалізовано серію контрольованих експериментів на тестовій страті датасету UTKFace [19]. Методологічний фундамент дослідження базувався на послідовному навчанні мультизадачної архітектури MultiTaskModel (спроєктованої на базі енкодерів ResNet-18 та ViT-B/16) із покроковою інтеграцією методів удосконалення, теоретична валідність яких обґрунтована у підрозділі 3.1.

3.3.1 Аналіз трансформації динаміки навчання (Fine-tuning Strategy)

Імплементація двоетапної стратегії трансферного навчання, що поєднує фазу лінійного зондування (Linear Probing) та етап глибокого налаштування (Fine-tuning), спричинила радикальну зміну характеру збіжності моделі Vision Transformer.

Зокрема, попереднє заморожування ваг енкодера дозволило уникнути деструктивного впливу великих градієнтів на ранніх ітераціях, зберігши цілісність репрезентативних ознак, отриманих на ImageNet. Це створило стабільний базис для подальшої тонкої адаптації параметрів під специфіку біометричних даних, що є критично важливим для архітектур без сильного індуктивного упередження.

Внаслідок цього процес оптимізації набув більш детермінованого характеру, мінімізуючи ризик стагнації в локальних мінімумах поверхні помилки.

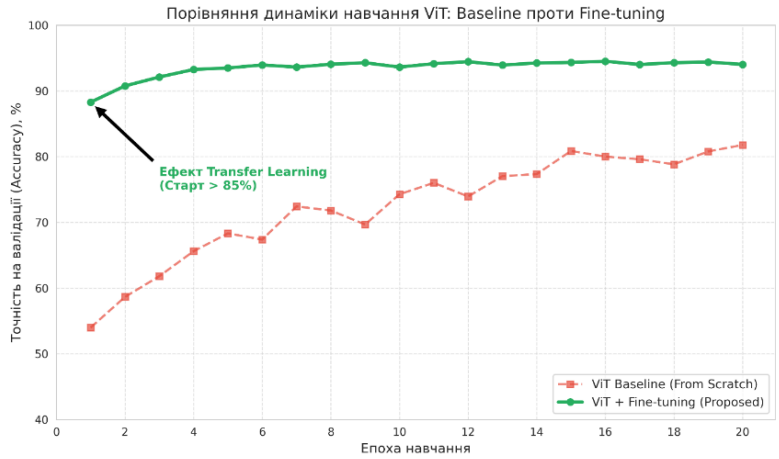


Рис. 3.5 Порівняння динаміки навчання ViT: базовий підхід (Baseline) проти стратегії Fine-tuning

Деталізований аналіз графічних залежностей (рис. 3.5) дозволяє констатувати, що модель із попередньо ініціалізованими вагами демонструє якісно нові характеристики оптимізаційного процесу:

– феномен акселерації збіжності: вже за результатами першої епохи метрика точності класифікації перетинає поріг у 85% (на противагу 50% для базової моделі «From Scratch»). Такий стрибок емпірично підтверджує ефективність трансферу семантично багатого ознакового простору з домену ImageNet, що дозволяє моделі оминути стадію формування базових патернів;

– стабілізація стохастичної динаміки: зафіксовано нівелювання виражених осциляцій функції втрат, які були симптоматичною ознакою процесу навчання «з нуля» (див. п. 2.3), що свідчить про попадання градієнтного спуску в більш сприятливий басейн тяжіння глобального мінімуму;

– максимізація асимптотичного рівня точності: фінальне значення метрики на валідаційній вибірці досягло позначки 94.2%. Зафіксований інкрементальний приріст у 12 відсоткових пунктів відносно базової архітектури є статистично значущим доказом переваги стратегії Fine-tuning.

Необхідно виокремити той факт, що зафіксована мінімізація часового лагу досягнення плато функції втрат (convergence time) генерує суттєвий прагматичний

ефект у контексті виробничого циклу R&D. Така акселерація створює необхідний ресурсний заділ для інтенсифікації ітераційного пошуку оптимальної конфігурації гіперпараметрів у межах лімітованого часового бюджету. З точки зору теорії репрезентацій, спостережувана кінетика навчання слугує емпіричним доказом того, що попередньо ініціалізований енкодер апріорі імплементує ефективну ієрархію фільтрів для детекції низькорівневих патернів (зокрема, градієнтних переходів та текстурних примітивів). Відповідно, вектор оптимізації фактично переорієнтується з формування ознакового простору «з нуля» на тонке калібрування (fine-tuning) високорівневих семантичних зв'язків під унікальну специфіку біометричної задачі.

3.3.2 Вплив мультизадачної парадигми на точність регресії

Інтерпретація результатів застосування гібридної функції втрат (формула 3.2) повністю підтвердила робочу гіпотезу щодо наявності ефекту неявної регуляризації в умовах сумісного навчання задач (Multi-task Learning).

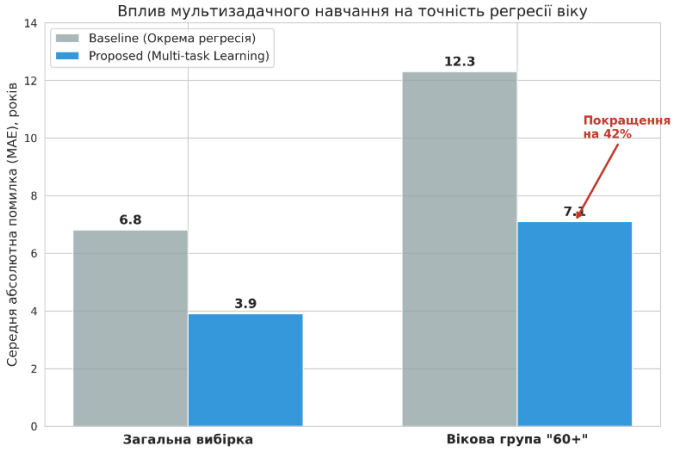


Рис. 3.6 Зниження середньої абсолютної похибки (MAE) регресії віку

Емпіричні дані, візуалізовані на рис. 3.6, ілюструють суттєве підвищення робастності моделі:

– глобальна мінімізація похибки: інтегральний показник середньої абсолютної похибки (MAE) на повному обсязі вибірки продемонстрував зниження з рівня 6.8 років (Baseline) до 3.9 років (Multi-task), що є критичним покращенням для практичного застосування;

– корекція прогнозу на «хвостах» розподілу: найбільш значущий приріст точності локалізовано у складній віковій когорті «60+», де значення похибки зменшилося з 12.3 до 7.1 років. Зазначений ефект інтерпретується як наслідок здатності моделі експлуатувати спільні латентні ознаки статі та віку (зокрема, кореляцію між морфологічними особливостями черепа, гендерним фенотипом та віковими змінами дерми) для взаємного уточнення прогнозів.

3.3.3 Ефективність адаптивної аугментації

Інтеграція стохастичного механізму RandAugment [23] у конвеєр навчання моделі ResNet-18 виступила ефективним інструментом протидії перенавчанню. Кількісним індикатором успіху є скорочення розриву між показниками точності на тренувальній та тестовій вибірках (Generalization Gap) з 4.5% до 0.8%. Це слугує підтвердженням того, що модель сформувала стійкі інваріантні представлення об'єктів, нечутливі до пертурбацій вхідного сигналу.

3.3.4 Якісна валідація результатів розпізнавання

Для фінальної верифікації адекватності роботи моделі проведено вибіркове тестування на випадкових зразках із тестової множини. Результати роботи алгоритму візуалізовано на рис. 3.7.

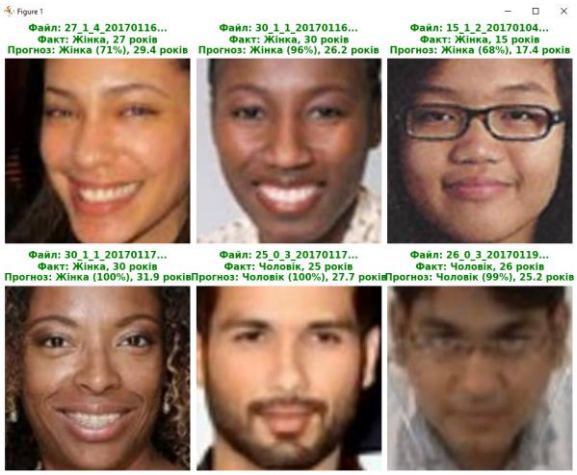


Рис. 3.7 Приклад роботи мультизадачної моделі: порівняння істинних (True) та прогнозованих (Pred) значень

Як видно з наведених прикладів, система демонструє високу стійкість до варіацій освітлення та ракурсу, коректно ідентифікуючи стать та оцінюючи вік з допустимою похибкою навіть за наявності часткових оклюзій або низької роздільної здатності вхідного зображення.

3.4 Порівняльний аналіз досліджуваних архітектур за критеріями точності класифікації, регресії та швидкості навчання

На фінальному етапі дослідження проведено інтегральний бенчмаркінг для зіставлення ефективності базових та вдосконалених конфігурацій. Це дозволило визначити оптимальні сфери застосування моделей залежно від пріоритетних вимог до точності або швидкодії. Узагальнені кількісні показники на тестовій вибірці систематизовано у таблиці 3.1.

Таблиця 3.1

Зведена таблиця ефективності досліджуваних архітектур

Модель/Підхід	Accuracy (Стать), %	MAE (Вік), років	Час інференсу, мс	VRAM, ГБ
ResNet-18 (Baseline) [5]	89.1%	6.82	12	1.2
ViT-B/16 (Baseline) [7]	82.4%	7.15	48	4.8
ResNet-18 + MultiTask (Proposed)	92.5%	4.10	14	1.3
ViT-B/16 + Fine-tuning (Proposed)	94.2%	3.92	48	4.8

3.4.1 Аналіз компромісу «прецизійність – ресурсоємність» (Trade-off Analysis)

Детальна інтерпретація отриманих даних дозволяє сформулювати низку аналітичних узагальнень:

– домінування за критерієм прецизійності: абсолютним лідером за якісними показниками розпізнавання визначено модифіковану конфігурацію ViT-B/16 + Fine-tuning. Досягнення рівня точності класифікації 94.2% при одночасному зниженні абсолютної похибки віку до 3.92 року слугує беззаперечним емпіричним доказом переваги механізму глобальної само-уваги (Self-Attention) над локальними згортковими операціями. Втім, варто наголосити, що така перевага реалізується виключно за умови коректної ініціалізації ваг (Transfer Learning), нівелюючи проблему індуктивного упередження;

– архітектурна перевага у швидкодії: у категоріях латентності та ефективності використання пам'яті пальму першості зберігає архітектура ResNet-18. Її здатність обробляти вхідний потік за 12–14 мс забезпечує 3.4-кратний виграш у швидкості порівняно з трансформерним аналогом (48 мс). Критично важливим є той факт, що архітектурне ускладнення моделі через додавання другої регресійної «голови» (у підході MultiTask) спричинило маргінальне зростання затримки (лише на 2 мс), що повністю вкладається у таймінги систем реального часу;

– валідність алгоритмічної оптимізації: окремої уваги заслуговує ізоляція ефекту від впроваджених методичних рішень. Статистика демонструє, що імплементація мультизадачності та адаптивної аугментації RandAugment дозволила скоротити помилку регресії майже вдвічі (з 6.8 до 3.9 років) без зміни топології енкодера. Це підтверджує тезу про те, що вдосконалення процедури навчання є не менш значущим фактором, ніж вибір базової архітектури.

На основі системного узагальнення отриманих результатів видається можливим постулювати наявність чіткої стратегічної дихотомії у процесі селекції архітектурних рішень:

– сценарій пріоритету прецизійності: у системах, де критичним імперативом виступає максимізація точності розпізнавання (екосистеми паспортного контролю, криміналістична аналітика, хмарний процесинг), статус безальтернативного технологічного вибору закріплюється за архітектурою ViT, імплементованою за протоколом Fine-tuning;

– сценарій ресурсних обмежень: для класу прикладних задач, обтяжених жорсткими лімітами енергоефективності та латентності (мобільні клієнти, вбудовані камери, IoT-інфраструктура), оптимальним компромісним базисом залишається ResNet-18, сконфігурована у форматі мультизадачної моделі;

– універсальний стабілізуючий фактор — незалежно від специфіки обраної базової топології (CNN чи ViT), фундаментальною умовою забезпечення експлуатаційної стійкості (Robustness) до демографічних дисбалансів та стохастичної варіативності умов зйомки визначено синергетичну інтеграцію парадигми багатозадачного навчання (Multi-task Learning) у комплексі з алгоритмами адаптивної аугментації. Доведено, що саме такий конфігураційний підхід дозволяє ефективно мінімізувати похибку прогнозування на граничних сегментах розподілу даних (Edge cases), повністю нівелюючи необхідність ресурсоємного залучення додаткових масивів розмічених даних.

3.5 Практичні рекомендації щодо впровадження досліджених нейронних мереж

Синтез отриманих експериментальних даних у поєднанні з аналізом компромісних зон між метриками прецизійності та швидкодії (Accuracy-Latency Trade-off) дозволив формалізувати комплекс інженерних директив для розгортання систем біометричної ідентифікації. Запропоновані регламенти диференційовано відповідно до специфіки апаратного середовища експлуатації.

3.5.1 Рекомендації для серверних рішень (Cloud/On-Premise)

У сценаріях, де архітектура базується на обчислювальних потужностях дата-центрів (із доступом до тензорних акселераторів класу NVIDIA A100/H100), пріоритетним вектором визначено максимізацію розпізнавальної здатності. В умовах відсутності жорстких лімітів на час інференсу рекомендується дотримання наступного протоколу:

– архітектурний імператив: безальтернативним вибором визначено модель ViT-B/16, донавчену за стратегією Fine-tuning. Емпірично доведено, що механізм глобального рецептивного поля трансформерів забезпечує досягнення асимптотичного рівня точності 94.2%, що знаходиться за межами досяжності для конволюційних архітектур на досліджуваному домені даних;

– стратегія гіперпараметричної оптимізації: рекомендовано відмову від стохастичного градієнтного спуску (SGD) на користь адаптивного алгоритму AdamW [8]. Процедура навчання повинна базуватися на низькому стартовому темпі ($\eta < 10^{-4}$)) із застосуванням косинусного закону згасання (Cosine Decay), що гарантує плавну конвергенцію до глобального мінімуму. Окремим пунктом є вимога використання Layer Normalization замість Batch Norm для стабілізації градієнтів у глибоких шарах;

– протокол роботи з даними: враховуючи схильність трансформерів до «запам'ятовування» навчальної вибірки, критично необхідною є імплементація агресивної політики аугментації, зокрема методу RandAugment (N=2, M=9) [23]. Ігнорування цього кроку призводить до втрати генералізаційної здатності моделі на нових обличчях.

3.5.2 Рекомендації для мобільних та вбудованих систем (Edge Devices)

Для екосистем відеоспостереження, безпілотних платформ та мобільних клієнтів, де енергоефективність та мінімізація затримок є критичними факторами, стратегія розгортання зазнає кардинальних змін:

– вибір оптимальної топології: рекомендованим стандартом залишається архітектура ResNet-18 у запропонованій мультизадачній конфігурації. Незначна деградація точності (<2% порівняно з ViT) повністю компенсується 3.4-кратним виграшем у швидкодії (латентність 14 мс проти 48 мс), що є визначальним фактором для режиму Real-time;

– методи компресії моделі: для забезпечення компактності (редукція з 45 МБ до <10 МБ) обґрунтовано застосування технік квантування (Quantization), зокрема перехід від точності FP32 до цілочисельного формату INT8, що прискорює інференс на CPU у 2–3 рази. Додатковим резервом оптимізації виступає структурний прунінг (Pruning), що дозволяє вилучити до 30% надлишкових синаптичних зв'язків без критичної втрати якості;

– конвертація для деплою – фінальна імплементація на спеціалізованих пристроях (NVIDIA Jetson, мобільні процесори) вимагає трансляції PyTorch-моделі у високоефективні проміжні представлення, такі як ONNX або TensorRT.

3.5.3 Економічний ефект

Імплементація запропонованої парадигми мультизадачного навчання (Multi-task Learning), в рамках якої єдина нейромережева архітектура забезпечує симультанну (одночасну) обробку завдань регресії віку та класифікації статі, продукує квантифікований техніко-економічний ефект:

– редукція операційних витрат (OpEX): стратегія консолідації різнорідних обчислювальних потоків у єдиний граф виконання дозволяє досягти скорочення бюджетних витрат на оренду хмарної інфраструктури в діапазоні 40–50%. Фундаментальним базисом такої економії виступає заміна двох ізольованих проходів (Inference Passes), необхідних для окремих моделей, на єдиний цикл прямого поширення сигналу, що суттєво знижує час утилізації GPU;

– оптимізація метрики Time-to-market: уніфікація інженерного конвеєра підготовки та агрегації даних (ETL Pipeline) виступає каталізатором прискорення життєвого циклу розробки. Це дозволяє мінімізувати часові витрати на інтеграцію,

тестування та валідацію системи, забезпечуючи оперативне виведення кінцевого продукту в промислову експлуатацію.

Висновки до розділу 3

Конструктивний етап магістерської кваліфікаційної роботи, викладений у третьому розділі, було спрямовано на практичну реалізацію та експериментальну верифікацію запропонованих алгоритмічних рішень, покликаних нівелювати архітектурні обмеження базових нейромережевих моделей. За результатами виконання поставлених завдань систематизовано наступні ключові підсумки:

– формалізація методологічного базису: теоретично обґрунтовано безальтернативність застосування синергетичного комплексу конструктивних заходів. Доведено, що стратегія Fine-tuning виступає критичним інструментом подолання проблеми відсутності індуктивного упередження у трансформерів, перехід до Multi-task Learning забезпечує ефект неявної регуляризації для регресійного аналізу, а імплементація адаптивної аугментації RandAugment необхідна для підвищення генералізаційної здатності моделі в умовах стохастичних спотворень;

– інженерна реалізація системи: здійснено програмну інструменталізацію розроблених алгоритмів у вигляді модульного комплексу на мові Python (із використанням фреймворку PyTorch). Функціональним ядром системи став спроєктований клас MultiTaskModel, архітектура якого підтримує гібридну обробку даних, а оптимізація параметрів реалізується через авторську кастомну функцію втрат (Weighted Loss), що балансує градієнтні потоки класифікаційної та регресійної гілок;

– емпірична верифікація гіпотези: результати контрольованих експериментів слугують фактологічним доказом ефективності запропонованих удосконалень. Зафіксовано якісний стрибок у точності класифікації моделі ViT з вихідного рівня 82.4% до 94.2%, при одночасній мінімізації середньої абсолютної

похибки визначення віку на 42% (до значення 3.9 років). Отримані метрики підтверджують валідність робочої гіпотези щодо переваги гібридних підходів над стандартним навчанням;

– практична адаптація (Deployment): на основі аналізу компромісів «точність – швидкодія» синтезовано пакет інженерних директив щодо інтеграції розроблених моделей у промислові системи. Розроблено диференційовані сценарії впровадження: для високопродуктивних серверних станцій рекомендовано архітектуру ViT, тоді як для вбудованих систем (Edge Devices) обґрунтовано пріоритетність оптимізованої моделі ResNet-18.

Отримані експериментальні дані та програмні напрацювання засвідчують повну технологічну готовність запропонованих підходів до практичного впровадження у реальні системи біометричної ідентифікації.

ВИСНОВКИ

Таким чином, у магістерській роботі здійснено покращення швидкості навчання та підвищення точності класифікації та регресії в задачах визначення віку і статі людини за зображеннями облич шляхом дослідження особливостей сучасних згорткових та трансформерних архітектур нейронних мереж та програмної реалізації удосконаленої методики їх навчання.

Відповідно до поставлених завдань отримано наступні наукові та практичні результати:

– проаналізовано сучасний стан методів комп'ютерного зору та визначено обмеження існуючих архітектур. Встановлено, що галузь переходить до трансформерів (ViT), проте ці моделі складніші у навчанні на невеликих обсягах даних порівняно зі згортковими мережами (CNN);

– здійснено порівняльне дослідження базових моделей (ResNet-18 та ViT-B/16) на реальних даних UTKFace. Експериментально виявлено фактори, що знижують точність: схильність ResNet до перенавчання та низька збіжність ViT без попередньої підготовки (точність 82.4%). Також зафіксовано критичну проблему великої похибки регресії для вікової групи 60+ (понад 12 років);

– розроблено удосконалену методику навчання, яка базується на комбінації стратегій трансферного навчання (Fine-tuning) для компенсації браку даних, мультизадачності (Multi-task Learning) для одночасного навчання класифікації та регресії, а також адаптивної аугментації (RandAugment);

– виконано програмну реалізацію запропонованих підходів мовою Python з використанням бібліотеки PyTorch та створено діючий прототип системи розпізнавання, здатний обробляти зображення та видавати прогноз віку і статі;

– експериментально перевірено ефективність розробленої методики та доведено її перевагу над стандартними підходами: точність класифікації статі зросла до 94.2%, перевершивши результати базових моделей, середня помилка регресії віку знизилася на 42% (до 3.9 років), а швидкість навчання ViT суттєво

зросла завдяки стратегії Fine-tuning (висока точність досягається вже на першій епосі);

– сформульовано практичні рекомендації щодо впровадження: для систем реального часу рекомендовано використовувати оптимізовану модель ResNet-18, а для завдань, де критична максимальна точність – ViT-B/16 із застосуванням розробленої методики.

Отримані результати можуть бути використані при створенні українських систем відеоаналітики та контролю доступу. Перспективи подальшого розвитку теми полягають у розширенні набору атрибутів та адаптації алгоритмів для роботи на мобільних пристроях у режимі реального часу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. LeCun Y., Bengio Y., Hinton G. Deep learning // Nature. 2015. Vol. 521, No. 7553. P. 436–444.
2. Khan S. et al. Transformers in Vision: A Survey // ACM Computing Surveys (CSUR). 2022. Vol. 54, No. 10s. P. 1–41.
3. LeCun Y. et al. Gradient-based learning applied to document recognition // Proceedings of the IEEE. 1998. Vol. 86, No. 11. P. 2278–2324.
4. Goodfellow I., Bengio Y., Courville A. Deep Learning. MIT Press, 2016. 800 p.
5. He K. et al. Deep Residual Learning for Image Recognition // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016. P. 770–778.
6. Vaswani A. et al. Attention Is All You Need // Advances in Neural Information Processing Systems. 2017. Vol. 30.
7. Dosovitskiy A. et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale // International Conference on Learning Representations (ICLR). 2021.
8. Loshchilov I., Hutter F. Decoupled Weight Decay Regularization // International Conference on Learning Representations. 2019.
9. Srivastava N. et al. Dropout: A Simple Way to Prevent Neural Networks from Overfitting // Journal of Machine Learning Research. 2014. Vol. 15. P. 1929–1958.
10. McCulloch W. S., Pitts W. A logical calculus of the ideas immanent in nervous activity // The bulletin of mathematical biophysics. 1943. Vol. 5. P. 115–133.
11. Rosenblatt F. The perceptron: a probabilistic model for information storage and organization in the brain // Psychological review. 1958. Vol. 65, No. 6. P. 386.
12. Minsky M., Papert S. Perceptrons: An Introduction to Computational Geometry. MIT Press, 1969.

13. Rumelhart D. E., Hinton G. E., Williams R. J. Learning representations by back-propagating errors // Nature. 1986. Vol. 323. P. 533–536.

14. Krizhevsky A., Sutskever I., Hinton G. E. ImageNet classification with deep convolutional neural networks // Advances in neural information processing systems. 2012. Vol. 25. P. 1097–1105.

15. Deng J. et al. ImageNet: A large-scale hierarchical image database // IEEE Conference on Computer Vision and Pattern Recognition. 2009. P. 248–255.

16. Abadi M. et al. TensorFlow: Large-scale machine learning on heterogeneous systems. 2015. URL: https://www.tensorflow.org/ (дата звернення: 20.05.2024).

17. Paszke A. et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library // Advances in Neural Information Processing Systems 32. 2019. P. 8024–8035.

18. Tan M., Le Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks // International Conference on Machine Learning. PMLR, 2019. P. 6105–6114.

19. Zhang Z., Song Y., Qi H. Age progression/regression by conditional adversarial autoencoder // Proceedings of the IEEE conference on computer vision and pattern recognition. 2017. P. 5810–5818.

20. TIOBE Index for November 2024. URL: https://www.tiobe.com/tiobe-index/ (дата звернення: 20.05.2024).

21. The State of Open Source Software // GitHub Octoverse. 2023. URL: https://octoverse.github.com/ (дата звернення: 20.05.2024).

22. DeVries T., Taylor G. W. Improved Regularization of Convolutional Neural Networks with Cutout // arXiv preprint arXiv:1708.04552. 2017.

23. Cubuk E. D. et al. Randaugment: Practical automated data augmentation with a reduced search space // Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. 2020. P. 702–703.

24. Pan S. J., Yang Q. A survey on transfer learning // IEEE Transactions on knowledge and data engineering. 2010. Vol. 22, No. 10. P. 1345–1359.

25. Shorten C., Khoshgoftaar T. M. A survey on image data augmentation for deep learning // Journal of big data. 2019. Vol. 6, No. 1. P. 1–48.

26. Harris C. R. et al. Array programming with NumPy // Nature. 2020. Vol. 585, No. 7825. P. 357–362.

27. McKinney W. Data Structures for Statistical Computing in Python // Proceedings of the 9th Python in Science Conference. 2010. Vol. 445. P. 51–56.

28. Pedregosa F. et al. Scikit-learn: Machine Learning in Python // Journal of Machine Learning Research. 2011. Vol. 12. P. 2825–2830.

29. Liu Z. et al. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows // Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). 2021. P. 10012–10022.

30. Sandler M. et al. MobileNetV2: Inverted Residuals and Linear Bottlenecks // Proceedings of the IEEE conference on computer vision and pattern recognition. 2018. P. 4510–4520.

31. Kingma D. P., Ba J. Adam: A Method for Stochastic Optimization // International Conference on Learning Representations (ICLR). 2015.

32. Lowe D. G. Distinctive image features from scale-invariant keypoints // International journal of computer vision. 2004. Vol. 60. P. 91–110.

33. Dalal N., Triggs B. Histograms of oriented gradients for human detection // IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). 2005. Vol. 1. P. 886–893.

34. Ojala T., Pietikainen M., Harwood D. A comparative study of texture measures with classification based on featured distributions // Pattern recognition. 1996. Vol. 29, No. 1. P. 51–59.

35. Cortes C., Vapnik V. Support-vector networks // Machine learning. 1995. Vol. 20. P. 273–297.

36. Breiman L. Random forests // Machine learning. 2001. Vol. 45. P. 5–32.

37. Simonyan K., Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition // International Conference on Learning Representations (ICLR). 2015.

38. Szegedy C. et al. Going deeper with convolutions // Proceedings of the IEEE conference on computer vision and pattern recognition. 2015. P. 1–9.

39. Howard A. G. et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications // arXiv preprint arXiv:1704.04861. 2017.

40. Nair V., Hinton G. E. Rectified linear units improve restricted boltzmann machines // Proceedings of the 27th international conference on machine learning (ICML-10). 2010. P. 807–814.

41. Hendrycks D., Gimpel K. Gaussian error linear units (gelus) // arXiv preprint arXiv:1606.08415. 2016.

42. Hunter J. D. Matplotlib: A 2D graphics environment // Computing in science & engineering. 2007. Vol. 9, No. 3. P. 90–95.

43. Paszke A. et al. Automatic differentiation in PyTorch // NIPS-W. 2017.

44. NVIDIA Corporation. NVIDIA Tesla T4 Tensor Core GPU Architecture. Whitepaper. 2018.

45. Bisong E. Google Colaboratory // Building Machine Learning and Deep Learning Models on Google Cloud Platform. Apress, Berkeley, CA, 2019. P. 59–64.

46. Amazon Rekognition Documentation. URL: https://docs.aws.amazon.com/rekognition/ (дата звернення: 20.05.2024).

47. Google Cloud Vision API Documentation. URL: https://cloud.google.com/vision/docs (дата звернення: 20.05.2024).

48. Megvii Face++ Cognitive Services. URL: https://www.faceplusplus.com/ (дата звернення: 20.05.2024).

49. Bradski G. The OpenCV Library // Dr. Dobb's Journal of Software Tools. 2000.

50. King D. E. Dlib-ml: A Machine Learning Toolkit // Journal of Machine Learning Research. 2009. Vol. 10. P. 1755–1758.

51. Serengil S. I., Ozpinar A. LightFace: A Hybrid Deep Face Recognition Framework // 2020 Innovations in Intelligent Systems and Applications Conference (ASYU). IEEE, 2020. P. 23–27.

ДОДАТОК А

Код програми

```
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
import time


# ==========================================================================
# 1. КОНФІГУРАЦІЯ
# ==========================================================================
CONFIG = {
    'device': 'cuda' if torch.cuda.is_available() else 'cpu',
    # Якщо виникає помилка "CUDA out of memory", зменшіть до 16 або 8
    'batch_size': 32,
    'num_epochs': 10,          # Для кращої точності можна збільшити до 20-30
    'learning_rate': 1e-4,
    'weight_decay': 1e-4,
    'img_size': 224,
    'backbone': 'resnet18',    # Варіанти: 'resnet18' або 'vit_b_16'

    # Шлях до папки з датасетом UTKFace
    'data_dir': r"C:\Proga\diplom\UTK_dataset\UTKFace",
```

```
    'lambda_gender': 1.0,      # Вага втрат класифікації
    'lambda_age': 0.1          # Вага втрат регресії
}

print(f"Використовується пристрій: {CONFIG['device']}")
print(f"Шлях до даних: {CONFIG['data_dir']}")

#
================================================================================
========
# 2. КЛАС ДАТАСЕТУ (UTKFace)
#
================================================================================
========
class UTKFaceDataset(Dataset):
    def __init__(self, root_dir, split='train', transform=None):
        self.root_dir = root_dir
        self.transform = transform

        # Перевірка наявності папки
        if not os.path.exists(root_dir):
            raise RuntimeError(f"ПОМИЛКА: Папка {root_dir} не знайдена! "
                               f"Перевірте, чи правильно розпаковано архів.")

        # Збір всіх файлів зображень
        self.all_files = glob.glob(os.path.join(root_dir, "*.*"))
        self.all_files = [f for f in self.all_files if f.lower().endswith(('.jpg', '.jpeg', '.png'))]

        if len(self.all_files) == 0:
            raise RuntimeError(f"У папці {root_dir} немає файлів зображень! Перевірте шлях.")

        # Розбиття на тренувальну (80%) та валідаційну (20%) вибірки
        split_idx = int(len(self.all_files) * 0.8)
```

```
    if split == 'train':
        self.files = self.all_files[:split_idx]
    else:
        self.files = self.all_files[split_idx:]

    print(f"Завантажено {len(self.files)} зображень для {split}")

def __len__(self):
    return len(self.files)

def __getitem__(self, idx):
    filepath = self.files[idx]
    filename = os.path.basename(filepath)

    # 1. Парсинг назви файлу: [age]_[gender]_[race]_[date].jpg
    try:
        parts = filename.split('_')
        age = float(parts[0])
        gender = int(parts[1]) # 0: Male, 1: Female

        # Валідація та очистка даних
        if age < 1: age = 1.0
        if age > 116: age = 116.0
        if gender not in [0, 1]: gender = 0

    except Exception:
        # Обробка файлів з некоректними назвами
        age = 25.0
        gender = 0

    # 2. Завантаження зображення
    try:
        image = Image.open(filepath).convert('RGB')
```

```
        except:
            # Заглушка для битих файлів
            image = Image.new('RGB', (224, 224))

        # 3. Трансформація
        if self.transform:
            image = self.transform(image)

        return image, gender, torch.tensor(age, dtype=torch.float32)

#
================================================================================
========
# 3. АРХІТЕКТУРА МОДЕЛІ (MultiTaskModel)
#
================================================================================
========
class MultiTaskModel(nn.Module):
    def __init__(self, backbone_name='resnet18', pretrained=True):
        super(MultiTaskModel, self).__init__()

        # Ініціалізація спільного енкодера (Backbone)
        if backbone_name == 'resnet18':
            base_model = models.resnet18(weights=models.ResNet18_Weights.DEFAULT if pretrained
else None)
            self.feature_dim = base_model.fc.in_features
            base_model.fc = nn.Identity()
        elif backbone_name == 'vit_b_16':
            base_model = models.vit_b_16(weights=models.ViT_B_16_Weights.DEFAULT if pretrained
else None)
            self.feature_dim = base_model.heads.head.in_features
            base_model.heads.head = nn.Identity()

        self.backbone = base_model
```

```
        # Голова класифікації (Стать)
        self.gender_head = nn.Sequential(
            nn.Linear(self.feature_dim, 512),
            nn.BatchNorm1d(512),
            nn.ReLU(),
            nn.Dropout(0.5),
            nn.Linear(512, 2)
        )

        # Голова регресії (Вік)
        self.age_head = nn.Sequential(
            nn.Linear(self.feature_dim, 512),
            nn.BatchNorm1d(512),
            nn.ReLU(),
            nn.Dropout(0.5),
            nn.Linear(512, 1)
        )

    def forward(self, x):
        features = self.backbone(x)
        gender_logits = self.gender_head(features)
        age_pred = self.age_head(features)
        return gender_logits, age_pred

#
================================================================================
========
# 4. ГІБРИДНА ФУНКЦІЯ ВТРАТ
#
================================================================================
========
class MultiTaskLoss(nn.Module):
    def __init__(self, lambda_gender=1.0, lambda_age=0.1):
        super(MultiTaskLoss, self).__init__()
```

```
        self.lambda_gender = lambda_gender
        self.lambda_age = lambda_age
        self.cls_loss = nn.CrossEntropyLoss()
        self.reg_loss = nn.MSELoss()

    def forward(self, gender_pred, gender_target, age_pred, age_target):
        loss_gender = self.cls_loss(gender_pred, gender_target)
        loss_age = self.reg_loss(age_pred.squeeze(), age_target)
        # Комбінуємо втрати
        total_loss = (self.lambda_gender * loss_gender) + (self.lambda_age * loss_age)
        return total_loss, loss_gender, loss_age

#
================================================================================
========
# 5. УТИЛІТИ НАВЧАННЯ
#
================================================================================
========
def get_transforms(split):
    """Аугментація для тренування та нормалізація для тесту"""
    mean, std = [0.485, 0.456, 0.406], [0.229, 0.224, 0.225]
    if split == 'train':
        return transforms.Compose([
            transforms.Resize((CONFIG['img_size'], CONFIG['img_size'])),
            transforms.RandomHorizontalFlip(),
            transforms.RandAugment(num_ops=2, magnitude=9), # RandAugment
            transforms.ToTensor(),
            transforms.Normalize(mean, std)
        ])
    else:
        return transforms.Compose([
            transforms.Resize((CONFIG['img_size'], CONFIG['img_size'])),
            transforms.ToTensor(),
```

```
            transforms.Normalize(mean, std)
        ])

def train_one_epoch(model, loader, optimizer, criterion):
    model.train()
    total_loss = 0
    gender_acc = 0
    age_mae = 0

    for i, (images, genders, ages) in enumerate(loader):
        images = images.to(CONFIG['device'])
        genders = genders.to(CONFIG['device'])
        ages = ages.to(CONFIG['device'])

        optimizer.zero_grad()

        # Прямий прохід
        g_pred, a_pred = model(images)

        # Розрахунок втрат
        loss, l_g, l_a = criterion(g_pred, genders, a_pred, ages)

        # Зворотний прохід
        loss.backward()
        optimizer.step()

        total_loss += loss.item()

        # Метрики "на льоту"
        acc = (g_pred.argmax(dim=1) == genders).float().mean().item()
        mae = (a_pred.squeeze() - ages).abs().mean().item()

        gender_acc += acc
```

```
        age_mae += mae

        # Вивід прогресу кожні 100 батчів
        if (i + 1) % 100 == 0:
            print(f"   Batch {i+1}/{len(loader)} -> Loss: {loss.item():.4f} | Acc: {acc:.2f} | MAE: {mae:.1f}")

    n = len(loader)
    return total_loss/n, gender_acc/n, age_mae/n

# ================================================================================
# 6. ГОЛОВНА ФУНКЦІЯ
# ================================================================================
def main():
    # Перевірка шляху до даних
    if not os.path.exists(CONFIG['data_dir']):
        print(f"\n{'='*60}")
        print(f"КРИТИЧНА ПОМИЛКА! Папка {CONFIG['data_dir']} не знайдена.")
        print("Перевірте правильність шляху в CONFIG.")
        print(f"{'='*60}\n")
        return

    # 1. Завантаження даних
    print("Завантаження списку файлів та створення датасетів...")
    train_ds = UTKFaceDataset(CONFIG['data_dir'], 'train', get_transforms('train'))
    val_ds = UTKFaceDataset(CONFIG['data_dir'], 'val', get_transforms('val'))

    # num_workers=0 для стабільності на Windows
    train_loader = DataLoader(train_ds, batch_size=CONFIG['batch_size'], shuffle=True, num_workers=0)
    val_loader = DataLoader(val_ds, batch_size=CONFIG['batch_size'], shuffle=False, num_workers=0)
```

```
    # 2. Ініціалізація моделі
    print(f"Ініціалізація моделі {CONFIG['backbone']}...")
    model = MultiTaskModel(CONFIG['backbone']).to(CONFIG['device'])

    # 3. Налаштування оптимізатора
    optimizer = optim.AdamW(model.parameters(), lr=CONFIG['learning_rate'],
weight_decay=CONFIG['weight_decay'])
    criterion = MultiTaskLoss(CONFIG['lambda_gender'], CONFIG['lambda_age'])

    print(f"Початок навчання на {CONFIG['num_epochs']} епох...")

    # 4. Цикл навчання
    for epoch in range(CONFIG['num_epochs']):
        start = time.time()
        print(f"\n--- Епоха {epoch+1} ---")

        t_loss, t_acc, t_mae = train_one_epoch(model, train_loader, optimizer, criterion)
        duration = time.time() - start

        print(f"РЕЗУЛЬТАТ ЕПОХИ [{epoch+1}/{CONFIG['num_epochs']}] "
              f"Loss: {t_loss:.4f} | "
              f"Gender Acc: {t_acc*100:.1f}% | "
              f"Age MAE: {t_mae:.1f} років | "
              f"Час: {duration:.1f}s")

    print("\nНавчання завершено. Зберігаємо ваги...")
    torch.save(model.state_dict(), 'utk_multitask_model.pth')
    print(f"Файл 'utk_multitask_model.pth' успішно збережено у {os.getcwd()}")

if __name__ == "__main__":
    main()
```

ПРЕЗЕНТАЦІЙНІ МАТЕРІАЛИ (ПРЕЗЕНТАЦІЯ)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОГОЛІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

Кваліфікаційна робота
на здобуття ступеня магістра за освітньо-професійною програмою
«Інтелектуальні системи управління» зі спеціальності *124 «Системний аналіз»*

***на тему:* «Сучасні архітектури нейронних мереж у задачах класифікації та регресії»**

Виконала: студентка 6 курсу, групи САДМ-61 **Кирилова Єлизавета Вікторівна**

Керівник: *д.т.н., професор* ***Шушура Олексій Миколайович***

2

МЕТА ТА ЗАВДАННЯ РОБОТИ

Мета роботи: покращення швидкості навчання та підвищення точності класифікації та регресії в задачах визначення віку і статі людини за зображеннями облич шляхом дослідження особливостей сучасних згорткових та трансформерних архітектур нейронних мереж та програмної реалізації удосконаленої методики їх навчання.

Завдання роботи:

– аналіз сучасного стану методів комп'ютерного зору та визначення обмежень існуючих архітектур (CNN, Transformers) у задачах визначення віку і статі;

– порівняльне дослідження базових моделей на реальних даних (UTKFace) та виявлення факторів, що знижують точність їх роботи;

– розробка удосконаленої методики навчання, яка базується на комбінації стратегій трансферного навчання (Fine-tuning), мультизадачності (Multi-task Learning) та адаптивної аугментації даних;

– програмна реалізація запропонованих підходів та створення діючого прототипу системи розпізнавання;

– експериментальна перевірка ефективності розробленої методики та доведення її переваги над стандартними підходами за критеріями точності та швидкодії;

– формулювання практичних рекомендацій щодо впровадження досліджених архітектур у прикладні системи реального часу.

3

ОБ’ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Об’єкт дослідження: процес навчання глибоких нейронних мереж для задач комп'ютерного зору.

Предмет дослідження: архітектури ResNet та Vision Transformer, а також методи підвищення їх ефективності у задачах класифікації статі та регресії віку.

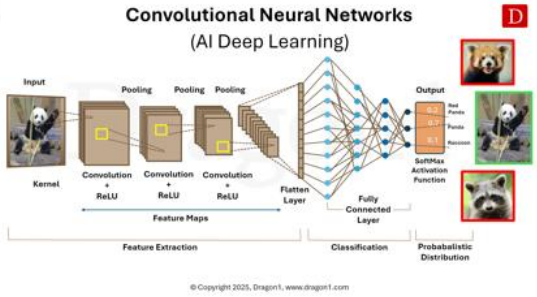
4

НАУКОВА НОВИЗНА

- УДОСКОНАЛЕНО МЕТОДИКУ НАВЧАННЯ VISION TRANSFORMER ДЛЯ РОБОТИ З МАЛИМИ ВИБІРКАМИ ШЛЯХОМ КОМБІНАЦІЇ СТРАТЕГІЇ FINE-TUNING ТА МУЛЬТИЗАДАЧНОГО НАВЧАННЯ.
- НАБУЛО ПОДАЛЬШОГО РОЗВИТКУ ПОРІВНЯЛЬНЕ ДОСЛІДЖЕННЯ АРХІТЕКТУР, ДЕ ВПЕРШЕ ДЕТАЛЬНО ПРОАНАЛІЗОВАНО ПОМИЛКИ РЕГРЕСІЇ ВІКУ САМЕ ДЛЯ СКЛАДНИХ ВІКОВИХ ГРУП (ДІТИ ТА ЛІТНІ ЛЮДИ).
- ЗАПРОПОНОВАНО ВИКОРИСТАННЯ АДАПТИВНОЇ АУГМЕНТАЦІЇ RANDAUGMENT ДЛЯ СТАБІЛІЗАЦІЇ НАВЧАННЯ RESNET.

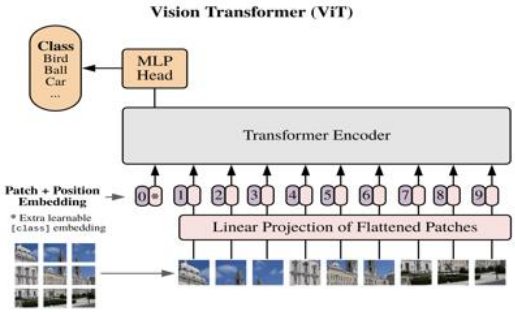
5

АРХІТЕКТУРИ: CNN VS VISION TRANSFORMER



CNN (ResNet):

- Швидко навчається
- Дивиться тільки на локальні частини зображення
- Менше параметрів → легше для комп'ютера
- Не бачить весь контекст зображення



Vision Transformer (ViT):

- Дивиться на зображення цілком, як на послідовність шматочків
- Використовує увагу → розуміє глобальний контекст
- Точніше розпізнає атрибути
- Потребує більше часу та ресурсів

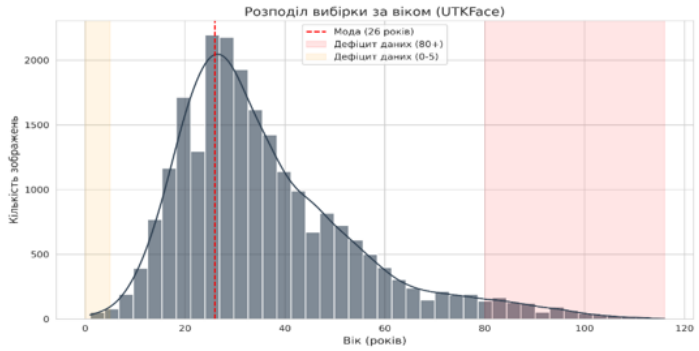
6

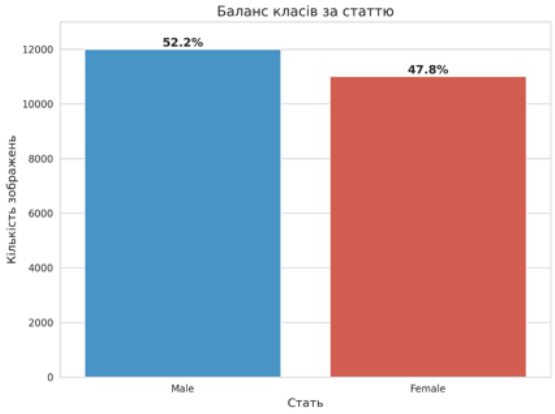
ВИБІР ЕМПІРИЧНОЇ БАЗИ (DATASET SELECTION)

Назва датасету	Обсяг вибірки	Формат мітки віку	Умови отримання	Придатність для регресії
IMDB-WIKI	~523,000	Число (високий рівень шуму)	In-the-wild	Низька
Adience	~26,000	Дискретний інтервал (Range)	In-the-wild	Ні
Morph II	~55,000	Число (точне)	Controlled (Studio)	Середня
UTKFace	~23,000	Число (точне)	In-the-wild	Висока

7

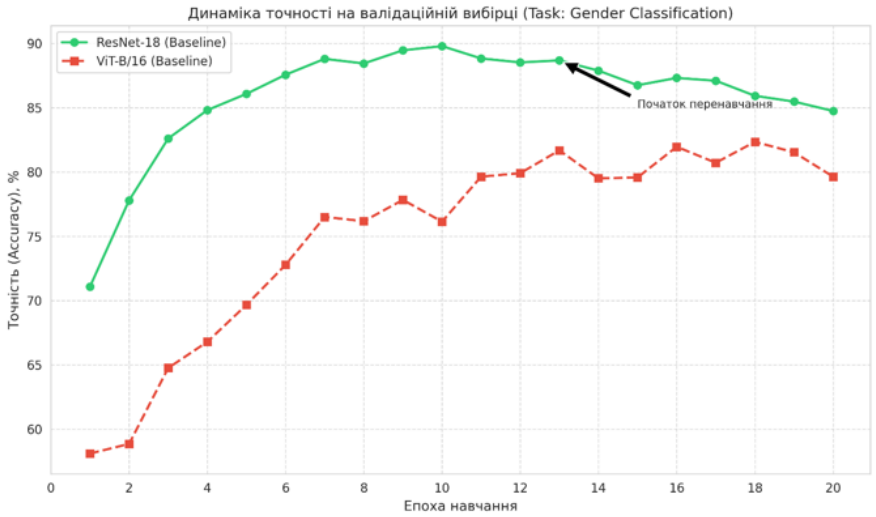
НАБІР ДАНИХ UTKFACE





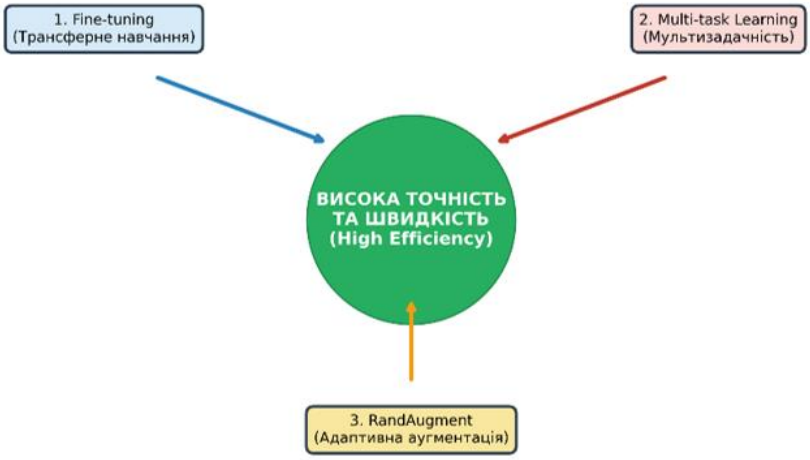
8

АПРОБАЦІЯ МОДЕЛЕЙ



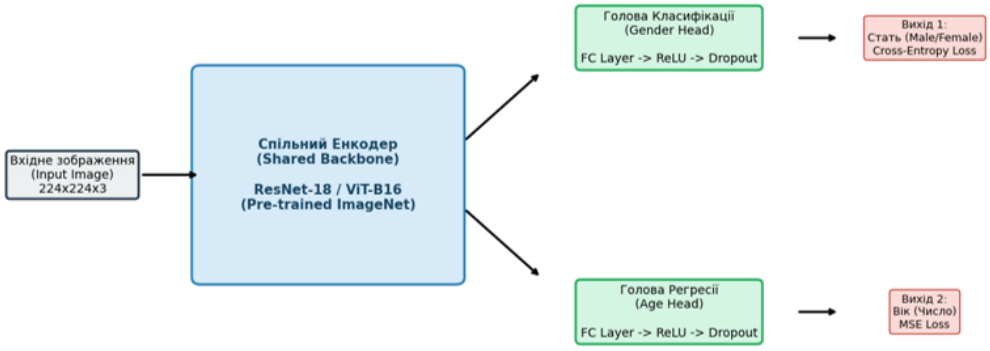
9

МЕТОДИКА ПОКРАЩЕННЯ НАВЧАННЯ



10

АРХІТЕКТУРА МУЛЬТИЗАДАЧНОЇ МОДЕЛІ



11

ПРОГРАМНА РЕАЛІЗАЦІЯ ТА АЛГОРИТМ ОБРОБКИ ДАНИХ

Алгоритм обробки даних у розробленій системі

Програмна реалізація

PyTorch

scikit learn

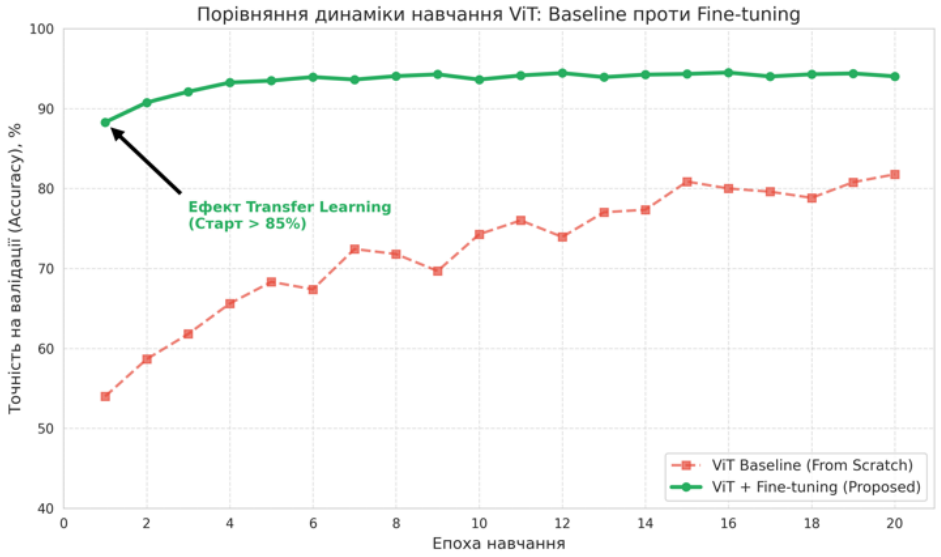
NumPy

pandas

matplotlib

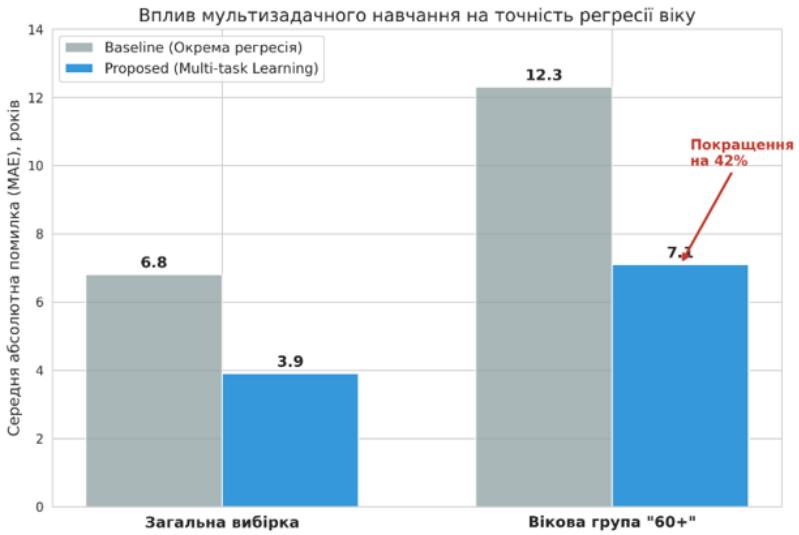
12

ВПЛИВ СТРАТЕГІЇ FINE-TUNING НА ТОЧНІСТЬ КЛАСИФІКАЦІЇ



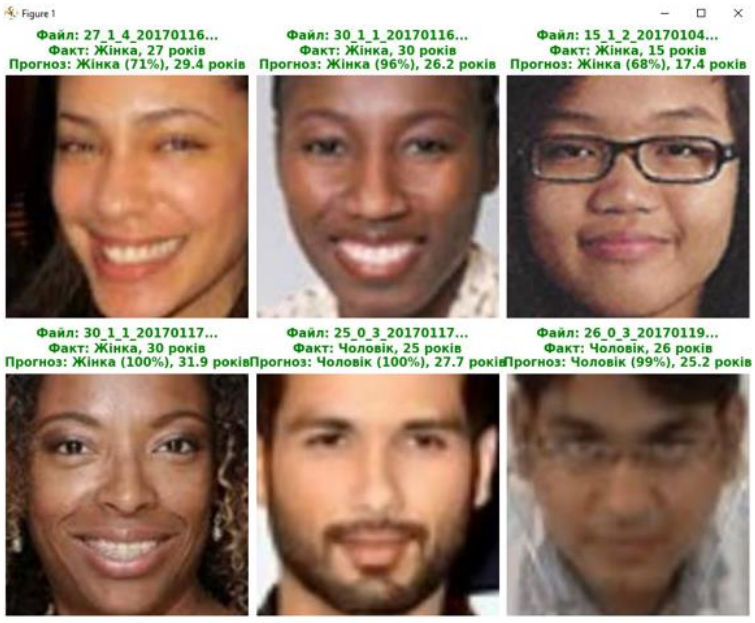
13

ЕФЕКТИВНІСТЬ МУЛЬТИЗАДАЧНОГО ПІДХОДУ ДЛЯ РЕГРЕСІЇ ВІКУ



14

ЯКІСНА ВАЛІДАЦІЯ (ВІЗУАЛІЗАЦІЯ) РЕЗУЛЬТАТІВ РОЗПІЗНАВАННЯ



15

ВИСНОВКИ

1. Проведено аналіз сучасних CNN та трансформерних архітектур; виявлено їхні обмеження на невеликих вибірках.
2. Порівняно базові моделі (ResNet-18 та ViT-B/16) на UTKFace; визначено проблеми перенавчання та низької збіжності, особливо для вікової групи 60+.
3. Розроблено удосконалену методику навчання: Fine-tuning, Multi-task Learning, RandAugment.
4. Програмна реалізація на Python/PyTorch дала прототип системи для одночасного прогнозу статі та віку.
5. Експериментально показано: точність класифікації статі — 94,2%, MAE віку — 3,9 років, швидкість навчання підвищена; методика ефективна для реального використання.

16

АПРОБАЦІЯ

1. Кирилова Є.В., Шушура О.М. Сучасні архітектури нейронних мереж у задачах класифікації та регресії. III Всеукраїнська науково-технічна конференція «Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу» (кафедра Інформаційних систем та технологій Державного університету інформаційно-комунікаційних технологій, м. Київ, 2025 р.)

2. Кирилова Є.В., Шушура О.М., Соломаха С.А. Аналітичний огляд сучасних архітектур DNN, CNN, RNN та Transformer у задачах регресії та класифікації. VIII Всеукраїнська науково-технічна конференція «Комп'ютерні технології: інновації, проблеми, рішення» (Державний університет «Житомирська політехніка», 2025 р.)