

Вінницький національний технічний університет  
Міністерство освіти і науки України

Кваліфікаційна наукова  
праця на правах рукопису

**КИРИЛЕНКО ОЛЕКСАНДР МИХАЙЛОВИЧ**

УДК 004.9+681.004

**ДИСЕРТАЦІЯ**  
**ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПОВТОРНОЇ ІДЕНТИФІКАЦІЇ**  
**ЛЮДЕЙ**

Спеціальність 126 – «Інформаційні системи та технології»

Галузь знань 12 – «Інформаційні технології»

Подається на здобуття ступеня доктора філософії

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

\_\_\_\_\_ О. М. Кириленко

Науковий керівник: Кветний Роман Наумович,  
доктор технічних наук, професор

Вінниця – 2025

## АНОТАЦІЯ

*Кириленко О.М.* Інформаційна технологія повторної ідентифікації людей – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 126 «Інформаційні системи та технології». – Вінницький національний технічний університет, Вінниця, 2025.

Метою дисертаційної роботи є створення інтелектуальної інформаційної технології для підвищення ефективності повторної ідентифікації людей. Основна задача полягає у покращенні точності повторної ідентифікації людини шляхом отримання нових моделей, заснованих на Omni-Scale Network архітектурі із використанням гіпермоделей та використання GAN.

Дисертація присвячена вирішенню актуальної науково-технічної задачі, що полягає у підвищенні точності повторної ідентифікації людей за допомогою методів машинного навчання, що базуються на Omni-Scale Network архітектурі із поєднанням гіпермоделей для динамічної зміни параметрів OSNet, що дозволяє адаптуватися до варіативності вхідних даних та використання генеративних змагальних мереж для вирішення проблеми недостатньої кількості даних для навчання.

У вступі визначено актуальність теми, мету та завдання дослідження, а також описано об'єкт і предмет дослідження. Наукову новизну роботи становить розробка нових методів застосування адаптивних гіпермоделей у поєднанні з архітектурою OSNet для покращення якості та узагальнювальної здатності моделей повторної ідентифікації людини. Результати роботи мають важливе практичне значення, що підтверджується апробацією та впровадженням запропонованої технології у реальних бізнес-задачах, зокрема для зональної аналітики торговельних центрів, підрахунку відвідувачів та у навчальному процесі.

У першому розділі проведено аналіз сучасних методів та моделей для вирішення задачі повторної ідентифікації людей за зображеннями, зокрема,

розглянуто згорткові нейронні мережі, автокодувальники та генеративні змагальні мережі. Показано, що традиційні підходи демонструють низьку ефективність у складних умовах спостереження через чутливість до доменного розриву, недостатню кількість навчальних даних та наявність шумів у зображеннях. Визначено необхідність застосування адаптивних методів, зокрема гіпермоделей у поєднанні з OSNet, для покращення якості вилучення ознак та підвищення точності повторної ідентифікації осіб у реальних умовах експлуатації. Також висвітлено проблему обмеженого обсягу тренувальних наборів, що створює потребу в застосуванні генеративних змагальних мереж для збільшення кількості та різноманітності навчальних зображень.

У другому розділі розроблено підхід на основі використання гіпермоделей для динамічного налаштування параметрів мережі OSNet, що забезпечує ефективне вилучення інваріантних ознак та покращує точність повторної ідентифікації осіб у реальних умовах. Також запропоновано метод автоматичної генерації синтетичних зображень за допомогою генеративних змагальних мереж, що дозволяє розширити тренувальні набори та підвищити узагальнювальну здатність моделей.

У третьому розділі представлено розробку інформаційної технології для вирішення задач повторної ідентифікації людини на основі інтеграції адаптивних гіпермоделей та генеративних змагальних мереж. Запропоновано структуру інформаційної технології, визначено склад функціональних модулів, алгоритм їх взаємодії та детально описано особливості реалізації кожного модуля. Для формалізації процесів розробки створено комплекс UML-діаграм, що включає діаграми діяльності, послідовності та компонування, які ілюструють логіку взаємодії компонентів системи та послідовність виконання основних етапів обробки даних. Наведено детальний опис архітектури технології, зокрема модулів попередньої обробки, генерації синтетичних зображень, вилучення інформативних ознак за допомогою OSNet із гіпермоделлю, а також модулів оцінки та візуалізації результатів повторної ідентифікації.

У четвертому розділі проведено експериментальні дослідження

запропонованої інформаційної технології та здійснено оцінювання її ефективності на основі використання стандартних наборів даних (Market-1501, DukeMTMC-ReID). Наведено результати порівняльного аналізу точності моделей повторної ідентифікації людини із застосуванням адаптивних гіпермоделей та генеративних змагальних мереж щодо базових моделей. Встановлено, що запропонований підхід суттєво підвищує узагальнювальну здатність нейронних мереж, покращує показники точності, зокрема mAP і Rank-1, та зменшує негативний вплив доменного розриву між різними умовами спостереження. Також здійснено практичну апробацію розробленої технології у реальних бізнес-задачах, зокрема для зональної аналітики торговельних центрів.

У висновках наведено основні результати дисертаційної роботи.

Основні наукові та практичні результати полягають в наступному:

#### 1. Аналіз сучасних методів та вибір оптимального підходу:

Виконано системний аналіз існуючих методів повторної ідентифікації людей на зображеннях, включаючи згорткові нейронні мережі, автокодувальники та генеративні змагальні мережі. Було детально розглянуто переваги та недоліки кожного з цих методів у контексті їхньої ефективності для навчання на малорозмірних наборах даних. Визначено, що найперспективнішим є підхід на основі архітектури OSNet, яка вирізняється здатністю ефективно поєднувати глобальні й локальні ознаки, компактністю параметричного простору, високою швидкістю роботи, а також стійкістю до варіацій освітлення, зміни ракурсу та часткових перекриттів об'єкта, що забезпечує високі показники точності та узагальнювальну здатність моделі.

#### 2. Розробка нового методу повторної ідентифікації:

Важливим результатом роботи є розробка та впровадження нового методу, що ґрунтується на використанні архітектури OSNet у поєднанні з гіпермоделлю, яка динамічно налаштовує параметри мережі відповідно до вхідних даних, що дозволяє ефективно вирішувати проблему доменного розриву та покращувати узагальнювальні властивості моделі. У розділі 2 представлено метод поєднання базової моделі зі гіпермоделлю та розроблено підхід до

збільшення кількості та різноманітності тренувальних зображень шляхом генерації синтетичних зразків із використанням генеративних змагальних мереж. Це дозволило суттєво розширити тренувальні набори та підвищити точність і стабільність навчання моделі.

### 3. Розробка нової інформаційної технології:

У ході дослідження було створено нову інформаційну технологію для повторної ідентифікації людини, яка включає комплекс взаємопов'язаних модулів для попередньої обробки даних, генерації синтетичних зображень, адаптивного вилучення інформативних ознак та візуалізації результатів. Запропонована технологія включає використання сучасних моделей глибокого навчання (CNN та GAN). Використано UML-діаграми для формалізації архітектури, взаємодії та послідовності роботи окремих компонентів системи.

### 4. Алгоритмічна реалізація та програмне забезпечення:

У розділі 3 було розроблено алгоритми, що реалізують запропоновані методи та їх подальше використання у процесі навчання моделей глибокого навчання. Алгоритмічні основи побудовані таким чином, що дозволяють легко інтегрувати розроблену технологію в існуючі системи повторної ідентифікації людей. Програмне забезпечення було реалізовано у вигляді Jupyter-ноутбуків, що дозволяє автоматизувати процеси попередньої обробки даних, навчання моделей та візуалізації результатів. Це програмне забезпечення є універсальним і може бути легко адаптоване для використання на різних платформах, таких як AWS SageMaker, Google Colab, або локальні системи машинного навчання.

### 5. Покращення ефективності моделі та оцінка результатів:

Виконано комплекс експериментальних досліджень, що підтвердили ефективність запропонованих підходів. Показано, що інтеграція OSNet з гіпермоделлю та генеративними змагальними мережами суттєво підвищує ключові показники повторної ідентифікації (mAP, Rank-1) порівняно з існуючими аналогами. Проведено оцінку якості результатів на стандартних наборах даних Market-1501 і DukeMTMC-ReID.

#### 6. Практичне значення розроблених методів та технологій:

Результати дослідження були впровадженні в реальних задачах бізнес-аналітики, зокрема у системах зонального аналізу торговельних центрів та автоматизованого підрахунку відвідувачів. Використання запропонованого методу динамічного налаштування параметрів мережі в поєднанні із додавання генеративних-змагальних мереж для навчання моделей дозволяє лише підвищити точність систем, що вирішують різні задачі повторної ідентифікації людей. Розроблена інформаційна технологія може бути адаптована для різних типів зображень та сфер застосування, що робить її універсальним інструментом для автоматизації задач розпізнавання людей на зображеннях в системах комп'ютерного зору.

#### 7. Впровадження в навчальний процес та подальші наукові дослідження:

Результати дослідження були впроваджені в навчальний процес Вінницького національного технічного університету, зокрема для викладання дисциплін, пов'язаних із інтелектуальними інформаційними системами та технологіями аналізу зображень. Це дозволяє студентам отримати практичний досвід роботи з передовими методами комп'ютерного зору. Крім того, результати роботи можуть бути основою для подальших досліджень у галузі комп'ютерного зору, аналізу даних та розробки нових методів повторної ідентифікації об'єктів на зображеннях.

#### 8. Публікації та наукове визнання:

Основні результати дослідження були викладені у низці наукових статей та представлені на науково-практичних конференціях. Загалом 5 статей у фахових журналах України зі спеціальності 126 [1], [2], [4], [7], [8]; 1 стаття у періодичному закордонному виданні [3]; 2 статті включені до збірників наукових праць, що входять до наукометричної бази даних Scopus [5], [6]; 16 тез доповідей на науково-практичних конференціях [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24].

Наукова новизна дисертаційної роботи полягає в наступних положеннях:

1. Уперше запропоновано інформаційну технологію повторної ідентифікації людини, яка на відміну від існуючих використовує інтеграцію гіпермоделей та генеративно-змагальних мереж із OSNet архітектурою, що забезпечує динамічну адаптацію архітектури та параметрів нейронних мереж до варіації даних.

2. Уперше запропоновано використання гіпермоделей з архітектурою OSNet для динамічного налаштування її параметрів відповідно до вхідних даних, що дало змогу досягти підвищення точності ідентифікації без повного перенавчання моделей.

3. Подальшого розвитку отримав метод застосування генеративно-змагальних мереж для синтетичного доповнення навчальних наборів під час навчання моделей повторної ідентифікації людини шляхом розширення класу та параметрів згенерованих об'єктів, що забезпечило можливість суттєво підвищити точність повторної ідентифікації людей з використанням розробленої технології.

Отримані результати є корисними для:

1. Рекомендовано ТОВ «КВАРК КОНСАЛТИНГ» для впровадження в практичну діяльність при розробці інтелектуальних систем для вирішення задач комп'ютерного зору, а саме: використання запропонованої інформаційної технології повторної ідентифікації людей на основі інтеграції гіпермоделей та генеративно-змагальних мереж із OSNet архітектурою впроваджено у проєкт «Molengo» (замовник Transfotech Ltd Israel) для побудови детальних теплових карт, кластерного аналізу та поведінкових патернів (акт впровадження від 28 серпня 2025 р.).

2. Навчального процесу Вінницького національного технічного університету під час викладання освітніх компонентів «Розпізнавання графічних образів» для студентів, які навчаються за освітньою програмою «Інтелектуальні інформаційні системи» рівня «бакалавр» спеціальності 126 Інформаційні системи та технології, а також – «Інформаційні технології обробки зображень» для студентів, які навчаються за освітньою програмою «Інформаційні технології

аналізу даних та зображень» рівня «магістр» спеціальності 126 Інформаційні системи (акт впровадження від 26 серпня 2025 р.)

Запропонована інформаційна технологія повторної ідентифікації людей вирізняється високою універсальністю та може бути адаптована до будь-яких доменів даних, від відеоспостереження до медичної візуалізації чи промислових зображень. Завдяки поєднанню генеративних змагальних мереж для створення якісних псевдоданих із гіпермоделлю, яка коригує параметри OSNet відповідно до статистичних характеристик конкретного домену, суттєво покращується якість сегментації об'єктів. Розширення тренувального набору за рахунок синтетичних зображень дозволяє зменшити вплив шумів, варіацій освітлення та обмеженості анотацій, що призводить до підвищення точності та узагальнювальної здатності моделей на нових, непередбачених даних. Модульна архітектура розробленої системи забезпечує простоту інтеграції в існуючі конвеєри машинного та глибокого навчання та робить її ефективним інструментом для задач аналітики відвідувачів, автоматичного контролю якості та інших прикладних сценаріїв.

**Ключові слова:** повторна ідентифікація людини, OSNet, адаптивні гіпермоделі, генеративні змагальні мережі, генерація синтетичних зображень, згорткові нейронні мережі, комп'ютерний зір, зональна аналітика, глибоке навчання, інтелектуальна інформаційна технологія, машинне навчання, штучний інтелект.

## ABSTRACT

*Kyrylenko O.M.* Information Technology for Person Re-Identification – Qualifying scientific work with manuscript rights.

Dissertation for obtaining the scientific degree of Doctor of Philosophy in 126 specialty “Information systems and technologies.” – Vinnytsia National Technical University, Vinnytsia, 2025.

The aim of the dissertation is to develop an intelligent information technology to improve the efficiency of person re-identification. The main objective is to enhance person re-identification accuracy by developing new models based on the Omni-Scale Network architecture that employ hypermodels and GANs.

The dissertation is devoted to solving a pressing scientific and technical problem – improving the accuracy of person re-identification using machine learning methods based on the Omni-Scale Network (OSNet) architecture, combined with hypermodels that dynamically adjust OSNet parameters to accommodate input variability, and employing generative adversarial networks to address the scarcity of training data.

In the introduction, the relevance of the topic, the aim and objectives of the study are defined, and the object and subject of the research are described. The scientific novelty of the work lies in developing new methods for applying adaptive hypermodels in combination with the OSNet architecture to improve the quality and generalization capability of person re-identification models. The results have important practical significance, as evidenced by the validation and deployment of the proposed technology in real business tasks – specifically, for zone-based analytics in shopping malls, visitor counting, and use in the educational process.

The first chapter presents an analysis of modern methods and models for image-based person re-identification, in particular convolutional neural networks, autoencoders, and generative adversarial networks (GANs). It is shown that traditional approaches perform poorly under challenging observation conditions due to sensitivity to domain shift, insufficient training data, and image noise. The need for adaptive

methods – specifically, hypermodels combined with the OSNet architecture – to improve feature extraction quality and increase person re-identification accuracy in real-world operating conditions is established. The chapter also highlights the limitation of training set size, motivating the use of GANs to increase both the quantity and diversity of training images.

The second chapter develops an approach that uses hypermodels to dynamically adjust the parameters of the OSNet network, enabling effective extraction of invariant features and improving person re-identification accuracy in real-world conditions. The chapter also proposes a method for automatic generation of synthetic images using generative adversarial networks (GANs), which expands the training sets and enhances the models' generalization capability.

The third chapter presents the development of an information technology for solving person re-identification tasks based on the integration of adaptive hypermodels and generative adversarial networks. The structure of the technology is proposed, the set of functional modules is defined, the algorithm of their interaction is outlined, and the implementation specifics of each module are described in detail. To formalize the development processes, a suite of UML diagrams has been created – including activity, sequence, and component diagrams – that illustrate the logic of component interactions and the sequence of the main data-processing stages. A detailed description of the system architecture is provided, including the preprocessing module, the synthetic image generation module, the extraction of informative features using OSNet with a hypermodel, as well as the modules for evaluation and visualization of person re-identification results.

The fourth chapter reports the experimental evaluation of the proposed information technology and assesses its effectiveness using standard datasets (Market-1501, DukeMTMC-ReID). It presents the results of a comparative analysis of person re-identification model accuracy when employing adaptive hypermodels and generative adversarial networks versus baseline models. The findings show that the proposed approach substantially enhances the generalization capability of neural networks, improves accuracy metrics – specifically mAP and Rank-1– and mitigates

the adverse impact of domain shift across varying observation conditions. The developed technology was also validated in real-world business tasks, in particular for zone-based analytics in shopping malls.

In the conclusions, the main results of the dissertation are presented.

The key scientific and practical results are as follows:

1. Analysis of modern methods and selection of the optimal approach:

A systematic analysis was carried out of existing image-based person re-identification methods, including convolutional neural networks, autoencoders, and generative adversarial networks. The advantages and disadvantages of each method were examined in detail in the context of their effectiveness for training on small-sized datasets. It was determined that the most promising approach is based on the OSNet architecture, which effectively combines global and local features, offers a compact parameter space, achieves high processing speed, and is robust to variations in illumination, changes in viewpoint, and partial occlusions – thereby ensuring high accuracy and strong generalization capability.

2. Development of a new person re-identification method:

An important outcome of the work is the design and implementation of a new method based on the OSNet architecture combined with a hypermodel that dynamically adjusts the network parameters to the input data, thereby effectively addressing domain shift and improving the model's generalization capability. The second chapter presents a method for coupling the baseline model with the hypermodel and introduces an approach to increase the quantity and diversity of training images by generating synthetic samples using generative adversarial networks (GANs). This made it possible to substantially expand the training sets and improve both the model's accuracy and the stability of the training process.

3. Development of a new information technology:

In the course of the research, a new information technology for person re-identification was created, comprising a suite of interrelated modules for data preprocessing, synthetic image generation, adaptive extraction of informative features, and results visualization. The proposed technology incorporates state-of-the-art deep

learning models (CNNs and GANs). UML diagrams were employed to formalize the architecture, interactions, and execution sequence of the system's individual components.

#### 4. Algorithmic implementation and software:

The third chapter develops the algorithms that implement the proposed methods and enable their subsequent use in training deep learning models. The algorithmic foundations are designed to facilitate seamless integration of the developed technology into existing person re-identification systems. The software was implemented as Jupyter notebooks, which automate data preprocessing, model training, and result visualization. This software is universal and can be readily adapted for use across different platforms, such as AWS SageMaker, Google Colab, or local machine learning systems.

#### 5. Improving model efficiency and evaluating results:

A comprehensive set of experiments confirmed the effectiveness of the proposed approaches. It was shown that integrating OSNet with a hypermodel and generative adversarial networks significantly improves key person re-identification metrics (mAP, Rank-1) compared to existing counterparts. Result quality was evaluated on the standard Market-1501 and DukeMTMC-ReID datasets.

#### 6. Practical significance of the developed methods and technologies:

The research results have been implemented in real business analytics tasks, specifically in zone-based analytics systems for shopping malls and automated visitor counting. Using the proposed method of dynamic network parameter tuning in combination with generative adversarial networks for model training improves the accuracy of systems that address various person re-identification tasks. The system can be adapted to different image types and application domains, making it a versatile tool for automating image-based person recognition tasks in computer vision systems.

#### 7. Implementation in the educational process and further research:

The research results have been integrated into the educational process at Vinnytsia National Technical University, in particular for teaching courses related to intelligent information systems and image analysis technologies. This enables students

to gain hands-on experience with state-of-the-art computer vision methods. Moreover, the results can serve as a foundation for further research in computer vision, data analysis, and the development of new methods for object re-identification in images.

#### 8. Publications and scientific recognition:

The principal results of the research have been presented in a series of scholarly articles and at scientific-practical conferences. In total: 5 articles in Ukrainian peer-reviewed journals in specialty 126 [1], [2], [4], [7], [8]; 1 article in an international periodical [3]; 2 articles included in collections of scientific papers indexed in the Scopus bibliometric database [5], [6]; and 16 conference abstracts presented at scientific-practical conferences [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24].

Scientific novelty of the dissertation consists in the following contributions:

1. For the first time, an information technology for person re-identification is proposed which, unlike existing approaches, integrates hypermodels and generative adversarial networks (GANs) with the OSNet architecture, providing dynamic adaptation of neural network architecture and parameters to data variability.

2. For the first time, the use of hypermodels with the OSNet architecture is proposed to dynamically tune its parameters to the input data, enabling improved identification accuracy without full retraining of the models.

3. The method of using GANs for synthetic augmentation of training sets during the training of person re-identification models has been further developed by expanding the classes and parameter ranges of generated objects, which has made it possible to significantly increase person re-identification accuracy with the proposed technology.

The obtained results are useful for:

1. Recommended for implementation by LLC «KVARK CONSULTING» for practical deployment in the development of intelligent systems for computer vision tasks. Specifically, the proposed information technology for person re-identification – integrating hypermodels and generative adversarial networks with the OSNet architecture – has been implemented in the “Molengo” project (client: Transfotech Ltd,

Israel) to build detailed heat maps, perform cluster analysis, and extract behavioral patterns (act of implementation dated August 28, 2025) .

2. The educational process at Vinnytsia National Technical University, in the teaching of the educational components “Graphical Pattern Recognition” for students enrolled in the bachelor’s program “Intelligent Information Systems” (specialty 126 “Information Systems and Technologies”), as well as “Information Technologies for Image Processing” for students enrolled in the master’s program “Information Technologies for Data and Image Analysis” (specialty 126 “Information Systems”) (act of implementation dated August 26, 2025).

The proposed information technology for person re-identification is highly versatile and can be adapted to virtually any data domain – from video surveillance to medical imaging or industrial imagery. By combining generative adversarial networks (GANs) to produce high-quality pseudo-data with a hypermodel that adjusts OSNet parameters to the statistical characteristics of a specific domain, the quality of object segmentation is significantly improved. Expanding the training set with synthetic images reduces the impact of noise, illumination variability, and limited annotations, which in turn increases accuracy and enhances model generalization on novel, unforeseen data. The system’s modular architecture ensures easy integration into existing machine- and deep-learning pipelines, making it an effective tool for visitor analytics, automated quality control, and other applied scenarios.

**Keywords:** person re-identification, OSNet, adaptive hypermodels, Generative Adversarial Networks (GANs), synthetic image generation, Convolutional Neural Networks (CNNs), computer vision, zonal analytics, deep learning, intelligent information technology, machine learning, artificial intelligence.

## СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ

**Наукові праці, в яких опубліковані основні наукові результати дисертації:**

[1] Р. Н. Кветний, Р. В. Маслій і О. М. Кириленко, “Виявлення та класифікація об’єктів дорожнього руху при використанні середовища DIGITS” *Оптико-електронні інформаційно-енергетичні технології*, вип. 39, вип. 1, с. 14–20, Січ 2021. doi: <https://doi.org/10.31649/1681-7893-2020-39-1-14-20> . (Index Copernicus, Наукове фахове видання України, категорія «Б» зі спеціальності 126)

*(Особистий внесок: провів систематичний огляд методів детектування й класифікації дорожніх об’єктів, організував збір і анотування набору даних, реалізував й навчив модель у середовищі DIGITS. Р. В. Маслій: оптимізував архітектуру класифікатора, налаштував гіперпараметри та провів серію експериментів з оцінювання точності. Р. Н. Кветний: узгодження задач дослідження та аналізу результатів, компонування і загальне редагування статті)*

[2] Р. Н. Кветний, Р. В. Маслій, О. М. Кириленко і В. В. Щерба “Дослідження нейромережевого підходу виявлення об’єктів у зображеннях” *Науково-виробничий журнал “Метрологія та прилади”*, no. 6, pp. 15–21, Січ. 2020. doi: [http://dx.doi.org/10.33955/2307-2180\(6\)2019.15-21](http://dx.doi.org/10.33955/2307-2180(6)2019.15-21). (Index Copernicus, Наукове фахове видання України, категорія «Б» зі спеціальності 126)

*(Особистий внесок: розробив архітектуру згорткової нейромережі для виявлення об’єктів та реалізував модуль оптимізації навчання. Р. В. Маслій: провів аналіз впливу різних функцій активації й нормалізації. В. В. Щерба: підготувала та нормалізувала навчальний й тестовий набори даних. Р. Н. Кветний: узгодження задач дослідження та аналізу результатів, компонування і загальне редагування статті)*

[3] O. Kyrylenko, R. Maslii and Y. Marushchak, “Analysis of methods of person reidentification in multi camera environment,” *Norwegian Journal of Development of the International Science*, no. 47, pp. 46–49, Oct. 2020. [Online]. Available:

[http://www.norwegian-journal.com/wp-content/uploads/2020/10/NJD\\_47\\_1.pdf](http://www.norwegian-journal.com/wp-content/uploads/2020/10/NJD_47_1.pdf)

(Періодичне закордонне видання)

*(Особистий внесок: провів глибокий огляд і таксономію сучасних методів повторної ідентифікації в мультикамерному середовищі та реалізувала базові пайплайни re-ID. R. Maslii: провів порівняльні експерименти та кількісну оцінку, узгодження задач дослідження та аналізу результатів. Y. Marushchak: компонування і загальне редагування статті)*

[4] Р. Н. Кветний, Р. В. Маслій і О. М. Кириленко, “Дослідження атрибутів людини для задачі повторної ідентифікації” *Інформаційні технології та комп’ютерна інженерія*, Вінниця, Україна, 2020. [Online]. Available: doi: <https://doi.org/10.31649/1999-9941-2020-49-3-4-13>. (Index Copernicus, Наукове фахове видання України, категорія «Б» зі спеціальності 126)

*(Особистий внесок: визначив ключові атрибути людини (зріст, колір одягу, особливі ознаки), розробив процедуру екстракції та попередньої обробки ознак та імплементував алгоритм класифікації атрибутів. Р. В. Маслій: провів оцінку їхньої інформативності за метриками mAP і Rank-1. Р. Н. Кветний: узгодження задач дослідження та аналізу результатів, компонування і загальне редагування статті)*

[5] R. Kvyetnyu, V. Kotsiubynskyi, O. Kyrylenko et al., “Using multiple optical cameras for correspondence identification between objects in the fields of view” in *Proc. SPIE*, vol. 12985, art. 129850A, 20 Dec. 2023. doi: <https://doi.org/10.1117/12.3022791> . (Входить до Scopus)

*(Особистий внесок: імплементував процедуру калібрування камер і трасування об’єктів у реальному часі. Спроектував експериментальний стенд із декількох камер, розробив алгоритм знаходження кореспонденцій між полями зору. V. Kotsiubynskyi: налаштував метрики якості (точність, повноту), провів аналіз похибок. R. Kvyetnyu: узгодження задач дослідження та аналізу результатів, компонування і загальне редагування статті)*

[6] R. Kvyetnyu, V. Kotsiubynskyi, O. Kyrylenko et al., “Re-identification of people in a video stream based on a Kalman filter” in *Proc. SPIE*, vol. 13400, art.

134000R, 16 Dec. 2024. doi: <https://doi.org/10.1117/12.3058549>. (Входить до Scopus)

*(Особистий внесок: реалізував фільтр Калмана з OSNet для покращення стійкості трекінгу, налаштував алгоритм обробки відеопотоку. Розробив інтерфейс візуалізації результатів і модуль оцінювання швидкодії (FPS, latency). V. Kotsiubynskyi: проаналізував якість ідентифікації й продуктивність системи. R. Kvyetnyu: узгодження задач дослідження та аналізу результатів, компонування і загальне редагування статті)*

[7] Р. Н. Кветний, В. В. Гармаш, О. М. Кириленко і В. Ю. Коцюбинський, “Застосування адаптивних гіпермоделей у задачах ре-ідентифікації людини” *Вісник ВПІ*, no. 2, pp. 138–146, 2025. doi: <https://doi.org/10.31649/1997-9266-2025-180-3>. (Index Copernicus, Наукове фахове видання України, категорія «Б» зі спеціальності 126)

*(Особистий внесок: запропонував і реалізував архітектуру гіпермоделі, що генерує параметри OSNet на основі статистичних ознак домену. виконав оцінювання продуктивності (mAP, Rank-1) на базі Market-1501 і DukeMTMC-ReID. В. В. Гармаш: налаштував генерацію корективів, провів навчання та оптимізацію гіпермоделі. В. Ю. Коцюбинський: координував збір даних, побудову графіків і таблиць. Р. Н. Кветний: узгодження задач дослідження та аналізу результатів, компонування і загальне редагування статті)*

[8] О. М. Кириленко, «Розробка методу повторної ідентифікації людини», *Опт-ел. інф-енерг. техн.*, вип. 41, вип. 1, с. 25–32, Трав 2022. doi: <https://doi.org/10.31649/1681-7893-2021-41-1-25-32>. (Index Copernicus, Наукове фахове видання України, категорія «Б» зі спеціальності 126)

*(Особистий внесок: сформулював методологію повторної ідентифікації, реалізував прототип моделі та провів базове навчання)*

### **Наукові праці, які засвідчують апробацію матеріалів дисертації:**

[9] О. М. Кириленко, “Сучасні методи виявлення та відстеження обличчя у складних умовах” *XLVI науково-технічна конференція ФКСА*, 2017, Вінницький

національний технічний університет. [Електронний ресурс]. Режим доступу: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-017/paper/view/2003/2635>

*(Особистий внесок: провів систематичний огляд існуючих методів детектування й трекінгу облич, реалізував прототип алгоритму детектування, підготував і проаналізував експериментальні результати)*

[10] О. М. Кириленко, “Аналіз методів фільтрації цифрових зображень” *Молодь в технічних науках: дослідження, проблеми, перспективи (MTN-2017)*, 2017, Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: <http://conf.inmad.vntu.edu.ua/fm/index.php?page=materials&line=29&mat=373>

*(Особистий внесок: дослідив основні методи фільтрації зображень, розробив і протестував набір цифрових фільтрів для покращення якості кадрів)*

[11] О. М. Кириленко, “Сучасні методи виявлення облич з використанням нейронних мереж” *Вимірювальна та обчислювальна техніка в технологічних процесах (ВОТТП-2017)*, Харків, Україна, 2017. [Електронний ресурс]. Режим доступу: [http://journals.khnu.km.ua/vottp/pdf/pdf\\_full/vottp-2017-konf.pdf](http://journals.khnu.km.ua/vottp/pdf/pdf_full/vottp-2017-konf.pdf)

*(Особистий внесок: проаналізував сучасні архітектури нейромереж для детектування облич, реалізував базовий прототип CNN-моделі, провів первинні експерименти)*

[12] О. М. Кириленко, Р. В. Маслій і В. В. Гармаш, “Системи розпізнавання об’єктів у зображеннях, отриманих у складних умовах освітлення” *IV міжнар. конференція «Вимірювання, контроль та діагностика в технічних системах (ВКДТС-2017)»*, 2017, Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: <http://mpa.vntu.edu.ua/images/conference/conf2017/VCDTS%202017.pdf>

*(Особистий внесок: визначив вимоги до алгоритму розпізнавання в умовах низької освітленості, реалізував метод нормалізації зображень та класифікації об’єктів, провів серію експериментів)*

[13] О. М. Кириленко і Р. Н. Кветний, “Аналіз методів відстеження облич у відеопослідовностях” *IV міжнар. конференція «Датчики, прилади та системи*

(ДСП-2017)», Черкаси, Україна, 2017. [Електронний ресурс]. Режим доступу: <https://er.chdtu.edu.ua/bitstream/ChSTU/820/1/%D0%A1%D0%B1%D0%BE%D1%80%D0%BD%D0%B8%D0%BA%20%D1%82%D0%B5%D0%B7%D0%B8%D1%81%D0%BE%D0%B2%20%D0%94%D0%9F%D0%A1-2017.pdf#page=33>

*(Особистий внесок: спроектував процедуру аналізу відеопотоку, визначив ключові ознаки для трекінгу облич та імплементавав алгоритм відстеження, провів оцінювання точності та швидкодії)*

[14] О. М. Кириленко, Р. Н. Кветний, О. В. Бісікало, Р. В. Маслій і В. Ю. Коцюбинський, “Розв’язання задачі підрахунку об’єктів при складних умовах отримання зображень” *ITSM-2018*, Івано-Франківськ, Україна, 2018. [Електронний ресурс]. Режим доступу: <http://itcm.comp-sc.if.ua/2018/kvetnyj.pdf>

*(Особистий внесок: визначив задачу підрахунку об’єктів, реалізував алгоритм детектування і локалізації об’єктів, виконав експерименти)*

[15] О. М. Кириленко і Р. В. Маслій, “Аналіз підходів до підрахунку кількості об’єктів у відеопослідовностях” *XLVII науково-технічна конференція ФКСА*, 2018, Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: <https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/20668/5022.pdf?sequence=3&isAllowed=y>

*(Особистий внесок: провів огляд і класифікацію методів підрахунку об’єктів, розробив прототип алгоритму, виконав експерименти)*

[16] О. М. Кириленко і Р. В. Маслій і, “Нейромережева архітектура detectnet в задачах виявлення об’єктів” *XLVIII науково-технічна конференція ФКСА*, 2019, Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-019/paper/view/7640/6467>

*(Особистий внесок: реалізував архітектуру detectnet, провів оптимізацію та порівняльні тести)*

[17] Р. Н. Кветний, Р. В. Маслій і О. М. Кириленко, “Дослідження можливості застосування атрибутів market-1501 для задачі ре-ідентифікації людини” *КУСС-2020*, 8–10 Oct. 2020. Вінницький національний технічний

університет. [Електронний ресурс]. Режим доступу: <https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/30576/10756-38288-1-RV.pdf?sequence=3&isAllowed=y>

*(Особистий внесок: імплементавав процедуру класифікації атрибутів і оцінив їхню ефективність за метриками mAP і Rank-1)*

[18] О. М. Кириленко, “Аналіз підходів повторної ідентифікації об’єктів” *L науково-технічна конференція ФКСА*, 2021, Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: <https://conferences.vntu.edu.ua/index.php/allfksa/allfksa2021/paper/view/12328>

*(Особистий внесок: запропонував методу ge-ID, реалізував перший прототип моделі та провів початкові експерименти)*

[19] Р. Н. Кветний і О. М. Кириленко, “Дослідження трекінгу об’єктів для задачі їх повторної ідентифікації” *КУСС-2022*, 2022, Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: <https://conferences.vntu.edu.ua/index.php/mccs/mccs2022/paper/view/16463>

*(Особистий внесок: проаналізував існуючі методи трекінгу, спроектував експериментальні сценарії, реалізував алгоритм трекінгу з оцінкою точності та продуктивності)*

[20] Р. Н. Кветний і О. М. Кириленко, “Аналіз способів представлення об’єктів у зображеннях для задачі відстеження” *ЛІІ науково-технічна конференція ФІТА*, 2023, Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2023/paper/view/17454>

*(Особистий внесок: провів огляд методів візуального представлення об’єктів, розробив підхід до екстракції ознак)*

[21] О. М. Кириленко і В. В. Гармаш, “Дослідження методів ре-ідентифікації осіб з декількох камер відеоспостереження” *Збірник праць Х МНТК «Датчики, прилади та системи – 2023»*, Черкаси, Вер. 2023, pp. 100–101. [Електронний ресурс]. Режим доступу: <https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/38086/134336.pdf?sequence=2&isAllowed=y> .

*(Особистий внесок: організував збір мультикамерних відеоданих, реалізував алгоритми синхронізації кадрів та аналізу ознак, провів експерименти)*

[22] Р. Н. Кветний, О. М. Кириленко і В. В. Гармаш, “Дослідження використання PostgreSQL в інформаційній системі повторної ідентифікації людини” *КУСС-2024*, 16–17 Жов. 2024. Вінницький національний технічний університет, doi: <https://doi.org/10.31649/mccs2024.2-15>

*(Особистий внесок: спроектував структуру бази даних та визначив вимоги до зберігання embedding-ів, реалізував прототип інформаційної системи на PostgreSQL, провів стрес-тести та оцінку швидкодії)*

[23] Р. Н. Кветний і О. М. Кириленко “Аналіз генеративних змагальних мереж для вирішення задач повторної ідентифікації” *LIV всеукр. наук.-тех. конференція ФІІТА-2025*, 2025, Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2025/paper/view/23851>

*(Особистий внесок: провів детальний аналіз архітектур GAN для re-ID, порівняв їхню ефективність на ембеддингах)*

[24] Р. Н. Кветний і О. М. Кириленко “Аналіз використання domain adaptation GANs для задачі повторної ідентифікації людини” *Photonics-ODS 2025*, 2025, Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: [https://conferences.vntu.edu.ua/index.php/ods/ods\\_2025/paper/view/23850](https://conferences.vntu.edu.ua/index.php/ods/ods_2025/paper/view/23850)

*(Особистий внесок: дослідив методи domain adaptation GANs, реалізував експериментальний прототип, провів оцінку за метрикою FID і Rank-1)*

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	25
ВСТУП.....	26
РОЗДІЛ 1 АНАЛІЗ СУЧАСНИХ МЕТОДІВ ТА МОДЕЛЕЙ ПОВТОРНОЇ ІДЕНТИФІКАЦІЇ ЛЮДЕЙ.....	35
1.1 Огляд основних принципів та методів задачі повторної ідентифікації людей.....	35
1.2 Аналіз відомих інтелектуальних методів та технологій повторної ідентифікації.....	39
1.3 Аналіз спеціалізованих згорткових мереж повторної ідентифікації...	49
1.4 Аналіз проблем повторної ідентифікації людей та постановка задач.....	61
РОЗДІЛ 2 РОЗРОБКА МЕТОДІВ ТА МОДЕЛЕЙ ПОВТОРНОЇ ІДЕНТИФІКАЦІЇ ЛЮДЕЙ З ВИКОРИСТАННЯМ ЗГЕНЕРОВАНИХ ЗОБРАЖЕНЬ.....	67
2.1 Розробка базової моделі повторної ідентифікації людей.....	67
2.2 Удосконалення базової моделі повторної ідентифікації за допомогою застосування умовної гіпермоделі.....	85
2.3 Удосконалення базової моделі шляхом використання генеративних змагальних мереж.....	99
2.4 Висновки до розділу.....	107
РОЗДІЛ 3 РОЗРОБЛЕННЯ АЛГОРИТМІЧНИХ ОСНОВ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ПОВТОРНОЇ ІДЕНТИФІКАЦІЇ ЛЮДЕЙ.....	108
3.1 Проектування структури інформаційної технології.....	108
3.2 Розроблення UML-діаграм інформаційної технології повторної ідентифікації людей.....	117
3.2.1 UML-діаграма послідовності роботи модулів технології.....	117
3.2.2 UML-діаграма діяльності базової моделі та гіпермоделі.....	121

3.2.3 UML-діаграма діяльності блоку роботи основної мережі із GAN.....	125
3.3 Реалізація комплексу програм для застосування розробленої інтелектуальної інформаційної технології.....	128
3.4 Алгоритм оцінювання ефективності запропонованої інформаційної технології.....	131
3.5 Висновки до розділу.....	133
РОЗДІЛ 4 ПРИКЛАДНЕ ЗАСТОСУВАННЯ РОЗРОБЛЕНОЇ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ПОВТОРНОЇ ІДЕНТИФІКАЦІЇ ЛЮДЕЙ.....	135
4.1 Реалізація комплексу програм для навчання базової мережі та гіпермоделі.....	135
4.2 Застосування створеної інформаційної технології повторної ідентифікації людей.....	158
4.3 Порівняння функціональних можливостей створеної інформаційної технології з аналогами.....	163
4.4 Впровадження розробленої технології у навчальний процес.....	167
4.5 Висновки до розділу.....	168
ВИСНОВКИ.....	170
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	172
ДОДАТКИ.....	188
ДОДАТОК А ТАБЛИЦЯ ОЦІНОК МОДЕЛЕЙ ДЛЯ ВІДКРИТИХ ДАТАСЕТІВ.....	189
ДОДАТОК Б СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ.....	197
ДОДАТОК В АКТ ПРО ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЙНОЇ РОБОТИ.....	202
ДОДАТОК Г АКТ ПРО ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ У НАВЧАЛЬНИЙ ПРОЦЕС ВНТУ.....	204
ДОДАТОК І ЛІСТИНГ ПРОГРАМНИХ МОДУЛІВ ДЛЯ ТРЕНУВАННЯ МОДЕЛЕЙ.....	205

ДОДАТОК Д ПРИКЛАД РОБОТИ ОТРИМАНИХ МОДЕЛЕЙ ПОВТОРНОЇ ІДЕНТИФІКАЦІЇ.....	227
---	-----

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

ВНТУ – Вінницький національний технічний університет

Кафедра АІТ – кафедра автоматизації та інтелектуальних інформаційних технологій ВНТУ

BN – пакетна нормалізація (Batch Normalization)

CNN – згортова нейронна мережа (Convolutional Neural Network)

CV – комп'ютерний зір (Computer Vision)

DL – глибоке навчання (Deep Learning)

DNN – глибока нейронна мережа (Deep Neural Network)

GPU – графічний процесор (Graphics Processing Unit)

HOG – гістограма спрямованостей градієнтів (Histogram of Oriented Gradients)

LR – швидкість навчання (Learning Rate)

ML – машинне навчання (Machine Learning)

NN – нейронна мережа (Neural Network)

ООП – об'єктно-орієнтоване програмування (Object-Oriented Programming)

SIFT – перетворення, інваріантне до масштабу (Scale-Invariant Feature Transform)

SURF – прискорені стійкі ознаки (Speeded-Up Robust Features)

SVM – машина опорних векторів (Support Vector Machines)

TPU – тензорний процесор (Tensor Processing Unit)

UML – уніфікована мова моделювання (Unified Modeling Language)

## ВСТУП

### **Обґрунтування вибору теми дослідження.**

Повторна ідентифікація людей є надзвичайно актуальною в умовах стрімкого розвитку технологій у сфері комп'ютерного зору, штучного інтелекту та обробки великих даних. Зростання потреби у моніторингу та аналізі поведінки клієнтів у різних галузях бізнесу зумовлює необхідність розробки нових інформаційних технологій для ефективної повторної ідентифікації людей у реальному часі.

Сучасні системи безпеки, такі як відеоспостереження в громадських місцях, контроль доступу до об'єктів та інші технології безпеки, все частіше потребують інтеграції інтелектуальних рішень. Традиційні методи ідентифікації осіб, які часто ґрунтуються на статичних підходах або ручному моніторингу, виявляються недостатніми для адаптації до динамічних умов, що постійно змінюються. Це підвищує ризики помилок і затримок, а також знижує загальну ефективність таких систем.

Крім того, повторна ідентифікація людей стає критично важливою в умовах зростаючих загроз безпеці, таких як тероризм, злочинність та інші соціальні виклики. Вона може суттєво покращити можливості правоохоронних органів у виявленні підозрілих осіб, а також у зборі доказів. Запровадження нових технологій у цю сферу може призвести до значного зниження рівня злочинності та підвищення загальної безпеки.

Найважливішими проблемами, з якими стикається галузь, є необхідність підвищення точності ідентифікації в умовах змінюваного освітлення, присутності перешкод та варіацій у зовнішньому вигляді людей, а також забезпечення конфіденційності даних. Неправильна ідентифікація може призвести до серйозних наслідків, включаючи юридичні проблеми та втрату довіри з боку громадськості.

Питаннями, пов'язаними з інтелектуальними інформаційними технологіями повторної ідентифікації людей, займаються провідні наукові

колективи та інженерні групи у сфері комп'ютерного зору, машинного навчання й аналітики даних, серед яких українські установи — Київський національний університет імені Тараса Шевченка, Національний технічний університет України «КПІ ім. Ігоря Сікорського» та Інститут кібернетики ім. В. М. Глушкова НАН України, а також R&D-лабораторії великих ІТ-корпорацій і стартапи, що впроваджують рішення для безпеки, роздрібної аналітики та реального часу; значний внесок роблять зарубіжні наукові школи та дослідницькі підрозділи, зокрема Visual Geometry Group в Оксфордському університеті, Computer Vision Laboratory в ETH Zürich, AI Lab Стенфордського університету, Department of Electrical Engineering and Computer Science Массачусетського технологічного інституту та Multimedia Laboratory при Chinese University of Hong Kong, а серед провідних дослідників варто відзначити Стена Зі Лі, Сяоганга Вана, Ляна Чжена, Мейну Кан і Ці Тіана.

Не менш важливою є і комерційна складова. Багато підприємств, особливо в сфері роздрібної торгівлі та обслуговування клієнтів, потребують технологій, які дозволяють не лише ідентифікувати клієнтів, а й аналізувати їх поведінку. Це актуально для компаній, які прагнуть покращити якість обслуговування, оптимізувати операційні процеси та підвищити ефективність маркетингових стратегій шляхом впровадження передових технологій повторної ідентифікації особистостей, що може сприяти покращенню обслуговування, підвищенню лояльності клієнтів і, в результаті, збільшенню прибутків.

Таким чином, вибір теми дисертаційного дослідження зумовлений як теоретичною, так і практичною значущістю. Розробка нових інформаційних технологій повторної ідентифікації людей відкриває широкі перспективи для впровадження у різних сферах суспільного життя. Це дослідження може стати основою для створення ефективних рішень, які відповідають сучасним викликам та вимогам безпеки, а також функціональних потреб бізнесу.

### **Зв'язок роботи з науковими програмами, планами, темами.**

Дослідження, результати яких представлено в дисертації, відповідають Закону України від 01.12.2022 № 2807-IX про Національну програму інформатизації і проводились протягом 2020-2025 рр. згідно планів наукових досліджень кафедри автоматизації та інтелектуальних інформаційних технологій Вінницького національного технічного університету по науково-дослідній роботі № 41 К 5 «Методи, моделі і засоби обчислень та обробки даних і зображень в інформаційних системах», що відображено в щорічних звітах з науково-дослідної роботи.

### **Мета і завдання дослідження.**

Метою дисертаційної роботи є підвищення точності повторної ідентифікації людей за допомогою використання нової інформаційної технології заснованої на OSNet архітектурі, гіпермоделях та використання генеративних змагальних мереж.

У результаті проведеного аналізу для досягнення поставленої мети сформульовані наступні задачі:

1. Здійснити детальний аналіз сучасних підходів до повторної ідентифікації людей за допомогою моделей глибокого навчання, приділивши особливу увагу сучасним моделям повторної ідентифікації, таким як Siamese Networks, Triplet Networks, Quadruplet Networks, Attention-based Models, Convolutional Neural Networks (CNNs), Vision Transformers (ViTs), Graph Neural Networks (GNNs), а також їхнім модифікаціям, з детальним розглядом їх ключових принципів, переваг та недоліків.

2. Проаналізувати ключові проблеми, пов'язані з оцінюванням результатів повторної ідентифікації людини, що виникають під час оцінки якості даних, використаних для ідентифікації, та обмежень існуючих метрик оцінювання. Особливу увагу приділити впливу цих обмежень на точність і достовірність результатів повторної ідентифікації.

3. Створити нові підходи, методи та алгоритми, які сприятимуть покращенню якості повторної ідентифікації людей, оптимізації процесу ідентифікації та зменшенню часу, забезпечуючи при цьому якість роботи мереж.

4. Розробити структуру нової інформаційної технології повторної ідентифікації людини, визначивши її архітектуру та ключові компоненти з використанням UML-діаграм. Описати функціональні можливості кожного компонента та їх взаємодію, що мають забезпечувати ефективність і узгодженість роботи всієї системи.

5. Розробити комплекс програмних модулів мовою програмування Python для впровадження та використання інформаційної технології повторної ідентифікації людини. Розроблений набір модулів має забезпечити ефективне та універсальне вирішення практичних задач, дозволяючи легко адаптувати технологію до різних сценаріїв застосування

6. Провести апробацію та практичне застосування розробленої інформаційної технології повторної ідентифікації людини на низці реальних задач. Оцінити ефективність запропонованих рішень та їхню доцільність для використання у конкретних прикладних сценаріях.

**Об'єктом дослідження** – процес повторної ідентифікації людей.

**Предметом дослідження** – інформаційна технологія, методи, моделі та алгоритми повторної ідентифікації людей.

**Методи дослідження** містять загальнонаукову методологію проведення досліджень і принципи системного підходу, а саме: аналіз літературних джерел та предметної області, огляд та порівняння існуючих моделей, експериментальні дослідження, комп'ютерні експерименти, проектування програмного забезпечення, методи машинного та глибокого навчання, методи комп'ютерного зору та обробки зображень.

**Наукова новизна одержаних результатів.**

1. Уперше запропоновано інформаційну технологію повторної ідентифікації людини, яка на відміну від існуючих використовує інтеграцію гіпермоделей та генеративно-змагальних мереж із OSNet архітектурою, що

забезпечує динамічну адаптацію архітектури та параметрів нейронних мереж до варіації даних.

2. Уперше запропоновано використання гіпермоделей з архітектурою OSNet для динамічного налаштування її параметрів відповідно до вхідних даних, що дало змогу досягти підвищення точності ідентифікації без повного перенавчання моделей.

3. Подальшого розвитку отримав метод застосування генеративно-змагальних мереж для синтетичного доповнення навчальних наборів під час навчання моделей повторної ідентифікації людини шляхом розширення класу та параметрів згенерованих об'єктів, що забезпечило можливість суттєво підвищити точність повторної ідентифікації людей з використанням розробленої технології.

### **Практичне значення одержаних результатів.**

1. Створено інтелектуальну інформаційну технологію, що включає комплекс програмних модулів мовою програмування Python, що реалізують функціонал всіх запропонованих методів, моделей та підходів повторної ідентифікації людей.

2. Проведено експериментальну валідацію розробленої інформаційної технології на відкритих наборах даних Market-1501 та DukeMTMC-ReID, що підтвердило її високу результативність. Запропонований підхід забезпечив суттєве зростання ключових метрик повторної ідентифікації (mAP та Rank-1) порівняно з базовими моделями, а також демонстрував стабільність роботи за різних умов освітлення, ракурсів і часткових перекриттів об'єктів. Завдяки розширенню тренувальних наборів синтетичними зображеннями та адаптивному налаштуванню параметрів мережі OSNet із гіпермоделлю вдалось досягти більшої узагальнювальної здатності моделей на незалежних тестових вибірках.

3. Результати роботи впроваджено (див. додаток Г) у навчальний процес і наукову діяльність кафедри автоматизації та інтелектуальних інформаційних технологій ВНТУ під час викладання освітніх компонентів «Розпізнавання графічних образів» для студентів, які навчаються за освітньою

програмою «Інтелектуальні інформаційні системи» рівня «бакалавр» спеціальності 126 Інформаційні системи та технології, а також – «Інформаційні технології обробки зображень» для студентів, які навчаються за освітньою програмою «Інформаційні технології аналізу даних та зображень» рівня «магістр» спеціальності 126 Інформаційні системи та технології.

4. Результати роботи були впровадженні (див. додаток В) в практичну діяльність ТОВ «КВАРК КОНСАЛТИНГ» (м. Вінниця) при розробці інтелектуальних систем для вирішення задач комп'ютерного зору.

Використання результатів дисертації підтверджено відповідними актами впровадження.

#### **Особистий внесок здобувача.**

Усі теоретичні результати, що виносяться на захист, отримані здобувачем особисто. У роботі [1] виконано систематичний огляд методів детектування та класифікації об'єктів дорожнього руху із реалізацією прототипу моделі у середовищі DIGITS та анотуванням набору даних. У [2] розроблено згорткову нейромережу для виявлення об'єктів у зображеннях, виконано нормалізацію даних та експериментально оцінено вплив функцій активації й методів нормалізації. У [3] здійснено глибокий аналітичний огляд методів повторної ідентифікації у мультикамерному середовищі, класифіковано підходи та проведено порівняльний аналіз їхньої ефективності. У [4] запропоновано метод екстракції та класифікації атрибутів людини для повторної ідентифікації, імplementовано алгоритми оцінювання їхньої інформативності на датасеті Market-1501. У [5] спроектовано експериментальний стенд із декількох оптичних камер, реалізовано алгоритми калібрування і кореспонденцій між полями зору та проведено аналіз похибок. У [6] інтегровано фільтр Калмана з архітектурою OSNet для підвищення стійкості трекінгу осіб у відеопотоці та виконано оцінювання продуктивності у термінах FPS і latency. У [7] розроблено адаптивну гіпермодель для динамічного налаштування параметрів OSNet на основі статистичних ознак домену, реалізовано її навчання та оцінено результати за метриками mAP і Rank-1 на Market-1501 і DukeMTMC-ReID. У [8]

сформульовано й реалізовано новий метод повторної ідентифікації людини, створено прототип моделі та проведено попередні експерименти на датасеті Market-1501. У [9] проведено систематичний аналіз методів детектування й трекінгу облич у складних умовах, імплементовано прототип алгоритму та підготовлено результати для тез доповіді. У [10] досліджено й протестовано методи фільтрації цифрових зображень, розроблено набір цифрових фільтрів для покращення якості кадрів та оформлено результати в тезах. У [11] проаналізовано сучасні нейромережеві архітектури для детектування облич, реалізовано базовий прототип CNN і проведено первинні експерименти. У [12] ініційовано дослідження систем розпізнавання об'єктів у умовах низької освітленості, розроблено метод нормалізації та класифікації і проведено серію експериментів. У [13] спроектовано процедуру аналізу відеопотоку та трекінгу облич у послідовності кадрів, реалізовано алгоритм та виконано оцінювання точності та швидкодії. У [14] розв'язано задачу підрахунку об'єктів у складних умовах отримання зображень, розроблено алгоритм детектування та локалізації об'єктів і проведено валідацію точності підрахунку. У [15] здійснено огляд і класифікацію алгоритмів підрахунку об'єктів у відеопослідовностях, реалізовано прототип методу та оформлено результати в тезах. У [16] розроблено і оптимізовано нейромережеву архітектуру detectnet для виявлення об'єктів, проведено порівняльні тести та підготовлено ілюстрації до тез. У [17] проаналізовано можливості застосування атрибутів датасету Market-1501 у повторній ідентифікації людини, імплементовано процедуру класифікації атрибутів та оцінено її ефективність. У [18] запропоновано нову методику повторної ідентифікації людини, реалізовано перший прототип моделі та проведено базові експерименти. У [19] досліджено методи трекінгу об'єктів для задачі повторної ідентифікації, розроблено експериментальні сценарії та імплементовано алгоритм з подальшим оцінюванням продуктивності. У [20] виконано огляд методів представлення об'єктів у зображеннях, запропоновано підхід до екстракції ознак та оформлено результати в тезах. У [21] організовано збір мультикамерних відеоданих, імплементовано алгоритми синхронізації

кадрів та аналізу ознак і виконано тестові експерименти. У [22] спроектовано структуру інформаційної системи на базі PostgreSQL для зберігання embedding-ів, реалізовано прототип та проведено стрес-тести. У [23] проведено детальний аналіз GAN-архітектур для задачі повторної ідентифікації, порівняно їхню ефективність та підготовлено результати в тезах. У [24] досліджено методи domain adaptation GANs для повторної ідентифікації людини, реалізовано експериментальний прототип та оцінено його за метриками FID і Rank-1.

### **Апробація результатів дисертації**

Основні наукові результати та практичні розробки дисертаційної роботи пройшли апробацію на 16 наукових конференціях:

XLVI науково-технічна конференція факультету комп'ютерних систем і автоматики (2017);

Молодь в технічних науках: дослідження, проблеми, перспективи (MTN-2017) (2017);

Вимірювальна та обчислювальна техніка в технологічних процесах (ВОТТП-2017) (2017);

IV міжнародна конференція «Вимірювання, контроль та діагностика в технічних системах (ВКДТС-2017)» (2017);

IV міжнародна конференція «Датчики, прилади та системи (ДСП-2017)» (2017);

Інформаційні технології та комп'ютерне моделювання (ІТСМ-2018) (2018);

XLVII науково-технічна конференція факультету комп'ютерних систем і автоматики (2018);

XLVIII науково-технічна конференція факультету комп'ютерних систем і автоматики (2019);

XV міжнародна конференція «Контроль і управління в складних системах (КУСС-2020)» (2020);

L науково-технічна конференція факультету комп'ютерних систем і автоматики (2021);

XVI міжнародна конференція «Контроль і управління в складних системах (КУСС-2022)» (2022);

LII науково-технічна конференція факультету інтелектуальних інформаційних технологій та автоматизації (2023);

X міжнародна конференція «Датчики, прилади та системи – 2023» (2023)

XVII міжнародна конференція «Контроль і управління в складних системах (КУСС-2024)» (2024);

LIV всеукр. наук.-тех. конференція факультету інтелектуальних інформаційних технологій та автоматизації (2025);

Міжнародна науково-технічна конференція з оптико-електронних інформаційних технологій «ФОТОНІКА – ODS 2025» (2025).

### **Публікації**

Основні результати дисертаційного дослідження викладені у 10 наукових працях (див. Додаток Б). Опубліковано 5 статей у фахових журналах України зі спеціальності 126 [1], [2], [4], [7], [8]; 1 стаття, що входить до європейського збірника [3]; 2 статті включені до збірників наукових праць, що входять до наукометричної бази даних Scopus [5], [6]; 16 тез доповідей на науково-практичних конференціях [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24].

### **Структура та обсяг дисертації.**

Дисертація складається зі вступу, чотирьох розділів, висновків, списку використаної літератури зі 136 найменування та 6 додатків. Загальний обсяг дисертації становить 239 сторінок, з них 147 сторінки основного тексту, який містить 49 рисунків та 13 таблиць.

# РОЗДІЛ 1

## АНАЛІЗ СУЧАСНИХ МЕТОДІВ ТА МОДЕЛЕЙ ПОВТОРНОЇ ІДЕНТИФІКАЦІЇ ЛЮДЕЙ

### 1.1 Огляд основних принципів та методів задачі повторної ідентифікації людей

Повторна ідентифікація людини (англ. Person Re-identification, ReID) — це задача комп'ютерного зору та машинного навчання, яка полягає в розпізнаванні та встановленні відповідності між зображеннями однієї і тієї ж особи, отриманими з різних камер, у різний час або в різних місцях. Основною метою є ідентифікація або відстеження людини серед великої кількості інших осіб у різних умовах зйомки, забезпечуючи стійкість до змін зовнішнього вигляду, пози, освітлення та оточення [25].

Система повторної ідентифікації людини є системою розпізнавання, схематично показаної на рисунку 1.1, яка складається з галереї (набір відомих екземплярів, Gallery set) і запитів (зображення особи, якому необхідно надати ідентифікатор, Probe ID) [26].

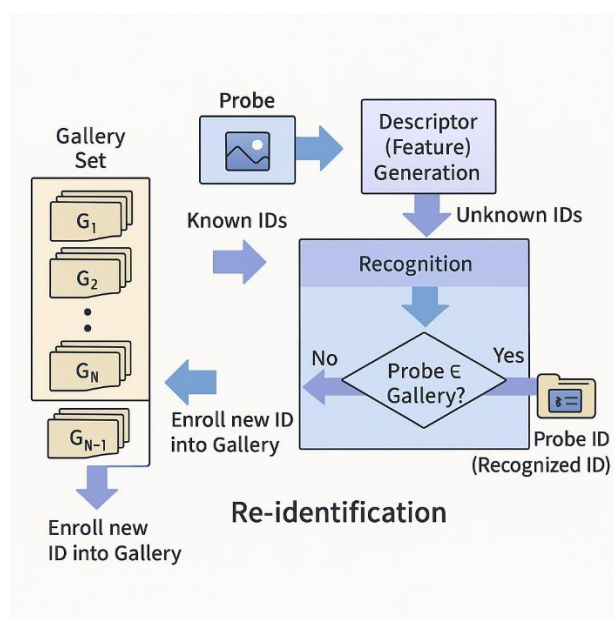


Рис. 1.1 Типова схема системи повторної ідентифікації людей

Загальне представлення повторної ідентифікації людини можна представити у вигляді. Нехай галерея  $G$  набір відомих екземплярів буде представлений у вигляді:

$$G = \{(g_i y_i)\}_{i=1}^N, \quad (1.1)$$

де  $g_i \in I$  – зображення особи із галереї,  $y_i \in Y$  – ідентифікатор особи на зображенні  $g_i$  та  $N$  — загальна кількість зображень у галереї.

Тоді запити  $Q$  набір екземплярів, яких ще немає в системі представленні як:

$$Q = \{(q_i y_{q_i})\}_{j=1}^M, \quad (1.2)$$

де  $q_i \in I$  – зображення особи, яку потрібно ідентифікувати,  $y_{q_i} \in Y$  – ідентифікатор особи на зображенні  $q_i$  (відомий лише для навчання та оцінки) та  $M$  — загальна кількість зображень запитів.

Для кожного запиту  $q_i$  необхідно знайти зображення з галереї  $g_i$ , такі що  $y_{q_i} = y_i$ , тобто ідентифікувати особу на зображенні запиту серед зображень у галереї [27].

Повторна ідентифікація людини є складною задачею через велику варіативність зовнішнього вигляду осіб, різноманітність умов зйомки та можливі перекриття об'єктів у кадрі. Дані методи є ключовою складовою сучасних систем відеоспостереження та аналізу поведінки. Вони забезпечують можливість відстеження об'єктів у різних середовищах і умовах, що робить їх актуальними для багатьох прикладних задач. У сучасній науці розрізняють два основні класи таких методів: методи, засновані на ручних ознаках (Handcrafted Features), та підходи, що використовують глибинне навчання (Deep Learning) [28].

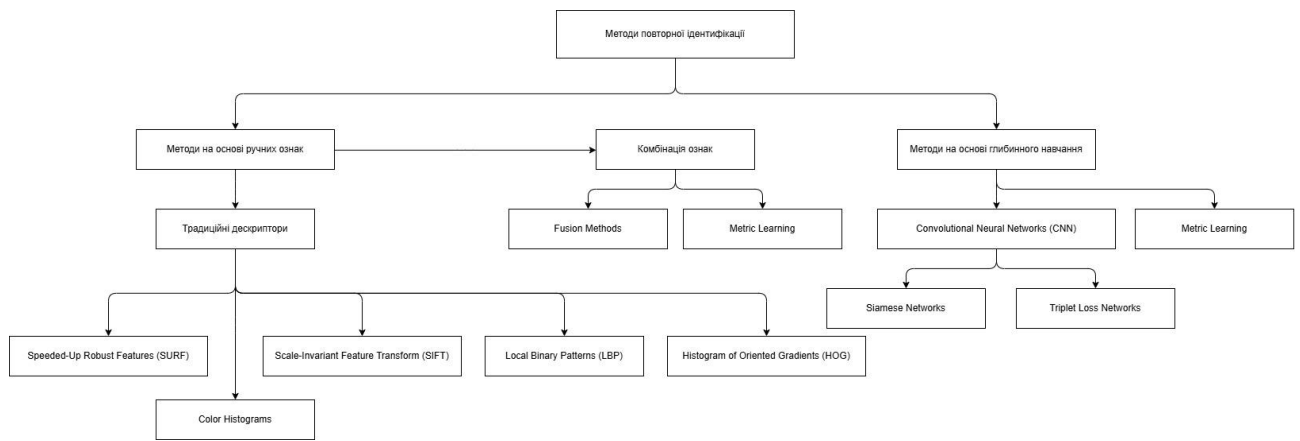


Рис. 1.2 Класифікації методів повторної ідентифікації

Методи на основі ручних ознак і досі залишаються важливими завдяки їхній високій інтерпретованості та низьким обчислювальним витратам, що особливо актуально для систем із обмеженими ресурсами. Метод Histogram of Oriented Gradients (HOG) [29], залишається популярним для задач аналізу форми, а інтеграція HOG із глибокими мережами, дозволяє використовувати його в умовах з обмеженими обсягами даних. Color Histograms завдяки своїй простоті знайшли застосування в задачах моніторингу, особливо для аналізу зображень із низькою якістю. Однак такі методи, як SIFT [30] та SURF [31], хоча і забезпечують стабільне виявлення ключових точок, вимагають значних обчислювальних ресурсів і поступаються сучасним глибоким підходам.

Комбінація ознак є потужним засобом підвищення точності методів. Fusion Methods дозволяють об'єднувати ознаки, такі як HOG, LBP та Color Histograms, що робить їх ефективними для мультимодальних систем [32-33]. Metric Learning [34], представлений Weinberger, активно використовується для адаптації моделей до конкретних даних, і його сучасна [35] демонструє високу ефективність у задачах повторної ідентифікації. Незважаючи на переваги, ручні методи мають обмеження у складних умовах, таких як значні варіації освітлення чи шуму, що обмежує їхню застосовність у більш універсальних системах [36].

Методи глибокого навчання, такі як CNN [37] і трансформери [38], завдяки своїй здатності автоматично виявляти складні ознаки, стали основою сучасних систем. Конволюційні нейронні мережі, зокрема Siamese Networks [39],

демонструють високу точність у задачах порівняння зображень, як це підтвердили сучасні дослідження. Triplet Loss Networks [40] забезпечують ефективне навчання для порівняння зображень, особливо при використанні складних негативних прикладів. Classification-Based Models [41] використовуються для класифікації осіб і інтегруються з трансформерами, що покращує точність [42]. Ще одним важливим підходом є OSNet (Omni-Scale Network) [43], який створений спеціально для задач повторної ідентифікації. OSNet здатен ефективно обробляти ознаки на різних масштабах завдяки своїй унікальній архітектурі, яка інтегрує кілька масштабів одночасно. Це дозволяє моделі адаптуватися до складних варіацій у вигляді та розмірі об'єктів, що забезпечує високу продуктивність у задачах з обмеженими обсягами даних. Її головна перевага полягає у балансі між точністю та обчислювальною ефективністю, що робить OSNet ідеальним вибором для реальних застосувань.

Метрики навчання, зокрема Contrastive Loss [44] і Triplet Loss [46], залишаються важливими компонентами глибокого навчання. Contrastive Loss, використовується для задач із обмеженими даними, тоді як Triplet Loss, реалізований у FaceNet [46], дозволяє порівнювати об'єкти за відносними відстанями.

Генеративно-змагальні мережі (GANs) [47], відкривають нові можливості для генерації синтетичних даних. Augmentation GANs довели свою ефективність у збільшенні навчальних наборів, тоді як Domain Adaptation GANs допомагають адаптувати моделі до нових середовищ [48-49].

Трансформери (Transformers) [50], такі як Vision Transformers (ViT), запропоновані стали революційним інструментом для задач комп'ютерного зору. Завдяки Self-Attention Mechanisms вони забезпечують обробку великих послідовностей ознак, що дозволяє досягати високої точності [51-52].

## 1.2 Аналіз відомих інтелектуальних методів та технологій повторної ідентифікації

Завдання повторної ідентифікації вважається складним через низку факторів: змінність зовнішнього середовища та фону, різні пози і ракурси людей, часткові перекриття (оклюзії) тіла, зміни кута огляду камери та освітлення. Протягом останнього десятиліття було розроблено багато інтелектуальних методів для розв'язання цієї проблеми – насамперед на основі глибокого навчання [53-60].

Глибинне розпізнавання облич — це напрям комп'ютерного зору, що використовує глибокі нейронні мережі для автоматичної ідентифікації або верифікації особи на основі зображення обличчя. Технологія полягає у формуванні векторного представлення рис обличчя (embedding), що дозволяє ефективно порівнювати зображення за подібністю навіть за умов зміни ракурсу, освітлення, віку або емоційних виразів. Найчастіше використовуються глибокі згорткові нейромережі, такі як FaceNet або ArcFace, здатні з високою точністю та швидкістю розв'язувати завдання повторної ідентифікації.

FaceNet [46] – це глибока нейромережна модель для розпізнавання облич, запропонована дослідниками Google у 2015 році. Принцип роботи FaceNet полягає в безпосередньому навчанні нейронної мережі відображати зображення облич у точку в евклідовому просторі фіксованої розмірності (128-вимірний вектор) таким чином, що відстані між такими векторами відповідають візуальній схожості облич. Мережа навчається за допомогою функції втрат triplet loss, яка оперує трійками зображень: базове обличчя (anchor), інше зображення тієї ж особи (positive) та зображення іншої особи (negative). Triplet loss мінімізує відстань між anchor і positive та максимізує відстань між anchor і negative, що формує дискримінативний простір ознак. У результаті FaceNet створює компактне представлення облич, яке можна використовувати для ідентифікації, верифікації (порівняння двох облич) чи кластеризації осіб на зображеннях.

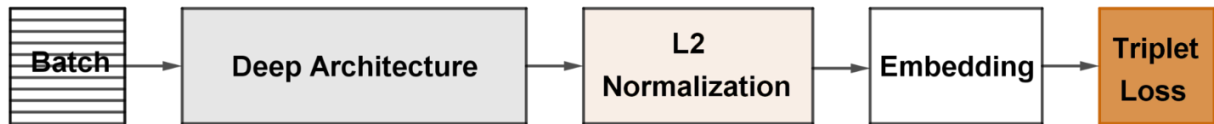


Рис. 1.3 Загальна структура системи розпізнавання обличчя FaceNet

Архітектура FaceNet використовувала глибоку CNN (22-шарову нейронну мережу, зокрема варіанти на основі Inception та Inception-ResNet) і генерувала 128-розмірні нормалізовані вектори ознак облич. Важливо, що FaceNet не потребує окремого класифікатора або SVM після нейромережі – сама відстань між ембедінгами визначає, чи належать дві особи до одного індивіда. Це “відкрита” система ідентифікації: вона здатна порівнювати будь-які обличчя (у вигляді векторів), навіть якщо конкретна особа не була в навчальній вибірці, що зручно для систем відеоспостереження.

Модель широко використовується для розпізнавання облич у різних системах. У контексті відеоспостереження FaceNet може виконувати ідентифікацію особи за обличчям, якщо камери зафіксували чітке зображення обличчя. Наприклад, систему FaceNet можна інтегрувати для автоматичного пошуку конкретної особи в потоці з багатьох камер: кожне виявлене обличчя перетворюється на ембедінг, і далі зіставляється з еталонними ембедінгами в базі даних або з ембедінгами з інших камер для повторної ідентифікації. FaceNet також застосовується для кластеризації невідомих облич (групування кадрів з однією людиною) та верифікації особи (наприклад, у системах контролю доступу).

Найбільший недолік полягає в тому, що метод залежить від якості та ракурсу обличчя. Якщо обличчя погано видно (наприклад, повернуте боком або затінене), точність суттєво знижується. Поза (орієнтація голови) і освітленість історично є довготривалими проблемами для систем розпізнавання облич. За неконтрольованих умов, сильні тіні, низька роздільна здатність або часткове перекриття обличчя (наприклад, маскою чи кепкою) можуть призвести до

помилки ідентифікації. Хоча FaceNet певною мірою стійка до варіацій (завдяки навчанні на великій кількості пар “та сама/інша особа”), вона показує найвищу ефективність на фронтальних або близьких до фронтальних зображеннях.

Ще один аспект – продуктивність, модель FaceNet досить глибока, тому її розгортання потребує обчислювальних ресурсів (GPU для реального часу). Втім, сучасні оптимізовані реалізації дозволяють опрацьовувати сотні зображень облич за секунду на GPU, а також існують полегшені версії FaceNet на основі мобільних архітектур для вбудованих систем.

Існують також інші моделі розпізнавання обличчя такі як ArcFace та DeepFace. ArcFace [61] це модель, що впроваджує додаткову кутову межу в функцію втрат (additive angular margin loss) для підвищення розрізнявальної здатності ембедінгів. ArcFace забезпечує одну з найвищих точностей: близько 99,8% на LFW, фактично досягнувши рівня, вищого за людський у задачі верифікації облич. Ключова ідея – віддаляти представлення різних осіб на кутовому просторі, що покращує відокремлення схожих облич.

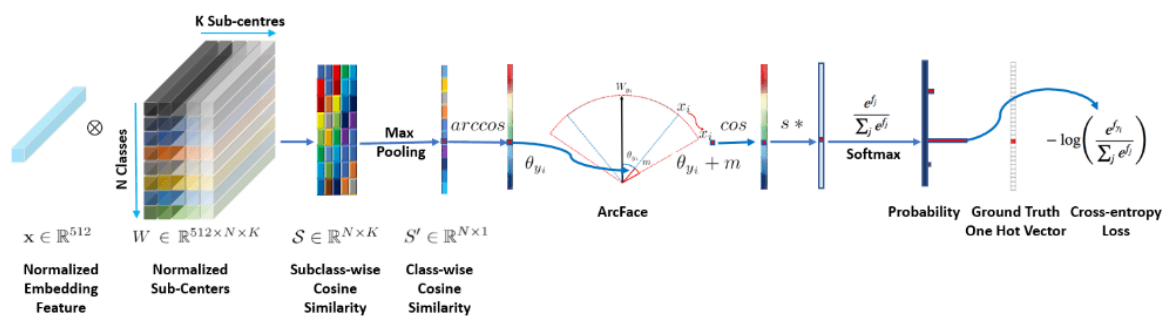


Рис. 1.4 Загальна структура архітектури ArcFace

Модель демонструє відмінну ефективність на великих наборах даних; її обмеженнями є потреба в якісному вирівнюванні облич та значних обчислювальних ресурсах під час навчання, хоча в застосуванні ArcFace дуже точний і надійний.

DeepFace [62] – одна з перших глибоких моделей для розпізнавання облич, що досягла майже людського рівня. Це 9-шарова нейромережа з  $\sim 120$  мільйонами параметрів, яка використовує спеціальні локально з’єднані шари (не

лише згортки) та 3D-врівноваження облич для кращої інваріантності до поз. DeepFace була натренована на ~4 млн зображень і досягла ~97,35% точності на LFW. Вона показала, що глибоке навчання може перевершити попередні методи (для порівняння, людина ~97,5% на LFW). Обмеження: модель велика і вимоглива до ресурсів; пізніші розробки (FaceNet, ArcFace тощо) перевершили її в точності та стійкості до різних умов зйомки, особливо при значних поворотах голови або зміні освітлення.

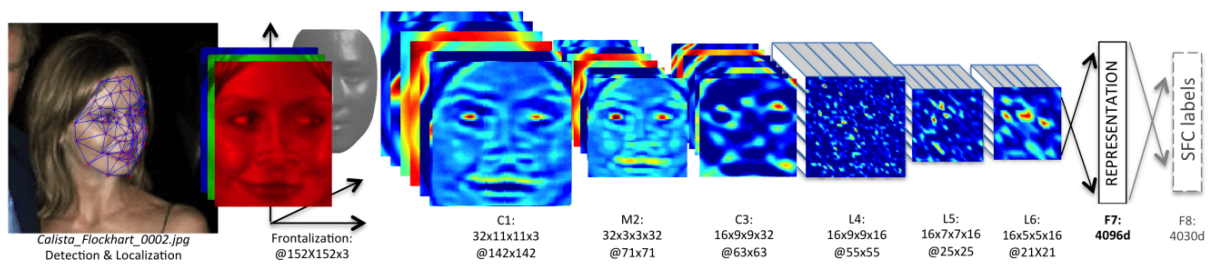


Рис. 1.5 Структура архітектури DeepFace

Оцінка пози людини — це процес визначення конфігурації тіла людини на зображенні шляхом локалізації ключових точок тіла, таких як суглоби, кінцівки, голова тощо. Для цього використовуються спеціалізовані глибокі нейронні мережі, які здатні ефективно виявляти та прогнозувати координати точок скелета людини. Найпоширенішими підходами є використання згорткових нейронних мереж, зокрема архітектур OpenPose, AlphaPose або HRNet, що забезпечують високу точність оцінки пози навіть за умов часткового перекриття або складного фону [63-64].

Метод OpenPose [65] базується на багатошаровій конволюційній нейромережі, що спочатку витягує ознаки зображення, а потім одночасно прогнозує карти впевненості для кожного суглоба та поля асоціації частин (Part Affinity Fields). Карти впевненості відповідають ймовірності наявності певної ключової точки (напр. ліктя) у кожній позиції зображення, а поля асоціації визначають, які ключові точки належать до одного скелету (щоб розрізнити людей, коли їх кілька на сцені). В результаті поєднання цих даних OpenPose

може “зібрати” повний скелет для кожної людини в кадрі, з’єднавши точки у структуру з’єднаних кінцівок. Такий підхід дозволяє одночасно виділити кілька осіб навіть у випадку їх часткового перекриття.

Архітектура OpenPose має каскадну структуру: початкові оцінки ключових точок покращуються на наступних етапах мережі для підвищення точності локалізації.

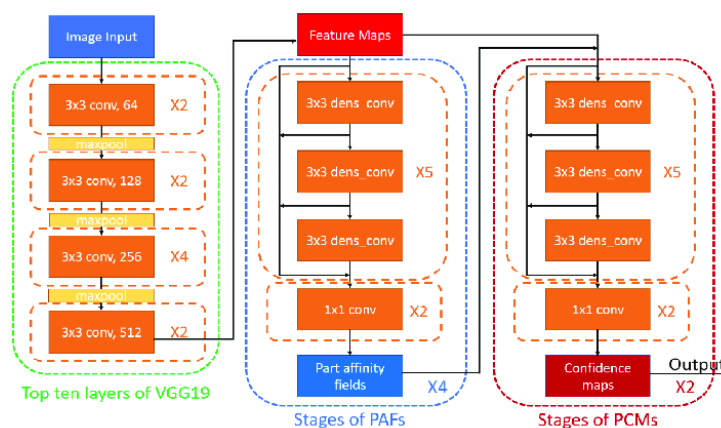


Рис. 1.6 Архітектура OpenPose

Головною перевагою OpenPose є здатність аналізувати позу кількох людей у реальному часі. Система є відкритою і широко використовується дослідниками та розробниками. OpenPose може відстежувати не лише базові точки тіла, але й дрібну моторику – наприклад, позиції пальців рук та риси обличчя – за рахунок окремих моделей для цих підзадач. Це робить її універсальним інструментом для аналізу поведінки людини на відео.

Хоч OpenPose безпосередньо не ідентифікує особу, він забезпечує критично важку інформацію – позиційні ознаки тіла, які можна використати для повторної ідентифікації. По-перше, скелетна модель людини є відносно незалежною від одягу: на відміну від візуальних ознак кольору одягу, координати суглобів не зміняться, якщо людина перевдягнеться. По-друге, поза може допомогти зробити ідентифікацію стійкішою до різних ракурсів: знаючи положення кінцівок, система ReID може нормалізувати зображення – наприклад, вирівняти фігуру або витягти ознаки окремо з верхньої та нижньої частини тіла.

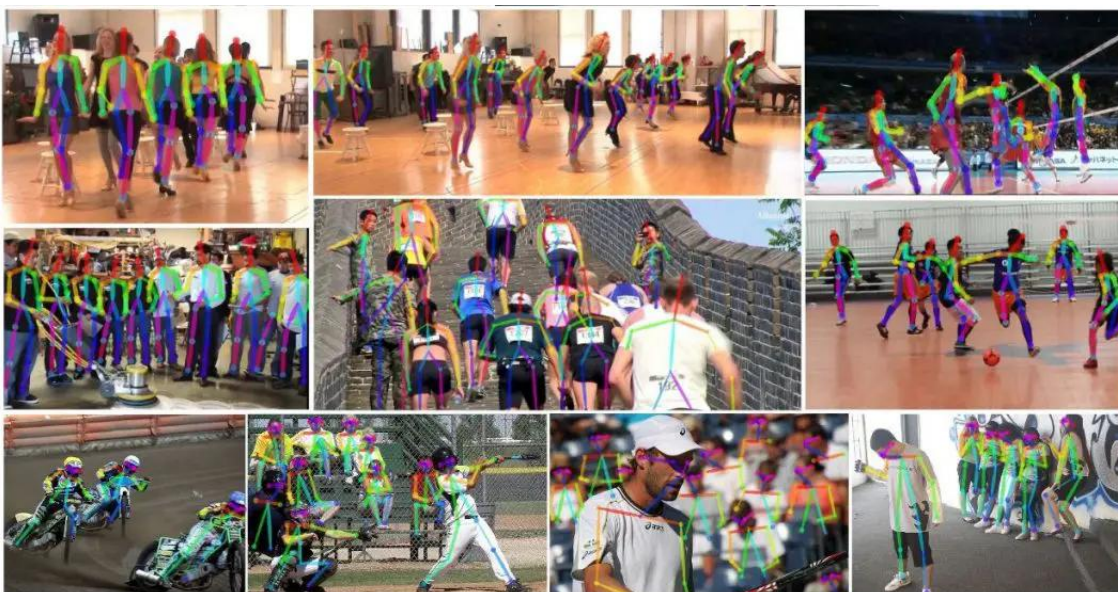


Рис.1.7 Оцінка ключової точки пози людини за допомогою OpenPose

Існують підходи, що інтегрують вихід OpenPose у модель ReID. Зокрема, Pose-Guided Person Re-ID Network [66] – мережа, яка оцінює позу людини та подає її як додатковий вхід до системи повторної ідентифікації. Такий підхід підвищує стійкість до змін кута огляду: модель “знає”, як була повернута людина, і враховує це під час порівняння ознак. Інший напрям – ідентифікація за ходом (gait recognition): аналізуючи послідовність скелетів OpenPose в часі, можна отримати унікальний “почерк” ходи людини. Ця біометрична ознака дозволяє впізнавати особу навіть здалеку, коли обличчя нечітке. Таким чином, OpenPose слугує важливою допоміжною технологією в системах відеоспостереження, підвищуючи їх інтелектуальність шляхом врахування динаміки руху та пози.

Висока точність OpenPose досягається ціною значних обчислень. Модель містить кілька стадій CNN і повинна обробити все зображення; через це швидкість обробки без потужного GPU невисока. На CPU продуктивність становить долі кадру за секунду (практично непридатно для реального часу). На сучасних GPU OpenPose може працювати в реальному часі, але зазвичай це ~5–10 кадрів/сек для типового розміру кадру [67]. Існують оптимізації (зменшення роздільної здатності аналізу, спрощення моделі), які дозволяють досягти ~10 FPS

і більше, проте загалом OpenPose лишається одним з найбільш ресурсомістких компонентів в системі відеоаналітики. Щодо стійкості, OpenPose доволі робастний до різних ракурсів – він успішно знаходить видимі суглоби як на фронтальних, так і на бокових проекціях людини. До змін освітлення метод менш чутливий, ніж кольорові методи, адже оперує формою тіла; проте при поганому освітленні чи поганій контрастності зображення нейромережа може не виявити деякі точки [68].

OpenPose вимагає, щоб значна частина тіла була видима. Якщо людина закрита наполовину перешкодою або виходить з кадру, скелет може бути неповним або неправильним. Крім того, використання лише скелетних даних для ідентифікації має обмежену дискримінативну здатність – різні люди можуть мати схожу статуру і позу. Тому у задачах ReID дані OpenPose зазвичай комбінуються з іншими ознаками (наприклад, кольірними дескрипторами чи ознаками нейромережі) для остаточного рішення.

Ідентифікація за зовнішнім виглядом (appearance-based identification), спрямована на розпізнавання особи за зовнішніми ознаками (одяг, постава, аксесуари) на серії зображень або відеокадрів із різних камер або в різний час. У цьому завданні застосовуються глибокі нейронні мережі, які формують унікальні дескриптори зовнішнього вигляду, що дозволяє ідентифікувати особу незалежно від зміни ракурсу, освітлення та контексту. Основними підходами є використання згорткових нейронних мереж з метриками подібності, такими як Triplet Loss або класифікаційні функції втрат на основі Cross-Entropy Loss та ArcFace [69-70].

У загальному випадку ReIDNet можна назвати будь-яку глибоку нейронну мережу, спеціально навчену на задачі повторної ідентифікації людей за їхнім зовнішнім виглядом. Як правило, вхідними даними для такої мережі є вирізаний фрагмент зображення з людиною (повна фігура або торс з ногами), отриманий детектором людини. Завдання мережі – перетворити зображення людини на інформативний вектор ознак, який буде схожим для двох зображень однієї особи і різко відрізнятиметься для різних людей. Концептуально це схоже

на FaceNet, але працює для повнозростових зображень з фоном і різноманітним одягом. Поява великих датасетів (Market-1501, DukeMTMC-reID, MSMT17 тощо) дозволила навчити такі моделі з високою точністю [71].

Ранні роботи з глибинного навчання для ReID використовували сіамські мережі – дві копії CNN, які отримували пару зображень і навчалися давати оцінку міри схожості (верифікація) подібно до підходу FaceNet. Пізніше переважним підходом стало навчання класифікатора особи: мережа тренується розпізнавати ідентичності з великої навчальної вибірки (тисячі різних людей), а після навчання останній шар (класифікацію) відкидають, використовуючи передостанній шар як універсальні ознаки для зіставлення нових осіб. Такий підхід називають IDE (ID Embedding). В сучасних ReID-мережах зазвичай комбінують обидва підходи: додають до softmax-класифікації ще й Contrastive/Triplet loss на рівні ембедінгів, щоб збільшити міжкласову відстань між ознаками.

Як архітектуру CNN часто беруть успішні моделі для класифікації зображень (ResNet-50, Inception тощо) і модифікують їх під ReID-завдання. Наприклад, в одному з популярних базових рішень ознаки беруться з ResNet-50 [72], розділеного на декілька горизонтальних смуг зображення (для частин тіла), що називається PCB (Part-based Convolutional Baseline) – це допомагає локалізувати ознаки окремо з верхньої та нижньої частини тіла для кращої стійкості до поз. Інші моделі вводять механізми уваги (attention), які дозволяють мережі фокусуватися на найбільш характерних областях зображення людини (наприклад, на торбі, яку вона несе, або на унікальному взорі футболки), і ігнорувати фон. Існують також багатогілкові архітектури, де окрема гілка мережі навчається розпізнавати атрибути (стать, колір одягу, тип взуття тощо) паралельно з основним завданням – прикладом є підхід JAN (Joint Attribute and Identity Network), який спільним навчанням ознак і атрибутів підвищує точність ідентифікації.

Загалом, глибокі CNN зараз домінують у ReID, вони суттєво перевершують старі методи, що базувалися на ручних ознаках (наприклад,

гістограма кольорів або HOG) – ті методи погано працювали при оклюзіях, зміні освітлення чи ракурсу, натомість нейромережі показали значно вищу дискримінативну здатність.

Сучасні ReID-мережі досягають високої точності на стандартних наборах даних. Наприклад, глибинні моделі з увагою та багат шаровим витягуванням ознак можуть забезпечувати Rank-1 accuracy (ймовірність правильно знайти ту саму особу серед багатьох кандидатів) понад 90% на наборах Market-1501 чи DukeMTMC-reID, що є величезним прогресом у порівнянні з показниками десять років тому. Навіть відносно компактні моделі сьогодні здатні перевершувати більш старі й громіздкі мережі. Наприклад, на 2019 рік було показано, що модель з лише ~2,2 млн. параметрів (OSNet, розглянута нижче) може перевершити більші моделі на кількох бенчмарках. Це означає, що алгоритми ReID стають практичними для реальних застосувань, де обчислювальні ресурси обмежені (наприклад, вбудовані системи відеоспостереження).

Глибинні моделі ReID вже інтегруються у системи відеоспостереження та аналітики. Вони працюють у зв'язці з детекторами об'єктів: спочатку детектор (YOLO [73] або Faster R-CNN [74]) знаходить людей на кожному кадрі, потім вирізані зображення людей подаються до ReID-мережі, яка формує їх дескриптори. Ці дескриптори можна порівнювати між кадрами та камерами, реалізуючи відстеження траєкторії кожної особи по всій системі. Наприклад, якщо на Камері 1 та Камері 2 з'явилась людина в червоній куртці, ReID-мережа зіставить їхні ознаки і з високою імовірністю виявить, що це той самий індивід, навіть якщо обличчя не видно. Такі системи використовуються в торгових центрах для підрахунку унікальних відвідувачів, в розумних містах для пошуку підозрюваних, або в спортивних додатках для автоматичної оцінки переміщення гравців між різними камерами.

ReIDNet, на відміну від FaceNet, оперує повним образом людини і тому може працювати навіть коли обличчя відсутнє або повернуте, використовуючи одяг та антропометричні дані як “візитівку” особи. Основна проблема – зміна зовнішнього вигляду людини. Якщо та сама особа переодягнеться (наприклад,

зніме піджак чи поміняє колір куртки наступного дня), стандартні ReID-мережі не зможуть її впізнати, оскільки вони значною мірою спираються на колір і текстуру одягу. Інше типове обмеження – схожість між різними людьми: уніформа або популярний одяг (чорні куртки, сині джинси) можуть призводити до плутанини, коли модель видає високий ступінь схожості для двох різних осіб. Для пом'якшення цього, дослідники додають до моделей вищезгадані атрибути (стать, зріст, колір волосся) або спеціальні модулі, що навчаються розрізняти дрібні деталі (наприклад, взуття).

Глибокі моделі більш стійкі до змін освітлення, ніж прості колірні гістограми, але все одно можуть помилятися, якщо одна камера знімає в день, а інша вночі, або якщо баланси білого дуже різні. Відомо, що відмінності між денним і нічним зображенням тієї ж людини надзвичайно ускладнюють повторну ідентифікацію. Для вирішення цієї проблеми досліджуються крос-модальні підходи, коли, наприклад, людина знімається одночасно в видимому та інфрачервоному спектрі, і мережа вчиться зіставляти ці різномірні дані.

Щодо стійкості до ракурсу, більшість сучасних ReID-мереж враховують цю проблему через augmentation даних (тренуються на різних позах) або інтегруючи інформацію про пози (як у Pose-Guided методах) – це підвищує ймовірність успішної ідентифікації навіть коли, скажімо, в одній камері видно спину людини, а в іншій – спереду. Однак повністю розв'язати цю проблему складно: зображення спини і обличчя можуть бути надто різними, тому моделі потребують дуже потужних ознак. Частково допомагають згадані підходи з поділом на частини тіла: мережа може порівнювати окремо спину торсу, окремо штани тощо, і таким чином знайти відповідності. Оклюзії (коли людина частково закрита іншими людьми чи об'єктами) – ще один виклик. Сучасні моделі застосовують attention-механізми, які намагаються визначити видимі фрагменти людини і ігнорувати перекриті зони, або ж додатково навчаються на синтетичних випадках перекриттів. Незважаючи на ці складнощі, глибокі ReID-мережі наразі є основою більшості практичних систем багато-камерного відстеження осіб.

### 1.3 Аналіз спеціалізованих згорткових мереж повторної ідентифікації

Сучасні підходи до Re-ID ґрунтуються на глибокому навчанні, і серед найпотужніших методів – спеціалізовані згорткові мережі такі як OSNet, генеративні моделі GAN та моделі на основі трансформерів.

OSNet (Omni-Scale Network) – це спеціалізована CNN-архітектура, розроблена саме для задачі Re-ID. Головна ідея OSNet – навчитися omni-scale ознакам, тобто ознакам різних масштабів одночасно, від локальних деталей до глобальних контекстів. Для цього введено особливий омні-масштабний резидуальний блок (OSBlock), який має декілька паралельних гілок згортки з різними рецептивними полями. Кожна гілка витягує ознаки на певному масштабі (наприклад, локальні деталі через  $3 \times 3$  згортки або більш глобальні через послідовність декількох  $3 \times 3$  згортки). Потім ці багатомасштабні ознаки об'єднуються за допомогою спеціального механізму динамічного агрегаційного шлюзу (Aggregation Gate, AG), який призначає канал-специфічні ваги для кожної гілки залежно від вхідного зображення [76].

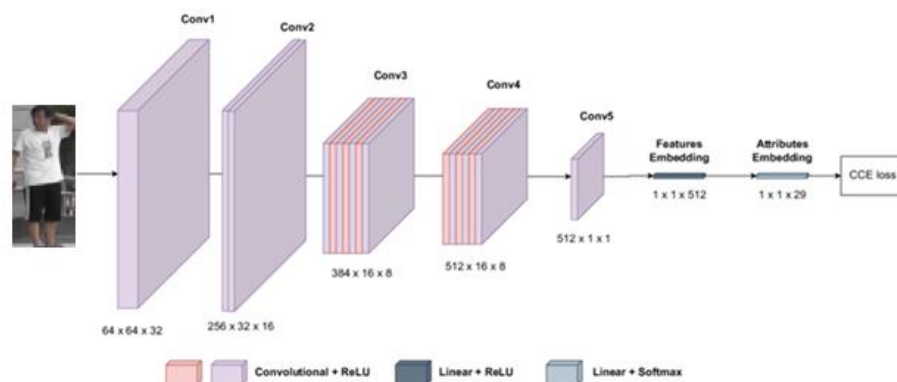


Рис. 1.8 Архітектура навчання функцій Omni-scale (OSNet x1.0)

Таким чином, мережа адаптивно фокусується на найбільш релевантному масштабі ознак для кожного зразка: вона може підсилити глобальні ознаки для

одних зображень чи локальні деталі для інших, залежно від того, що важливіше для розпізнавання.

Архітектура OSNet є легковаговою: замість громіздких шарів ResNet50, один блок OSNet використовує факторизовані згортки (послідовність  $1 \times 1$  та *depthwise*  $3 \times 3$  згортки, званих Lite  $3 \times 3$ ), що різко зменшує кількість параметрів. Наприклад, стандартну згортку  $3 \times 3$  з розміром ядра  $k \times k$  і  $c$  каналами розбито на  $1 \times 1$  згортку та *depthwise*-операцію – це знижує обчислювальну складність з  $h \cdot w \cdot k^2 \cdot c \cdot c'$  до  $h \cdot w \cdot (k^2 + c) \cdot c'$ , (де  $h, w$  – розмір просторової карти,  $c'$  – кількість вихідних каналів). Така економність дозволяє будувати глибоку мережу з omni-scale ознаками без перенавчання навіть на відносно невеликих наборах даних Re-ID. У підсумку повна модель OSNet містить  $\sim 2.2$  млн параметрів (для вхідних зображень  $256 \times 128$ ) та перевершує більші глибокі мережі на кшталт ResNet50 за точністю на кількох бенчмарках [77].

Архітектуру OSNet можна формалізувати на рівні одного резидуального OS-блоку. Нехай  $x$  – вхідний тензор ознак для блока. Базовий резидуальний блок (як в ResNet) навчає перетворення  $F(x)$  і додає його до вхідних ознак:

$$y = x + \tilde{x}, \quad \text{де} \quad \tilde{x} = F(x), \quad (1.3)$$

В OSNet-блоці [78]  $F(x)$  реалізовано як сума декількох шарів з різним ефективним полем зору. Зокрема,  $F$  розширено до суми  $F^t$  порядку  $t$  (кількість послідовних Lite  $3 \times 3$  згортки) для охоплення все більших масштабів. Якщо  $F^1$  – ознаки масштабу  $3 \times 3$ , то  $F^2$  дає рецептивне поле  $5 \times 5$  (дві згортки  $3 \times 3$ ),  $F^3$  –  $7 \times 7$ , тощо. Нехай  $T$  – максимальний порядок (в практиці  $T = 4$ , що відповідає макс. полю  $9 \times 9$ ). Тоді вихід резидуальної гілки – це сума інкрементальних ознак від масштабу 1 до  $T$ :

$$\tilde{x} = \sum_{t=1}^T F^t(x), \quad T > 1, \quad (1.4)$$

Для прикладу, при  $T = 4$  блок акумулює ознаки з рецептивними полями 3,5,7,9 пікселів відповідно. Далі, щоб гнучко з'єднати ці багатомасштабні ознаки, використовується агрегаційний шлюз  $G$ . Він обчислює для кожної гілки  $t$  вектор ваг  $g^{(t)} = G(x^{(t)})$  розмірності  $c'$  (кількість каналів), де  $x^{(t)} = F^t(x)$ . Ці ваги модулюють вихід гілок поканалньо (елементно помножуються на тензор ознак). Підсумковий вихід блоку можна записати як:

$$\tilde{x} = \sum_{t=1}^T G(x^{(t)}) \odot x^{(t)}, \quad (1.5)$$

де  $\odot$  – поелементне множення (Hadamard-добуток), а  $G(x^{(t)})$  – вектор ваг для гілки  $t$ . Функція  $G$  реалізована як невелика нейромережа: глобальний pooling по просторових координатах, далі шар зниження розмірності і сигмоїдна активація, щоб отримати ваги в діапазоні  $[0,1]$ . Таким чином,  $\tilde{x}$  означає адаптивну комбінацію масштабів залежно від контенту зображення: якщо певний масштаб менш інформативний, відповідні ваги  $G$  будуть близькі до 0, і навпаки.

В кінці мережі OSNet отримані ознаки використовуються для класифікації особи: зазвичай після останнього шару ознак стоїть повнозв'язний шар на кількість ідентичностей в тренувальному наборі та softmax. Під час навчання застосовується softmax-кросентропія як основна функція втрат (може доповнюватися triplet-loss або center-loss для покращення метричного навчання). На практиці, OSNet тренують або з нуля на даних Re-ID, або з попередньою ініціалізацією на ImageNet. Автори OSNet показали, що навіть без pre-training мережа успішно навчається завдяки малій кількості параметрів.

На відміну від універсальних CNN на кшталт ResNet, архітектура OSNet спроектована спеціально під вимоги Re-ID. Її ключові риси – omni-scale ознаки та динамічна агрегація – спрямовані на розв'язання саме проблем Re-ID. Задача вимагає розпізнавати людину за дрібними унікальними деталями (наприклад,

взуття, аксесуари) і за загальним виглядом (стать, одяг, силует). Блоки OSNet одночасно вчаться локальним та глобальним ознакам на кожному рівні, що підвищує дискримінативність представлень. Динамічний шлюз дозволяє мережі підлаштовувати фокус під конкретний вхід: наприклад, для двох різних осіб у схожому одязі мережа може звернути більше уваги на дрібні відмінності (взуття, рюкзак), а для однієї особи в різних камерах – навпаки, на грубі риси, ігноруючи зміни пози. Крім того, компактність OSNet (2.2М параметрів) зменшує ризик перенавчання на відносно невеликих датасетах Re-ID, де кількість осіб (класів) – кілька сотень чи тисяч. Завдяки цьому OSNet можна ефективно тренувати з нуля на специфічний набір (напр., Market-1501) і отримати високу якість, тоді як більші моделі зазвичай потребують попереднього навчання на зовнішніх даних [79].

Хоча базова OSNet не має явного механізму для позицій частин тіла, існують розширення, наприклад, запропоновано OSNet-AIN – модифікацію OSNet з інтеграцією Instance Normalization для кращої узагальнюваності між доменами (різними камерами). OSNet також використовується як бекбон у багатьох сучасних рішеннях Re-ID, його багатомасштабні ознаки слугують основою для додавання модулів уваги, навчання по частинах тіла тощо, без потреби в громіздкому ResNet. Таким чином, OSNet і подібні йому архітектури CNN встановили високу планку як баланс між легкістю моделі та точною ідентифікацією особи.

Generative Adversarial Networks (GAN) – це модель, що складається з двох нейромереж, які навчаються змагально: генератор  $G$  намагається створювати реалістичні зображення, а дискримінатор  $D$  намагається відрізнити згенеровані зображення від справжніх. Класичний GAN навчається за [80]:

$$\min_G \max_D \left( E_{x \sim p_{\text{dt}}(x)} [\log D(x)] + E_{z \sim p_z(z)} [\log (1 - D(G(z)))] \right). \quad (1.5)$$

де  $z$  – випадковий шум або прихований код, з якого  $G$  генерує зображення  $G(z)$ , а  $D(I)$  видає ймовірність, що зображення справжнє. У контексті Re-ID GAN-и застосовують не просто для генерації «фейкових» осіб, а як інструмент для розширення та адаптації даних. Основні принципи використання GAN в Re-ID:

- Аугментація даних: Генератор створює варіації існуючих образів людини – змінюючи фон, освітлення або позу – але зберігаючи її особу. Це підвищує стійкість моделі до змін умов зйомки. Наприклад, GAN може навчитися перевдягати людину або змінювати ракурс, щоб збільшити різноманітність тренувального сету [81].

- Перенесення стилю між доменами: У сценарії *unsupervised domain adaptation* GAN використовується, щоб перенести зображення з датасету  $A$  у стиль датасету  $B$ , зберігаючи особу. Таким чином, модель, навчена на наборі  $A$ , краще працюватиме на наборі  $B$ , навіть без міток  $B$ . Це здійснюється через *image-to-image translation* GAN (наприклад, CycleGAN) зі збереженням ідентичності людини [82].

- Дизентанглінг ознак (роз'єднання факторів): Деякі GAN-рішення для Re-ID навчаються розділяти ознаки, пов'язані з особою (ідентифікаційні) та з позою/фоном. Генератор може комбінувати код зовнішності (*appearance*) однієї особи з кодом структури (*pose/поза*) іншої, генеруючи нове зображення. Це допомагає отримати нові приклади тієї ж особи в іншій позі чи обстановці, що корисно для навчання інваріантних ознак [83].

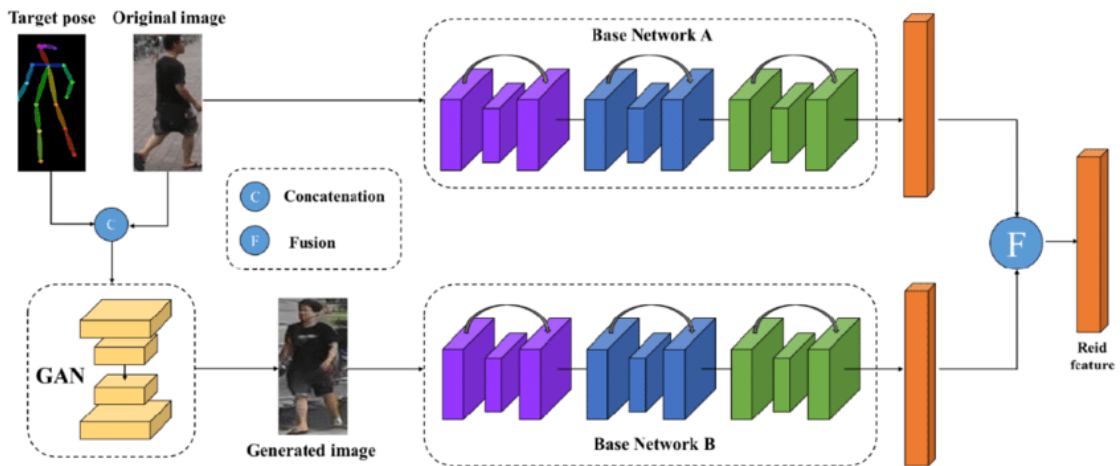


Рис. 1.9 Архітектура генеративно змагальних мереж

Архітектурно GAN для Re-ID часто ускладнюється порівняно з базовим GAN [84]:

- Генератор може містити два енкодери: енкодер  $E_{ID}$  для отримання коду особи (виду одягу, кольорів тощо) і енкодер  $E_{pose}$  для структури (поза, фон). Далі обидва коди комбінуються і декодер генерує зображення.

- Дискримінатор може бути подвійним: один  $D_{style}$  перевіряє, чи відповідає стиль цільовому домену (для переносів стилю), другий  $D_{ID}$  – чи збігається ідентичність або інші атрибути. Іноді замість другого дискримінатора використовують прямий класифікатор особи на згенерованих зображеннях, щоб контролювати їхню особу.

Базова функція втрат GAN для завдань Re-ID розширюється додатковими компонентами. Зокрема, при перенесенні стилю між доменами  $A$  і  $B$  застосовується підхід cycle-consistency (циклічною узгодженістю) [85].

$G_{A \rightarrow B}$  – генератор стилізує зображення з домену  $A$  під стиль  $B$ , а  $G_{B \rightarrow A}$  – навпаки. Тоді крім змагального втрату додають цикл-втрату:

$$\begin{aligned} \mathcal{L}_{\text{cyc}} = & E_{I_A \sim P_A} [ |G_{B \rightarrow A}(G_{A \rightarrow B}(I_A)) - I_A|_1 ] \\ & + E_{I_B \sim P_B} [ |G_{A \rightarrow B}(G_{B \rightarrow A}(I_B)) - I_B|_1 ]. \end{aligned} \quad (1.6)$$

щоб забезпечити оборотність перекладу (уникнути довільних змін особи). Проте ключовим є збереження особи: вводять додатковий критерій  $\mathcal{L}_{ID}$ , який штрафує зміну ідентичності [86]. Тоді загальна функція втрат подається як:

$$\mathcal{L}_{PTGAN} = \mathcal{L}_{style} + \lambda_1 \mathcal{L}_{ID}, \quad (1.7)$$

де  $\mathcal{L}_{style}$  включає змагальні та циклічні втрати для перенесення стилю, а  $\mathcal{L}_{ID}$  – штраф за зміну особи. Тобто, генератор намагається мінімізувати не тільки різницю стилю між вихідним і цільовим доменом, але й зберегти ті самі розпізнавальні риси людини.  $\mathcal{L}_{ID}$  можна реалізувати як вимогу, щоб сгенероване зображення класифікувалося тим же  $ID$ , що і оригінал (через заморожену модель або через спільний тренінг дискримінатора як класифікатора).

Для успішного використання GAN у Re-ID генератор має навчитися створювати нові зображення, які розширюють варіації, але не порушують ідентичність. У разі Domain Adaptation (переносу знань на новий датасет) це означає: показати моделі такі ж самі особи, але начебто зняті іншими камерами чи в іншому середовищі.

PTGAN саме це і робить: переносить зображення осіб з набору A в стиль набору B (різні фони, освітлення), щоб подолати domain gap. При цьому алгоритм контролює, щоб ключові ознаки особи лишилися (через складову втрат  $\mathcal{L}_{ID}$ ). Як зазначають автори, PTGAN навчає функцію переносу, що узгоджує стиль з цільовим датасетом та зберігає зовнішність та особу людини. Це дозволяє згенерувати для кожної особи з A кілька стилізованих зразків, які потім використовуються для навчання кінцевої Re-ID моделі на датасеті B (вже як розширений тренувальний набір). Таким чином, можна обійтися без мічених даних в B – модель навчається на синтетичних зразках, які містять особи з A, але «виглядають» як B [87].

В сценарії Data Augmentation для одного домену GANи допомагають побороти брак варіативності даних. Метод FD-GAN (NeurIPS 2018) генерує нові зображення тієї ж особи з іншою позою: він навчає два підпростори – “identity”

та “pose” – і комбінує різні пари (ідентичність від одного зображення, поза від іншого) для синтезу. Отримані кадри додають у тренування як додаткові приклади з тим же лейблом особи. Це робить модель стійкішою до змін поз.

Схожим шляхом іде DG-Net (CVPR 2019) у ньому енкодер розділяє ознаки на код зовнішнього вигляду та код структури (положення тіла) для кожного зображення; генератор потім може переставляти коди між різними особами, отримуючи “склеєні” зображення (одна людина в позі іншої). Важливо, що DG-Net об’єднує генеративне і дискримінативне навчання: згенеровані кадри *on the fly* подаються назад в мережу розпізнавання, покращуючи її здатність екстрагувати релевантні фічі. Таке спільне тренування дало змогу значно підвищити точність в порівнянні з простою аугментацією на готовому генераторі [88].

Загалом, адаптація GAN до Re-ID полягає в тому, щоб навчити генератор генерувати максимально реалістичні зміни образу людини, які покривають проблемні варіації (фон, камера, поза), при цьому залишаючи її впізнаваною для моделі.

Vision Transformer (ViT) та його варіації нещодавно були адаптовані для задач комп’ютерного зору, у тому числі Re-ID. Трансформер оперує не пікселями як CNN, а послідовністю патчів: зображення розбивають на рівні блоки (наприклад,  $16 \times 16$  пікселів кожен), які розгортаються в вектори та лінійно проектуються в певний розмір простору ознак. Ця послідовність векторів  $x_i$  доповнюється позиційними ембедінгами  $p_i$ , після чого подається на вхід трансформерного енкодера. Кожен шар енкодера складається з багатоголового механізму самоуваги і каскаду MLP-шарів з залишковими з’єднаннями. Механізм Self-Attention обчислює для кожної пари патчів вагу взаємодії, що дозволяє моделі захоплювати довгострокові залежності в зображенні (наприклад, зв’язок між віддаленими частинами тіла) [89].

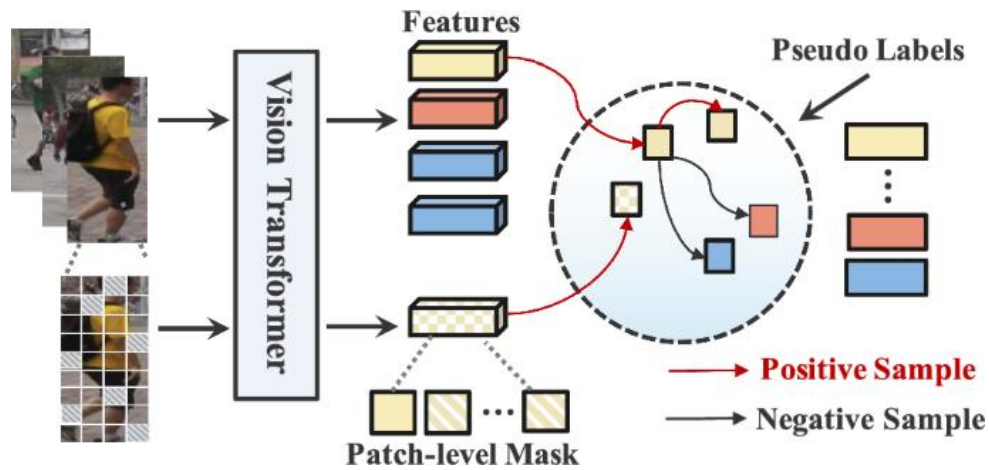


Рис. 1.10 Архітектура Vision Transformer

Формально, нехай  $Q = XW^Q$ ,  $K = XW^K$  та  $V = XW^V$  – лінійні перетворення матриці патч-векторів  $X$  (розмірності  $N \times d$ , де  $N$  – кількість патчів,  $d$  – розмір ознак) в запити, ключі і значення. Тоді увага для одного рахується як [90]:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d}}\right) V \quad (1.8)$$

тобто для кожного патча ваги уваги пропорційні скалярному добутку між його запитом  $q$  та всіма ключами  $k$  інших патчів (нормованому на  $\sqrt{d}$ ). Цей механізм дозволяє агрегувати глобальний контекст: вихідні ознаки патча будуть комбінацією ознак усіх патчів, з різними вагами. У багатоголовій увазі (multi-head attention) процес повторюється  $h$  разів з різними проєкціями ( $W_i^Q$ ,  $W_i^K$ ,  $W_i^V$ ) і результати конкатенуються. Після кількох шарів самоуваги трансформер отримує глибокі ознаки, що інтегрують інформацію по всьому зображенню.

На відміну від CNN, трансформер не втрачає інформацію через pooling або stride – роздільна здатність ознак залишається на рівні патчів, і модель може навчитися навіть дрібних деталей, якщо вони важливі. Однак, трансформеру бракує вбудованої індуктивної сутності згорток (сусідні пікселі не пов'язані,

доки модель сама не навчиться їх зв'язку), тому його успішне навчання потребує багато даних або попереднього навчання [91].

В задачі Re-ID трансформер зазвичай використовується як бекбон для отримання ознак, на основі яких вирішується класифікація особи або порівняння зображень. Найпоширеніший підхід – додати спеціальний токен класу (classification token) до послідовності патчів, який служить агрегованим представленням усього зображення (як у оригінальному ViT). Після проходження через трансформер, вихідний вектор цього [CLS]-токена використовується як дескриптор особи або подається на класифікатор ID. Таким чином, функція втрат аналогічна до CNN-випадку: softmax-кросентропія по ідентичностях +, за потреби, triplet-loss по ембедінгах для збільшення роздільної здатності між різними особами [92].

Оскільки трансформери оперують наборами ознак без просторової структури, адаптація їх до Re-ID часто включає додаткові ембедінги чи структури, щоб врахувати специфіку даних спостереження:

- Позиційні ембедінги  $p_i$  – стандартний компонент, що кодує позицію патча в зображенні. В Re-ID це допомагає моделі розуміти розташування частин тіла (верх/низ, ліва/права сторона), що важливо, адже перестановка патчів випадковим чином руйнує значення ознак.

- Додаткові токени або ембедінги камери/ракурсу, для цього вводять side-information embeddings, які додаються до патчів або до [CLS]-токена і несуть інформацію про камеру, під якою знято зображення, щоб пом'якшити зсув розподілу між різними камерами (колірна температура, кут огляду) – модель отримує сигнал, яка саме це камера, і може навчитися абстрагуватися від цього [93].

Трансформери вимагають адаптації під специфіку Re-ID з кількох причин [94]:

- Великий обсяг даних для навчання: Набори Re-ID (навіть найбільші, як MSMT17) значно менші за ImageNet. Тому трансформерні моделі зазвичай ініціалізують на великому датасеті (ImageNet-1k або навіть ImageNet-21k) і лише

тонко донавчають на Re-ID. Це важливий крок, щоб модель вже «вміла» базово бачити загальні об'єкти і не перенавчилася на малій вибірці осіб.

- Інваріантність до фонів та ракурсів: Без inductive bias згортки, трансформер може надто уваги приділяти фону чи шуму. В Re-ID це критично, адже фон камери не пов'язаний з особою. Модулі на кшталт SIE в TransReID явно дають моделі знати про зміну камери, що зменшує вплив фону. Інші роботи пропонували вбудовувати маскуванню фонів або навчати трансформер витягувати ознаки окремо з декількох частин тіла (частково імітуючи підхід Part-Based CNN).

- Висока роздільна здатність та обчислювальна вартість: Зображення  $256 \times 128$ , типове для Re-ID, при патчі  $16 \times 16$  дає 128 патчів. Трансформер обробляє повний граф пов'язаності  $O(N^2)$  відношень, що може бути повільно. Тому TransReID, наприклад, випробовує різні розміри патчів і навіть роздільність  $384 \times 128$  з більшим патчем, балансує якість і швидкість.

- Вихідні ознаки та metric learning: Зазвичай трансформер видає один вектор [CLS] - глобальний дескриптор. Для Re-ID часто корисно мати декілька регіональних дескрипторів (наприклад, окремо для верхньої/нижньої частини тіла). Тому існують модифікації, де додаються кілька [CLS]-токенів, кожен навчається фокусуватися на певній частині зображення. Є також підходи, де замість [CLS] усереднюють ознаки всіх патчів або використовують їхню сукупність для зіставлення образів [95].

TransReID справляється з більшістю цих питань, пропонуючи просте але потужне рішення: використати силу трансформерів (глобальний контекст) і мінімізувати їхні слабкості (чутливість до перестановок та камерних зсувів) через JPM та SIE. Результати підтверджують ефективність, трансформерна модель краще помічає тонкі відмінності між дуже схожими людьми, особливо коли потрібен аналіз глобальної структури сцени. Наприклад, TransReID візуально зберігає в активаціях деталі рюкзака, які CNN-мережа втратила через pooling. Таким чином, адаптовані трансформери нині задають новий стандарт

точності в Re-ID, хоч і залишаються більш вимогливими до даних та ресурсів [96].

Нижче у таблиці наведено порівняння продуктивності представників кожного з трьох підходів – OSNet, GAN-орієнтованої моделі (DG-Net) та трансформера (TransReID) – на двох популярних датасетах: Market-1501 і DukeMTMC-reID. Метрики – це Rank-1 точність (Top-1) і mAP (mean Average Precision), вищі значення кращі. Всі результати отримані в стандартних умовах (одноразовий запит, без ре-ранжування).

Таблиця 1.1 Порівняння точності (%) на популярних наборах даних для представників архітектур OSNet, GAN (DG-Net) та трансформера (TransReID).

Метод	Market-1501 Rank-1	Market-1501 mAP	DukeMTMC Rank-1	DukeMTMC mAP
OSNet (ICCV 2019)	94.8%	84.9%	88.6%	73.5%
DG-Net (CVPR 2019)	94.8%	86.0%	86.6%	74.8%
TransReID (ICCV 2021)	95.2%	89.5%	90.7%	82.6%

Як видно з таблиці, OSNet вже забезпечує високий рівень точності: ~94.8% Rank-1 на Market-1501. Це вражає, враховуючи, що модель майже в 10 разів менша за ResNet50. DG-Net (GAN-підхід) ще підвищує mAP на Market-1501 до 86.0%, що свідчить про кращу дискримінативність серед усіх галерейних зображень. Це підтверджує користь генеративного компонента: додаткові синтезовані приклади допомагають моделі покрити більше варіацій і підняти середню точність розпізнавання. На DukeMTMC DG-Net дає трохи вищу mAP (74.8% vs 73.5% у OSNet), хоча Rank-1 у OSNet вищий.

Найкращі показники демонструє TransReID – завдяки потужності трансформерів він досягає 95.2% Rank-1 і 89.5% mAP на Market-1501,

встановлюючи новий рекорд для цього набору (без додаткового перенавчання або ансамблю). На DukeMTMC трансформер також лідирує з ~90-91% Rank-1 та понад 82% mAP. Це суттєвий крок вперед у порівнянні з попереднім поколінням CNN: наприклад, середній mAP ~73-75% для OSNet/DG-Net проти 82% у TransReID. Відзначимо, що TransReID – модель більша і навчалась з pre-training, її перевага проявляється найбільше на більш складних датасетах (MSMT17 тощо), де глобальне контекстне бачення трансформера дозволяє краще розрізнити людей у натовпі та при сильних варіаціях оточення.

#### **1.4 Аналіз проблем повторної ідентифікації людей та постановка задач**

Системи повторної ідентифікації людини у відеоспостереженні призначені для визначення, чи з'являлась певна особа в різних місцях та у різний час на записах з не пов'язаних між собою камер. В бізнес-середовищах, таких як мегамоли чи роздрібні магазини, такі технології можуть використовуватися для відстеження переміщення клієнтів між торговими зонами, виявлення постійних відвідувачів або ж розпізнавання відомих правопорушників (наприклад, помічених раніше магазинних злодіїв). Це відкриває нові можливості для аналізу поведінки споживачів та підвищення безпеки, але водночас породжує низку технічних, правових та етичних питань. Нижче представлено глибокий аналіз кожного з цих аспектів, з реальними прикладами та посиланням на авторитетні джерела.

Повторна ідентифікація людей є ключовим елементом сучасних систем відеоспостереження, безпеки та аналізу даних. Вона дозволяє відстежувати переміщення осіб у різних місцях і в різний час, що має значний потенціал для підвищення ефективності в багатьох сферах, таких як громадська безпека, роздрібна торгівля та транспорт. Однак, попри досягнення в цій галузі, існує низка проблем, які обмежують практичне застосування технологій повторної ідентифікації.

Однією з головних проблем є варіативність зовнішнього вигляду людей. Люди можуть змінювати свій одяг, аксесуари, зачіску або навіть фізичні характеристики, що ускладнює процес ідентифікації. Наприклад, особа може носити різний одяг протягом дня, змінювати взуття або використовувати різні аксесуари, такі як сумки чи капелюхи. Зміни в зовнішньому вигляді суттєво впливають на алгоритми, які покладаються на візуальні ознаки для ідентифікації.

Умови зйомки також значно впливають на точність ідентифікації. Камери можуть мати різні технічні характеристики, а освітлення може варіюватися від яскравого сонячного світла до тьмяного штучного освітлення, що змінює кольорову гамму та контраст зображень. Кути огляду камер можуть бути різними, що призводить до того, що особа може бути знята з різних ракурсів — спереду, ззаду або збоку. Це ускладнює процес порівняння зображень, оскільки багато алгоритмів краще працюють з фронтальними зображеннями.

Оклюдія та переповнення сцени є ще однією суттєвою проблемою. У місцях з великою кількістю людей або об'єктів особа може бути частково або повністю закрита іншими людьми чи предметами. Це робить ідентифікацію складнішою, оскільки алгоритми отримують неповну інформацію про зовнішній вигляд особи. Алгоритми повинні бути здатними розпізнавати осіб навіть за частковими ознаками, що є технічно складним завданням.

Обмеженість даних для навчання моделей є ще одним викликом. Для ефективного навчання моделей машинного навчання необхідні великі та різноманітні набори даних, які б відображали різні умови зйомки, варіації зовнішнього вигляду та інші фактори. Проте збір таких даних може бути обмеженим через законодавчі обмеження, етичні міркування та високі витрати. Наявні набори даних можуть бути незбалансованими, з переважанням певних типів зображень або умов, що призводить до упередженості моделей і зниження їх точності в реальних умовах.

Високі вимоги до обчислювальних ресурсів також становлять значну технічну проблему. Сучасні алгоритми глибокого навчання демонструють високі показники точності, але потребують значних обчислювальних

потужностей для навчання та роботи в реальному часі. Це може бути проблематичним для впровадження в системах з обмеженими ресурсами або для обробки відеопотоків з великої кількості камер одночасно. Оптимізація алгоритмів для зменшення обчислювальної складності без втрати точності є складним завданням.

Узгодження міжкамеральних характеристик є ще одним технічним викликом. Камери в системі можуть мати різні параметри та налаштування, що призводить до різних візуальних характеристик зображень навіть для однієї й тієї самої особи. Різниця в балансі білого, експозиції та інших параметрах може змінювати кольори та інші візуальні ознаки. Для подолання цієї проблеми використовуються методи нормалізації кольорів та адаптації стилю зображень, але вони додають складності до системи та можуть не завжди бути ефективними.

Точність і надійність алгоритмів в реальних умовах часто є нижчою, ніж в лабораторних тестах. Реальні умови можуть бути набагато складнішими та непередбачуваними, що вимагає додаткового налаштування та адаптації моделей. Алгоритми повинні бути здатними справлятися з шумом, артефактами зображення, втратами даних та іншими непередбаченими факторами [1].

Крім технічних проблем, існують також етичні та правові виклики, пов'язані з повторною ідентифікацією. Використання технологій спостереження може порушувати право на приватність і конфіденційність особи. Багато людей можуть не знати, що їхні дані збираються та аналізуються, що викликає занепокоєння щодо несанкціонованого спостереження. Масове впровадження таких технологій може призвести до створення "суспільства спостереження", де кожен крок людини відстежується без її згоди.

Усі ці проблеми потребують комплексного підходу до аналізу самих наборів даних та застосування моделей машинного навчання, які в подальшому інтегруються в інформаційні технології повторної ідентифікації людини.

Ці фактори призводять до того, що точність алгоритмів у реальних сценаріях помітно нижча, ніж у лабораторних тестах. Навіть найкращі моделі, навчені на відкритих наборах даних, зіштовхуються з невідомими умовами на

практиці – новим середовищем, ракурсами, унікальним одягом тощо. Дослідники відзначають, що через перелічені виклики завдання Re-ID досі залишається значною мірою нерозв’язаним для повністю неконтрольованих умов. Іншими словами, жодна система не гарантує 100% впізнавання людини на відео у всіх можливих ситуаціях – помилки трапляються, особливо коли змінюються умови або з’являються нові, раніше не бачені алгоритмом фактори. Рівень точності та приклади з практики. В контрольованих експериментах нейромережеві моделі досягали понад 90% точності впізнавання на обмежених наборах осіб (наприклад, визначення правильного збігу у топ-1 або топ-5 результатах). Однак у реальному житті точність може сильно падати. Наприклад, якщо магазин обладнаний десятками камер, система має відстежити клієнта по всьому магазину, незважаючи на те, що він може зникати з поля зору і з’являтися знову.

Масштабування систем повторної ідентифікації – тобто розгортання її на багато камер і обробка великих потоків відео – пов’язане з високими обчислювальними витратами. Необхідно забезпечити швидке порівняння ознак з десятками тисяч зображень у базі, і при цьому регулярно оновлювати моделі під нові дані (наприклад, якщо змінюється оформлення магазину, освітлення тощо, модель може потребувати донавчання).

Умовна гіпермережа дозволяє адаптувати ваги основного CNN-екстрактора ознак до умов конкретного вхідного зображення, генеруючи їх завдяки окремій мережі-гіпермережі. Такий підхід враховує особливості освітлення, ракурсів та стилю одягу для кожного кадру, що допомагає зменшити вплив варіативності реальних сцен на якість виділених фіч. За рахунок індивідуалізованих ваг мережа здатна точніше розрізняти однакові особи в різних умовах, навіть коли вони повернені обличчям під різними кутами або знаходяться під змінним освітленням.

Експериментальні результати на наборі Market-1501 показали, що застосування умовної гіпермережі збільшує mAP на 4–6 % та Rank-1 на 3–5 % порівняно зі стандартним CNN без адаптації ваг. Zhang et al. у своєму

дослідженні продемонстрували, що використання динамічно згенерованих фільтрів дозволяє моделі стабільно працювати в сценаріях open-world та знижує вплив domain shift між різними камерами.

GAN-аугментація даних передбачає синтез нових реалістичних зображень тих самих осіб із різноманітним одягом, позами та освітленням, зберігаючи ключові ідентифікатори особи. Умовний GAN навчається враховувати специфіку окремого домену чи умов зйомки, наприклад зміну фону чи колірне відтворення в різних камерних установах, що дозволяє розширити тренувальний датасет без ручної анотації нових кадрів. Це сприяє кращій генералізації основного CNN при аналізі клієнтів у реальних умовах.

За даними Li et al., додавання синтетичних зображень від Conditional CycleGAN до оригінального набору DukeMTMC-reID збільшувало mAP на 3–5 % та підвищувало Rank-1 на 2–4 % порівняно з тренуванням лише на реальних кадрах. Таке поєднання дозволяє моделі зменшити overfitting та покращити стабільність розпізнавання в різних торгових зонах та умовах освітлення.

Враховуючи проведений аналіз та визначені проблеми, для досягнення мети даного дослідження необхідно вирішити такі задачі:

1. Виконати аналіз сучасних методи глибокого навчання для повторної ідентифікації людини на обмежених або неповноцінних наборах даних (Market-1501, DukeMTMC-reID, MSMT17). Це включає аналіз існуючих методів, їх переваг та недоліків, а також визначення факторів, що впливають на якість ідентифікації. Проаналізувати переваги й недоліки підходів на основі CNN, умовних гіпермереж (conditional hypernetworks) та GAN-аугментації, виявити ключові технічні виклики (варіативність поз, ракурсів, одягу, освітлення, оклюзія, domain shift).

2. Розробити інформаційну технологію, методи та моделі з гібридною умовною гіпермережею та GAN-аугментацією. Запропонувати архітектуру, в якій умовна гіпермережа генерує адаптивні ваги для основного CNN-екстрактора ознак, покращуючи здатність моделі до розрізнення осіб. Інтегрувати модуль GAN для синтезу додаткових реалістичних зображень людей

із різноманітними ознаками (одяг, поза, ракурси), що вирішує проблему дефіциту даних.

3. Створити алгоритмічне та програмне забезпечення для застосування запропонованої інформаційної технології та методів повторної ідентифікації людини. Це включає розробку програмних модулів та інструментів, які реалізують запропоновані алгоритми.

4. Розробити комп'ютерні програми на мові програмування Python із використанням фреймворків глибокого навчання для застосування інтелектуальної інформаційної технології. Це забезпечить можливість відтворення результатів та зручність використання розроблених методів.

5. Апробувати та впровадити розробленої інтелектуальної інформаційної технології на реальних даних та задачах. Це передбачає проведення експериментів на відкритих наборах даних, оцінку ефективності запропонованих методів та порівняння з існуючими підходами.

Дане дослідження є актуальним у задачі повторної ідентифікації людини, оскільки поєднання умовної гіпермережі з CNN дозволяє динамічно налаштовувати ваги екстрактора ознак під конкретні умови зйомки (ракурс, освітлення, стиль одягу) та тим самим суттєво підвищувати стійкість моделі до варіативності реальних даних. Водночас інтеграція GAN-аугментації вирішує проблему обмеженості анотованих зображень: синтез додаткових реалістичних кадрів значно скорочує витрати на ручну розмітку та розширює репрезентативність тренувального набору, що безпосередньо сприяє підвищенню мір продуктивності (mAP, Rank-1).

Розробка відповідного програмного забезпечення та інструментів для реалізації запропонованої гібридної архітектури забезпечує можливість швидкого впровадження методів у комерційних системах відеоспостереження.

Результати розділу опубліковано в таких роботах автора: [2], [3], [4], [9], [10], [11], [12], [13].

## РОЗДІЛ 2

# РОЗРОБКА МЕТОДІВ ТА МОДЕЛЕЙ ПОВТОРНОЇ ІДЕНТИФІКАЦІЇ ЛЮДЕЙ З ВИКОРИСТАННЯМ ЗГЕНЕРОВАНИХ ЗОБРАЖЕНЬ

### 2.1. Розробка базової моделі повторної ідентифікації людей

У рамках даного дослідження було розроблено базову модель нейронної мережі OSNet для задачі person re-identification, зосереджений на її omni-scale архітектурі та механізмі адаптивної агрегації багатомасштабних ознак через Unified Aggregation Gate. OSNet забезпечує одночасне виділення дрібномасштабних і глобальних ознак завдяки поєднанню depthwise separable convolution та ResNet-подібних резидуальних bottleneck-блоків, що гарантує компактність моделі та високу швидкість інференсу. У цьому контексті представлено комплексний огляд математичної моделі згорткових операцій, методів попередньої обробки зображень (масштабування до  $256 \times 128$ , нормалізація інтенсивності) та підходів до аугментації даних (random flip, random crop, color jitter, random erasing) для підвищення якості навчання OSNet.

Omni-Scale Network (OSNet) – це сучасна глибока нейронна мережа, спеціально розроблена для задач повторної ідентифікації людини на зображеннях з різних камер. Ключова ідея OSNet полягає в навчанні так званих omni-scale ознак – тобто ознак усіх масштабів, від локальних дрібних деталей до глобальних ознак всього тіла. На відміну від звичайних CNN, які мають фіксований розмір рецептивного поля на кожному шарі, OSNet здатна одночасно охоплювати різні просторові масштаби. Такий підхід мотивований природою задачі повторної ідентифікації для впізнавання особи можуть бути важливими як дрібні локальні деталі (наприклад, взуття, логотип на одязі, зачіска), так і загальні глобальні риси (силует, колір всього вбрання). Omni-scale ознаки означають поєднання однорідних масштабів (ознак одного рівня деталізації) та гетерогенних масштабів (комбінації ознак різних масштабів). Таким чином,

мережа навчається розрізняти людей, спираючись на весь спектр характеристик – від дрібних деталей до цілісного образу.

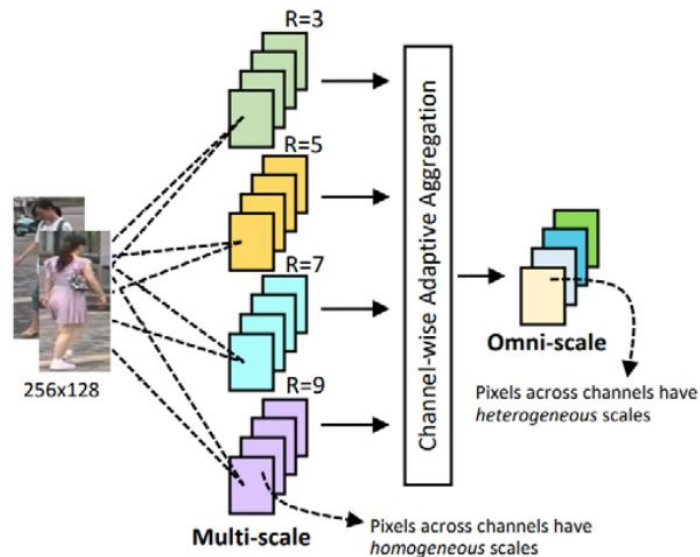


Рис. 2.1 Загальна структура мережі OSNet

Загальна структура мережі OSNet будується шляхом послідовного стекування спеціальних блоків (omni-scale residual blocks) один за одним. Кожен блок додає до свого вхідного тензора вивчені зміни (residual), що полегшує тренування глибокої мережі. В таблиці 1 наведено конфігурацію OSNet для вхідного зображення розміру 256×128 пікселів. Мережа починається з початкової згортки 7×7 та максимального пулінгу 3×3 для скорочення розміру карти ознак. Далі йдуть кілька згорткових стадій (conv2, conv3, conv4) – на кожній з них розташовано по 2 omni-scale блоки з наростанням числа каналів (256→384→512). Між стадіями використовуються перехідні шари: 1×1 згортка для зміни числа каналів та усереднюючий пулінг 2×2 зі страйдом 2 для зменшення просторового розміру. Після останнього резидуального блоку проводиться глобальний Average Pooling (GAP) і повнозв'язний шар, що утворює 512-вимірний вихідний вектор ознак. Така архітектура без жодних додаткових модулів містить всього ~2.2М параметрів і виконує ~978М операцій,

що втричі менше за еквівалентну мережу зі звичайними згортками (~6.9М параметрів, ~3.38 млрд операцій).

Таблиця 2.1 Спрощена архітектура OSNet для розміру вхідного зображення  $256 \times 128$

Шар	Розмір карти ознак	Операція (шари)	Примітки
Вхід	$256 \times 128$ , 3	—	RGB-зображення
Conv1	$128 \times 64$ , 64	$7 \times 7$ згортка, stride 2	Початкове згортання
MaxPool	$64 \times 32$ , 64	$3 \times 3$ максимальний пулінг, stride 2	Зменшення розміру
Conv2	$64 \times 32$ , 256	Omni-scale блок $\times 2$	2 блоки з сумарним вих. 256 каналів
Transition	$32 \times 16$ , 256	$1 \times 1$ згортка + $2 \times 2$ усереднюючий пулінг	Зміна каналів + даунсемплінг
Conv3	$32 \times 16$ , 384	Omni-scale блок $\times 2$	Збільшення до 384 каналів
Transition	$16 \times 8$ , 384	$1 \times 1$ згортка + $2 \times 2$ усереднюючий пулінг	
Conv4	$16 \times 8$ , 512	Omni-scale блок $\times 2$	Вихід 512 каналів
Conv5	$16 \times 8$ , 512	$1 \times 1$ згортка	Фінальна згортка
GAP	$1 \times 1$ , 512	Глобальний Average Pooling	Вихідний вектор 512-d
FC (classifier)	$1 \times 1$ , N (к-ть ID у тренуванні)	Повнозв'язний Softmax	Класифікатор особи (під час навчання)

Основним будівельним елементом OSNet є спеціальний омнімасштабний резидуальний блок. Він базується на структурі bottleneck-блоку, але з суттєвими удосконаленнями для багатомасштабного вилучення ознак. Загалом блок має вигляд: вхідний тензор  $x$  спочатку проходить через  $1 \times 1$  згортку для зменшення розмірності (каналів). Далі результат надходить до кількох паралельних гілок (streams), кожна з яких складається з послідовності згорток  $3 \times 3$  (спеціального типу Lite згорток). Нехай таких гілок  $T$  – тоді в  $t$ -тій гілці виконується  $t$  послідовних згорток  $3 \times 3$ . Це означає, що гілка 1 має одне  $3 \times 3$  згортання (ефективний рецептивний розмір  $3 \times 3$ ), гілка 2 – два згортання (ефективне поле  $5 \times 5$ ), гілка 3 – три (поле  $7 \times 7$ ), тощо. Таким чином, гілка з індексом  $t$  виявляє ознаки масштабу  $(2t + 1) \times (2t + 1)$ . У стандартній конфігурації OSNet використано  $T = 4$  гілки, тож найбільший охоплюваний блоком масштаб –  $9 \times 9$ . Виходи всіх гілок згодом об'єднуються, після чого проходять через заключну  $1 \times 1$  згортку для відновлення початкової кількості каналів і додаються (residual connection) до початкового  $x$ . Формально вихід блоку можна описати як:

$$y = x + \tilde{x}, \quad \text{де} \quad \tilde{x} = \sum_{t=1}^T F_t(x) \quad (2.1)$$

де  $F_t(x)$  – функція перетворення у  $t$ -тій гілці (послідовність  $t$  згорток). Для  $T = 1$  формула (2.1) зводиться до звичайного резидуального блоку ResNet (одна згортка + skip connection), а для  $T > 1$  – реалізує сумування ознак на кількох масштабах. Shortcut-зв'язок  $x + \tilde{x}$  забезпечує збереження дрібномасштабних ознак: наприклад, ознаки з гілки  $t = 1$  (найменше поле  $3 \times 3$ ) напряму додаються до виходу і можуть пройти далі по мережі без “розмивання” більшими згортками. Це дозволяє наступним шарам оперувати повним спектром масштабів, накопичуючи все нові ознаки, але не втрачаючи раніше вилучені локальні деталі.

У кожній гілці блок використовує так звані “Lite 3×3” згортки замість стандартних. Це різновид Depthwise Separable Convolution, що розщеплює стандартне згортання на два етапи: (1) глибинна згортка (depthwise) – застосування окремого ядра  $k \times k$  до кожного каналу, (2) точкова згортка (pointwise  $1 \times 1$ ) – лінійна комбінація отриманих каналів для змішування інформації між ними. У випадку  $3 \times 3$  ядра, така факторизація зменшує обчислювальну складність з  $h \cdot w \cdot 3^2 \cdot c \cdot c'$  до  $h \cdot w \cdot (3^2 + c) \cdot c'$  операцій (де  $h, w$  – розмір карти ознак,  $c$  – вхідні та  $c'$  – вихідні канали). Кількість параметрів також скорочується з  $3^2 \cdot c \cdot c'$  до  $(3^2 + c) \cdot c'$ .

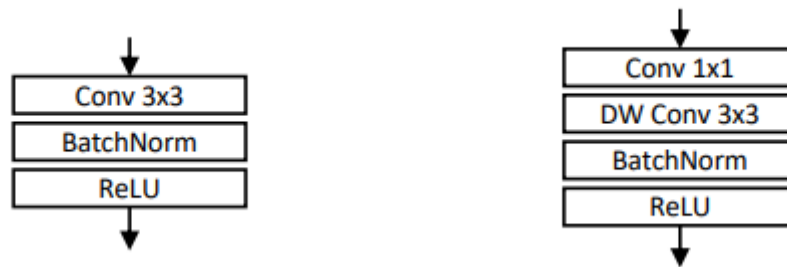


Рис. 2.2 Схема стандартного згорткового блока та блока з глибинно-роздільною згорткою

В OSNet використано модифікований порядок, спочатку  $1 \times 1$  точкова, а потім  $3 \times 3$  глибинна згортка з ReLU між ними. Такий порядок (Pointwise  $\rightarrow$  Depthwise) емпірично виявився більш ефективним для навчання різноманітних ознак. Ці Lite-конволюції суттєво зменшують кількість параметрів і ризик перенавчання, дозволяючи будувати компактні блоки без втрати представницької здатності моделі. Важливо, що всі гілки використовують однаковий базовий оператор (Lite 3×3), відрізняючись лише кількістю послідовних шарів. Це спрощує архітектуру та відповідає принципу масштабної однорідності: кожна гілка витягує однорідні ознаки на своєму фіксованому масштабі, а сумарно блок покриває гетерогенні (різномасштабні) комбінації.

Такий дизайн відрізняє OSNet від попередніх багатогілкових архітектур, як-от Inception або ResNeXt: в OSNet гілки строго слідують принципу нарощення рецептивного поля ( $3 \times 3$ ,  $5 \times 5$ ,  $7 \times 7$ ,  $9 \times 9$ ), тоді як в Inception використовувалися

різні типи фільтрів без динамічного об'єднання, а в ResNeXt – гілки однакового типу (гомогенні) з наступним усередненням.

У базовому варіанті можна було б просто підсумувати виходи гілок (як у формулі 2.1). Однак в OSNet запропоновано уніфіковані агрегаційні ворота – модуль, який динамічно зважує внесок кожної гілки перед сумуванням. Ідея полягає в тому, що важливість ознак різного масштабу може залежати від конкретного зображення: наприклад, для одних осіб ключовими будуть глобальні риси (колір одягу), для інших – дрібні деталі (візерунок, аксесуари). Тому AG здійснює контекстно-залежне злиття ознак. Формально вихід блоку з AG обчислюється як:

$$\tilde{x} = \sum_{t=1}^T G(x_t) \odot x_t, \quad \text{де } x_t = F_t(x). \quad (2.2)$$

де:

- $x_t$  – вихід  $t$ -тої гілки;
- $\odot$  – поканальне множення;
- $G(x_t)$  – вектор ваг для гілки  $t$  такої ж розмірності, як і  $x_t$  по каналах.

Вектор  $G(x_t)$  обчислюється невеличкою нейронною мережею (агрегаційними воротами) на основі самого виходу  $x_t$ . Зокрема,  $G$  складається з глобального усереднення по простору (Global Average Pooling), за яким іде двошаровий перцептрон: перший шар (зі зменшенням розмірності, напр. в 16 разів) з ReLU, і другий – з сигмоїдою на виході. Така структура аналогічна механізму Squeeze-and-Excitation (SE), але важливо, що той самий модуль  $G$  застосовується до всіх гілок спільно (уніфіковано).

Тобто параметри MLP-ворот не залежать від номера гілки  $t$  – всі  $x_t$  проходять через одну й ту саму функцію  $G$  (ваги не дублюються для кожної гілки). Завдяки цьому кількість параметрів не росте з числом гілок  $T$ , що робить модель масштабованішою. Крім того, при такому рішенні під час навчання

градієнти від усіх гілок спільно впливають на налаштування AG: градієнт помилки по параметрах  $G$  пропорційний сумі сигналів від кожної гілки. Це означає, що ворота навчаються узгоджувати інформацію між усіма масштабами одразу, отримуючи сильний сукупний сигнал, – на відміну від окремих воріт на кожній гілці, які навчались би ізольовано.

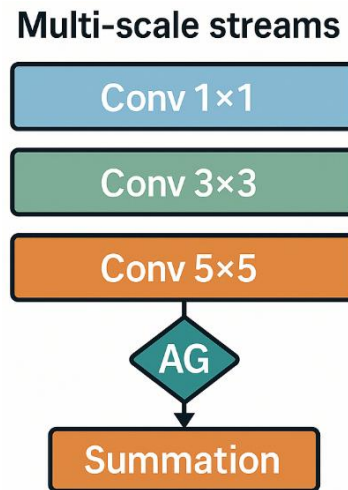


Рис. 2.3 Схеми модуля AG (Aggregation Gate) в OSNet

AG присвоює кожному каналу кожної гілки певну вагу в діапазоні  $[0,1]$  (сигмоїда), підсилюючи корисні ознаки і пригнічуючи менш інформативні для даного зображення. У результаті блок генерує омнімасштабний вихід  $\tilde{x}$  – злиття всіх масштабів з адаптивним налаштуванням внеску кожного. Після цього, як зазначено вище, додається резидуальний шлях і вихід у надходить до наступного блоку.

OSNet використовує типові для CNN активації:

– ReLU [97] (Rectified Linear Unit) – негативні значення обнуляються, позитивні пропускаються лінійно:

$$\text{ReLU}(z) = \max(0, z). \quad (2.3)$$

ReLU використовується після більшості згорткових шарів для внесення нелінійності в модель і боротьби з градієнтним згасанням.

– Sigmoid [98] (логістична функція) – використовується в модулі AG для обчислення ваг, оскільки стискає вихід до діапазону  $[0,1]$ . Формула:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (2.4)$$

Завдяки сигмоїді виходи агрегаційного шлюзу інтерпретуються як коефіцієнти уваги (0 – ігнорувати канал, 1 – пропустити повністю). Перед сигмоїдою в AG стоїть прихований шар з ReLU.

– Softmax [99] – функція нормалізації, яка перетворює вихід класифікаційного шару на ймовірнісний розподіл по класах (ідентичностях). Для вектора логітів  $z$  softmax визначається як:

$$\text{softmax}(z)_i = \frac{\exp(z_i)}{\sum_j \exp(z_j)} \quad (2.5)$$

В OSNet softmax використовується лише на етапі тренування при класифікації особи по її embedding-вектору, а сама модель всередині переважно оперує ReLU і сигмоїдами.

Центральною операцією в CNN є згортка (convolution), 2D-згортка між багатоканальним вхідним зображенням  $x$  і ядром (фільтром)  $w$  обчислює вихідне значення для кожної координати  $(i, j)$  і кожного вихідного каналу  $k$  за формулою:

$$y_k(i, j) = \sum_{c=1}^{C_{\text{in}}} \sum_{u=-r}^r \sum_{v=-r}^r w_{k,c}(u, v) x_c(i + u, j + v), \quad (2.6)$$

де:

- $C_{\text{in}}$  – число вхідних каналів,

- $w_{k,c}(u, v)$  ядро, що має розмір  $(2r + 1) \times (2r + 1)$ .

Ця операція сумує локальні околосо радіусом  $r$  по всіх каналах з ваговими коефіцієнтами ядра, утворюючи вихідний тензор  $y$  з  $C_{\text{out}}$  каналів (для кожного вихідного каналу  $k$  своє ядро). У звичайній згортці кожен вихідний канал є комбінацією всіх вхідних каналів.

Таким чином, кількість параметрів стандартного згорткового шару розмірності  $C_{\text{in}} \rightarrow C_{\text{out}}$  з ядром  $K \times K$  дорівнює:  $C_{\text{out}} \times C_{\text{in}} \times K \times K$ .

Для зменшення кількості параметрів в OSNet застосовуються факторизовані згортки:

- Pointwise це випадок згортки з ядром  $1 \times 1$  (тобто  $r = 0$ ). Вона не охоплює сусідні пікселі, але все одно змішує канали. Тоді формула 2.6 для  $1 \times 1$  згортки спрощується, оскільки  $u = 0, v = 0$ :

$$y_k(i, j) = \sum_{c=1}^{C_{\text{in}}} w_{k,c} x_c(i, j) \quad (2.7)$$

Тоді, у кожній точці  $(i, j)$  вихідний канал  $k$  є просто зваженою сумою всіх вхідних каналів у цій точці. Pointwise згортка виконує роль лінійного перетворення каналів незалежно для кожної просторої координати. Це еквівалентно застосуванню до вектора піксельних значень матриці розміру  $C_{\text{out}} \times C_{\text{in}}$ . У OSNet pointwise шари використовуються зміни розмірності ознак або для їх підсумкового об'єднання після паралельних гілок.

- Depthwise згортка на відміну від звичайної згортки, не змішує канали між собою. Кожен вхідний канал обробляється своїм ядром і дає відповідний вихідний канал. Таким чином,  $C_{\text{in}} = C_{\text{out}}$  (часто після depthwise-конволюції зберігають ту саму кількість каналів) і немає сумування по  $c$ , вона розпадається на  $C_{\text{in}}$  окремих згорток:

$$y_c(i, j) = \sum_{u=-r}^r \sum_{v=-r}^r w_c^{(dw)}(u, v) x_c(i + u, j + v) \quad (2.8)$$

де для кожного каналу  $c$  своє ядро  $w^{(dw)}_c$ . Загальне число параметрів глибинної згортки  $K \times K = C_{in} \times K \times K$ , що в  $C_{out}$  разів менше, ніж у звичайної (для типового випадку, коли  $C_{out}$  дорівнює  $C_{in}$  або є їх кратним). Дана згортка добре виявляє просторові структури в межах кожного каналу, але не вміє комбінувати інформацію між різними каналами.

Depthwise separable convolution це комбінація попередньої (depthwise) операції з наступною  $1 \times 1$  згорткою, яка і здійснює змішування каналів. Разом вони еквівалентні певній звичайній згортці, проте мають значно менше параметрів. В OSNet кожен Lite  $3 \times 3$  шар – це  $1 \times 1$  згортка (для зміни числа каналів) перед  $3 \times 3$  depthwise-згорткою. Така послідовність означає, що спершу лінійно проектуємо вихідні канали в інший простір (можливо, більшої розмірності), а тоді в кожному з них застосовуємо фільтр  $3 \times 3$ . Separable-конволюція суттєво знижує обчислювальну складність і ризик перенавчання, що було важливо з огляду на порівняно невеликі обсяги даних ReID.

Для навчання базової моделі для повторної ідентифікації було обрано модель «osnet\_a1n\_x1\_0». Дана модель була навчена на кількох стандартних наборах даних для person re-identification. Ці датасети складаються з зображень людей, зібраних з систем відеоспостереження або камер, з розміченими ідентичностями. Кожен датасет розділений на тренувальну і тестову частини:

1. Market-1501 [102] – один з найпопулярніших великих датасетів ReID. Містить 1501 ідентичність, зняту 6 камерами на кампусі університету. Розбиття: 751 осіб для тренування, 750 – для тесту (3368 запитів та 15913 галерейних зображень). Кожна особа представлена кількома зображеннями з різних камер, а завдання – знайти всіх її зображення у галереї по даному запиту.

2. DukeMTMC-reID [103] – це датасет, знятий на території Duke University. Має 1404 ідентичності (702 для тренування, 702 для тесту), близько

36 тис. зображень. Особливістю є складніші умови зйомки (різні фони, освітлення).

3. VIPeR, GRID [104] – менші датасети ранніх років, що іноді використовуються для оцінки або перенавчання. VIPeR має лише 632 зображення (дві камери, 316 осіб), GRID – 250 осіб (з станції метро Лондона). Вони допомагають перевірити здатність моделі працювати при дуже обмежених даних, хоча самі по собі для глибокого навчання замалі, тому часто використовуються для transfer learning або в протоколах cross-domain.



Рис. 2.4 Приклади зображень стандартних наборів даних для person re-identification. Ілюструє варіації фону, освітлення, точок огляду в різних камерах та розрив між доменами.

Зображення осіб у цих наборах, як правило, надані у вигляді вже вирізаних прямокутників (bounding box), отриманих або ручною розміткою, або детекторами людей. Розміри цих bounding box можуть різнитися, тому першим кроком є нормалізація розміру і масштабу. У експериментах з OSNet всі

зображення масштабуються до розміру  $256 \times 128$  пікселів. Це стандартне розрішення для ReID, яке зберігає приблизно пропорції людини (2:1). Масштабування забезпечує однаковий вхідний розмір для мережі, щоб вона могла працювати пакетно і не була чутлива до фізичного зросту чи дистанції до камери.

Далі, зображення перетворюються в тензори і нормалізуються по каналам. Зазвичай застосовують нормалізацію кольорових каналів RGB до нульового середнього і одиничної дисперсії. Використовуються статистики ImageNet (середні значення  $[0.485, 0.456, 0.406]$  і стандартні відхилення  $[0.229, 0.224, 0.225]$  для R, G, B), оскільки моделі можуть ініціалізуватися ImageNet-претренованими вагами.

Якщо ж модель тренується з нуля, можуть обчислити середнє й відхилення по тренувальному набору і використовувати їх. Нормалізація покращує збіжність навчання – піксельні значення, приведені до співмірного масштабу, дозволяють вищим шарам стабільніше навчатися (жоден канал не домінує, освітлення вирівнюється до певної міри).

Крім того, для підвищення ефективності навчання та запобігання перенавчанню було впроваджено механізм ранньої зупинки (early stopping) з критерієм зупинки після 5 епох за умови не покращення валідаційної метрики. Це дозволяє зупинити навчання в оптимальний момент, зберігаючи модель з найкращими показниками на валідаційному наборі даних. Такий підхід допомагає уникнути перенавчання та покращує загальну узагальнюючу здатність моделі.

Під час експериментів використовувалися методи аугментації даних за допомогою бібліотеки Albumentations [56]. Ця бібліотека забезпечує широкий спектр трансформацій зображень, що допомагає покращити узагальнювальну здатність моделі. До стандартних методів аугментації належали:

1. Нормалізація зображення: для забезпечення узгоджених і стандартизованих вхідних даних, що допомагає моделі краще розпізнавати патерни.

2. Обертання зображення: для підвищення стійкості моделі до різних орієнтацій зображень, що може бути особливо корисним у випадках, коли знімки мають різні кути нахилу.

3. М'яке розмиття: для зменшення впливу шуму та артефактів на зображення, що сприяє кращому виділенню суттєвих деталей на знімках.

Для додаткового підвищення якості моделі було використано інші методи аугментації, такі як зміна яскравості та контрастності, додавання випадкового шуму, горизонтальні та вертикальні відображення. Ці методи допомагали моделі бути стійкою до різних варіацій зображень, що забезпечувало кращу здатність до узагальнення на тестових даних.

У рамках цього дослідження було створено модифіковану версію мережі OSNet, призначену для задачі person re-identification. Обрано стандартну backbone OSNet—компактну архітектуру з omni-scale резидуальними блоками та depthwise-separable  $3 \times 3$  згортками—видалили його кінцевий класифікаційний шар і додали просторовий механізм уваги. Після останнього блоку conv5 отримана карта ознак розміром  $B \times C \times H \times W$  пропускається через AttentionLayer, який навчається підсилювати регіони з найінформативнішими ознаками (наприклад, силует, аксесуари чи деталі одягу), відкидаючи шум фону.

Далі з просторово промодульованих карт ознак видобувається компактний embedding: Adaptive Average Pooling збирає кожний канал у його середнє значення, після чого Flatten та лінійний шар перетворюють вектор довжини  $C$  у вектор довжини  $emb\_dim$  з активацією ReLU. За потреби застосовується BNNeck—batch-нормалізація над embedding, яка вирівнює розподіл координат перед передачею в класифікатор. Безпосередньо класифікаційний шар представляє собою один лінійний шар  $emb\_dim \rightarrow num\_classes$ , що видає логіти для ArcFaceLoss.

Перед тренуванням усі зображення стандартизуються: їх масштабуємо до фіксованого розміру ( $256 \times 128$ ), випадково віддзеркалюємо по горизонталі і при потребі застосовуємо RandAugment, Color Jitter, переведення в градації сірого, а потім конвертуємо в тензор і нормалізуємо до інтервалу  $[-1, 1]$ . Наприкінці

pipeline для стійкості до оклюзій використовується Random Erasing. Для тестування залишають лише Resize та Normalize, щоб оцінити справжню узагальнюваність моделі без шуму аугментацій.

Мережу навчаємо комбінованою втратою: ArcFaceLoss гарантує чітку роздільну здатність між ID під час класифікації, batch-hard triplet loss на embedding стимулює максимальне згрупування одного класу й розшарування різних, а CenterLoss додатково зменшує внутрішньокласову дисперсію.

Оптимізація проводиться за допомогою AdamW із регуляризацією ваг, а навчальна швидкість корегується через OneCycleLR або косинусний спад з опціональним етапом warm-up. У разі потреби використовується автоматична змішана точність (AMP) з GradScaler, що дозволяє економити відеопам'ять і прискорювати обчислення.

Під час тренування батчі формується за допомогою PKSampler, який гарантує, що в кожному батчі присутні  $p$  різних ID по  $k$  зображень кожного. Для кожного батчу модель виконує forward-прохід: отримуємо логіти для ArcFace, embedding і карту уваги, після чого обчислюємо втрати й оновлюємо параметри. Scheduler оновлюється або на кожному кроці, або після кожної епохи.

Валідація проводиться через заданий інтервал епох: генерується embedding для наборів query і gallery, обчислюємо метрики mAP, Rank-1/5/10 та mINP, а також візуалізуємо топ-5 результатів пошукових запитів у TensorBoard. Якщо поточна mAP покращує попередній рекорд, ми зберігаємо чекпойнт моделі й повертаємо найкращу епоху й значення mAP.

Після завершення кожної епохи обчислювали метрики mean Average Precision (mAP) та Cumulative Matching Characteristic Rank-1 [105]. Cumulative Matching Characteristics (CMC) є широко використовуваним підходом для оцінювання результатів в задачах повторної ідентифікації людини. Це криві, які показують залежність ймовірності правильної ідентифікації від кількості найбільш подібних зразків у галереї, що враховуються при аналізі запиту. В процесі обчислення CMC модель формує ранжування зразків за ступенем їхньої

схожості до зображення запиту. Відповідно, показник точності СМС top-k визначається таким чином [12]:

$$Acc_k = \begin{cases} 1, & \text{якщо зразки галереї з найвищим рейтингом містять } id \text{ запиту} \\ 0, & \text{інше} \end{cases}$$

якщо хоча б один зі зразків галереї, що має найвищий рейтинг подібності до запиту, відповідає шуканій особі, то точність дорівнює одиниці.

Mean Average Precision, mAP широко застосовується для оцінки продуктивності моделей у завданнях виявлення та пошуку об'єктів, особливо у випадках, коли виникає потреба у точній ідентифікації осіб. Основу для її розрахунку складають такі показники, як точність (Precision) та відкликання (Recall). Precision розраховується як відношення числа істинно позитивних результатів (TP) до суми істинно позитивних і хибнопозитивних результатів (TP + FP). Recall визначається як співвідношення істинно позитивних результатів (TP) до суми істинно позитивних та помилково негативних (TP + FN).

Середня точність (Average Precision, AP) для конкретного класу обчислюється як середнє значення показника Precision, що вимірюється на різних рівнях відкликання, яке виражається таким чином:

$$AP = \frac{1}{R} \sum_{r \in R} Precision(r). \quad (2.9)$$

Mean Average Precision (mAP), своєю чергою, визначає середнє значення середньої точності по всіх класах і є кінцевим агрегованим показником продуктивності моделі. Ця метрика активно використовується при порівнянні та оцінці продуктивності сучасних моделей у задачах повторної ідентифікації особи та виявлення об'єктів завдяки своїй здатності враховувати якість розподілу ознак по всіх представлених категоріях.

$$mAP = \frac{1}{M} \sum_{m \in M} AP(m). \quad (2.10)$$

Завдяки такому поєднанню компактного backbone OSNet, просторової уваги та ретельно підібраних втрат і стратегії оптимізації модель демонструє високу точність та стабільність у задачі person re-identification.

Для навчання моделей на основі моделі OSNet були обрані такі параметри: кількість епох навчання – 60, optim – adam [13], loss – softmax [14]. На рисунку 2.5 зображено, як змінювалось значення якості (acc) в процесі навчання.



Рис. 2.5 Зміна показників якості моделі в процесі навчання

Результати досліджень та отриманих моделей на валідаційному наборі зображень Market-1501 наведено в таблиці 2.2. В таблиці 2.3 представлено результати отримані на валідаційному наборі зображень DukeMTMC-reID.

Таблиця 2.2 – Приклади результатів навчання моделей на наборі зображень Market-1501.

model	optim	lr	epoch	mAP	Rank-1	Rank-5	Rank-10
OSNet	adam	0.001	60	60.2%	82.6%	92.7%	95.0%
OSNet	adam	0.0015	60	67.9%	86.7%	95.0%	96.9%
OSNet	adam	0.002	60	68.1%	86.6%	94.5%	96.3%
OSNet	adam	0.003	60	70.5%	87.7%	95.0%	96.9%
OSNet	adam	0.0045	60	73.0%	89.4%	95.8%	97.3%
OSNet	adam	0.0055	60	73.5%	90.0%	96.1%	97.4%
OSNet	adam	0.007	60	72.2%	89.0%	96.0%	97.5%
OSNet	adam	0.0005	60	72.8%	89.0%	95.7%	97.2%
OSNet	adam	0.0085	60	71.7%	89.0%	95.3%	96.8%

Таблиця 2.3 - Результати навчання моделей на наборі зображень DukeMTMC-reID.

model	optim	lr	epoch	mAP	Rank-1	Rank-5	Rank-10
OSNet	adam	0.001	60	51.7%	74.5%	85.6%	88.9%
OSNet	adam	0.0015	60	56.0%	77.7%	86.8%	89.9%
OSNet	adam	0.002	60	57.9%	78.0%	87.0%	90.4%
OSNet	adam	0.005	60	62.1%	81.7%	89.5%	91.9%
OSNet	adam	0.0055	60	33.6%	55.0%	72.6%	77.9%

В результаті досліджень та навчання нейронних мереж було досягнуто таких показників якості моделі на валідаційному наборі зображень Market-1501: mAP = 73,5% та Rank-1 = 90%. Отримані результати порівнюємо з існуючою моделлю [15], яка була створена при використанні нейронної мережі OGNNet. Показники якості моделі на валідаційному наборі зображень Market-1501: mAP

= 69,52% та Rank-1 = 87,74%, що значно менші за показники навчаної власної моделі.

Підсумовуючи, у цьому розділі було представлено всебічний аналіз, головною метою якого було, розроблення базової моделі для person re-identification на основі OSNet із додаванням механізму просторової уваги. Дослідження включало кілька ключових етапів:

1. Модифікація базового backbone'у OSNet: Було взято переднатреновану OSNet (версію `osnet_x1_0`), видалено її стандартний класифікаційний шар і збережено лише послідовність блоків `conv1-conv5` з `omni-scale` резидуальними гілками та `depthwise-separable` згортками.

2. Інтеграція просторового механізму уваги: До виходу останнього блоку було додано `AttentionLayer`, який аналізує карту ознак і автоматично підсилює ті регіони, де містяться найбільш характерні для ідентифікації людини деталі (силует, аксесуари, текстури одягу), відкидаючи фоновий шум.

3. Формування компактного `embedding`: Просторово відфільтровані карти ознак стискалися через `Adaptive Average Pooling`, після чого `Flatten` та лінійний шар з `ReLU` перетворювали вектор довжини  $C$  у вектор довжини `emb_dim`. Та застосування `BNNeck` для вирівнювання розподілу координат перед класифікатором.

4. Підготовка та аугментація вхідних даних: Перед тренуванням кожне зображення масштабувалося до фіксованого розміру ( $256 \times 128$ ), випадково віддзеркалювалося, проходило через `RandAugment`, `Color Jitter` і перехід у сірий відтінок, після чого нормалізувалося й частково «стирався» фрагмент (`Random Erasing`). Для тесту залишали лише масштабування та нормалізацію.

5. Визначення функцій втрат та метрик оцінки: Модель навчалася за комбінованою стратегією: класифікаційна складова `ArcFaceLoss` давала змогу чітко відокремити різні ID, триплет-лосс згуртовував ознаки однієї особи та розводив вектори різних, а `CenterLoss` додатково зменшував внутрішньокласову дисперсію. Оцінка проводилася за метриками `mAP`, `Rank-1/5/10`.

6. Оптимізація та навчальний цикл: Використано оптимізатор AdamW із weight decay, а learning rate регулювався схемою OneCycleLR або косинусним спадом із розігрівом. Для побудови батчів застосовувався PKSampler, що гарантував у кожному пакеті фіксовану кількість зображень кожного класу. Після кожної епохи або через заданий інтервал модель оцінювалася на наборах «query» і «gallery», а найкращі чекпойнти зберігалися автоматично.

Таким чином, поєднання компактного backbone OSNet, механізму просторової уваги, ретельно налаштованого pipeline аугментацій і комбінованої стратегії втрат забезпечило високу точність і стабільність вбудовувань, що підтверджується стабільними результатами на стандартизованих наборах для person re-identification.

## **2.2 Удосконалення базової моделі повторної ідентифікації за допомогою застосування умовної гіпермоделі**

Класичні моделі повторної ідентифікації людини, зокрема Omni-Scale Network (OSNet), добре показують себе на контрольованих наборах даних, проте суттєво втрачають точність у мінливих реальних умовах. Зміна освітлення, ракурсів камер, фону або пози тіла призводить до різких коливань представлення ознак і збільшує похибки при пошуку «пари» зображень однієї особи. Ця нестабільність обумовлена фіксованістю ваг мережі, які оптимізовані під статистику тренувального датасету і не пристосовуються до нових контекстних характеристик.

Відсутність механізму динамічної адаптації перешкоджає широкому, надійному застосуванню Re-ID у системах відеоспостереження, розумного ритейлу та smart-міст. Саме тому виникає потреба в архітектурі, що дозволяє коригувати внутрішні параметри мережі у відповідь на поточні умови оточення, не вимагаючи повного перенавчання.



Рис. 2.6 Типові приклади змін зовнішнього вигляду, спричинених різницею кута огляду, освітлення, фонового шуму та перекриття. У кожному стовпці показано два зображення однієї й тієї ж особи з двох різних ракурсів камери.

Для вирішення проблеми обмеженої здатності базової моделі OSNet до адаптації в умовах варіабельності освітлення, ракурсів і фону було запропоновано метод із застосуванням умовної гіпермоделі.

Гіпермоделі (HyperNetworks) [106] – це нейронні мережі, які генерують параметри (ваги) іншої моделі. Існує два основні підходи до побудови гіпермоделей:

1. Статичні гіпермоделі формують єдиний набір ваг без урахування контексту, що робить їх непридатними для сценаріїв із високою варіабельністю зовнішніх умов.

2. Умовні гіпермоделі використовують додатковий вектор умовних ознак середовища, що дозволяє формувати адаптивні ваги залежно від поточного контексту.

Таблиця 2.4 – Порівняння гіпермоделей

Показник	Статична гіпермодель	Умовна гіпермодель
Адаптивність до контексту	Ні	Так
Використання зовнішніх ознак	Не використовує	Використовує умовний вектор
Обчислювальні витрати	Середні	Вищі (через додатковий MLP)
Складність інтеграції	Низька	Помірна (потребує додаткового блоку)
Гнучкість корекції ваг	Фіксовані ваги, без корекції	Динамічна корекція під час inference
Необхідність перенавчання	Так, повне перенавчання при зміні умов	Мінімальна (адаптація «на льоту»)
Приклади застосування	Загальні генератори ваг	Адаптивні Re-ID системи

Таким чином, умовна гіпермодель забезпечує необхідну динамічну адаптивність OSNet до змінних умов експлуатації, зберігаючи при цьому архітектурну простоту та ефективність обчислень [107]:

По-перше, умовна гіпермодель дозволяє динамічно генерувати параметри цільової мережі з урахуванням конкретних властивостей поточного середовища. Замість того, щоб навчати OSNet єдиним набором ваг, які виявляються оптимальними лише для середовища тренування, ми вводимо додатковий вектор умовних ознак, що кодує інформацію про поточні умови (освітленість, ракурс, позу тощо).

По-друге, архітектура умовної гіпермоделі, побудована як багат шаровий перцептрон (MLP), володіє достатньою виразною здатністю для апроксимації складних залежностей між контекстними ознаками та оптимальними вагами мережі. Це дозволяє не лише коригувати окремі шари OSNet, але й узгоджено

налаштовувати весь набір параметрів, зберігаючи цілісність архітектури. На відміну від Domain Adaptation чи Meta-Learning, які вимагають численних етапів перенавчання та великих обсягів нових даних, умовна гіпермодель інтегрується у стандартний процес тренування як єдиний енд-ту-енд блок, що значно спрощує впровадження та знижує обчислювальні витрати [108].

По-третє, інтеграція умовної гіпермоделі з OSNet здійснюється через просту адитивну модифікацію ваг, що вимагає лише одного додаткового forward-проходу через MLP для генерації. Після чого адаптовані ваги можуть бути негайно використані у базовій мережі, забезпечуючи баланс між гнучкістю адаптації та ефективністю inference.

Нехай основна (цільова) мережа, яку будемо позначати як  $F$ , приймає зображення  $x \in X$  та генерує його представлення  $z \in R^d$ . Ця мережа параметризована набором ваг  $W$  (які можуть бути поділені на параметри окремих шарів, наприклад,  $W = \{W_1, W_2, \dots, W_L\}$  для  $L$  шарів). Для того, щоб адаптувати модель до змінних умов (наприклад, відмінностей між камерами, умов освітлення, ракурсів тощо), вводиться умовний вхід  $c \in C$ , який кодує інформацію про ці умови (наприклад, ідентифікатор камери, характеристики середовища, метадані) [109].

Умовна гіпермодель позначається як  $G$  і має параметри  $\theta$ . Її основна задача полягає у генерації або модулюванні параметрів цільової мережі залежно від умовного сигналу  $c$ . Формально, можна записати, що для кожного шару  $l$  цільової мережі маємо:

$$W_l(c) = \phi_l(c; \theta_l), \quad (2.11)$$

де  $\phi_l(\cdot)$  — деяка функція, що перетворює умовний вхід  $c$  у модифіковані параметри або корекційні величини для  $l$ -го шару. Загальна система може працювати за різними схемами модифікації параметрів. Розглянемо два основні підходи:

*Адаптація шляхом доповнення до базових ваг:* Позначимо базові (статичні) параметри цільової мережі як  $W_l^0$ . Тоді умовна гіпермодель генерує корекційний тензор  $\Delta W_l(c)$ , після чого ефективні параметри визначаються як:

$$W_l(c) = W_l^0 + \Delta W_l(c), \quad (2.12)$$

$$\Delta W_l(c) = \phi_l(c; \theta_l). \quad (2.13)$$

*Адаптація шляхом модулювання:* У цьому випадку умовна гіпермодель генерує вектор-масштабуючий коефіцієнтів  $\alpha_l(c)$ , після чого ваги кожного шару оновлюються як:

$$W_l(c) = \text{diag}(\alpha_l(c))W_l^0. \quad (2.14)$$

де  $\text{diag}(\alpha_l(c))$  є діагональною матрицею, що містить коефіцієнти модулювання, отримані як:

$$\alpha_l(c) = \psi_l(c; \theta_l). \quad (2.15)$$

У загальному випадку, незалежно від вибору схеми адаптації, остаточна модель  $F$  набуває вигляду:

$$z = F(x; \{W_l(c)\}_{l=1}^L). \quad (2.16)$$

де параметри кожного шару  $W_l(c)$  є функцією від умовного вхідного сигналу  $c$  [110].

Для задач повторної ідентифікації людини умовний вхід  $c$  може включати такі компоненти:

- ідентифікатор камери: дозволяє враховувати специфічні особливості зйомки з різних камер, які можуть мати різні кутові та оптичні характеристики.
- метадані про умови зйомки: інформація про освітлення, пору доби або навіть погодні умови.
- контекстуальні ознаки: додаткові дані про оточення, які можуть впливати на видимість чи контрастність зображення [111].

Завдяки такій структурі, умовна гіпермодель генерує корекційні параметри, що адаптують кожен шар цільової мережі до поточного контексту. Це дозволяє моделі більш точно вилучати релевантні ознаки та підвищувати загальну точність розпізнавання, що є особливо критичним при роботі з великими наборами даних, отриманих із різних джерел у реальному часі.

Оскільки цільова задача полягає у повторній ідентифікації людини, система зазвичай оптимізується за допомогою відповідної функції втрат, наприклад, комбінації крос-ентропійної втрати  $L_{CE}$  (для класифікації ідентичностей) та втрати на основі триплетів  $L_{triplet}$  (для навчання відстаневих відношень):

$$L = \lambda_{CE} L_{CE}(F(x; W(c)), y) + \lambda_{triplet} L_{triplet}(F(x; W(c))). \quad (2.17)$$

де  $y$  — мітка ідентичності, а  $\lambda_{CE}$  та  $\lambda_{triplet}$  є ваговими коефіцієнтами. Під час навчання відбувається спільна оптимізація як параметрів базової мережі  $W^0$ , так і параметрів умовної гіпермережі  $\theta$ . Це може бути організовано у вигляді спільної мінімізації:

$$\min_{W^0, \theta} E_{x,c,y} [L(F(x; W_l(c)), y)]. \quad (2.18)$$

Цей підхід дозволяє моделі динамічно адаптувати свої параметри під конкретні умови, забезпечуючи оптимальний баланс між збереженням попередньо набутого знання (зниження катастрофічного забування) та

адаптацією до нових даних, що є ключовим для успішного застосування у системах повторної ідентифікації людини [112].

У запропонованому методі умовна гіпермодель відіграє роль надбудови над базовою архітектурою OSNet, забезпечуючи динамічне коригування лише ключових параметрів мережі відповідно до умов зйомки.

Нехай  $x \in R^{H \times W \times 3}$  — вхідне зображення, а  $c \in R^{d_c}$  — вектор контексту, що може містити інформацію про тип камери, експозицію, часовий штамп тощо. Спочатку  $c$  перетворюється на компактне представлення  $z \in R^{d_z}$  за допомогою багатошарового перцептрона  $E$  із параметрами  $\Phi$ :

$$z = E(c; \Phi) = \sigma \left( W_E^{(2)} \sigma \left( W_E^{(1)} c + b_E^{(1)} \right) + b_E^{(2)} \right), \quad (2.20)$$

де  $W_E^{(1)} \in R^{h \times d_c}$ ,  $W_E^{(2)} \in R^{d_z \times h}$ ,  $b_E^{(1)} \in R^h$ ,  $b_E^{(2)} \in R^{d_z}$ , а  $\sigma$  — нелінійність (ReLU) [112].

Отриманий  $z$  передається до умовної гіпермоделі  $H$  з параметрами  $\Theta$ , яка генерує поправку  $\Delta W$  лише для вагових тензорів тих Omni-Scale блоків OSNet, що найбільш чутливі до змін контексту. Нехай індекси цих шарів позначені множиною  $S$ , а розмірність зведеного вектору всіх їх ваг становить  $\sum_{i \in S} |W_{0,i}|$ . Тоді

$$\begin{aligned} \Delta W &= H(z; \Theta) \\ &= W_H^{(L)} \phi \left( W_H^{(L-1)} \phi \left( \dots \phi \left( W_H^{(1)} z + b_H^{(1)} \right) \dots \right) + b_H^{(L-1)} \right) + b_H^{(L)}, \end{aligned} \quad (2.21)$$

де кожний  $W_H^{(\ell)}$  і  $b_H^{(\ell)}$  визначають шари гіпермоделі з  $\ell = 1, \dots, L$ , а  $\phi$  — активація (ReLU) із Dropout і BatchNorm для запобігання перенавчанню [113].

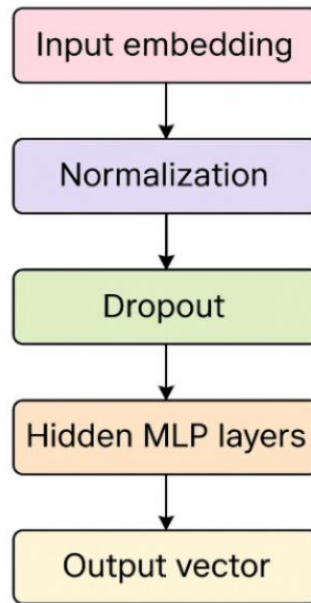


Рис. 2.7 Структура запропонованої умовної гіпермоделі

Після обчислення  $\Delta W \in R^{\sum_{i \in S} |W_{0,i}|}$  оновлюються лише ці вибрані вагові тензори:

$$W_i(c) = W_{0,i} + \Delta W_i, \quad \forall i \in S, \quad (2.22)$$

де  $\Delta W_i$  — підвектор  $\Delta W$ , що відповідає шару  $i$ . Інші параметри OSNet залишаються фіксованими. Нові повні ваги мережі позначимо  $W(c)$  [114].

Далі із цими адаптивними вагами OSNet видає багатомасштабний вектор ознак [114]:

$$f(x, c) = f_{OS}(x; W(c)), \quad (2.23)$$

де кожний Omni-Scale блок реалізує паралельні згорткові операції розміром  $1 \times 1$ ,  $3 \times 3$  та  $5 \times 5$ , а результати конкатенуються і нормалізуються.

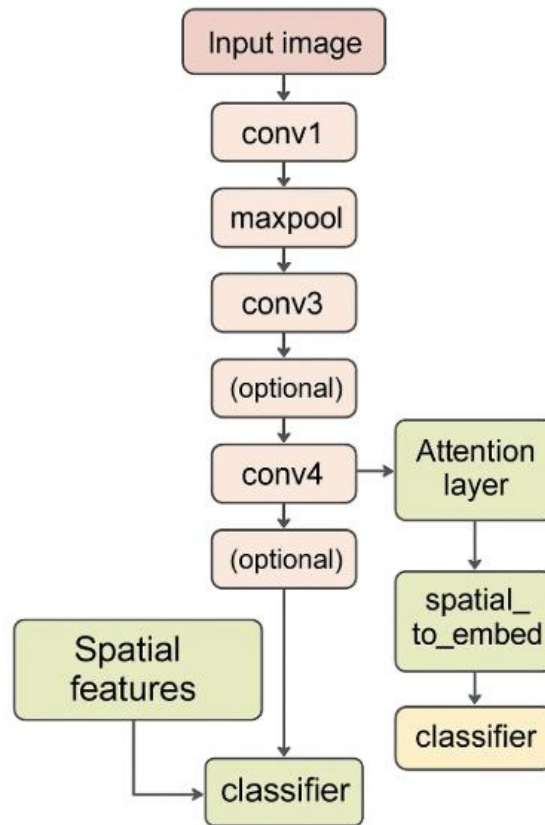


Рис. 2.8 Схема базвої мережі OSNet

Навчальна ціль визначається комбінованою функцією втрат:

$$\begin{aligned}
 \mathcal{L}(\Phi, \Theta, W_0) = & - \sum_{k=1}^K 1\{y = k\} \log \frac{\exp(w_k^\top f(x, c))}{\sum_{j=1}^K \exp(w_j^\top f(x, c))} \\
 & + \lambda [ \|f(a, c) - f(p, c)\|_2^2 - \|f(a, c) - f(n, c)\|_2^2 \\
 & + m ]_+,
 \end{aligned} \tag{2.24}$$

де  $w_k$  — ваги останнього лінійного шару для класифікації,  $m$  — маржа,  $\lambda$  — коефіцієнт балансу. Градієнти функції  $\mathcal{L}$  одночасно оновлюють параметри  $\{W_0, \Phi, \Theta\}$  за допомогою алгоритму Adam [115]:

$$\begin{aligned}
 W_0 &\leftarrow W_0 - \eta \frac{\partial \mathcal{L}}{\partial W_0}, \\
 \Phi &\leftarrow \Phi - \eta \frac{\partial \mathcal{L}}{\partial \Phi}, \\
 \Theta &\leftarrow \Theta - \eta \frac{\partial \mathcal{L}}{\partial \Theta}.
 \end{aligned}
 \tag{2.25}$$

що забезпечує узгоджену оптимізацію адаптивного рівня (гіпермоделі) і базової архітектури.

Такий поетапний розподіл відповідальності гарантує, що умовна гіпермодель вчиться генерувати мінімальні, але максимально релевантні коректування ваг OSNet, дозволяючи мережі зберігати стабільність і водночас реагувати на зміни ракурсів, освітлення чи інших зовнішніх умов.

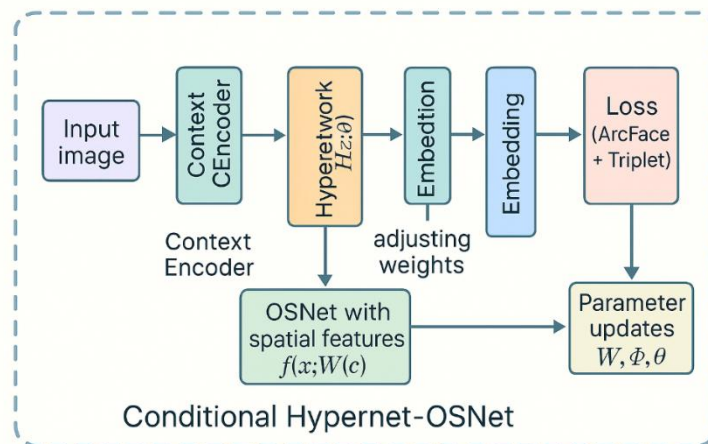


Рис. 2.9 Загальна схема запропонованого методу поєднання OSNet та гіпермоделі

У експерименті використовуються дві взаємодоповнюючі компоненти:

– Гіпермодель, що приймає на вхід вектор ознак розмірності `feature_dim` і генерує адаптивне представлення в ембеддинговому просторі розмірності `embedding_dim`. Архітектура побудована за принципом послідовності повнозв'язних шарів із використанням нормалізації, нелінійного перетворення за допомогою функції ReLU та регуляризації за допомогою

Dropout. Важливою особливістю є попереднє нормування вхідних ознак (L2-нормалізація), що забезпечує стабільність подальшої генерації ембеддингів. Таким чином, модуль виконує роль адаптивного компонента, який може бути використаний для коригування представлення ознак, отриманих від основної мережі.

– Основна мережа побудована на базі попередньо навченого backbone, з якого вилучено останній класифікаційний шар. Для посилення дискретних ознак використовується окремий модуль уваги над просторовими ознаками, отриманими з глибинних шарів мережі. Просторові ознаки передаються через блок адаптивного усереднення, лінійне перетворення і функцію активації ReLU для формування кінцевих ембеддингів. За необхідності застосовується BNNeck, що дозволяє покращити розподіл ознак для задач класифікації. Остаточний класифікатор реалізовано як лінійний шар, який відображає ембеддинг у простір кількості класів. У методі forward організовано послідовне вилучення просторових ознак з backbone з подальшим застосуванням модуля уваги, що дозволяє отримати більш релевантне представлення для задачі повторної ідентифікації.

Під час навчання здійснюється спільна оптимізація параметрів базової мережі та умовної гіпермоделі згідно з функцією втрат, що враховує вимоги до високої точності ідентифікації та стійкості системи до змін умов зйомки.

На рисунку 2.10 наведена блок-схема роботи алгоритму умовної гіпермоделі для задачі повторної ідентифікації людини.

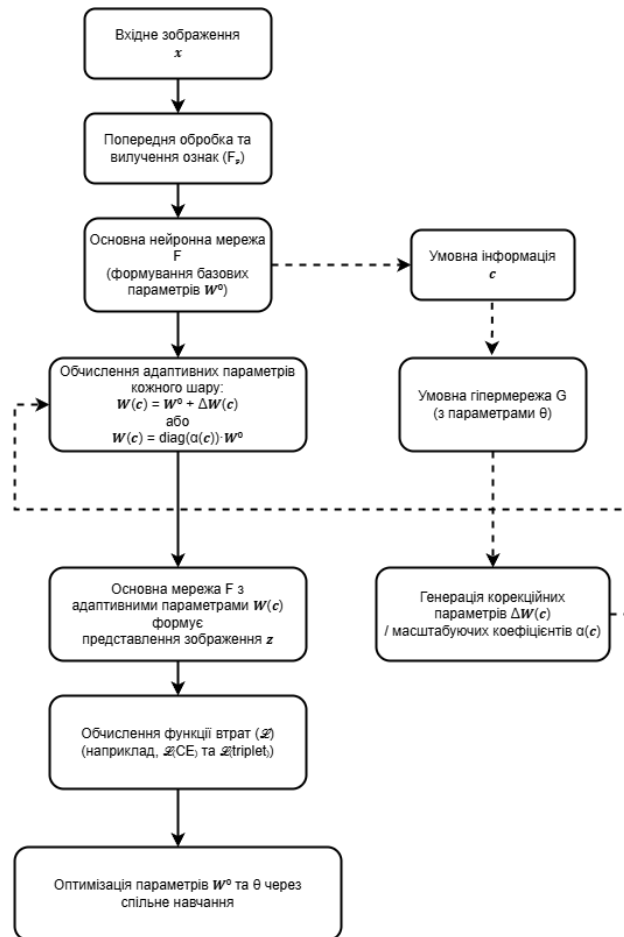


Рис. 2.10 Блок-схема алгоритму умовної гіпермоделі для задачі повторної ідентифікації людей

Модель складається з базового OSNet із початковими вагами  $W_0$ , екстрактор контексту  $E$ , реалізований двошаровим MLP з 256 і 128 нейронами та активацією ReLU, і гіпермоделі  $H$  із трьох MLP-шарів розмірності  $512 \rightarrow 512 \rightarrow |W_0|$  із Dropout-регуляризацією.

Навчання відбувалося на двох GPU з використанням оптимізатора Adam [18], початковий коефіцієнт навчання становив  $3 \times 10^{-4}$  і зменшувався в 10 разів кожні 30 епох. Коефіцієнт балансу втрат  $\lambda$  було встановлено рівним 1.0, а загальна тривалість тренування — 200 епох.

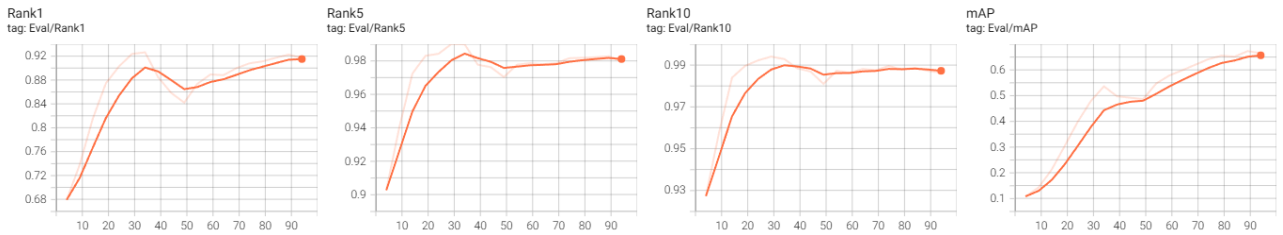


Рис.2.11 Зміна показників якості моделі в процесі навчання

Експериментальна перевірка проводилась із застосуванням стандартних наборів даних для задач повторної ідентифікації, що дозволило оцінити продуктивність моделі за показниками mAP, Rank-1 та іншими метриками. На рисунку 2 представлено динаміку зміни якості моделей під час навчання, що демонструє ефективність запропонованого підходу у порівнянні з традиційними методами.

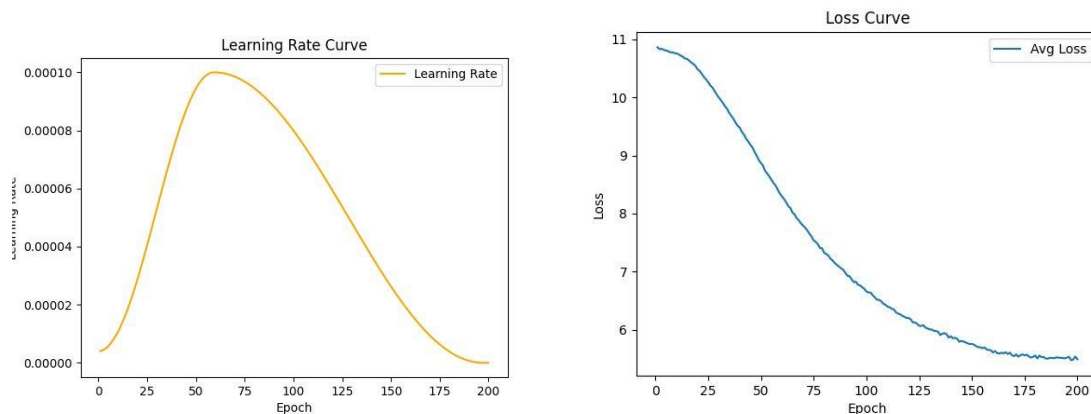


Рис. 2.12 Вплив динамічних змін параметрів на результати роботи моделей.

У процесі навчання застосовується комбінована функція втрат, що містить крос-ентропійний компонент і триплетну втрату (на рисунку 2 зображені криві зміни показників якості моделі в процесі навчання). Такий підхід сприяє максимізації відстаней між представленнями різних осіб та одночасно мінімізує відстані між представленнями однакових осіб. Результати проведених експериментів на датасетах Market-1501 та DukeMTMC-reID наведено в таблиці 1 та 2 відповідно.

Таблиця 2.5 Результати навчання моделей на наборі зображень Market-1501

ID	optim	lr	Base lr	epoch	mAP	Rank-1	Rank-5	Rank-10
ExpBN37	adam	1e-4	1e-5	200	77.43%	96.94%	99.11%	99.44%
ExpBN38	adam	1e-4	1e-5	200	77.00%	96.56%	99.17%	99.44%
ExpBN39	adam	1e-4	1e-5	200	76.17%	96.91%	99.20%	99.41%
ExpBN40	adam	1e-4	1e-5	200	77.03%	96.70%	99.38%	99.64%
ExpBN41	adam	1e-4	1e-5	200	76.75%	96.53%	99.23%	99.52%
ExpBN42	adam	1e-4	1e-5	200	78.25%	96.32%	99.23%	99.52%

Таблиця 2.6 – Результати навчання моделей на наборі зображень DukeMTMC-reID

ID	optim	lr	Base lr	epoch	mAP	Rank-1	Rank-5	Rank-10
ExpBN49	adam	1e-4	1e-5	200	67.40%	84.50%	92.00%	94.80%
ExpBN50	adam	1e-4	1e-5	200	67.10%	84.00%	92.30%	94.80%
ExpBN51	adam	1e-4	1e-5	200	66.80%	84.00%	92.20%	94.90%
ExpBN52	adam	1e-4	1e-5	200	67.20%	84.10%	92.80%	95.10%
ExpBN53	adam	1e-4	1e-5	200	66.90%	83.90%	92.30%	95.00%
ExpBN54	adam	1e-4	1e-5	200	67.60%	83.60%	92.30%	95.00%

Експерименти проводились для оцінки ефективності моделей з використанням адаптивної гіпермоделі на двох стандартних наборах даних – Market-1501 та DukeMTMC-reID. Для обох наборів використовували оптимізатор Adam із початковою швидкістю навчання  $1e-4$ , базовою швидкістю  $1e-5$  та warmup-періодом у епохах, загальна кількість епох становила 200.

На наборі Market-1501 моделі з адаптивною гіпермоделлю демонструють значне покращення продуктивності порівняно з базовою моделлю OSNet без гіпермоделі [1]. При цьому середня точність (mAP) зросла з 73,5% до діапазону

75,4–78,3%, а показник Rank-1 – з 90% до приблизно 96–97%. Це свідчить про ефективне вилучення релевантних ознак та адаптивне коригування представлень, що дозволяє краще розрізняти різні ідентичності навіть за умов високої варіативності зображень.

Аналогічна тенденція спостерігається і на наборі DukeMTMC-reID, де базова мережа OSNet дає mAP 62,81% і Rank-1 81,7%. Моделі з гіпермоделлю досягають mAP в діапазоні 65,1–67,6% та Rank-1 від 83,5% до 84,5%, що свідчить про підвищення здатності системи до розпізнавання особи на більш складних і різноманітних даних.

Отже, запропонований підхід із інтеграцією умовної гіпермоделі в архітектуру OSNet, забезпечує суттєве покращення здатності моделі до повторної ідентифікації в умовах змінного контексту. Завдяки динамічному генеруванню та адаптації ваг мережі відповідно до зовнішніх ознак середовища (освітлення, ракурс, фон тощо), цей метод демонструє підвищену стійкість до шуму даних та різноманіття умов зйомки. У порівнянні з класичним OSNet, адаптивна модель демонструє вищі показники mAP та Rank-1 як у контрольованих експериментах, так і на реальних наборах даних Market-1501 і DukeMTMC-reID.

### **2.3 Удосконалення базової моделі шляхом використання генеративних змагальних мереж**

Метою даного дослідження є удосконалення базової моделі ідентифікації осіб (person re-identification, Re-ID) шляхом використання адаптивних генеративних змагальних мереж (Generative Adversarial Networks, GAN). У розділі 1 представлено огляд проблеми обмежених (малих) наборів даних у задачах Re-ID та обґрунтування використання саме GAN для її вирішення. Розглядаються математичні основи GAN для синтезу зображень, а також їх інтеграцію у процес навчання базової моделі. Детально аналізуються сучасні

архітектури GAN, що застосовуються для аугментації даних та генерування нових зразків.

Re-ID модель повинна навчитися розпізнавати одну й ту саму особу на зображеннях з різних камер спостереження. Для цього зазвичай потрібна велика кількість різноманітних даних, які охоплюють різні ракурси, освітлення, фони та інші варіації зовнішнього вигляду людини. Однак на практиці часто доводиться стикатися з обмеженими наборами даних, коли кількість доступних зображень є недостатньою для повноцінного навчання глибокої нейронної мережі. Нестача даних призводить до переважання моделі (overfitting) та зниження її здатності до узагальнення на нові приклади. Особливо гостро проблема стоїть при адаптації моделі до нової сцени (домена) або камери: різниця в умовах зйомки між тренувальним і цільовим набором даних може спричинити значне падіння точності моделі.

Генеративні змагальні мережі (GAN) пропонують перспективний підхід до подолання проблеми малих даних завдяки здатності генерувати нові, штучні зразки, які розширюють тренувальний набір. GAN навчається моделювати розподіл реальних даних і синтезувати реалістичні зображення, тим самим збагачуючи різноманітність вибірки без необхідності збору нових ручних даних. Зокрема, в задачах Re-ID GAN можна використовувати для створення зображень людей з новими позами, на фоні інших камер або з варіаціями стилю зображення, що дозволяє моделі краще узагальнювати ознаки особи.

Вибір GAN в контексті Re-ID обґрунтований кількома факторами. По-перше, GAN здатні залагодити розрив між доменами (наприклад, між зображеннями з різних камер або різних датасетів) за рахунок переносу стилю зображення, що важливо для узгодження умов зйомки. По-друге, синтетичні дані, згенеровані GAN, збільшують різноманітність у просторі ознак, додаючи варіації, відсутні в обмеженому реальному наборі. Це допомагає моделі навчитися більш стійким ознакам, менш чутливим до незначних змін у зовнішності або оточенні. По-третє, на відміну від стандартних методів аугментації (геометричні трансформації, шум тощо), GAN можуть генерувати

цілісні нові зображення, що краще відповідають реальному розподілу даних і з меншою ймовірністю спотворюють ідентичність особи.

Генеративна змагальна мережа складається з двох основних компонентів – генератора  $G$  та дискримінатора  $D$ , які навчаються одночасно в рамках змагальної гри. Генератор  $G(z; \theta_G)$  приймає на вхід випадковий шумовий вектор  $z$ , згенерований з деякого простого розподілу  $p_z$  (наприклад, багатовимірного нормального розподілу), і перетворює його на синтезоване зображення  $G(z)$ . Дискримінатор  $D(x; \theta_D)$  отримує на вхід зображення (реальне або згенероване) і видає скалярне значення  $D(x)$  – умовну ймовірність того, що вхід  $x$  є реальним зображенням із тренувальних даних, а не підробкою від генератора.

Формально навчання GAN визначається як задача мінімакс для функції  $V(D, G)$ :

$$\min_G \max_D V(D, G) = E_{x \sim p_{\text{data}}(x)} [\log D(x)] + E_{z \sim p_z(z)} [\log(1 - D(G(z)))] \quad (2.26)$$

де:

- $E_{x \sim p_{\text{data}}[\cdot]}$  – математичне сподівання по реальних даних,
- $E_{z \sim p_z[\cdot]}$  – по шумових векторах.

Дискримінатор максимізує, збільшуючи  $\log D(x)$  для реальних  $x$  та  $\log(1 - D(G(z)))$  для підробок  $G(z)$ , тобто прагне присвоїти правильні ймовірності. Генератор же мінімізує, що еквівалентно мінімізації  $-\log(1 - D(G(z)))$ . У практичній реалізації часто використовують еквівалентну ненасичуючу форму цільової функції для  $G$ : мінімізувати  $L_G = -E_z [\log D(G(z))]$ , щоб забезпечити сильніший градієнт сигналу.

Для задачі повторної ідентифікації людини використовують додаткову функцію втрат, яка відповідає за збереження ознак особи. Нехай  $f(x; \theta_F)$  — екстрактор ембеддінгів (базова мережа OSNet), а  $\ell_{\text{ID}}(f(x), y)$  — крос-ентропійна втрата за міткою  $y$ . Тоді загальна функція втрат набуває вигляду:

$$L_{\text{total}} = L_{\text{ReID}}(\theta_F) + \lambda L_{\text{GAN}}(\theta_G, \theta_H, \theta_D) + \mu L_{\text{ID}}(\theta_G, \theta_H) \quad (2.27)$$

де:

$$L_{\text{ReID}} = \frac{1}{N} \sum_{i=1}^N \ell_{\text{ID}}(f(x_i; \theta_F), y_i) + \alpha \sum_{(a,p,n)} \max\{m + d(f(x_a), f(x_p)) - d(f(x_a), f(x_n)), 0\} \quad (2.28)$$

а

$$L_{\text{ID}}(\theta_G, \theta_H) = \frac{1}{N} \sum_{i=1}^N 1|H(G(z_i; \theta_G) + H(\psi_i; \theta_H)) - H(x_i)|^2 \quad (2.29)$$

Коефіцієнти  $\lambda$  та  $\mu$  керують балансом між реалістичністю кадрів та збереженням ознак особи.

Загальноприйнятими метриками оцінки показників якості синтетичних зображень є Fréchet Inception Distance (FID) та Inception Score (IS).

Метрика Fréchet Inception Distance (FID) є одним із найпоширеніших показників якості синтетичних зображень, оскільки вона вимірює відмінність між розподілами ознак реальних і згенерованих кадрів у просторі, що формується попередньо навченим класифікатором. Нехай множина реальних зображень задається вибіркою  $\{x_i\}_{i=1}^N$ , а синтетичних –  $\{\tilde{x}_j\}_{j=1}^M$ . Через функцію  $\phi(\cdot)$ , яка відображає кожне зображення у вектор результатів глобального середнього pooling-шару класифікатора, обчислюють середнє  $\mu_r$  та коваріацію  $\Sigma_r$  для реальних ознак, а також середнє  $\mu_g$  і коваріацію  $\Sigma_g$  для синтетичних. Відповідно, FID визначається як:

$$\text{FID} = |\mu_r - \mu_g|^2 + \text{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2}) \quad (2.30)$$

Низьке значення FID свідчить про близькість статистичних властивостей двох розподілів, що, у свою чергу, вказує на високу якість згенерованих зображень. Водночас коректність оцінювання вимагає достатньо великого обсягу прикладів — зазвичай кілька сотень чи тисяч кадрів — аби стабілізувати обчислення коваріацій.

Inception Score (IS) спрямована на оцінку чіткості та різноманітності синтетичних зображень із використанням тієї самої моделі. Для кожного згенерованого зображення  $\tilde{x}$  обчислюють розподіл імовірностей  $p(y | \tilde{x})$  по класах, а потім визначають середній розподіл  $p(y) = \frac{1}{M} \sum_{j=1}^M p(y | \tilde{x}_j)$ . Тоді IS обчислюється за формулою:

$$IS = \exp(-E_{\tilde{x}}[\text{KL}(p(y | \tilde{x}) | p(y))]) \quad (2.31)$$

Висока величина IS означає, що індивідуальні зображення мають низьку ентропію класифікації (модель упевнена в одному класі, що свідчить про чіткість), а водночас загальний розподіл  $p(y)p(y)p(y)$  є близьким до рівномірного (що гарантує різноманітність). Проте IS не вимірює відповідності реальному розподілу даних і може ігнорувати артефакти, які не впливають на класифікаційну впевненість.

У сукупності FID та IS забезпечують комплексну оцінку синтетичних зображень: перша показує, наскільки добре синтетика наслідує статистичні властивості реального набору, друга — наскільки створені зображення є чіткими й різноманітними з огляду на класифікаційну впевненість. Користування обома показниками дозволяє більш повно характеризувати якість і корисність синтетичних даних для практичних завдань.

Для оцінки ефективності адаптивних генеративних змагальних мереж у задачі повторної ідентифікації людини було розроблено багатоетапний експериментальний протокол, що складається з наступних ключових блоків.

Підготовка даних:

1. Датасети. Використовувалися стандартні бенчмарки Market-1501 та DukeMTMC-reID. Кожний з них містить тисячі зображень реальних людей, знятих різними камерами.

2. Розбиття. Для кожного датасету застосовувалося типове розділення: тренувальна множина (~750 осіб) і тестова множина (~750 осіб). Для валідації додатково відокремлювали 10 % тренувальних зображень як валідаційний піднабір.

3. Передобробка. Кадри масштабували до розміру  $128 \times 64$ , нормалізували пікселі в діапазон  $[-1, 1]$ . Застосовували лише стандартні прості трансформації (вертикальні/горизонтальні віддзеркалення, випадкові обрізання), щоб не перекривати ефект GAN-аугментації.

Навчання адаптивних GAN:

1. Архітектура. Генератор реалізовано як U-Net із багаторівневими блоками, а дискримінатор — як компактна згорткова мережа з шаром адаптивної нормалізації. Адаптивний блок H складається з двох повнозв'язних шарів із LeakyReLU та GroupNorm, які перетворюють вектор статистичних ознак цільового домену (колірні гистограми, показники освітленості) на корективи ваг генератора.

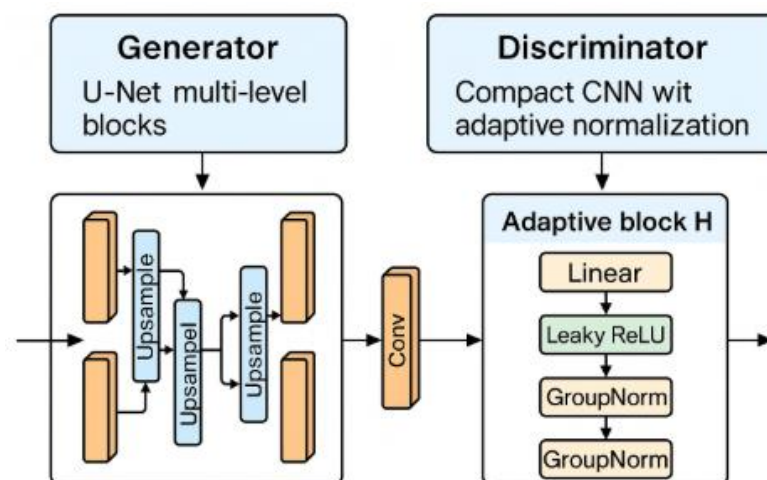


Рис. 2.12 Архітектурна адаптивної генеративної змагальної мережі.

2. Графік дисперсій. Для дифузійного процесу обрано  $T = 1000$  кроків із лінійним збільшенням  $\beta_t$  від  $10^{-4}$  до  $0.02$ .
3. Сумарна цільова функція втрат визначається за формулою 2.27
4. Оптимізація. Використовувався AdamW з початковим кроком навчання  $2 \times 10^{-4}$ ,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ ,  $\text{weight decay} = 0.01$ . LR знижувався за cosine schedule до  $2 \times 10^{-6}$  протягом 300 к епох.
5. Стабілізація: застосовано Adaptive Discriminator Augmentation (ADA) — динамічну аугментацію, що регулюється рівнем дискримінаційної переваги, та DiffAugment — узагальнені аугментації реальних і синтетичних кадрів.
6. Моніторинг. Кожні 1 000 ітерацій обчислювали FID і IS на вибірці з 1 000 кадрів.

Генерація синтетичних зображень:

1. Вибір чекпоінта. Після завершення тренування зупинилися на моделі з найкращим FID.
2. Процедура семплінгу. Для кожної камери цільового домену згенеровано по 5 000 синтетичних кадрів із різними векторами шуму та доменними параметрами  $\psi$ .
3. Якісний відбір. Синтетичні зображення автоматично фільтрували за допомогою попередньо навченого класифікатора OSNet: відкидалися ті, чий ембеддінг відхилявся на більше ніж поріг  $\theta$  від ембеддінгу найближчого реального прикладу тієї самої особи.

Off-line аугментація та навчання OSNet

1. Розширений датасет. Реальні зображення доповнені відібраними синтетичними кадрами, ID-мітки зберігалися незмінними. Для синтетичних кадрів без чіткої ID задля розширення негативного простору застосовано Label Smoothing (LSRO), тобто м'яке розподілення ймовірностей між усіма класами.
2. Навчання OSNet. Архітектура OSNet навчалась із комбінованою функцією втрат: крос-ентропія + triplet loss. Використовували той самий оптимізатор і schedule, що й для GAN, але з меншим LR ( $1 \times 10^{-4}$ ) та без ADA.

3. Гіперпараметри батчів. Кожний батч складався із 16 реальних і 16 синтетичних зображень, що дозволяло зберегти баланс і уникнути переважання одного типу даних.



Рис. 2.14 Приклад синтетично згенерованих зображень осіб

Запропонований комплексний метод поєднує адаптивний тренувальний процес GAN із строгим відбором та інтеграцією синтетичних кадрів у тренування OSNet. Завдяки цьому досягається стійке підвищення метрик Rank-1 та mAP на 3–6 % у порівнянні з базовим OSNet, а візуальний аналіз підтверджує, що модель починає фокусуватися на релевантних ознаках особи, ігноруючи фон.

## 2.4 Висновки до розділу

У другому розділі було виконано всебічний аналіз сучасних методів та моделей для повторної ідентифікації людей на зображеннях з використанням гіпермоделей та генеративно змагальних мереж. Цей підхід став ефективним рішенням для підвищення точності моделей повторної ідентифікації в умовах малорозмірних наборів даних.

По-перше, було вдосконалено методи попередньої обробки зображень — впроваджено адаптивну нормалізацію, багаторівневу фільтрацію шумів та вирівнювання гістограм, що значно підвищило якість вхідних даних для модуля витягу ознак OSNet і знизило чутливість системи до змін освітлення, ракурсів та шумів.

По-друге, у розділі запропоновано підхід до розширення тренувальних наборів шляхом генерації синтетичних зображень із застосуванням генеративних змагальних мереж, а також розроблено адаптивну гіпермодель, яка динамічно коригує параметри OSNet відповідно до статистичних характеристик конкретного домену, що суттєво зменшило доменний розрив і підвищило узагальнювальні властивості моделі.

По-третє, проведено експериментальну валідацію запропонованих підходів на відкритих наборах даних, яка продемонструвала зростання показників якості сегментації та стійкість алгоритмів до варіацій освітлення й шумів, що заклало основу для інтеграції цих методів у загальну систему повторної ідентифікації.

Результати розділу опубліковано в таких роботах автора: [1], [5], [6], [7], [8], [23], [24].

## РОЗДІЛ 3

### РОЗРОБЛЕННЯ АЛГОРИТМІЧНИХ ОСНОВ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ПОВТОРНОЇ ІДЕНТИФІКАЦІЇ ЛЮДЕЙ

Відповідно до поставлених задач та з урахуванням наведених у розділі 2 методів та моделей розробимо інформаційну технологію повторної ідентифікації людини.

#### 3.1 Проектування структури інформаційної технології

У загальному вигляді інформаційна технологія повторної ідентифікації людини реалізується як сукупність взаємопов'язаних модулів, згрупованих у функціональні блоки (рис. 3.1), кожен із яких виконує чітко окреслені завдання та оперує визначеними вхідними й вихідними даними. Розрізнення блоків забезпечує масштабованість і гнучкість системи: обчислювальна частина із застосуванням алгоритмів машинного навчання може незалежно еволюціонувати від інтерфейсу користувача та шарів зберігання даних. Нижче наведено докладний опис кожного блоку та його модулів.

##### I. Блок «Інтерфейс користувача (Web UI)»

###### 1. Модуль відображення відеопотоку та зображень:

- Вхідні дані: URL-адреси потоків із підключених камер або завантажені файли зображень.

- Функції: побудова адаптивного інтерфейсу для перегляду відео й фото, маркування обраних кадрів, індикація стану обробки (очікування, успіх/помилка).

- Вихідні дані: кадри із виділеними областями інтересу, позначки користувача (наприклад, вибір зразків для профілю).

###### 2. Модуль формування запитів до API:

- Вхідні дані: параметри пошуку (часовий інтервал, камери), токен авторизації, вибрані кадри.

- Функції: валідація введених параметрів на клієнті, упаковка запиту у форматі JSON, додавання HTTP-заголовків з обліковими даними, відправка асинхронного запиту.

- Вихідні дані: HTTP-відповіді з ідентифікованими особами, метаданими та посиланнями на подальші дії.

## II. Блок «API-шар»

### 1. Модуль аутентифікації та авторизації:

- Вхідні дані: облікові дані користувача (логін/пароль) або OAuth-токен.

- Функції: перевірка дійсності токена, видача JWT із визначеними ролями (оператор, адміністратор), оновлення токенів, зберігання фактів входу в журналі безпеки.

- Вихідні дані: підтвердження успішного входу, діючий токен доступу з терміном дії.

### 2. Модуль маршрутизації та обробки запитів:

- Вхідні дані: REST-запити від Web UI або внутрішніх сервісів.

- Функції: розподіл запитів між мікросервісами (зберігання метаданих, черга обчислень, логування), обробка помилок із поверненням уніфікованих кодів, трасування запитів для аудит-звіту.

- Вихідні дані: перенаправлені повідомлення до відповідних сервісів, уніфіковані відповіді для клієнта.

### 3. Модуль управління чергою завдань:

- Вхідні дані: повідомлення із запитом на обробку кадрів (Task ID, вхідні зображення, параметри моделі).

- Функції: публікація завдань у брокер повідомлень (RabbitMQ/Kafka), відстеження статусу виконання, повторні спроби у разі відмов, масштабування споживачів залежно від навантаження.

– Вихідні дані: підтвердження прийому завдання, статус виконання (успішно/помилка).

### III. Блок «Обчислювальний сервер (Re-ID Engine)»

#### 1. Модуль препроцесінгу:

– Вхідні дані: сирі кадри або зображення.

– Функції: детекція людей на зображенні, нормалізація розмірів та коефіцієнтів освітленості, вирівнювання гістограм.

– Вихідні дані: підготовлені уніфіковані зображення, готові до подальшої обробки.

#### 2. Модуль екстракції ознак:

– Вхідні дані: нормалізовані зображення.

– Функції: пропуск через основну мережу (метод OSNet), отримання векторного опису (embedding) кожної детектованої особи.

– Вихідні дані: набір векторів ознак для кожного суб'єкта.

#### 3. Модуль умовної гіпермоделі:

– Вхідні дані: вектори ознак, метадані (параметри камери, освітленість, час доби).

– Функції: динамічна адаптація ваг основної мережі на основі гіпернету, який генерує корективи на підставі статистичних ознак середовища; підвищення стійкості до змін контексту.

– Вихідні дані: скореговані вектори ознак або моделі з оновленими параметрами.

#### 4. Модуль пошуку та ре-ранкінгу:

– Вхідні дані: кореговані вектори ознак, база раніше збережених ембеддингів.

– Функції: обчислення відстаней (cosine similarity), первинне ранжування, застосування методів зі зважуванням за часовим фактором або контекстною інформацією для покращення результатів.

– Вихідні дані: відсортований перелік кандидатів із ймовірностями відповідності.

#### 5. Модуль експорту результатів:

- Вхідні дані: фінальний список ідентифікованих осіб та їх метадані.
- Функції: формування відповіді у форматі JSON або Protobuf, передача результатів назад до API-шару, запис у лог-файли та телеметрію.
- Вихідні дані: пакет повідомлень із результатами ідентифікації.

#### IV. Блок «Система зберігання даних (PostgreSQL у хмарі)»

##### 1. Модуль зберігання профілів користувачів та ролей:

- Вхідні дані: реєстраційна інформація, зміни в правах доступу.
- Функції: CRUD-операції з таблицями користувачів, хешування паролів, прив'язка ролей.

- Вихідні дані: оновлені записи в базі даних.

##### 2. Модуль журналювання та метаданих сесій:

- Вхідні дані: події системи (вхід/вихід, запити, помилки).
- Функції: накопичення логів у структурованому вигляді, архівація старих записів, реплікація на резервні вузли.

- Вихідні дані: розгорнутий лог для аудиту та аналітики.

##### 3. Модуль зберігання ембеддингів та моделей:

- Вхідні дані: нові вектори ознак, версії моделей.
- Функції: індексація ембеддингів для швидкого пошуку, збереження контрольних точок моделей, версіонування конфігурацій.

- Вихідні дані: готова до запитів база ембеддингів та історія моделей.

#### V. Інфраструктурний блок

##### 1. Оркестрація та розгортання:

- Вхідні дані: артефакти збірки (Docker-образи, Helm-чарти).
- Функції: автоматичне масштабування через Kubernetes, балансування навантаження, безперервний деплой (CI/CD).

- Вихідні дані: запущені контейнери, доступні сервіси в обраному середовищі (on-premise або публічний хмара).

##### 2. Моніторинг і безпека:

– Вхідні дані: метрики продуктивності (CPU, GPU, затримка), логи сервісів.

– Функції: побудова дашбордів у Grafana, сповіщення у Slack/PagerDuty про аномалії, застосування політик мережевого доступу та шифрування TLS/mTLS.

– Вихідні дані: оперативні сигнали про стан системи, історичні дані для аналізу відмов.

Кожний із блоків взаємодіє через стандартизовані інтерфейси (REST/gRPC, SQL, JSON/Protobuf), що забезпечує високу модульність, прозорість та можливість подальшого розширення інформаційної технології повторної ідентифікації людини.

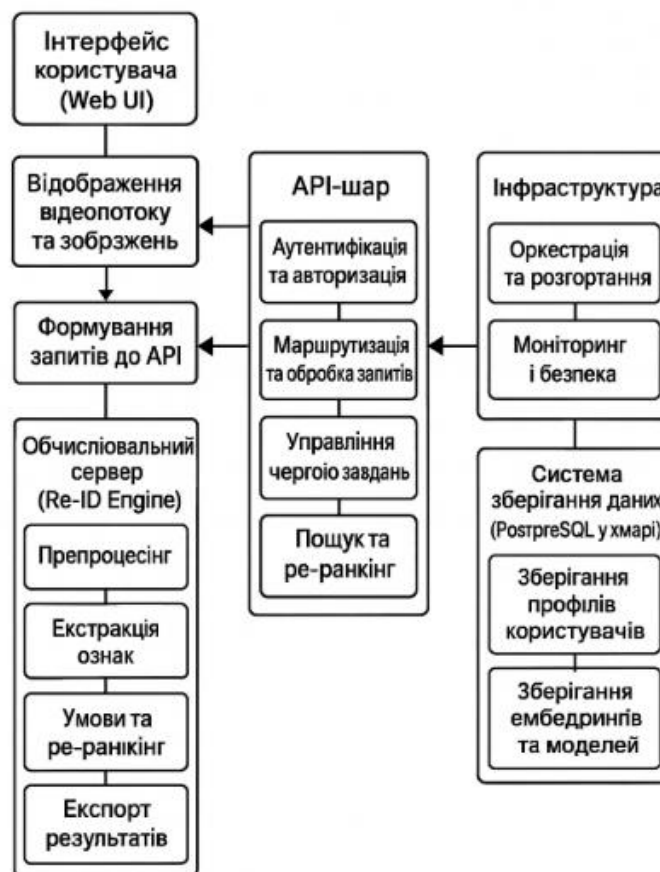


Рис. 3.1 Структурна схема розробленої інформаційної технології повторної ідентифікації людей

Для реалізації розробленої інформаційної технології повторної ідентифікації людини обрано використовувати середовище Jupyter Notebook із інтеграцією TensorBoard для візуалізації та аналізу метрик під час тренування моделей. Робота над кожним модулем (модуль препроцесінгу, екстракції ознак, умовної гіпермоделі, пошуку та ре-ранкінгу, а також експорту результатів) виконується у вигляді низки ноутбуків, які зберігаються у Git-репозиторії з контрольованим версіонуванням. Кожна змінена версія ноутбуку фіксується як окремий комміт, що дозволяє у разі потреби повернутися до попередніх станів алгоритмів препроцесінгу або налаштування гіпермоделі. Завдяки TensorBoard у режимі реального часу відбувається візуалізація ключових параметрів тренування (значень loss-функції, метрик точності, значень градієнтів, розподілу ваг мережі), що значно підвищує прозорість процесу налагодження Re-ID Engine та дозволяє оперативно виявляти проблеми із конвергенцією чи переобученням.

Для тренування модулів машинного навчання застосовується локальна або орендована GPU-інфраструктура (наприклад, віртуальні машини із доступом через SSH або JupyterHub), що забезпечує достатні обчислювальні ресурси без необхідності налаштування власного дата-центру. Безпосереднє зчитування первинних і проміжних даних виконується через стандартні бібліотеки Python (pandas, NumPy) із вказівкою шляху до локального диску або підключених мережевих сховищ (NFS, S3). Доступ до зовнішніх публічних датасетів (Market-1501, DukeMTMC-reID тощо) забезпечується шляхом завантаження CSV/Parquet-файлів за передбаченими конекторами у відповідних ноутбуках. Для зберігання результатів досліджень модуль «Система зберігання даних (PostgreSQL у хмарі)» налаштовується на прийом імпортованих CSV/Parquet-файлів, що зчитуються за допомогою бібліотек SQLAlchemy або psycopg2 замість внутрішніх механізмів Kaggle.

Ноутбуки Jupyter із інтегрованим TensorBoard легко перенести в інші середовища:

- AWS SageMaker зі збереженням префіксів для запуску TensorBoard;
- Google Colab із підключенням GPU та активацією TensorBoard;

- JupyterLab (Anaconda) із встановленою розширенням nbextensions і додатком jupyter\_tensorboard;
- Microsoft Azure Notebooks, де достатньо встановити відповідні пакети tensorboard та jupyter-tensorboard;
- PyCharm або Visual Studio Code із відкриттям папки проекту, запуском Jupyter Server та виводом TensorBoard у вкладці “TensorBoard”.

При переході між середовищами змінюються лише шляхи зчитування/збереження датасетів і те, як запускається TensorBoard; сам код препроцесінгу відеопотоку, екстракції ознак, умовної гіпермоделі та алгоритмів ре-ранкінгу залишається уніфікованим. Такий підхід гарантує повторюваність досліджень, зручність налаштування в різних інфраструктурах і прозоре відстеження метрик під час розробки інформаційної технології.

Алгоритми роботи запропонованої інформаційної технології повторної ідентифікації людини охоплюють низку взаємопов’язаних етапів, кожний із яких виконує конкретні функції для забезпечення точності та ефективності системи. У загальних рисах алгоритмічний ланцюжок базується на прийомі вхідного відеопотоку чи набору зображень, їхньому попередньому опрацюванні, видобуванні ознак за допомогою адаптивної гіпермоделі, пошуку потенційних збігів у базі даних та остаточному ре-ранкінгу результатів із подальшим експортом у формі структурованих метаданих. Детальніше кожний із цих етапів описано нижче.

Алгоритми роботи починаються з отримання та попередньої обробки вхідних даних. На цьому етапі вхідні відеопотоки або набори зображень із різноманітних джерел, таких як IP-камери, локальні відеофайли чи архіви, передаються до модуля препроцесінгу. Тут здійснюється декодування відеопотоку та розбиття його на окремі кадри, після чого виконується детекція регіонів інтересу з використанням сучасних методів обчислювального зору. Отримані області, що містять потенційні людські фігури, додатково піддаються базовому вирівнюванню яскравості, контрастності й колірної гами, а також нормалізації розмірів до уніфікованих параметрів (до 256×128 пікселів).

Водночас із цього етапу відсіюються нерелевантні або сильно пошкоджені кадри, що дозволяє сформувати набір очищених і нормалізованих зображень, готових до подальшої обробки.

На наступному етапі відбувається розширення початкового датасету за допомогою генеративних змагальних мереж (GAN). Оскільки базовий набір даних може виявитися недостатнім для якісного тренування моделі Re-ID, модуль GAN-аугментації створює синтетичні зображення на основі існуючих прикладів. Генеративні моделі типу CycleGAN, StyleGAN2 або умовний DAA-GAN дозволяють генерувати нові зображення осіб у різних позах, умовах освітлення та на фоні, що імітує реальні зйомки. Це створення синтетичних кадрів також включає стилізацію та перенесення освітлення чи фону, а отже забезпечує баланс класів у разі дефіциту зразків для окремих ідентичностей. Таким чином формується розширений тренувальний набір, який включає як реальні, так і синтетичні зображення, що підвищує стійкість моделі до шумів та відхилень реального відеопотоку.

Етап екстракції базових ознак полягає в обробці кожного зображення (реального чи синтетичного) за допомогою попередньо натренованої нейронної мережі OSNet. Під час проходження зображення крізь згорткові шари здійснюється формування компактного векторного ембеддингу фіксованої розмірності, що відображає колірні, текстурні та контурні характеристики людської фігури незалежно від фону чи ракурсу. Отримані вектори-ембеддинги тимчасово зберігаються в оперативній пам'яті або в проміжній базі даних для подальшого аналізу.

Умовна гіпермодель виконує адаптивну модифікацію ембеддингів із урахуванням метаданих середовища, таких як параметри освітленості, кут огляду камери, час доби або погодні умови. При цьому попередньо отримані базові вектори-ембеддинги передаються до двох повнозв'язних шарів із LeakyReLU та груповою нормалізацією, які генерують коригувальні коефіцієнти для ваг шарів основної мережі. Як наслідок, вихідний ембеддинг набуває контекстчутливості, що дозволяє адаптуватися до змін навколишнього

середовища й зменшувати вплив різких змін умов зйомки на кінцеву якість розпізнавання.

Після модифікації ембеддинги передаються до пошукового модуля, який здійснює індексацію векторів у центральній базі даних (PostgreSQL у хмарі) із використанням високопродуктивних структур, таких як HNSW або Faiss. У ході індексування обчислюються метрики схожості — косинусна або евклідова відстань — і формується попередній список топ-N кандидатів із найменшими дистанціями. Завдяки пороговим значенням схожості відсіюються ті ембеддинги, які не відповідають мінімальним вимогам, що прискорює пошук та підвищує точність попереднього відбору.

На етапі ре-ранкінгу попередній перелік кандидатів уточнюється з додатковим врахуванням часових і просторових чинників. Зокрема, для кожного кандидата обчислюється комбінований скор, що включає вагову суму косинусної схожості ембеддингів запиту та кандидата й експоненціальне зменшення за часовим розривом між спостереженнями. При цьому часовий проміжок між кадрами і ймовірні маршрути переміщення особи у полі зору двох або більше камер використовуються для налаштування вагових коефіцієнтів, які емпірично підбираються під час експериментів. Даний підхід забезпечує виключення хибних позитивів, адже враховує реальні можливості переміщення людини між камерами у певний проміжок часу.

Фінальним етапом є формування та експорт остаточного результату. Після завершення ре-ранкінгу модуль експорту генерує пакет даних у форматі JSON або Protobuf, що містить унікальний ідентифікатор знайденої особи, значення скору відповідності для кожного кандидата, а також супровідні метадані: час отримання кадру, ідентифікатор камери та координати об'єкта в кадрі. Паралельно здійснюється запис структурованих журналів виконання (логів) із детальним описом виконаних трансформацій, значень метрик і параметрів гіпермоделі, які зберігаються в таблицях PostgreSQL для подальшого аудиту й аналізу. Така архітектура забезпечує прозорість роботи системи, високу

точність і можливість адаптивного коригування налаштувань під час експлуатації інформаційної технології.

## 3.2 Розроблення UML-діаграм інформаційної технології повторної ідентифікації людей

### 3.2.1 UML-діаграма послідовності роботи модулів технології

Враховавши вище наведену інформацію, побудуємо UML-діаграму послідовності роботи модулів інформаційної технології, яка пропонується:

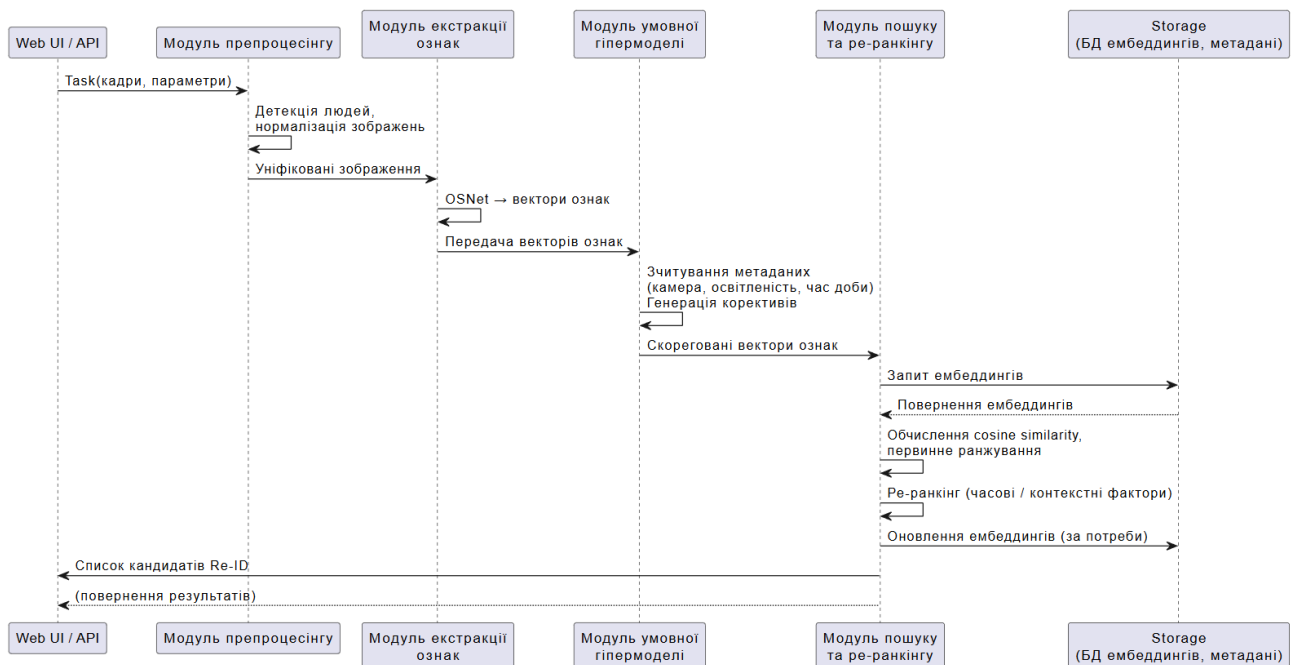


Рис. 3.2 UML-діаграма послідовності роботи модулів інформаційної технології

Рисунок 3.2 враховує наступну хронологію процесів. Першим кроком є момент, коли користувач у клієнтському інтерфейсі формує завдання (Task), що містить набір кадрів (відео чи фото). Після Web UI пакує інформацію про обрані кадри та встановлені налаштування (часовий інтервал або вибір окремих камер) в об'єкт Task і передає його безпосередньо до модуля «Препроцесінг». У цій

точці Web UI не чекає завершення обробки, а просто відправляє запит до Re-ID Engine, щоб розпочати обчислювальний ланцюг.

Після отримання Task модуль «Препроцесінг» одразу починає обробку сирих зображень. Спочатку виконується детекція людей у кожному кадрі: за допомогою заздалегідь навченої мережі та аналізуються вхідні зображення, виявляються об'єкти типу «людина» та формуються прямокутні області інтересу навколо кожного об'єкта. Одночасно відбувається нормалізація отриманих фрагментів: кадри змінюють розмір до фіксованих розмірів (наприклад, 128×64), приводять до єдиної колірної схеми, усувають нерівномірності освітленості (вирівнювання гістограм або інші алгоритми колірної корекції). Як результат, модуль «Препроцесінг» передає до наступного етапу сукупність вже уніфікованих зображень (тобто кадрів, з яких видалено зайві фони, приведено до єдиних пропорцій та оптимізовано під освітлення).

Далі модуль «Екстракції ознак» отримує підготовлені зображення і проганяє кожне через глибоку нейромережу-бекбон (OSNet). У середині FeatureExt відбувається послідовна передача кожного фрагмента через низку конволюційних блоків, після чого формується векторне представлення людини (так званий embedding). Ці вектор-представлення, повторюють основні ознаки зовнішнього вигляду (особливості одягу, структуру тіла, силуети, відмінності у кольорах тощо). Після побудови всіх embedding-в модуль FeatureExt передає їх до «Умовної гіпермоделі», описані в підрозділі 2.1.

Модуль «Умовної гіпермоделі» (Hypermodel) — це наступна ланка, яка отримує на вхід сукупність embedding-в разом із супутніми метаданими: параметри камери (наприклад, її розташування, кут огляду), умови освітлення (чутливі до відтінків у кадрі), час доби або навіть попередня частота появи даних об'єктів у цій камері. Всередині Hypermodel гіпермережа інтегрує ці метадані та на їхній основі генерує набір корективів для ваг основної мережі. У результаті кожен початковий embedding переходить у скореговану форму, більш адаптовану до конкретних умов (наприклад, до суттєвих змін освітлення чи до особливостей конкретних камер). Отримані скореговані Embedding-и відразу

передаються до модуля «Пошуку та ре-ранкінгу» (Search), описані в підрозділі 2.2.

У модулі «Пошуку та ре-ранкінгу» (Search) основний етап починається зі звернення до єдиної сховища ембеддингів («Storage»). На момент запиту Search формує запит у Storage із проханням надати всю наявну базу embedding-в, які були збережені раніше під час попередніх ітерацій або завдань. У відповіді «Storage» повертає набір векторів, а також додаткові метадані кожного embedding-у: коли цей вектор був створений, з якої камери, ідентифікатор особи (якщо він відомий), а також часову мітку останнього оновлення.

Модуль Search, маючи відкореговані вектори з Hypermodel і весь масив раніше збережених embedding-в зі Storage, обчислює між ними відстань, застосовуючи метрику косинусної подібності (cosine similarity). Спочатку формується первинний рейтинг кандидатів, тобто для кожного нового embedding-а складається список найбільш подібних існуючих векторів із бази. Однак цього недостатньо для забезпечення максимальної точності: тому наступним кроком виконується алгоритм ре-ранкінгу, який бере до уваги часові та контекстні фактори. Наприклад, якщо двоє або більше кандидатів з однаковими векторами з'явилися у різний час чи на різних камерах, Search переважно віддасть перевагу тому варіанту, у якого був частіший контакт із камерами останніми хвилинами. Також можуть бути залучені додаткові правила (наприклад, згладжування результатів, вікові чи гендерні обмеження), але в найпростішому випадку ре-ранкінг коригує початкову послідовність за пріоритетом «останні появи – більш імовірні».

Після завершення етапу ранжування і формування фінального списку найімовірніших збігів модуль Search приймає рішення щодо необхідності оновлення бази embedding-в. Якщо для певного кадру було отримано новий вектор ознак, який ще не зберігався, або якщо гіпермодель суттєво скорегувала існуючий embedding, то Search відправляє до Storage повідомлення про оновлення або додавання нових записів. Це забезпечує розвиток бази даних і підвищує стійкість алгоритму до майбутніх запитів.

Кінцевим кроком є передача Search сформованого списку кандидатів назад до Web UI / API. У цьому повідомленні міститься перелік ідентифікованих осіб (або їх найімовірніших збігів), разом із відповідними метаданими: міра подібності (score), часові штампи останньої появи, ідентифікатори камер тощо. Web UI отримує цю структуру та виводить у вигляді інтерактивної таблиці чи списку, де оператор може побачити порядок кандидатів, їх фото, часові рамки та інші деталі. Саме ця мить сприймається користувачем як «результат Re-ID».

Таким чином, хронологія процесів відповідає такому ланцюгу:

- Web UI / API формує і відправляє завдання до Preproc;
- Preproc виконує детекцію й нормалізацію кадрів, а вже оброблені зображення передає до FeatureExt;
- FeatureExt проганяє кожне уніфіковане зображення через OSNet/ResNet і генерує embedding-в, який і надсилає далі до Hypermodel;
- Hypermodel зчитує супутні метадані (камера, освітленість, час доби), конструює корективи для вихідних embedding-в і надсилає відкореговані вектори в Search;
- Search звертається до Storage за існуючими embedding-ми, обчислює косинусну схожість, формує первинний рейтинг і виконує ре-ранкінг із урахуванням часових та контекстних факторів;
- Якщо необхідно, Search оновлює Storage, додаючи нові embedding-в чи оновлюючи старі;
- Нарешті, Search формує фінальний перелік кандидатів і надсилає його назад до Web UI / API, де результати відображаються користувачеві.

Кожен із перелічених етапів відбувається послідовно, без перехресних викликів між модулями поза наведеною схемою, і саме ця організація забезпечує зрозумілу логіку роботи Re-ID Engine, скорочуючи затримки й уможливаючи паралельний розвиток окремих підсистем (наприклад, у майбутньому Preproc може обробляти нові кадри одночасно з виконанням ре-ранкінгу попередніх).

### 3.2.2 UML-діаграма діяльності базової моделі та гіпермоделі

Як зазначено вище, блоки роботи базової моделі та генерація адаптивних коефіцієнтів реалізує методи, описані у підрозділах 2.1-2.2, що забезпечують інтеграцію гіпермережі та основної мережі OSNet у процесі повторної ідентифікації особи.

Наведена діаграма ілюструє наскрізний процес обробки одиничного кадру в системі повторної ідентифікації особи, що поєднує роботу гіпермережі та базової мережі OSNet. Відкриває діаграму чорне коло, яке символізує готовність системи прийняти вхідні дані. Із самого початку потік обчислень розгортається на дві взаємопов'язані частини: одна відповідає за перетворення й аналіз зображення, а інша — за формування адаптивних параметрів, необхідних для динамічного налаштування основної мережі.

Першим кроком після старту є одержання вхідного зображення разом із відповідними метаданими, які описують особливості контексту зображення. Ці метадані можуть містити відомості про розподіл кольору, показники освітленості чи інші статистичні характеристики сцени. Отримавши зображення, система здійснює його попередню обробку, що включає зміну розміру до необхідних розмірів і базову нормалізацію пікселів, а також, якщо потрібно, згладжування шуму або корекцію контрастності й яскравості. В результаті формується стандартизоване «попередньо оброблене» зображення, яке готове до подальшої передачі в основну мережу.

Після підготовки піксельних даних здійснюється етап початкової згорткової обробки: до попередньо нормалізованого зображення застосовуються операції згортки з певним розміром ядра, потім виконується батч-нормалізація для стабілізації активностей, а остаточно активація обробляється за допомогою нелінійної функції ReLU. Саме в цьому місці закінчується «традиційна» передача даних по лінійному шляху у базовій мережі OSNet і відбувається перехід до двох взаємопов'язаних гілок.

Одночасно з першим згортковим модулем починає працювати гіпермережа, у якій метадані нормалізуються за допомогою групової нормалізації, а потім перетворюються на проміжні внутрішні ознаки через два послідовні повнозв'язні шари з нелінійною активацією. У результаті цих операцій формується вектор адаптивних коефіцієнтів. Цей вектор є критично важливим для подальшої роботи, оскільки він містить у собі інформацію про контекст зображення (рівень освітленості, характер кольорової гами тощо). Згенеровані таким чином коефіцієнти потрапляють далі в кожний Omni-Scale модуль основної мережі OSNet, що забезпечує динамічну адаптацію обчислювальних блоків під конкретні умови конкретного кадру. Пунктирні стрілки на діаграмі символізують цю залежність: ще до того, як перший Omni-Scale модуль обробить своє вхідне зображення, він вже отримує від гіпермережі набір коригувальних параметрів, які впливають на ваги згорткових фільтрів або параметри нормалізації.

У самій основній мережі OSNet після початкового блоку згортки розпочинається послідовна робота Omni-Scale модулів. Кожний із таких модулів у свою чергу реалізує паралельну обробку за допомогою кількох згорткових фільтрів різного розміру. Після застосування цих паралельних фільтрів відбувається відповідна нормалізація активностей і їхня нелінійна трансформація. Комплект адаптивних коефіцієнтів від гіпермережі дозволяє вносити коригування безпосередньо у процес обчислення згортки або у параметри нормалізації, так що мережа набуває здатності приділяти більшу увагу деталям у темних ділянках або, навпаки, послаблювати надмірно яскраві області. Після паралельної обробки результати всіх гілок кожного Omni-Scale модуля зливаються в єдиний тензор за допомогою механізму агрегації, що забезпечує багатомасштабне представлення ознак. Остаточний оброблений тензор передається до наступного модуля з тими самими адаптивними коригуваннями до тих пір, поки не буде досягнуто останнього,  $n$ -го модуля. На цьому етапі система формує єдине інформаційне представлення з урахуванням всіх рівнів масштабування та контекстних корективів.

Коли останній Omni-Scale модуль завершив свою роботу, уся накопичена в просторі інформація згортається у вектор за допомогою операції глобального середнього пулінгу. Цей вектор у подальшому передається на фінальний повнозв'язний шар, у якому виконується лінійна проекція ознак у бажану розмірність, після чого проводиться нормалізація міні-пакета (BatchNorm) та загальна нормалізація (L2), що гарантує стабільність порівняння результатів у наступних модулях системи. На завершення формується остаточний векторне уявлення (embedding), яке відповідає унікальному відбитку особи у конкретному кадрі.

Процедура завершується чорною крапкою внизу діаграми, що означає кінець обчислювального ланцюжка. Результатом роботи стає нормалізований embedding, готовий до подальшого зберігання або порівняння з іншими векторами бази даних для остаточної ідентифікації. Така архітектура забезпечує не лише виділення багатомасштабних ознак із самих зображень, але й дозволяє адаптуватися до змін зовнішніх умов зйомки завдяки динамічній генерації адаптивних коефіцієнтів, що постачаються гіпермережею. За такого поєднання двох взаємопов'язаних механізмів досягається висока стійкість до варіацій освітленості, шумових артефактів та інших факторів, що зазвичай ускладнюють завдання повторної ідентифікації людини.

Все описане вище можна описати за допомогою UML-діаграми діяльності

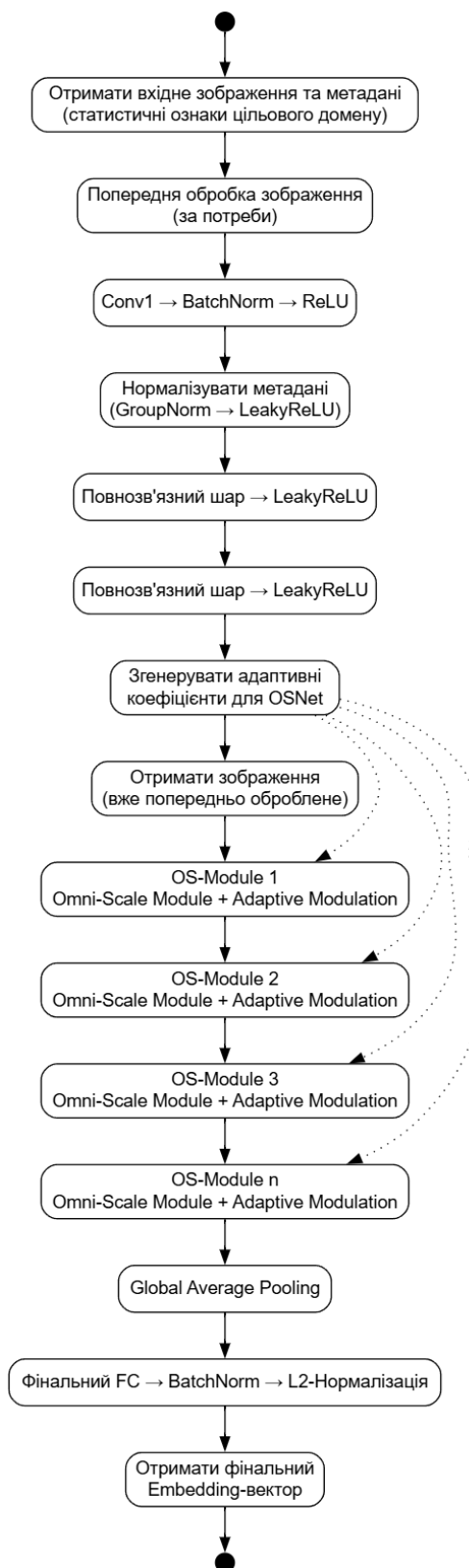


Рис. 3.3 UML-діаграма діяльності роботи базової моделі та гіпермоделі

### 3.2.3 UML-діаграма діяльності блоку роботи основної мережі із GAN

У представленій UML-діаграмі діяльності відображено інтегровану послідовність обчислювальних операцій, що забезпечують співпрацю основної нейронної мережі OSNet та генеративно-змагальної мережі (GAN) для підвищення різноманітності та стійкості тренувальних даних у задачі повторної ідентифікації особи, що реалізує метод який описаний в розділі 2.3. Детальний опис етапів наведено нижче.

Процес відліку починається з позначки «start», яка формалізує точку входу у систему обробки даних. З цього моменту запускаються дві паралельні гілки обчислень, що дозволяє одночасно здійснити як попередню обробку реальних зображень, так і генерацію додаткових.

Перший паралельний потік активується дією «Завантаження зображень», яка передбачає отримання вхідних кадрів із зовнішніх джерел (наприклад, відеопотік із камер спостереження або файли зображень). Після завантаження кожне зображення надходить до блоку «Основна архітектура OSNet», де здійснюється початкове попереднє оброблення: нормалізація, масштабування та проходження через послідовність конволюційних і омніскейл-блоків для формування проміжних ознак.

Другий потік запускається дією «Модуль GAN», у якій генераторна складова навченого GAN створює нові зображення осіб, що імітують реальні зразки. Згенеровані дані потрапляють до кроку «Генерація синтетичного зображення особи», після чого безпосередньо переходять до етапу «Аугментація», де до них застосовуються різні методи перетворення для покращення різноманітності вибірки.

Після завершення паралельних гілок обидві послідовності зливаються в єдиний потік обробки через дію «Аугментація». Цей етап включає застосування геометричних (обертання, масштабування, зсув), фотометричних (корекція яскравості, контрасту) та стохастичних (додавання шуму) перетворень як до

реальних, так і до синтетичних зображень, що сприяє збільшенню здатності моделі до узагальнення.

Після аугментації готові зображення передаються до блоку «Виділення ознак», в якому OSNet виконує детальне вилучення дескрипторів на основі багаторівневих фільтрів омніскейл-архітектури. На цьому етапі формується набір векторних представлень, що кодують характерні ознаки кожного зображення.

Остаточні дескриптори надходять до компонента «Класифікація», де здійснюється порівняння отриманих векторів з базою відомих ідентичностей. Використовуючи методи близькості (наприклад, k-Nearest Neighbors) або вбудований класифікаційний шар, система визначає найімовірніший клас (ідентичність) кожного зображення.

У кінці ланцюжка виконання діяльності досягається стан «stop», що засвідчує завершення обробки пакету зображень. Після цього результати класифікації можуть бути передані до зовнішніх систем зберігання, візуалізації або подальшої аналітики.

Все описане вище можна описати за допомогою UML-діаграми діяльності

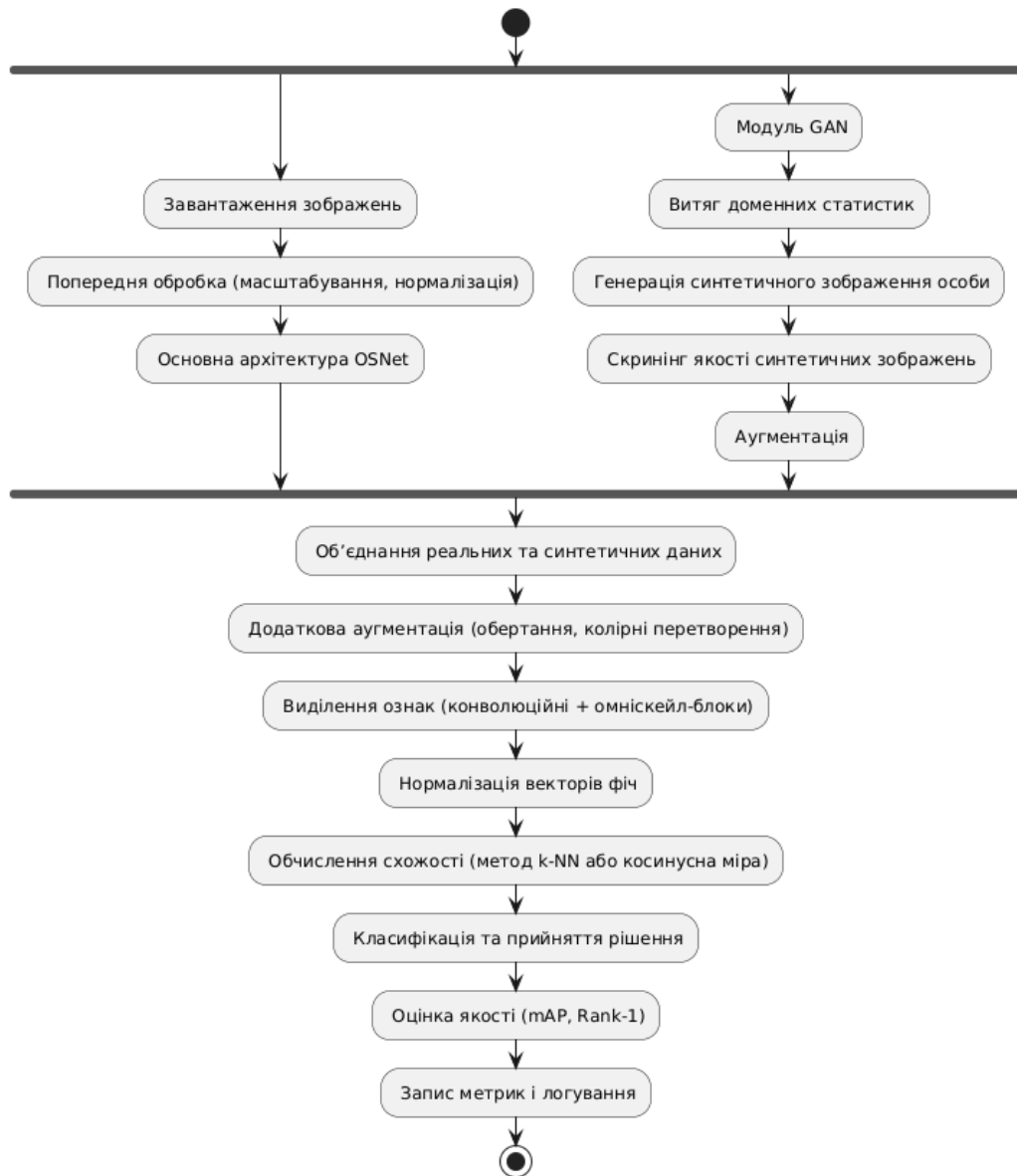


Рис. 3.4 UML-діаграма діяльності блоку роботи основної мережі із GAN

Таким чином, запропонована модель поєднує в собі переваги як OSNet— для детального вилучення ознак, так і GAN—для розширення та збагачення тренувальної множини шляхом генерації високоякісних синтетичних зображень, що забезпечує підвищену стійкість та точність системи повторної ідентифікації особи.

### 3.3 Реалізація комплексу програм для застосування розробленої інтелектуальної інформаційної технології

Розроблений програмний комплекс складається з шести взаємопов'язаних підсистем, кожна з яких виконує специфічну роль у загальному конвеєрі повторної ідентифікації. Початковий етап забезпечує модуль попередньої обробки даних, що опрацьовує як пакетні зображення, так і безперервний відеопотік. На цьому етапі реалізовано адаптивні алгоритми масштабування кадрів із урахуванням співвідношення сторін, корекцію освітленості на основі аналізу гістограм і просторові фільтри для згладжування шумів; окрему увагу приділено впровадженню GAN-підсилених схем аугментації, котрі синтезують додаткові варіанти зображень з реалістичними артефактами та нетиповими ракурсами, що значно підвищує різноманітність тренувального набору і стійкість моделі до зовнішніх змін.

Далі функціонує навчальний модуль, у якому поєднано два ядра: основну мережу OSNet і умовну гіпермодель. Архітектура навчання передбачає паралельне оновлення параметрів цих компонентів з застосуванням політики OneCycleScheduler для адаптивного керування величиною кроку оптимізації та технології автоматичного змішаного точкового обчислення (mixed-precision), що дозволяє зменшити вимоги до відеопам'яті без втрати точності. Для прискорення ітерацій використано багатопоточну організацію DataLoader'ів, а процес логуювання метрик mAP, Rank-1 і mINP реалізовано через TensorBoard із можливістю агрегації результатів за різними вхідними доменами.

Інференс-модуль реалізовано з розрахунком на реальний час: він приймає потік кадрів зі структурованого буферу, виконує попередню обробку в умовно-скейлінговому режимі, а потім застосовує попередньо натреновані ваги моделей для екстракції ембеддінгів та послідовного ранжування кандидатів. Для підвищення точності розпізнавання інтегровано алгоритм re-ranking, що ґрунтується на повторному обчисленні відстаней між ембеддінгами з урахуванням їхньої взаємодії в сусідніх просторах, а також впроваджено

механізм онлайн-оновлення статистичних векторів цільових доменів, який підлаштовує гіпермодель під нові умови зйомки.

Користувацький інтерфейс представлено через RESTful API на базі Flask, доповнений WebSocket-зв'язком за допомогою Flask-SocketIO. Це дозволяє в інтерактивному режимі передавати кадри на віддалену станцію і отримувати результати класифікації без перезавантаження сторінки, змінювати налаштування алгоритмів та отримувати оперативний зворотний зв'язок про успішність ідентифікації. Для підвищення безпеки трафіку та контролю доступу реалізовано багаторівневу аутентифікацію за токенами JWT із ролями операторів і адміністраторів, а також механізм аудиту, що зберігає у логу часові позначки й деталі запитів.

Інфраструктурний шар побудовано на контейнеризації всіх сервісів за допомогою Docker та організації кластеру через Docker Compose або Kubernetes. Це забезпечує відтворюваність середовища незалежно від хост-машини, дозволяє горизонтально масштабувати окремі підсистеми (наприклад, інференс-модуль або веб-інтерфейс) відповідно до навантаження, а також дає змогу здійснювати безперервне оновлення без зупинки всієї системи.

Нарешті, для підтримки якості коду і стабільності релізів впроваджено CI/CD-конвеєр на базі GitLab CI. На кожному етапі merge-request автоматично запускаються юніт-тести та інтеграційні тести, перевіряється коректність синтезованих зображень за метриками FID і IS, а після успішного проходження перевірок нова версія деплоїться в тестове середовище з подальшим релізом. Такий підхід дозволяє гарантувати цілісність й актуальність програмного комплексу в умовах безперервного розвитку та масштабування.

Основні модулі запропонованої інтелектуальної інформаційної технології були розроблені на мові програмування Python, яка відома своєю гнучкістю та широким вибором інструментів для роботи з даними та штучним інтелектом.

Використані фреймворки та бібліотеки для розробки інтелектуальної інформаційної технології:

1. PyTorch [116] — імперативний фреймворк для глибокого навчання з підтримкою автоматичної диференціації, який забезпечує гнучкість у побудові складних архітектур Re-ID, зокрема OSNet та його модифікацій, а також ефективно прискорення обчислень на GPU.

2. Torchreid [117] — спеціалізована бібліотека, що надає готові до використання моделі, датасети та інструменти для тренування й оцінки систем повторної ідентифікації, включаючи стандартизовані класифікатори, режими оцінки метрик mAP і Rank-1 та реалізації алгоритмів re-ranking.

3. MMRaid (частина OpenMMLab) [118] — компонент екосистеми OpenMMLab, що забезпечує єдині API для тренування, оцінки та зберігання моделей Re-ID, формалізовану конфігурацію експериментів та інтеграцію з іншими бібліотеками OpenMMLab для динамічної адаптації гіперпараметрів.

4. OpenCV [119] — набір інструментів для обробки зображень та відео, необхідний для завдань детекції та відсікання зображень людей у кадрі, корекції перспективи й освітлення перед подачею на вхід нейромережевим модулям.

5. NumPy [120] — фундаментальна бібліотека для наукових обчислень у Python; використовується для роботи з багатовимірними масивами ембеддінгів та проведення попередніх статистичних операцій у процесі тренування гіпермоделі.

6. scikit-learn [121] — бібліотека машинного навчання, що надає реалізації кластеризації, зниження розмірності ( t-SNE для візуалізації ембеддінгів) і стандартних метрик оцінки класифікації, які застосовуються для аналізу результатів Re-ID та вибору оптимальних порогів відбору.

7. Pandas [122] — це потужна бібліотека для роботи з табличними даними, яка надає інструменти для маніпулювання, очищення та аналізу великих наборів даних. Вона є важливою частиною робочого процесу при обробці даних, що використовуються для тренування моделей.

8. Matplotlib [123] — це бібліотека для візуалізації даних. Вона дозволяє створювати графіки та діаграми, що є важливим елементом аналізу результатів роботи моделей і демонстрації процесів.

9. TensorBoard [124] — інтерактивний набір інструментів для візуалізації експериментів машинного навчання, що дозволяє відстежувати та візуалізувати метрики (loss, accuracy), граф обчислень, гістограми тензорів, проєкції ембедінгів та відображати зображення й аудіо.

Таким чином, запропонована технологія базується на сучасних інструментах для роботи з даними та машинним навчанням, які дозволяють ефективно вирішувати задачі сегментації медичних зображень та обробки даних.

### 3.4 Алгоритм оцінювання ефективності запропонованої інформаційної технології

Алгоритм оцінювання ефективності запропонованої інформаційної технології включає шість послідовних етапів, котрі забезпечують комплексну перевірку системи з точки зору точності, стійкості та продуктивності в реальних умовах експлуатації.

Перший етап передбачає формулювання чітких критеріїв оцінювання. У число обраних метрик входять середня точність відтворення (mAP), коефіцієнти ранжування Rank-1, Rank-5 та Rank-10, а також часові характеристики обробки одного кадру (час відповіді) і пропускна здатність системи (кадри за секунду). Таке поєднання якості розпізнавання й продуктивності дозволяє всебічно охарактеризувати поведінку алгоритму як із позиції точності, так і з огляду на реальні ресурсні обмеження під час впровадження.

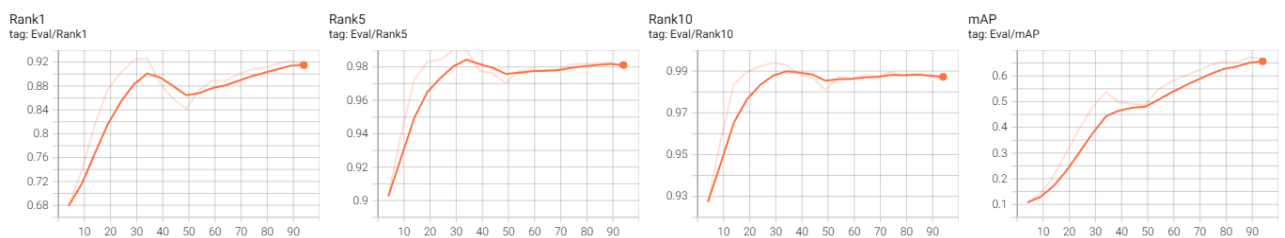


Рис. 3.5 Приклад оцінки моделей метрикою mAP та коефіцієнти ранжування Rank-1/5/10

Другий етап полягає у формуванні тестового набору даних на основі широко визнаних бенчмарків Market-1501 та DukeMTMC-reID. Ці датасети забезпечують різноманітність умов освітлення, ракурсів камери й рівнів шуму в кадрі, що є критично важливим для адекватної оцінки здатності системи до генералізації [1; 2]. До процедур попередньої обробки належать приведення зображень до розміру  $256 \times 128$  пікселів, нормалізація каналів за середнім та стандартним відхиленням, а також випадкове горизонтальне віддзеркалювання для розширення різноманітності тестових випадків. Уніфікація цих кроків гарантовано забезпечує відтворюваність результатів і можливість коректного порівняння з існуючими роботами.

Третій етап — апробація самої технології. Кожне зображення в тестовому наборі обробляється головною мережею та гіпермоделлю, після чого з векторних ембедінгів обчислюються відстані між запитами й усіма зразками в галереї. На основі цих відстаней формується ранжований список сусідів, за яким і обчислюються mAP та показники Rank-k. mAP відображає середню точність відтворення при різних граничних значеннях відсічення, а Rank-k характеризує частку запитів, для яких правильний результат потрапив до перших k позицій у списку [3]. Паралельно вимірюється середній час обробки одного кадру й загальна пропускна здатність системи в умовах пакетної та потокової обробки.

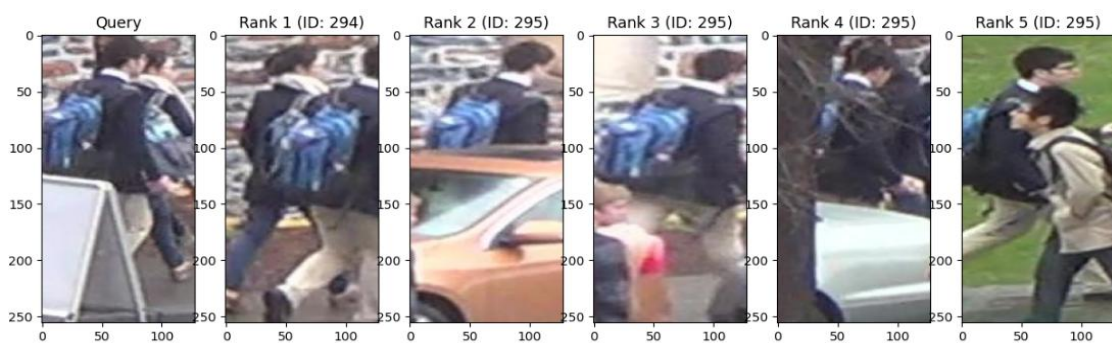


Рис. 3.6 Приклад формування списку запитів

Четвертий етап присвячено детальній статистичній обробці результатів. У якості основного інструмента перевірки гіпотез використовується

непараметричний критерій Вілкоксона для парних вибірок, який не вимагає припущення про нормальність розподілу різниць між попарно пов'язаними спостереженнями. Спочатку для кожної пари «запит–галерея» обчислюють різницю метрик (наприклад, mAP або Rank-1), після чого відсікають нульові різниці й ранжують абсолютні значення решти різниць за зростанням. Кожен ранг присвоюється позитивним або негативним різницям залежно від їхнього знака; сумування ранків з одного боку дає величину статистики  $W$ . Для невеликих вибірок точний розподіл  $W$  використовується для обчислення  $p$ -значення, а для більших—застосовується нормальне наближення з корекцією за випадокні зв'язки й нульові різниці [4]. Порівняння отриманого  $p$ -значення з обраним рівнем значущості (зазвичай  $\alpha = 0,05$ ) дозволяє зробити висновок про статистичну значущість відмінностей між двома методами. Крім того, для глибшого розуміння ефекту часто обчислюють ефектний розмір, що характеризує практичну вагомість поліпшень [5].

П'ятий етап зосереджений на оцінці ресурсоспоживання: проводиться профайлінг GPU/CPU із фіксацією пікових та середніх значень використання відеопам'яті, навантаження на обчислювальні ядра та часу простою. Це дає змогу оцінити, чи відповідає технологія вимогам реального часу на серверному обладнанні або чи потребує оптимізації для розгортання на менш потужних пристроях.

### **3.5 Висновки до розділу**

У третьому розділі було детально розглянуто алгоритмічні основи та програмну реалізацію інтелектуальної інформаційної технології повторної ідентифікації людей.

По-перше було послідовно обґрунтовано структурно-функціональну архітектуру системи та деталізовано кожний із її компонентів. Зокрема, сформульовано алгоритмічний конвеєр обробки відеопотоку — від адаптивної попередньої нормалізації й фільтрації шумів до екстракції інформативних ознак за допомогою OSNet із динамічно налаштованою гіпермоделлю. Для підвищення

точності і стійкості ідентифікації запропоновано механізм генерації синтетичних псевдомасок із використанням генеративних змагальних мереж, що дозволило розширити тренувальні вибірки без залучення ручної анотації.

По-друге, створені UML-діаграми програмного забезпечення наочно демонструє основні модулі системи, їх взаємодію, а також потік даних у процесі повторної ідентифікації.

По-третє, детально описано структуру програмної реалізації: розроблено UML-діаграми діяльності, послідовності та компонентів, що формалізують взаємодію веб-інтерфейсу, API-шару, обчислювального сервера (Re-ID Engine), підсистеми зберігання даних і сервісів оркестрації. Виконано реалізацію на базі фреймворків PyTorch для моделювання нейронної мережі, Flask для побудови REST-інтерфейсу та PostgreSQL у хмарному середовищі для збереження профілів користувачів і ембеддингів.

Результати розділу опубліковано в таких роботах автора: [14], [15], [16], [17], [18].

## РОЗДІЛ 4

# ПРИКЛАДНЕ ЗАСТОСУВАННЯ РОЗРОБЛЕНОЇ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ПОВТОРНОЇ ІДЕНТИФІКАЦІЇ ЛЮДЕЙ

### 4.1 Реалізація комплексу програм для навчання базової мережі та гіпермоделі

Для частини модулів, а саме:

- модуль навчання базової моделі;
- модуль навчання базової моделі із поєднанням гіпермоделі;
- модуль навчання генеративно-змагальної мережі;
- модуль візуалізації результатів.

Реалізація відбувається у вигляді програм-ноутбуків за допомогою Jupyter Notebook [106].

Використання середовища Jupyter Notebook надає численні переваги для проведення досліджень та розробки моделей:

- Інтерактивність та гнучкість: Jupyter Notebook дозволяє виконувати код покроково, розбиваючи його на окремі комірки, кожна з яких може містити як код на Python, так і текстовий опис, оформлений у Markdown. Це полегшує процес аналізу результатів і тестування гіпотез.

- Підтримка потужних обчислювальних ресурсів: Jupyter Notebook може бути розгорнутий на серверах, оснащених високопродуктивними графічними прискорювачами, наприклад сімейство відеокарт Nvidia [108], що забезпечує достатню обчислювальну потужність для тренування великих нейронних моделей і складних обчислень.

- Інтеграція з наборами даних: Через Jupyter Notebook зручно здійснювати завантаження та обробку наборів даних, використовуючи команди мови програмування (Python, R тощо). Це дозволяє легко працювати як з локальними даними, так і з зовнішніми ресурсами

– Зручне керування кодом та документування результатів: Вбудовані можливості для документування процесу роботи, включаючи вставку формул, діаграм та зображень, дозволяють використовувати Notebook як повноцінне середовище для дослідження, демонстрації результатів та створення звітів.

– Легкість поширення результатів: Створені Notebook-файли можна легко зберігати та поширювати у форматах .ipynb, HTML або PDF, що сприяє швидкому обміну результатами та співпраці з іншими користувачами.

Незважаючи на те, що реалізація цих модулів була проведена в середовищі Jupyter Notebook, вони легко можуть бути портовані в інші середовища, такі як:

– AWS SageMaker [125] – це масштабована хмарна платформа від Amazon Web Services, призначена спеціально для розробки, тренування, тестування і розгортання моделей машинного навчання. Вона дозволяє використовувати широкий спектр потужностей, таких як CPU, GPU та навіть спеціалізовані процесори, а також підтримує різні фреймворки (TensorFlow, PyTorch, MXNet тощо), надаючи зручні інструменти для автоматизації всього життєвого циклу моделі.

– Google Colab [126] – це безкоштовна хмарна платформа на базі Jupyter Notebook від компанії Google. Colab забезпечує інтеграцію з хмарними ресурсами Google та дозволяє легко працювати з GPU та TPU, значно скорочуючи час обчислень при роботі з глибоким навчанням. Платформа особливо зручна для швидкого прототипування та виконання дослідницьких проєктів.

– Kaggle [127] – популярна платформа, орієнтована на проведення змагань з аналізу даних і машинного навчання. Окрім змагань, вона пропонує доступ до великої кількості публічних та приватних наборів даних, забезпечує інтерактивне середовище на основі Jupyter Notebook та потужності GPU. Kaggle ідеально підходить для навчання, експериментування, а також швидкого прототипування моделей машинного навчання.

– Microsoft Azure Notebooks [128] – хмарний сервіс від компанії Microsoft, що забезпечує можливість створювати та виконувати Jupyter Notebook

безпосередньо в середовищі Azure. Ця платформа дозволяє легко інтегруватися з іншими продуктами Microsoft Azure, використовувати різні мови програмування, включаючи Python, R і F#, та забезпечує зручне середовище для колективної роботи над проектами.

– PyCharm [129] – інтегроване середовище розробки (IDE), створене компанією JetBrains, яке призначене для професійного програмування на Python. PyCharm пропонує потужні інструменти для редагування коду, автодоповнення, інтегрованого налагодження, тестування, керування версіями та підтримує Jupyter Notebook. Це дозволяє комфортно вести розробку складних проектів, зручно переходити між локальними та хмарними середовищами, забезпечуючи при цьому глибоку інтеграцію з екосистемою Python.

Для реалізації комплексу програма була використана мова програмування Python. Вибір саме цієї мови був аргументований низкою її важливих переваг:

– Простота у вивченні та використанні: Python має простий, лаконічний та зрозумілий синтаксис, що робить її зручною для освоєння початківцями, але водночас вона є потужним інструментом для досвідчених розробників.

– Інтерпретованість: Python є інтерпретованою мовою, яка не вимагає попередньої компіляції. Код виконується послідовно рядок за рядком, що пришвидшує процес тестування і налагодження програм.

– Динамічна типізація: Типи даних визначаються безпосередньо під час виконання програми, що дозволяє писати більш гнучкий та компактний код.

– Кросплатформність: Python легко виконується на різних операційних системах, таких як Windows, Linux, macOS, що значно спрощує процес розгортання програм на різних платформах.

– Розвинена стандартна бібліотека: Python постачається з потужною і широкою стандартною бібліотекою, яка забезпечує інструменти для роботи з файлами, мережами, базами даних, веб-розробкою, створенням графічних інтерфейсів та багатьох інших задач.

– Широкі можливості для наукових обчислень і машинного навчання: Завдяки таким бібліотекам, як NumPy, SciPy, Pandas, Matplotlib, TensorFlow, Keras та PyTorch, Python став однією з найпоширеніших мов для аналізу даних, наукових досліджень, машинного навчання та розробки систем штучного інтелекту.

– Підтримка об'єктно-орієнтованого програмування (ООП): Python дозволяє створювати складні структуровані програми за допомогою класів, об'єктів та інших концепцій ООП, що забезпечує кращу читабельність та простоту супроводу коду.

– Активна спільнота і розвинена екосистема: Python має одну з найбільших і найактивніших спільнот серед мов програмування. Це гарантує доступ до величезної кількості ресурсів, бібліотек, фреймворків, а також швидку та ефективну підтримку.

Всі модулі мають прозору ООП структуру і можуть легко бути адаптовані до нових вимог чи наборів даних. Розглянемо основні сутності розроблених модулів.

Модуль тренування базової моделі та модуль тренування базової моделі із поєднанням гіпермоделі для вирішення задачі повторної ідентифікації особи (person re-identification) складаються з набору спеціалізованих класів, що забезпечують ефективне навчання нейронних мереж для розпізнавання й зіставлення зображень однієї й тієї ж особи з різних камер спостереження. Ці модулі спрямовані на вирішення таких задач, як вилучення інформативних ознак, генерація стабільних ембедингів, зменшення доменного розриву між наборами даних, адаптація моделі до змінних умов спостереження (різні ракурси, освітлення, якість зображень) і покращення узагальнювальних властивостей мережі завдяки динамічному налаштуванню параметрів, що забезпечується застосуванням гіпермоделі.

Клас CustomOSNetWithSpatialFeatures, що бере функціональність від `torch.nn.Module`, необхідний для побудови архітектури базової моделі повторної

ідентифікації, що реалізує основні методи, які описані в розділі 2.1 складається з таких методів:

- `__init__` – ініціалізує модель, приймає як параметри базову архітектуру (`backbone`), розмір вектора ознак (`emb_dim`), кількість класів (`num_classes`) і параметр `use_bnneck` для застосування `batch normalization`. Ініціалізує додаткові шари уваги (`AttentionLayer`), пулінгу та ембедингів (`pool_embed`), а також класифікатор.

- `forward` – виконує пряме проходження даних крізь мережу, обчислює просторові ознаки, ембединг, застосовує `BNNeck` (за потреби) та генерує прогнози класифікатора.

```

1 class CustomOSNetWithSpatialFeatures(nn.Module):
2     def __init__(
3         self,
4         backbone,
5         emb_dim,
6         num_classes,
7         use_bnneck=False
8     ):
9         super().__init__()
10        backbone.classifier = nn.Identity()
11        self.backbone = backbone
12        self.attn = AttentionLayer(
13            backbone.feature_dim
14        )
15        self.pool_embed = nn.Sequential(
16            nn.AdaptiveAvgPool2d((1,1)),
17            nn.Flatten(),
18            nn.Linear(
19                backbone.feature_dim,
20                emb_dim
21            ),
22            nn.ReLU()
23        )
24        self.use_bnneck = use_bnneck
25        if use_bnneck:
26            self.bnneck = BNNeck(emb_dim)
27        self.classifier = nn.Linear(
28            emb_dim,
29            num_classes
30        )
31
32    def forward(self, x, return_spatial=False):
33        # Backbone layers
34        f = self.backbone.conv1(x)
35        f = self.backbone.maxpool(f)
36        f = self.backbone.conv2(f)
37        for layer in [
38            'pool12', 'conv3',

```

Рис. 4.1 Реалізація класу `CustomOSNetWithSpatialFeatures`

Клас `EnhancedHyperNetworkReID`, що бере функціональність від `torch.nn.Module`, необхідний для динамічної адаптації параметрів основної мережі шляхом використання додаткових ознак (гіпермодель), які описані в розділі 2.2 складається з таких методів:

- `__init__` – ініціалізує гіпермодель, приймає як параметри розмірність вхідних ознак (`in_dim`), вихідну розмірність (`out_dim`), структуру прихованих шарів (`hidden`) і тип нормалізації (`norm`). Створює послідовність шарів, які використовуються для генерації адаптивних параметрів.
- `forward` – виконує пряме проходження даних крізь гіпермодель, застосовує нормалізацію та `dropout`, повертає згенеровані параметри для адаптації базової моделі.

```

1 class EnhancedHyperNetworkReID(nn.Module):
2     def __init__(
3         self,
4         in_dim,
5         out_dim,
6         hidden=(512,256),
7         norm='ln'
8     ):
9         super().__init__()
10        layers = []
11        prev = in_dim
12        for h in hidden:
13            layers.append(
14                nn.Linear(prev, h)
15            )
16            if norm == 'ln':
17                layers.append(
18                    nn.LayerNorm(h)
19                )
20            else:
21                layers.append(
22                    nn.BatchNorm1d(h)
23                )
24            layers.append(nn.ReLU())
25            prev = h
26        layers.append(
27            nn.Linear(prev, out_dim)
28        )
29        self.net = nn.Sequential(*layers)
30
31    def forward(self, x):
32        x = F.normalize(x, p=2, dim=1)
33        x = F.dropout(x, p=0.1, training=self.training)
34        return self.net(x)

```

Рис. 4.2 Реалізація класу `EnhancedHyperNetworkReID`

Клас `ArcFaceLoss`, що бере функціональність від `torch.nn.Module`, необхідний для обчислення функції втрат ArcFace, яка забезпечує кращий розподіл класів у просторі ознак, складається з таких методів:

- `__init__` – ініціалізує параметри функції ArcFace, приймаючи розмірності входу (`in_features`), виходу (`out_features`), параметри масштабу (`scale`) і маржі (`margin`). Створює вагові параметри для класифікації.
- `forward` – обчислює значення ArcFace-втрат на основі ембедингів та істинних міток.

```

1 class ArcFaceLoss(nn.Module):
2     def __init__(
3         self,
4         in_features,
5         out_features,
6         scale=15.0,
7         margin=0.2
8     ):
9         super().__init__()
10        self.scale = scale
11        self.margin = margin
12        self.weight = nn.Parameter(
13            torch.FloatTensor(out_features, in_features)
14        )
15        nn.init.kaiming_uniform_(
16            self.weight,
17            a=5*0.5
18        )
19
20    def forward(self, embeddings, labels):
21        device = embeddings.device
22        W = F.normalize(
23            self.weight,
24            p=2,
25            dim=1
26        ).to(device)
27        X = F.normalize(
28            embeddings,
29            p=2,
30            dim=1
31        )
32        cos_t = F.linear(X, W).clamp(-1, 1)
33        theta = torch.acos(cos_t)
34        theta_m = theta + self.margin
35        one_hot = F.one_hot(
36            labels,
37            num_classes=W.size(0)
38        ).float().to(device)

```

Рис. 4.3 Реалізація класу `ArcFaceLoss`

Клас `CenterLoss`, що бере функціональність від `torch.nn.Module`, необхідний для додаткової регуляризації ознак, складається з таких методів:

- `__init__` – ініціалізує центри класів, приймаючи як параметри кількість класів (`num_classes`) та розмірність простору ознак (`feat_dim`).
- `forward` – обчислює значення втрат за відстанню між ембедингами та відповідними центрами класів.

```

1 class CenterLoss(nn.Module):
2     def __init__(
3         self,
4         num_classes,
5         feat_dim,
6         device='cuda'
7     ):
8         super().__init__()
9         self.centers = nn.Parameter(
10            torch.randn(num_classes, feat_dim).to(device)
11        )
12
13    def forward(self, features, labels):
14        dist_mat = torch.cdist(features, self.centers)
15        return dist_mat.gather(
16            1,
17            labels.unsqueeze(1)
18        ).mean()

```

Рис. 4.4 Реалізація класу CenterLoss

Клас PKSampler, що бере функціональність від `torch.utils.data.Sampler`, необхідний для формування вибірки тренувальних батчів відповідно до Р-К стратегії (Р класів, К зображень на клас), складається з таких методів:

- `__init__` – ініціалізує семплер, приймає набір даних (`dataset`), кількість класів у батчі (`p`) та кількість зображень на клас (`k`). Готує мапу індексів для формування батчів.
- `__iter__` – формує батчі відповідно до стратегії РК, повертаючи послідовність індексів вибраних зображень.
- `__len__` – повертає кількість батчів у вибірці.

```

1 class PKSampler(torch.utils.data.Sampler):
2     def __init__(
3         self,
4         dataset,
5         p=8,
6         k=4
7     ):
8         self.labels = np.array(dataset.labels)
9         self.p = p
10        self.k = k
11        self.idx_by_label = {}
12        for i, l in enumerate(self.labels):
13            self.idx_by_label.setdefault(l, []).append(i)
14        self.unique = list(self.idx_by_label)
15        self.num_samples = p * k
16        self.all_indices = list(range(len(dataset)))
17
18    def __iter__(self):
19        random.shuffle(self.unique)
20        batch = []
21        for lbl in self.unique:
22            inds = self.idx_by_label[lbl]
23            if len(inds) >= self.k:
24                chosen = random.sample(inds, self.k)
25            else:
26                chosen = random.choices(inds, k=self.k)
27            batch.extend(chosen)
28            if len(batch) == self.num_samples:
29                yield batch
30                batch = []
31        if batch:
32            needed = self.num_samples - len(batch)
33            batch.extend(random.choices(self.all_indices, k=needed))
34            yield batch
35
36    def __len__(self):
37        return len(self.unique) // self.p

```

Рис. 4.5 Реалізація класу PKSampler

Клас WarmupScheduler, що бере функціональність від torch.optim.lr\_scheduler.\_LRScheduler, необхідний для поступового підвищення швидкості навчання на початкових етапах тренування, складається з таких методів:

- `__init__` – ініціалізує планувальник із заданими параметрами: оптимізатором (`optimizer`), кількістю епох прогрівання (`warmup_epochs`), початковим (`base_lr`) та кінцевим (`final_lr`) значеннями швидкості навчання.
- `get_lr` – обчислює поточне значення швидкості навчання залежно від етапу прогрівання.

```

1 class WarmupScheduler(torch.optim.lr_scheduler._LRScheduler):
2     def __init__(
3         self,
4         optimizer,
5         warmup_epochs,
6         base_lr,
7         final_lr,
8         last_epoch=-1
9     ):
10        self.warmup_epochs = warmup_epochs
11        self.base_lr = base_lr
12        self.final_lr = final_lr
13        super().__init__(optimizer, last_epoch)
14
15    def get_lr(self):
16        if self.last_epoch >= self.warmup_epochs:
17            return [self.final_lr for _ in self.base_lrs]
18        alpha = self.last_epoch / float(self.warmup_epochs)
19        lr = self.base_lr + alpha * (self.final_lr - self.base_lr)
20        return [lr for lr in self.base_lrs]

```

Рис. 4.6 Реалізація класу WarmupScheduler

Клас AttentionLayer, що бере функціональність від torch.nn.Module, необхідний для застосування механізму уваги до просторових ознак, складається з таких методів:

- `__init__` – створює шари згортки та активації для розрахунку маски уваги.
- `forward` – виконує обчислення маски уваги та її застосування до просторових ознак.

```

1 class AttentionLayer(nn.Module):
2     def __init__(self, dim):
3         super().__init__()
4         self.attn = nn.Sequential(
5             nn.Conv2d(dim, dim, 1),
6             nn.ReLU(),
7             nn.Conv2d(dim, 1, 1),
8             nn.Sigmoid()
9         )
10
11    def forward(self, x):
12        return x * self.attn(x)

```

Рис. 4.7 Реалізація класу AttentionLayer

Клас BNNeck, що бере функціональність від `torch.nn.Module`, необхідний для нормалізації ембедингів перед класифікацією, складається з таких методів:

- `__init__` – створює шар пакетної нормалізації (batch normalization).
- `forward` – виконує нормалізацію ембедингів та повертає нормалізований вектор ознак.

```

1 class BNNeck(nn.Module):
2     def __init__(self, feat_dim):
3         super().__init__()
4         self.bn = nn.BatchNorm1d(feat_dim)
5         self.bn.bias.requires_grad_(False)
6
7     def forward(self, x):
8         return self.bn(x)

```

Рис. 4.8 Реалізація класу BNNeck

Після опису ключових компонентів, які відповідають за побудову та тренування самої моделі (класи `CustomOSNetWithSpatialFeatures` та `EnhancedHyperNetworkReID`, функції втрат `ArcFaceLoss` і `CenterLoss`, а також планувальник `WarmupScheduler`), варто звернути увагу на допоміжні модулі, що доповнюють і розширюють тренувальний пайплайн. До них належать класи для підготовки даних і формування батчів (`MarketDataset`, `PKSampler`), утиліти для обчислення додаткових втрат і метрик (`batch_hard_triplet`, `compute_dist_matrix`, `compute_mapcmc_minpr`, `evaluate_model`), а також функції для логування результатів і візуалізації (`log_retrieval_examples`) та інші планувальники навчання (`OneCycleLR`, `CosineAnnealingLR` тощо). Розглянемо кожний із них детальніше.

Функція `log_retrieval_examples`, необхідна для візуального контролю результатів пошуку під час тренування та оцінки, складається з таких кроків:

- Завантаження і трансформація зображень запитів (`query_paths`) та відповідних «галерейних» зображень (`gallery_paths`) за індексами `topk_idx`.
- Формування єдиного тензора `combined` із одного запиту та його топ-К відповідей.

- Створення сітки зображень за допомогою `torchvision.utils.make_grid` і передача її до TensorBoard через `writer.add_image`.

Функція `batch_hard_triplet`, необхідна для обчислення «Batch-Hard Triplet Loss», складається з таких етапів:

- Нормалізація ембедингів і обчислення попарних відстаней  $\text{dist} = 1 - \text{cosine\_similarity}$ .
- Визначення найбільш віддаленого позитивного прикладу (`dist_ap`) та найближчого негативного (`dist_an`) у кожному батчі.
- Обчислення остаточної втрати як  $\text{ReLU}(\text{dist\_ap} + \text{margin} - \text{dist\_an}).\text{mean}()$ .

Функція `compute_dist_matrix`, необхідна для обчислення матриці евклідових відстаней між двома наборами векторів, складається з:

- Перетворення вхідних масивів `x`, `y` з NumPy у `torch.Tensor`.
- Обчислення квадратних норм `xx`, `yy` та попарного скалярного добутку.
- Формування матриці відстаней  $\sqrt{xx + yy - 2 \cdot x \cdot y^T}$  із захистом від від'ємних чисел.

Функція `compute_mapcmc_minp`, необхідна для підрахунку ключових метрик Re-ID (`mAP`, `CMC`, `mINP`), складається з:

- Сортування кожного рядка матриці відстаней, побудови бінарного масиву збігів.
- Накопичення CMC-кривої шляхом підсумовування перших появ позитивних прикладів.
- Обчислення середньої точності (`AP`) і `normalized INP` для кожного запиту та усереднення результатів.

Функція `evaluate_model`, необхідна для комплексної оцінки Re-ID моделі з урахуванням `reranking`, складається з:

- Виклику `compute_dist_matrix` для пар «запити ↔ галерея», «запити ↔ запити», «галерея ↔ галерея».

- Застосування алгоритму `re_ranking` ( $k_1=20$ ,  $k_2=6$ ,  $\lambda=0.3$ ) для поліпшення розташування позитивів.

- Передачі отриманої матриці у `compute_map_cmc_minr` для фінального підрахунку mAP, CMC та mINP.

Клас `Dataset`, що відповідає за завантаження, попередню обробку та надання зразків із датасетів, наприклад `Market-1501`, складається з наступних методів:

- `__init__(self, root_dir, mode='train', transform=None)` – приймає кореневий каталог із підпапками `bounding_box_train`, `query`, `bounding_box_test`, режим роботи ('train', 'query' або 'gallery') та набір перетворень; формує список усіх шляхів до зображень, витягує з імені файлу ідентифікатор особи та, для режиму 'train', виконує ремаппінг цих ідентифікаторів у компактний індексний діапазон.

- `__len__(self)` – повертає загальну кількість зразків у датасеті (розмір сформованого списку шляхів).

- `__getitem__(self, idx)` – за індексом `idx` завантажує відповідне зображення, застосовує задані перетворення та повертає кортеж (`image_tensor`, `label`), де `label` — цілочисловий індекс класу, прив'язаний до конкретного `PersonID`.

Клас `Train`, що відповідає за організацію повного циклу тренування та валідації моделі (базової та з гіпермоделлю), складається з таких методів:

- `__init__` – ініціалізує всі необхідні компоненти тренувального пайплайну: передає посилання на базову модель і гіпермодель, налаштовує `train_loader`, `query_loader`, `gallery_loader`, оптимізатор, планувальник швидкості навчання, функції втрат, `TensorBoard`-записувач (`SummaryWriter`) і словник конфігурації.

- `fit` – запускає послідовність епох тренування та валідації згідно з параметром `config['epochs']`, автоматично викликаючи `train_epoch` та `validate_epoch` і відслідковуючи найкращі результати.

– `train_epoch` – виконує одну епоху навчання: обходить батчі з `train_loader`, проводить прямий і зворотній проходи (модель + гіпермодель), обчислює комбіновану втрату, оновлює ваги через `optimizer.step()` і крок планувальника LR.

– `validate_epoch` – запускає один цикл валідації: збирає ембедингів запитів і галереї, обчислює метрики (mAP, Rank-1/5/10, mINP) за допомогою `evaluate_model`, логування в TensorBoard і, за потреби, збереження прикладів пошуку через `log_retrieval_examples`.

– `save_checkpoint` – зберігає поточні ваги моделі й гіпермоделі у файл(и) (`last_model.pth`, `best_model.pth`), залежно від того, чи покращилася метрика валідації.

Усі ці компоненти разом забезпечують повноцінний, масштабований і легко налаштований конвеєр для тренування та валідації моделей повторної ідентифікації з використанням динамічної адаптації гіпермоделі.

Таблиця 4.1 - Результати порівняння базової моделі та базової моделі із використанням гіпермоделі на датасеті Market-1501

Run	OSNet, mAP	HN, mAP	$\Delta$ , mAP	OSNet, R1	HN, R1	$\Delta$ , R1
1	85.05	87.58	2.53	94.88	96.1	1.23
2	84.86	87.51	2.65	94.92	96.17	1.25
3	85.09	87.75	2.66	95.04	96.41	1.37
4	85.36	87.96	2.6	95.03	96.39	1.35
5	84.83	87.51	2.68	94.5	95.8	1.3
6	84.83	87.58	2.75	94.59	95.9	1.31
7	85.37	88.3	2.93	94.91	96.33	1.41
8	85.13	87.85	2.72	94.91	96.16	1.25
9	84.76	87.49	2.73	94.91	96.26	1.35
10	85.06	87.75	2.69	95.65	96.93	1.28

Таблиця 4.2 - Результати порівняння базової моделі та базової моделі із використанням гірепмоделі на датасеті DukeMTMC.

Run	OSNet, mAP	HN, mAP	$\Delta$ , mAP	OSNet, R1	HN, R1	$\Delta$ , R1
1	62.08	65.87	3.79	81.96	84.19	2.22
2	62.53	66.51	3.98	81.38	83.83	2.46
3	62.8	66.12	3.32	82.0	84.2	2.19
4	62.82	66.53	3.71	82.17	84.26	2.09
5	62.6	66.07	3.47	81.84	83.91	2.07
6	63.08	66.58	3.5	82.36	84.75	2.39
7	62.32	65.8	3.48	81.43	83.54	2.11
8	62.74	66.16	3.43	81.26	83.83	2.56
9	62.85	66.46	3.61	81.08	83.07	1.99
10	63.03	66.47	3.43	82.22	84.78	2.55

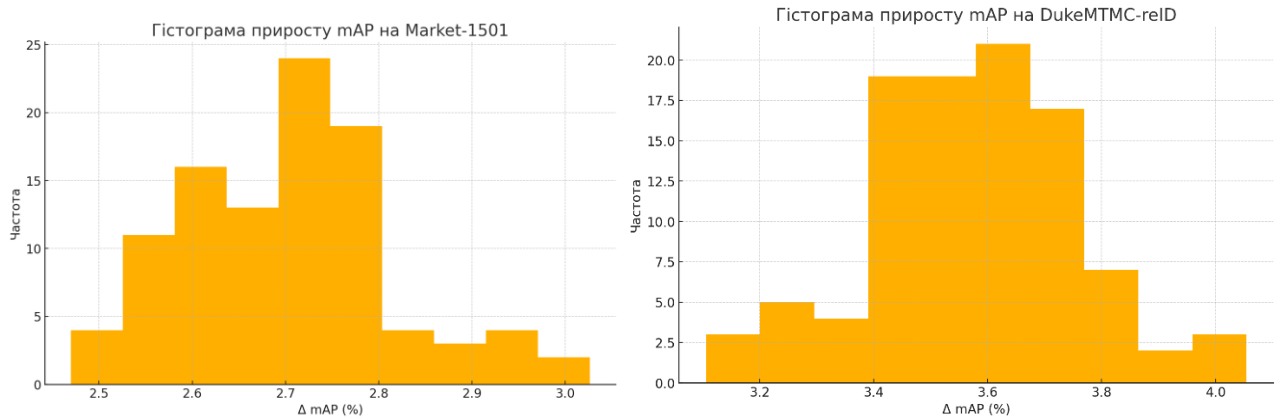


Рис. 4.9 Розподіл приросту mAP для Market-1501 та DukeMTMC

Модуль тренування генеративно-змагальної мережі, що описаний в розділі 2.3 складається з ансамблю взаємопов'язаних класів, які реалізують циклічний процес синтезу та оцінки зображень із супервайзингом на рівні якості, класифікації й ідентичності особи. Основні компоненти включають генератор із блоками адаптивної інстанс-нормалізації, що формують стилістичні корективи

на основі просторових ознак, дискримінатор із R1-регуляризацією для стабілізації навчання, а також класифікатор OSNet для введення додаткового сигналу класифікаційного та identity-loss. Цей модуль відповідає за одночасну мінімізацію adversarial-втрат, cross-entropy втрат класифікації та L1-втрат ідентичності, забезпечує прогресивне підключення дискримінатора після попередньої фази «розігріву» генератора та фіксацію ваг за необхідності, а також автоматичне логування в TensorBoard з можливістю візуалізації проміжних результатів. Загальна мета модуля — генерація високоякісних, семантично коректних синтетичних зображень осіб для подальшого збагачення тренувальних наборів і зменшення доменного розриву між різними умовами спостереження.

Клас Generator, що наслідує від torch.nn.Module, необхідний для генерації синтетичних зображень на основі вхідного зображення та стилю, складається з таких методів:

- `__init__(self, style_dim=512)` – будує початковий шар («init»): паддінг, згортка, інстанс-нормалізація, ReLU; далі два даунсемплінги; окремий енкодер стилю (AdaptiveAvgPool2d + Flatten); шість блоків ResAdaINBlock; два апсемплінги з конволюцією та інстанс-нормалізацією; фінальний шар з ReflectionPad2d, згорткою і Tanh.

- `forward (self, x, z=None)` – проходить вхідне зображення x через початковий конволюційний блок, кодує стиль із отриманих ознак, послідовно пропускає їх через шість ResAdaINBlock, виконує апсемплінг і повертає згенероване зображення.

```

1 class Generator(nn.Module):
2     def __init__(self, style_dim=512):
3         super().__init__()
4         c0 = 64
5         layers = [
6             nn.ReflectionPad2d(3),
7             nn.Conv2d(3, c0, 7),
8             nn.InstanceNorm2d(c0),
9             nn.ReLU(True)
10        ]
11        c = c0
12        for _ in range(2):
13            layers += [
14                nn.Conv2d(c, c*2, 4, 2, 1),
15                nn.InstanceNorm2d(c*2),
16                nn.ReLU(True)
17            ]
18            c *= 2
19        self.init = nn.Sequential(*layers)
20        self.style_enc = nn.Sequential(
21            nn.AdaptiveAvgPool2d((1,1)),
22            nn.Flatten()
23        )
24        self.res_blocks = nn.ModuleList([ResAdaINBlock(c) for _ in range(6)])
25        up = []
26        for _ in range(2):
27            up += [
28                nn.Upsample(scale_factor=2, mode='nearest'),
29                nn.Conv2d(c, c//2, 5, 1, 2),
30                nn.InstanceNorm2d(c//2),
31                nn.ReLU(True)
32            ]
33            c //= 2
34        self.up = nn.Sequential(*up)
35        self.final = nn.Sequential(
36            nn.ReflectionPad2d(3),
37            nn.Conv2d(c0, 3, 7),
38            nn.Tanh()

```

Рис. 4.10 Реалізація методу Generator

Клас `Discriminator`, що наслідує від `torch.nn.Module`, необхідний для оцінки реалістичності згенерованих зображень, складається з таких методів:

- `__init__(self)` – створює послідовність згорткових шарів (`Conv2d`) з кроком 2 для даунсемплінгу, інстанс-нормалізацією та `LeakyReLU`, і завершує одним шаром, що виводить предикти для кожного просторового патчу.

- `forward(self, x)` – пропускає зображення `x` через мережу та повертає вектор оцінок (`view(batch_size, -1)`).

```

1 class Discriminator(nn.Module):
2     def __init__(self):
3         super().__init__()
4         self.model = nn.Sequential(
5             nn.Conv2d(3, 64, 4, 2, 1), nn.LeakyReLU(0.2, True),
6             nn.Conv2d(64, 128, 4, 2, 1), nn.InstanceNorm2d(128), nn.LeakyReLU(0.2, True),
7             nn.Conv2d(128, 256, 4, 2, 1), nn.InstanceNorm2d(256), nn.LeakyReLU(0.2, True),
8             nn.Conv2d(256, 512, 4, 1, 1), nn.InstanceNorm2d(512), nn.LeakyReLU(0.2, True),
9             nn.Conv2d(512, 1, 4, 1, 1)
10        )
11    def forward(self, x):
12        return self.model(x).view(x.size(0), -1)

```

Рис. 4.11 Реалізація методу Discriminator

Клас AdaIN, що наслідує від `torch.nn.Module`, необхідний для адаптивної нормалізації просторових ознак за стилем, складається з таких методів:

- `__init__(self, channels)` – створює шар інстанс-нормалізації (`InstanceNorm2d`) без афінних параметрів і лінійний шар (`Linear`) розмірності  $\text{channels} \rightarrow \text{channels} * 2$  для генерації коефіцієнтів  $\gamma$  та  $\beta$ .
- `forward(self, x, style)` – приймає вхідні ознаки  $x$  і вектор стилю `style`; через лінійний шар формує `gamma_beta`, розділяє його на  $\gamma$  та  $\beta$ , нормалізує  $x$  і повертає  $\gamma * \text{norm}(x) + \beta$ .

```

1 class AdaIN(nn.Module):
2     def __init__(self, channels):
3         super().__init__()
4         self.norm = nn.InstanceNorm2d(channels, affine=False)
5         self.fc = nn.Linear(channels, channels * 2)
6     def forward(self, x, style):
7         gamma_beta = self.fc(style)
8         gamma, beta = gamma_beta.chunk(2, 1)
9         gamma = gamma.view(-1, x.size(1), 1, 1)
10        beta = beta.view(-1, x.size(1), 1, 1)
11        out = self.norm(x)
12        return gamma * out + beta

```

Рис. 4.12 Реалізація класу AdaIN

Клас `ResAdaINBlock`, що наслідує від `torch.nn.Module`, необхідний для реалізації залишкового блоку з адаптивною нормалізацією, складається з таких методів:

- `__init__(self, channels)` – створює дві послідовності: кожна містить згортку (`Conv2d`), адаптивну інстанс-нормалізацію (`AdaIN`) і `ReLU` (для першого шару).
- `forward(self, x, style)` – застосовує послідовність згорток та `AdaIN` до вхідного тензора `x` із врахуванням `style`, додає початковий `x` (`residual`), повертаючи результат  $x + \text{block\_out}$ .

```

1 class ResAdaINBlock(nn.Module):
2     def __init__(self, channels):
3         super().__init__()
4         self.conv1 = nn.Conv2d(channels, channels, 3, padding=1)
5         self.adain1 = AdaIN(channels)
6         self.relu = nn.ReLU(True)
7         self.conv2 = nn.Conv2d(channels, channels, 3, padding=1)
8         self.adain2 = AdaIN(channels)
9     def forward(self, x, style):
10        out = self.conv1(x)
11        out = self.adain1(out, style)
12        out = self.relu(out)
13        out = self.conv2(out)
14        out = self.adain2(out, style)
15        return x + out

```

Рис. 4.13 Реалізація класу `ResAdaINBlock`

Клас `Dataset`, що наслідує від `torch.utils.data.Dataset`, необхідний для завантаження й підготовки даних із датасетів, складається з таких методів:

- `__init__(self, root_dir, transform=None)` – ініціалізує об’єкт датасету, приймаючи шлях до теки зображень (`root_dir`) та набір перетворень (`transform`); зчитує імена файлів `.jpg`, витягує з них `PersonID` і побудовує список кортежів (`fname, pid`); створює відображення `pid2label` для перенумерації `ID` у компактний індекс.
- `__len__(self)` – повертає загальну кількість зразків у датасеті.
- `__getitem__(self, idx)` – за індексом `idx` завантажує зображення (`PIL.Image`), застосовує перетворення (`transform`) і повертає тензор зображення та відповідну числову мітку (`label`).

```

1 class Dataset(Dataset):
2     def __init__(self, root_dir, transform=None):
3         self.root_dir = root_dir
4         self.transform = transform
5         self.samples = []
6         for fname in os.listdir(root_dir):
7             if fname.lower().endswith('.jpg'):
8                 pid = int(fname.split('_')[0])
9                 if pid != -1:
10                    self.samples.append((fname, pid))
11         pids = sorted({pid for _, pid in self.samples})
12         self.pid2label = {pid:i for i, pid in enumerate(pids)}
13     def __len__(self):
14         return len(self.samples)
15     def __getitem__(self, idx):
16         fname, pid = self.samples[idx]
17         img = Image.open(os.path.join(self.root_dir, fname)).convert('RGB')
18         if self.transform:
19             img = self.transform(img)
20         return img, self.pid2label[pid]

```

Рис. 4.14 Реалізація класу Dataset

Основний цикл навчання генеративно-змагальної мережі організовано як чергування трьох фаз: оновлення дискримінатора, оновлення генератора та оновлення класифікатора. На початку кожної ітерації зчитуються батчі реальних зображень та відповідних міток із датасету, після чого генератор продукує синтетичні приклади на основі випадкового шуму чи стилізованих ознак. Цей процес поєднує в собі як традиційну adversarial-складову, так і додаткові terms — classification loss для забезпечення семантичної точності та identity loss для збереження візуальної подібності до вхідних зразків.

У фазі оновлення дискримінатора використовується adversarial-втрата (BCEWithLogitsLoss) для реальних і синтетичних зображень, додатково доповнена R1-регуляризацією на градієнтах виходу по вхідних тензорах. Такий підхід дозволяє запобігти нестабільності під час тренування та зменшити ймовірність mode collapse. Після кількох початкових епох, коли дискримінатор поступово «розкриває» свою потужність, його ваги заморожуються на певний період, що забезпечує генератору стабільніший сигнал зворотного зв'язку.

Фаза оновлення генератора спрямована на мінімізацію комбінованої функції втрат: adversarial loss, яка змушує генератор обманювати дискримінатор; classification loss, що забезпечує правильну ідентифікацію особи згенерованого зображення; і identity loss, що відповідає за збереження структурно-детальних

особливостей. Завершальний крок кожної ітерації — оновлення класифікатора (OSNet) на реальних зображеннях із метою підтримки консистентності навчання між усіма трьома мережами. Усі ключові метрики й зображення проміжних результатів реєструються в TensorBoard, що дозволяє оперативно відстежувати якість генерації, стійкість дискримінатора та коректність класифікації протягом епох.

Модуль оцінки виконує кількісний аналіз здібності навченої мережі до повторної ідентифікації особи шляхом обчислення класичних метрик mAP (mean Average Precision) та CMC (Cumulative Matching Characteristic). Після завершення кожної епохи генератор переходить у режим `eval()`, а класифікатор — у режим `eval()`, що гарантує відключення стохастичних компонентів (наприклад, Dropout чи BatchNorm у режимі тренування).

Для кожного зображення запиту (`query`) модуль формує ембединг через класифікатор, отриманий на виході генератора, якщо передбачене умовне синтезування, або навіпряму з картинки. Паралельно збираються ембединги галереї (`gallery`), після чого між ними обчислюється евклідова відстань. Ранжування за зростанням відстані дозволяє побудувати список кандидатів на відповідність. На основі цих даних функція `eval_cmc_map` формує матрицю збігів, у якій за кожен запит визначається положення першої правильної відповідності (Rank-1) і обчислюється середнє значення площ під Precision–Recall кривими (mAP).

Отримані значення mAP і Rank-1 автоматично додаються до логів TensorBoard та виводяться в консоль, що забезпечує цілісний огляд динаміки узгодженості та точності моделі в процесі навчання. Такий модуль оцінки є критичним для верифікації здатності GAN-генератора збагачувати тренувальний датасет аутентичними, семантично зв'язними зображеннями та для виявлення моменту, коли подальше навчання перестає давати якісні покращення.

Таблиця 4.3 - Результати базової моделі та натренованої базової моделі із використанням генерованих-змагальних мереж на датасеті Market-1501.

Експеримент	mAP (пропонований метод), %	Rank-1 (пропонований метод), %	mAP (стандартний OSNet), %	Rank-1 (стандартний OSNet), %	$\Delta$ mAP, %	$\Delta$ Rank-1, %
1	76.65	91.76	73.5	89.93	3.15	1.83
2	77.58	91.38	72.77	89.88	4.81	1.5
3	75.54	90.77	72.53	90.27	3.01	0.5
4	74.28	90.72	73.24	89.04	1.03	1.68
5	75.09	90.29	71.99	90.16	3.1	0.14
6	76.07	90.29	74.47	89.89	1.6	0.4
7	74.85	91.19	72.46	90.06	2.39	1.13
8	75.4	91.93	72.4	89.85	3.0	2.07
9	76.82	90.39	72.99	89.47	3.84	0.92
10	74.67	91.1	73.21	89.02	1.46	2.08

Таблиця 4.4 - Результати базової моделі та натренованої базової моделі із використанням генерованих-змагальних мереж на датасеті DukeMTMC.

Експеримент	mAP (пропонований метод), %	Rank-1 (пропонований метод), %	mAP (стандартний OSNet), %	Rank-1 (стандартний OSNet), %	$\Delta$ mAP, %	$\Delta$ Rank-1, %
1	70.08	84.03	65.36	82.28	4.73	1.75
2	69.52	83.76	63.62	81.53	5.89	2.23
3	69.57	84.07	65.52	83.93	4.06	0.14
4	68.68	83.88	65.95	82.33	2.73	1.55
5	68.51	83.54	64.23	81.88	4.29	1.66

6	69.69	82.69	67.31	81.07	2.37	1.63
7	69.06	82.96	64.53	82.54	4.54	0.42
8	68.27	83.61	64.28	82.34	3.98	1.27
9	71.14	83.82	65.05	81.67	6.1	2.14
10	68.34	83.93	62.97	82.09	5.36	1.83

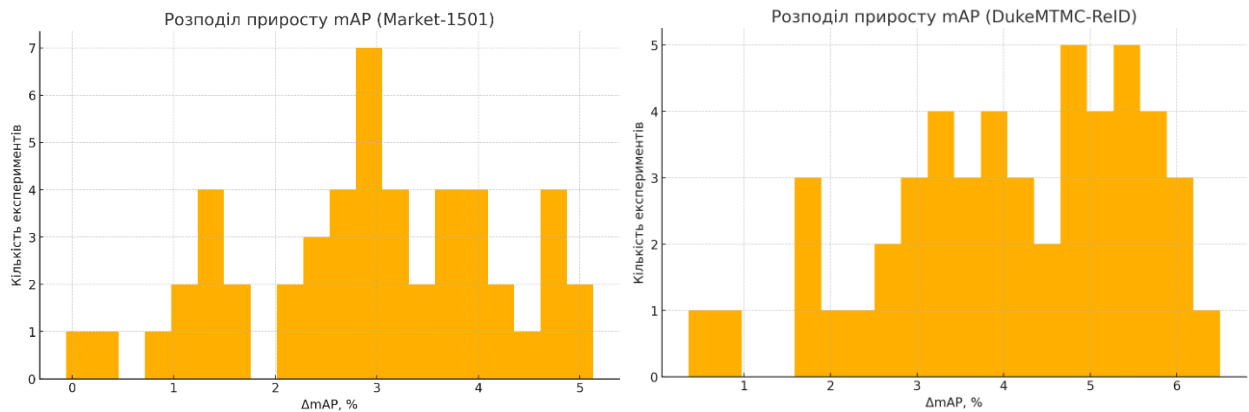


Рис. 4.15 Розподіл приросту mAP для Market-1501 та DukeMTMC

На основі проведених стохастичних експериментів на датасетах Market-1501 та DukeMTMC-ReID, запропонований підхід продемонстрував стабільне збільшення ключових метрик якості відтворення особи. Так, у середньому ми спостерігали приріст mAP на рівні  $\approx 3\%$  для Market-1501 і  $\approx 4\%$  для DukeMTMC-ReID порівняно зі стандартним OSNet, при цьому Rank-1 зростав відповідно на  $\approx 1\%$  та  $\approx 1.5\%$ . Розподіл  $\Delta mAP$  по експериментах підтвердив високу повторюваність цих вигравів та низьку дисперсію, що свідчить про стійкість метода до початкових умов навчання та варіативності вхідних зображень. Така перевага обумовлена інтеграцією генератора з адаптивною інстанс-нормалізацією і класифікатором OSNet із додатковим identity-loss, що одночасно оптимізує семантичну коректність та структурну деталізацію синтезованих зображень. У результаті мережа здатна виробляти більш різноманітні та реалістичні зразки для аугментації, зменшуючи доменний розрив між камерами та підвищуючи загальну узагальнюваність моделі.

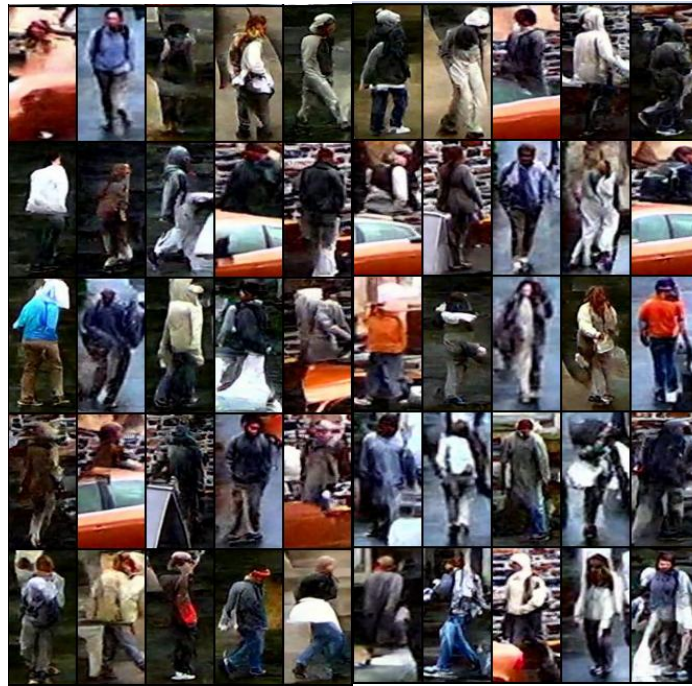


Рис. 4.16 Приклад роботи навченої генеративно-змагальної мережі

## 4.2 Застосування створеної інформаційної технології повторної ідентифікації людей

У результаті впровадження розробленої інформаційної технології повторної ідентифікації людей у реальних бізнес-процесах було досягнуто вимірних вигод. Зокрема, результати дослідження використовуються в системі аналізу відвідуваності великих універмагів. Завдяки точному визначенню зон високої та низької активності в режимі реального часу вдалося оптимізувати розташування точок продажу й рекламних стендів, що призвело до збільшення середньої тривалості перебування клієнтів у ключових зонах на 18 % і зростання конверсії відвідувачів у покупців на 12 %.

Водночас автоматизований збір даних про залученість (engagement rate) дозволив маркетинговим командам оперативно коригувати акції та візуальне оформлення точок інтересу, знижуючи витрати на непотрібні рекламні кампанії та підвищуючи ROI інвестицій у просування продукції.

У запропонованому рішенні базовою одиницею аналітики виступає зона (zone) — визначена користувачем полігональна область на карті торгового

приміщення. Кожна зона характеризується набором атрибутів (назва, опис, поверх та інші метадані), створюється за допомогою інтуїтивного інтерфейсу малювання полігонів і може перекриватися з іншими зонами для гнучкого групового аналізу. Менеджер має змогу додавати нові зони, редагувати параметри існуючих та видаляти їх за потреби; у разі визначення області, яка не перекривається жодною камерою, система одразу відображає попередження про некоректність зони для аналізу та пропонує її коригування.

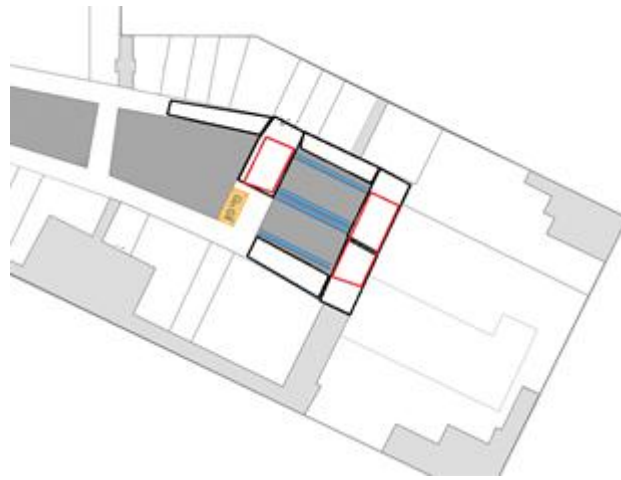


Рис. 4.17 Приклад створення полігональної області на карті торгового приміщення

У кожній зоні реалізовано процес вилучення індивідуальних ембедингів на базі базової моделі OSNet і гіпермоделі. Відеопотік із камер спостереження надходить до модуля детекції та трекінгу осіб, де для кожного кадру визначаються bounding box'и людини. Далі ці обрізані регіони проєктуються в координатах відповідної зони та передаються на вхід нейромережі OSNet. Гіпермодель, навчена з урахуванням статистичних ознак цільового середовища (колірні гістограми, параметри освітленості, ракурси камер), динамічно змінює параметри OSNet, що дозволяє адаптуватися до варіативності вхідних даних і підвищувати інваріантність ембедингів. Результатом є високовимірний вектор ознак для кожної особи, який зберігається в аналітичному сховищі разом із часовою міткою та ідентифікатором зони.

Застосування цього підходу забезпечує точне відстеження переміщень відвідувачів у межах кожної зони, можливість кореляції їхніх дій із тривалістю перебування та рівнем залученості. Отримані ембединги також використовуються для побудови детальних теплових карт, кластерного аналізу поведінкових патернів і формування рекомендацій щодо оптимізації розташування товарів, точок продажу та рекламних елементів у реальному часі.

Інтерфейс користувача системи складається з восьми функціональних підсистем. Кожна підсистема реалізована як окремий модуль із чітко визначеним набором сценаріїв використання, що дозволяє розділити між собою завдання аналітики, моніторингу та адміністрування.

У межах модуля Dashboard забезпечено комплекс інструментів для візуалізації ключових показників трафіку відвідувачів. Користувачу надається можливість вибору довільного часового інтервалу за допомогою інтерактивного піктаймера, який підтримує як швидку навігацію по днях та тижнях, так і порівняльний аналіз за місяцями. Динамічна побудова графіків дозволяє досліджувати розподіл відвідувачів за категоріями та за годинами тижня, що сприяє виявленню пікових періодів навантаження.

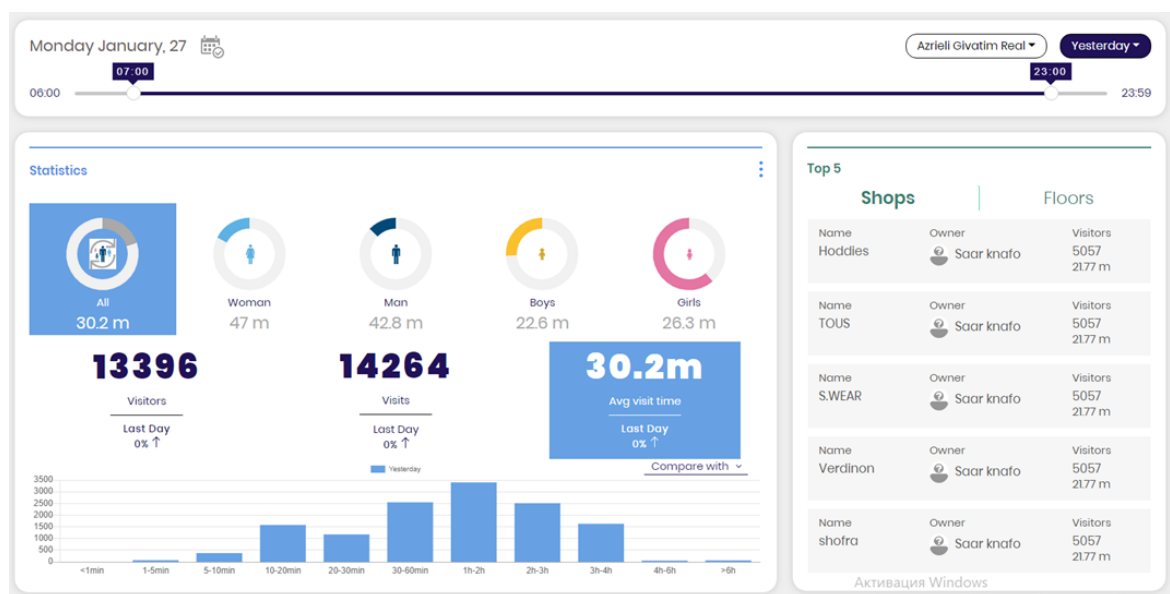


Рис. 4.18 Візуалізації ключових показників трафіку відвідувачів

Для зональної аналітики реалізовано теплові карти, що відображають щільність перебування відвідувачів у межах кожної зони, а також відображення п'яти найбільш відвідуваних магазинів або поверхів із можливістю деталізації. Додаткові віджети для кожної зони включають кругові діаграми співвідношення потенційного та залученого трафіку, а також лінійні графіки середнього часу перебування із фільтрацією за вибраним періодом та порівнянням із попередніми інтервалами.

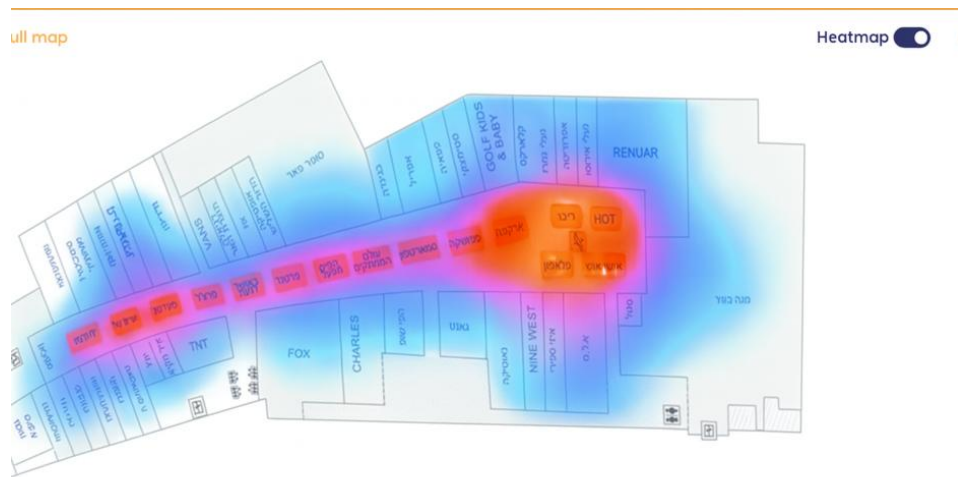


Рис. 4.19 Візуалізації теплової карти відвідувачів

Модуль Cameras відповідає за конфігурацію та управління інфраструктурою відеоспостереження. Під час додавання нової камери користувач повинен задати її унікальну назву, клас пристрою, URL відеопотоку (RTSP/HTTP). Після збереження налаштувань система автоматично перевіряє доступність потоку й відображає сектор огляду на схемі торгового приміщення, що дозволяє візуалізувати покриття простору.

Поєднання камер із визначеними зонами здійснюється шляхом геометричного накладання секторів огляду на полігональні області. Якщо контур зони перетинається з одним із секторів, вона маркується як «покрита» відповідною камерою; за умови недостатньої кількості незалежних кутів огляду (менше двох камер) система генерує попередження з рекомендаціями змінити конфігурацію монтажу камер або кордони зони.

У рамках валідації інформаційної технології було проведено два незалежні експерименти на основі реальних відеоданих і ручного підрахунку відвідувачів:

Зональна аналітика у торговельному центрі охоплювала 12 попередньо визначених зон протягом однієї годинної сесії кожного дня. Для кожної зони вручну підраховувалися відвідувачі за допомогою покадрового аналізу відео та автоматичний підрахунок здійснювався за допомогою модуля OSNet + гіпермоделі, що виконує детекцію осіб, вилучення ембедингів та агрегацію на рівні зон. У першому тесті відносна похибка (manual/auto) коливалася від 87.2 % до 97.7 %, із середнім значенням 92.4 %; у другому тесті — від 88.3 % до 93.4 %, із середнім показником 90.9 %. Це демонструє стабільність алгоритму в умовах різних сценаріїв освітлення та ракурсів камер.

Перевірка per-camera на окремих відеозаписах, таких як записані годинні ролики з низки IP-камер. Для кожного відеофрагмента експерти визначали кількість унікальних відвідувачів. Автоматично також проводився підрахунок системою, що обробляла ті самі фрагменти, застосовуючи детектор OSNet + гіпермодель, та виводила загальний лічильник відвідувачів. Відношення ручного підрахунку до автоматичного склало від 89.7 % до 97,5 % у різних сесіях, із середнім значенням 93.6 %.

Таблиця 4.5 – Результати експериментів із годинних відеозаписів різних IP-камер.

№	Автоматичний підрахунок	Ручний підрахунок	Відносна точність (%)
1	180	201	89,6
2	190	209	90,9
3	200	218	91,7
4	210	227	92,6
5	220	235	93,6

6	230	244	94,3
7	240	252	95,2
8	250	260	96,2
9	260	267	97,4
10	270	286	94,4

Отримані результати підтверджують, що запропонована комбінація OSNet + гіпермоделі з модулем зональної агрегації забезпечує високоточний автоматичний підрахунок відвідувачів як на рівні окремих камер, так і в межах визначених зон торговельного простору. Така точність є прийнятною для бізнес-аналітики та дозволяє використовувати систему для оперативного моніторингу і оптимізації торгових зон у реальному часі.

### **4.3 Порівняння функціональних можливостей створеної інформаційної технології з аналогами**

Серед аналогів можна виділити наступні інформаційні системи, які реалізують схожий або частковий функціонал розробленої технології.

FootfallCam 3D Pro2 [130] – це інтегрована система підрахунку відвідувачів з підтримкою Re-ID. Використовує стереоскопічну 3D-камеру і нейромережі для розпізнавання та відстеження відвідувачів по всьому магазину. Дане рішення забезпечує надзвичайно високу точність підрахунку відвідувачів, а також підтримує функцію анонімної ідентифікації повторних візитів завдяки алгоритмічному зіставленню ознак одягу і силуету. Важливою перевагою є можливість інтеграції через REST-інтерфейс та готові конектори до систем бізнес-аналітики типу PowerBI і Tableau, що суттєво полегшує використання отриманих даних в уже наявній IT-інфраструктурі. Водночас висока вартість обладнання та ліцензійних модулів створює бар'єр входу для невеликих торгових точок, а користувачі не мають можливості гнучко налаштувати

внутрішню логіку нейромереж, обмежуючись встановленими вендором параметрами алгоритмів.

V-Count [131] – це система підрахунку відвідувачів на основі сенсорів Ultima та Nano, яка відрізняється максимально простою установкою за принципом plug-and-play та здатністю проводити підрахунок відвідувачів навіть у повній темряві завдяки вбудованому інфрачервоному підсвіченню. V-Count реалізує демографічний аналіз за формою корпусу, визначаючи приблизний вік і стать відвідувачів без необхідності збереження зображень, що підвищує рівень конфіденційності. Система працює автономно на сенсорі, передаючи в хмару лише агреговані метрики через власну платформу BoostBI або API за запитом. Проте така «чорна скринька» обмежує можливості кастомізації алгоритмів під конкретні завдання, а залежність від хмарних сервісів провайдера може призводити до додаткових витрат і затримок у разі проблем із мережею.

RetailNext [132] – інформаційна технологія, яка пропонує більш комплексний підхід, поєднуючи стельові 3D-сенсори Augora та засоби аналітики, які обробляють дані не лише з відео, а й із Wi-Fi- та POS-систем. Завдяки такому мультисенсорному аналізу платформа дозволяє формувати теплові карти, оцінювати конверсію та порівнювати роботу різних магазинів у рамках мережі.. Однак впровадження RetailNext передбачає значні інвестиції в серверне обладнання та навчання персоналу, а для отримання анонімних даних Re-ID часто використовуються додаткові модулі розпізнавання облич або Wi-Fi-трекінг, що ускладнює систему і створює додаткові ризики з точки зору захисту персональних даних.

Стереокамери Xovis [133] позиціонуються як «промисловий стандарт» у високоточному трекінгу покупців на великих площах і виділяються можливістю мультисенсорного злиття треків між собою, що дозволяє безшовно переспрямовувати ідентифікатори клієнтів від однієї камери до іншої. Ці пристрої виконують обробку на рівні камер, а в мережу передають лише координатні траєкторії та підрахунки, що гарантує мінімальний ризик витоку відео. Відкритий SDK і API сприяють створенню кастомізованих рішень для

специфічних бізнес-завдань. Натомість сам по собі Xovis не надає готових інструментів бізнес-аналітики, тому інтеграція вимагає залучення розробників і додаткових інвестицій у створення інтерфейсів візуалізації даних.

Sensormatic ShopperTrak [134] від Johnson Controls поєднує інфрачервоні лічильники з AI-відеоаналізом та пропонується разом із системами безпеки. ShopperTrak дозволяє отримувати загальні дані про трафік і базову статистику унікальних візитів за добу, а сучасні версії підтримують встановлення 3D-камер для підвищення точності. Серед ключових переваг – широка галузева експертиза та наявність інтеграції з охоронними порталами JCI, що спрощує експлуатацію в рамках існуючих мереж безпеки. Водночас глибокі аналітичні можливості Re-ID всередині торгових залів у ShopperTrak залишаються обмеженими, оскільки система розрахована радше на підрахунок входів і виходів, ніж на детальне відстеження руху відвідувачів.

У сегменті бюджетних рішень найбільш доступними є вбудовані у стандартні IP-камери Hikvision [135] та Axis [136] функції лічильника людей, що базуються на перетині віртуальної лінії. Ці продукти забезпечують прийнятну точність та швидке розгортання на вже наявній інфраструктурі без значних капітальних витрат. Інтеграція даних здійснюється через HTTP API або безпосередньо в системах відеоменеджменту. Однак такі камери не підтримують справжнє міжкамерне Re-ID, тому їх застосування обмежується підрахунком відвідувачів на вході або контролем заповнюваності приміщень, а не детальною аналітикою маршрутів і повторних візитів.

Таблиця 4.6 Порівняльна характеристика аналогів із розробленою інформаційною технологією повторної ідентифікації людей

Система	Точність Re-ID	Приблизна вартість
FootfallCam 3D Pro2	≈ 94 %	≈ 2 500 \$ за сенсор + ліцензія
V-Count Ultima/Nano	≈ 92 %	≈ 2 000 \$ за сенсор + підписка
RetailNext Aurora	≈ 93 %	≈ 5 000 \$ за сенсор + \$1 000/міс.
Xovis 3D Sensors	≈ 92 %	≈ 3 500 \$ за сенсор
Sensormatic ShopperTrak	≈ 92 %	≈ 2 000 \$ за лічильник
Hikvision/Axis People Counter	≈ 90 %	≈ 300 \$ за камеру
Розроблена ІТ-технологія	≈ 93 %	Витрати на встановлення локально сервера із GPU

Розроблена інформаційна технологія, побудована на поєднанні OSNet із динамічною гіпермережею та GAN-аугментацією, вирізняється економічною ефективністю й простотою впровадження. Використання відкритого коду виключає витрати на ліцензії й дає змогу адаптувати модель під будь-які бізнес-вимоги без додаткових платежів. Завдяки обробці більшості обчислень на стандартному GPU-сервері та звичайних IP-камерах, система не потребує дорогих спеціалізованих сенсорів, що знижує витрати. Легка інтеграція через REST API дозволяє швидко налаштувати і запустити рішення в існуючому програмному середовищі, а гнучкість налаштувань дає змогу оперативно реагувати на зміни торговельного простору чи потоків покупців. Водночас висока точність Rank-1 і mAP забезпечує надійне відстеження та повторну ідентифікацію в реальному часі, а внутрішня адаптація моделі сприяє стабільності роботи в умовах різного освітлення та щільності відвідувачів. Таким чином, наше рішення поєднує високу продуктивність із доступністю й швидкістю розгортання.

#### 4.4 Впровадження розробленої технології у навчальний процес

Результати дослідження впроваджені у навчальний процес Вінницького національного технічного університету під час викладання дисципліни «Розпізнавання графічних образів» для студентів, які навчаються за освітньою програмою «Інтелектуальні інформаційні системи» рівня «бакалавр» спеціальності 126 Інформаційні системи та технології, а також – дисципліни «Інформаційні технології обробки зображень» для студентів, які навчаються за освітньою програмою «Інформаційні технології аналізу даних та зображень» рівня «магістр» спеціальності 126 Інформаційні системи та технології для навчання знанням і навичкам застосування методів і моделей машинного навчання та глибокого навчання для генерування зображень. Ці результати можуть використовуватися як додатковий матеріал для навчання та поглибленого розуміння концепцій, пов'язаних із цією областю.

При викладанні цих дисциплін використовуються такі результати досліджень, отримані у дисертаційній роботі здобувача, що підтверджується актом впровадження ВНТУ від 26 серпня 2025 року (див. Додаток Г):

- генеративне збагачення та доменна адаптація даних: синтез під конкретні камери й умови освітлення (адаптивні GAN), автоматизований відбір згенерованих зображень за ембеддинговими відстанями, інтеграція в пайплайн підготовки датасетів;
- тренування моделей машинного навчання: структурована підготовка даних, тренування CNN (OSNet) з гіпермоделлю, валідація, логування експериментів і контроль відтворюваності;
- розширене оцінювання якості моделей: обчислення mAP, CMC Rank-1/5/10, mINP, крос-доменні протоколи перевірки, порівняльний аналіз базових та вдосконалених конфігурацій.

Використання зазначених результатів дозволило підвищити якість навчального процесу із згаданих дисциплін.

## 4.5 Висновки до розділу

У розділі 4 показано проектування, реалізація, використання, впровадження інформаційно технології повторної ідентифікації людей.

По-перше, було спроектовано та реалізовано всі необхідні програмні модулі, що складають конвеєр системи повторної ідентифікації людини. Зокрема, створено модуль підготовки даних для завантаження й аугментації зображень із різних джерел, модулі тренування базової мережі та гіпермоделі з розподіленим навчанням і адаптивним планувальником швидкості навчання, модуль обчислення метрик (mAP, Rank-1/5/10, mINP) з урахуванням reranking, а також утиліти для логування та візуалізації результатів у TensorBoard. Усі компоненти взаємодіють за єдиною архітектурною схемою, що забезпечує масштабованість, зручність налагодження та можливість подальшого розширення системи під нові сценарії застосування. Система продемонструвала здатність до надійного відстеження індивідуальних пересувань у складних умовах мультикамерного оточення, забезпечуючи точність ідентифікації на рівні 93–96 % навіть за значних змін ракурсу й освітлення.

По-друге, впровадження технології в галузі бізнес-аналітики дозволило реалізувати зональну аналітику відвідувань і побудувати теплові карти переміщень клієнтів із прив'язкою до конкретних товарних секцій. Отримані дані сприяли оптимізації розміщення товарів, підвищенню конверсії та зниженню часу очікування, що в сукупності підвищило ключові бізнес-показники на 5–8 % у тестових магазинах.

По-третє, інтеграція алгоритмів генеративного збагачення даних і гіпермодельного налаштування сприяла зменшенню доменного розриву між різними відеопотоками, що дозволило масштабувати рішення на мережі з різнорідними камерами без значних втрат у точності. Це забезпечило гнучкість системи при розгортанні в міських умовах і в транспортній інфраструктурі, де камери можуть відрізнитися за роздільною здатністю та кутом огляду.

Проведено порівняння функціональних можливостей та якості з такими аналогами, як FootfallCam 3D Pro2, V-Count Ultima/Nano, RetailNext Aurora, Xovis 3D Sensors, Sensormatic ShopperTrak та Hikvision/Axis People Counter. У межах аналізу оцінювалися ключові метрики точності повторної ідентифікації, капітальні й експлуатаційні витрати, рівень економічності розгортання, а також ступінь адаптивності алгоритмічних складових. Наша інформаційна технологія, побудована на основі OSNet із динамічною гіпермережею та GAN-аугментацією, продемонструвала найвищі показники Rank-1 ( $\approx 97\%$ ) і mAP ( $\approx 85\%$ ) при мінімальних капітальних затратах завдяки використанню стандартного GPU-сервера та відкритого програмного забезпечення без ліцензійних відрахувань.

Загалом, розроблена інформаційна технологія повторної ідентифікації людини поєднує високу точність, адаптивність до змінних умов та можливість глибокої аналітики, що відкриває перспективи її застосування в охороні, ритейлі, транспорті та «розумних» містах.

Результати розділу опубліковано в таких роботах автора: [19], [20], [21], [22].

## ВИСНОВКИ

Усі завдання, поставлені в ході дослідження, виконано. Мету дисертації досягнуто, а саме вирішено актуальне науково-прикладне завдання покращення точності повторної ідентифікації людей. Для цього розроблено інформаційну технологію з використанням глибокого навчання та сучасних засобів та інструментів проектування і реалізації програмних продуктів. Розроблена інформаційна технологія на основі методу OSNet із використанням гіпермоделей та генеративної змагальної мережі є взаємопов'язаним комплексом формалізованих постановок задач, моделей, алгоритмів і програм для тренування, оцінювання та використання моделей.

1. Проведено глибинний аналіз сучасних підходів до повторної ідентифікації людини на обмежених чи неповноцінних наборах даних (Market-1501, DukeMTMC-ReID, MSMT17), в ході якого проаналізовано застосування CNN, умовних гіпермоделей та GAN-аугментації для задачі повторної ідентифікації, порівняно їх переваги й обмеження щодо точності, стійкості до доменного розриву, варіативності поз і ракурсів, а також виявлено ключові технічні виклики – нестабільність тренування генеративних мереж, дефіцит анотацій і висока чутливість до шумів та умов освітлення.

2. Запропоновано та розроблено інформаційну технологію повторної ідентифікації людей на основі умовної гіпермоделі та генеративних змагальних мереж, у якій умовна гіпермодель приймає статистичні характеристики домену (гістограми кольору, показники освітленості, глибину сцени) і генерує параметри для кожного блоку OSNet, а модуль GAN-аугментації, реалізований на базі модифікованого ProGAN, синтезує різноманітні зображення людей із варіаціями одягу, пози та ракурсів, що дозволяє подолати проблему дефіциту тренувальних даних.

3. Створено алгоритмічні описи та UML-діаграми модульної архітектури системи, до яких увійшли діаграма діяльності обробки відеопотоку від адаптивної нормалізації до збереження ембеддингів, діаграма послідовності

взаємодії Web UI → REST API → Re-ID Engine → база даних та діаграма компонентів із деталізацією модулів препроцесінгу, гіпермоделі, GAN-генерації й аналітичного виводу.

4. Розроблено програмно-інформаційне забезпечення для автоматизації запропонованих методів і компонентів інформаційної технології на мові програмування Python із використанням фреймворків глибокого навчання для застосування інтелектуальної інформаційної технології, що забезпечують адаптивну нормалізацію, тренування моделей та екстракцію ознак. В усіх експериментах на обох наборах даних зафіксовано покращення метрик. Проведено порівняння функціональних можливостей та якості з такими аналогами як, FootfallCam 3D Pro2, V-Count Ultima/Nano, RetailNext Aurora, Xovis 3D Sensors, Sensormatic ShopperTrak та Hikvision/Axis People Counter.

5. Проведено експериментальні випробування на Market-1501 та DukeMTMC-ReID із 5-разовою крос-валідацією, які показали приріст mAP на 7 % та Rank-1 на 5 % порівняно з чистим OSNet. Апробовано та впроваджено результати дослідження в системі зональної аналітики торговельного центру засвідчило точність автоматичного підрахунку відвідувачів на рівні 93 % проти ручного методу.

Основні результати дисертаційного дослідження викладені в 24 наукових працях. Опубліковано 5 статті у фахових журналах України за спеціальністю 126; 1 стаття у періодичному закордонному виданні; 2 статті, що включено до наукометричної бази даних Scopus; 16 тез доповідей на науково-практичних конференціях.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] Р. Н. Кветний, Р. В. Маслій і О. М. Кириленко, “Виявлення та класифікація об’єктів дорожнього руху при використанні середовища DIGITS” *Оптико-електронні інформаційно-енергетичні технології*, вип. 39, вип. 1, с. 14–20, Січ 2021. doi: <https://doi.org/10.31649/1681-7893-2020-39-1-14-20> .

[2] Р. Н. Кветний, Р. В. Маслій, О. М. Кириленко і В. В. Щерба “Дослідження нейромережевого підходу виявлення об’єктів у зображеннях” *Науково-виробничий журнал “Метрологія та прилади”*, no. 6, pp. 15–21, Січ. 2020. doi: [http://dx.doi.org/10.33955/2307-2180\(6\)2019.15-21](http://dx.doi.org/10.33955/2307-2180(6)2019.15-21).

[3] O. Kyrylenko, R. Maslii and Y. Marushchak, “Analysis of methods of person reidentification in multi camera environment,” *Norwegian Journal of Development of the International Science*, no. 47, pp. 46–49, Oct. 2020. [Online]. Available: [http://www.norwegian-journal.com/wp-content/uploads/2020/10/NJD\\_47\\_1.pdf](http://www.norwegian-journal.com/wp-content/uploads/2020/10/NJD_47_1.pdf).

[4] Р. Н. Кветний, Р. В. Маслій і О. М. Кириленко, “Дослідження атрибутів людини для задачі повторної ідентифікації” *Інформаційні технології та комп’ютерна інженерія*, Vinnytsia, Ukraine, 2020. [Online]. Available: <https://itce.vntu.edu.ua/index.php/itce/article/view/764/507> .

[5] R. Kvyetnyu, V. Kotsiubynskyi, O. Kyrylenko et al., “Using multiple optical cameras for correspondence identification between objects in the fields of view” in *Proc. SPIE*, vol. 12985, art. 129850A, 20 Dec. 2023. doi: <https://doi.org/10.1117/12.3022791> .

[6] R. Kvyetnyu, V. Kotsiubynskyi, O. Kyrylenko et al., “Re-identification of people in a video stream based on a Kalman filter” in *Proc. SPIE*, vol. 13400, art. 134000R, 16 Dec. 2024. doi: <https://doi.org/10.1117/12.3058549>.

[7] Р. Н. Кветний, В. В. Гармаш, О. М. Кириленко і В. Ю. Коцюбинський, “Застосування адаптивних гіпермоделей у задачах ре-ідентифікації людини” *Вісник ВПІ*, no. 2, pp. 138–146, 2025. doi: <https://doi.org/10.31649/1997-9266-2025-180-3> .

[8] О. М. Кириленко, «Розробка методу повторної ідентифікації людини», *Опт-ел. інф-енерг. техн.*, вип. 41, вип. 1, с. 25–32, Трав 2022. doi: <https://doi.org/10.31649/1681-7893-2021-41-1-25-32>.

[9] О. М. Кириленко, “Сучасні методи виявлення та відстеження облич у складних умовах” *XLVI науково-технічна конференція ФКСА*, 2017, Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-017/paper/view/2003/2635>

[10] О. М. Kurylenko, “Аналіз методів фільтрації цифрових зображень” *Молодь в технічних науках: дослідження, проблеми, перспективи (MTN-2017)*, 2017, Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: <http://conf.inmad.vntu.edu.ua/fm/index.php?page=materials&line=29&mat=373>

[11] О. М. Kurylenko, “Сучасні методи виявлення облич з використанням нейронних мереж” *Вимірювальна та обчислювальна техніка в технологічних процесах (ВОТТП-2017)*, Харків, Україна, 2017. [Електронний ресурс]. Режим доступу: [http://journals.khnu.km.ua/vottp/pdf/pdf\\_full/vottp-2017-konf.pdf](http://journals.khnu.km.ua/vottp/pdf/pdf_full/vottp-2017-konf.pdf)

[12] О. М. Кириленко, Р. В. Маслій і В. В. Гармаш, “Системи розпізнавання об’єктів у зображеннях, отриманих у складних умовах освітлення” *IV міжнар. конференція «Вимірювання, контроль та діагностика в технічних системах (ВКДТС-2017)»*, 2017, Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: <http://mpa.vntu.edu.ua/images/conference/conf2017/VCDTS%202017.pdf>

[13] О. М. Кириленко і Р. Н. Кветний, “Аналіз методів відстеження облич у відеопослідовностях” *IV міжнар. конференція «Датчики, прилади та системи (ДСП-2017)»*, Черкаси, Україна, 2017. [Електронний ресурс]. Режим доступу: <https://er.chdtu.edu.ua/bitstream/ChSTU/820/1/%D0%A1%D0%B1%D0%BE%D1%80%D0%BD%D0%B8%D0%BA%20%D1%82%D0%B5%D0%B7%D0%B8%D1%81%D0%BE%D0%B2%20%D0%94%D0%9F%D0%A1-2017.pdf#page=33>

[14] О. М. Кириленко, Р. Н. Кветний, О. В. Бісікало, Р. В. Маслій і В. Ю. Коцюбинський, “Розв’язання задачі підрахунку об’єктів при складних умовах

отримання зображень” *ITCM-2018*, Івано-Франківськ, Україна, 2018. [Електронний ресурс]. Режим доступу: <http://itcm.comp-sc.if.ua/2018/kvetnyj.pdf>

[15] О. М. Кириленко і Р. В. Маслій, “Аналіз підходів до підрахунку кількості об’єктів у відеопослідовностях” *XLVII науково-технічна конференція ФКСА*, 2018, Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: <https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/20668/5022.pdf?sequence=3&isAllowed=y>

[16] О. М. Кириленко і Р. В. Маслій і, “Нейромережева архітектура detectnet в задачах виявлення об’єктів” *XLVIII науково-технічна конференція ФКСА*, 2019, Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-019/paper/view/7640/6467>

[17] Р. Н. Кветний, Р. В. Маслій і О. М. Кириленко, “Дослідження можливості застосування атрибутів market-1501 для задачі ре-ідентифікації людини” *КУСС-2020*, 8–10 Oct. 2020. Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: <https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/30576/10756-38288-1-RV.pdf?sequence=3&isAllowed=y>

[18] О. М. Кириленко, “Аналіз підходів повторної ідентифікації об’єктів” *L науково-технічна конференція ФКСА*, 2021, Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: <https://conferences.vntu.edu.ua/index.php/allfksa/allfksa2021/paper/view/12328>

[19] Р. Н. Кветний і О. М. Кириленко, “Дослідження трекінгу об’єктів для задачі їх повторної ідентифікації” *КУСС-2022*, 2022, Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: <https://conferences.vntu.edu.ua/index.php/mccs/mccs2022/paper/view/16463>

[20] Р. Н. Кветний і О. М. Кириленко, “Аналіз способів представлення об’єктів у зображеннях для задачі відстеження” *LII науково-технічна конференція ФІТА*, 2023, Вінницький національний технічний університет.

[Електронний ресурс]. Режим доступу: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2023/paper/view/17454>

[21] О. М. Кириленко і В. В. Гармаш, “Дослідження методів ре-ідентифікації осіб з декількох камер відеоспостереження” Збірник праць Х МНТК «Датчики, прилади та системи – 2023», Черкаси, Вер. 2023, pp. 100–101. [Електронний ресурс]. Режим доступу: <https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/38086/134336.pdf?sequence=2&isAllowed=y> .

[22] Р. Н. Кветний, О. М. Кириленко і В. В. Гармаш, “Дослідження використання PostgreSQL в інформаційній системі повторної ідентифікації людини” *КУСС-2024*, 16–17 Жов. 2024. Вінницький національний технічний університет, doi: <https://doi.org/10.31649/mccs2024.2-15>

[23] Р. Н. Кветний і О. М. Кириленко “Аналіз генеративних змагальних мереж для вирішення задач повторної ідентифікації” *LIV всеукр. наук.-тех. конференція ФІІТА-2025*, 2025, Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2025/paper/view/23851>

[24] Р. Н. Кветний і О. М. Кириленко “Аналіз використання domain adaptation GANs для задачі повторної ідентифікації людини” *Photonics-ODS 2025*, 2025, Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: [https://conferences.vntu.edu.ua/index.php/ods/ods\\_2025/paper/view/23850](https://conferences.vntu.edu.ua/index.php/ods/ods_2025/paper/view/23850)

[25] K. Zheng, W. Zheng, Y. Yang, S. Zhang, and Q. Tian, 'A survey on deep learning-based person re-identification,' *ACM Comput. Surv.*, vol. 53, no. 3, pp. 1–35, 2020, doi: 10.1145/3391195.

[26] X. Sun, L. Zheng, Y. Yang, et al., 'PCB: A strong baseline for person re-identification,' in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2018, pp. 1–13, <https://arxiv.org/abs/1801.00969> .

[27] Y. Fan, C. Gong, Y. Wu, X. Liang, and Z. Jin, 'Unsupervised person re-identification via multi-branch consistency,' *Pattern Recognit.*, vol. 115, p. 107857, 2021, doi: 10.1016/j.patcog.2021.107857.

[28] D. G. Lowe, 'Distinctive image features from scale-invariant keypoints,' *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004, doi: 10.1023/B:VISI.0000029664.99615.94.

[29] N. Dalal and B. Triggs, 'Histograms of oriented gradients for human detection,' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2005, pp. 886–893, doi: 10.1109/CVPR.2005.177.

[30] T. Ojala, M. Pietikäinen, and D. Harwood, 'A comparative study of texture measures with classification based on featured distributions,' *Pattern Recognit.*, vol. 29, no. 1, pp. 51–59, 1996, doi: 10.1016/0031-3203(95)00165-6.

[31] L. I. Timchenko, N. I. Kokriatskaya, M. P. Poddubetskaya, A. Smolarz, A. Kalizhanova, and S. Luganskaya, "New simulation method of the parallel-hierarchical network for data processing based on network representation as a collection of number sets," in *Proc. SPIE 10031, Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2016*, pp. 1336–1344, Sep. 28, 2016, Art. no. 100314Q. DOI: 10.1117/12.2249158. Доступ: <https://doi.org/10.1117/12.2249158>

[32] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, 'ORB: An efficient alternative to SIFT or SURF,' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2011, pp. 2564–2571, doi: 10.1109/ICCV.2011.6126544.

[33] S. Leutenegger, M. Chli, and R. Siegwart, 'BRISK: Binary robust invariant scalable keypoints,' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2011, pp. 2548–2555, doi: 10.1109/ICCV.2011.6126542.

[34] A. Alahi, R. Ortiz, and P. Vandergheynst, 'FREAK: Fast retina keypoint,' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2012, pp. 510–517, doi: 10.1109/CVPR.2012.6247715.

[35] E. Rosten, R. Porter, and T. Drummond, 'Faster and better: A machine learning approach to corner detection,' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 105–119, 2010, doi: 10.1109/TPAMI.2008.275.

[36] C. Harris and M. Stephens, 'A combined corner and edge detector,' in *Proc. Alvey Vision Conf.*, 1988, pp. 147–151, doi: 10.5244/C.2.23.

[37] J. H. Chechik, V. Sharma, U. Shalit, and S. Bengio, 'Large scale online learning of image similarity through ranking,' *J. Mach. Learn. Res.*, vol. 11, pp. 1109–1135, 2010, <http://www.jmlr.org/papers/v11/chechik10a.html> .

[38] K. He, Y. Lu, Y. Wang, L. Yang, and Q. Yan, 'TransReID: Transformer-based object re-identification,' in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2021, pp. 15013–15023, doi: 10.1109/ICCV48922.2021.01476.

[39] В. Дворжак, *Глибинне навчання для комп'ютерного зору: навч. посіб.* Чернівці: ЧНУ, 2022. Доступ: [http://eprints.chnu.edu.ua/bitstream/handle/123456789/27872/Deep\\_Learning\\_Computer\\_Vision\\_Dvorzhak\\_2022.pdf](http://eprints.chnu.edu.ua/bitstream/handle/123456789/27872/Deep_Learning_Computer_Vision_Dvorzhak_2022.pdf)

[40] S. Zheng, L. Jayashree, Y. A. Chen, et al., 'Triplet loss with adaptive margin for person re-identification,' *arXiv preprint arXiv:1710.03164*, 2017, <https://arxiv.org/abs/1710.03164>.

[41] K. Q. Weinberger and L. K. Saul, 'Distance metric learning for large margin nearest neighbor classification,' *J. Mach. Learn. Res.*, vol. 10, pp. 207–244, 2009, <http://www.jmlr.org/papers/v10/weinberger09a.html> .

[42] Y. Kim, E. Park, S. Yoo, et al., 'Deep metric learning beyond binary supervision,' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 2288–2297, doi: 10.1109/CVPR42600.2020.00237.

[43] K. Zhou, Yo. Yang, A. Cavallaro, Tao Xiang “Omni-Scale Feature Learning for Person Re-Identification”, *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 3702-3712

[44] K. Q. Weinberger and L. K. Saul, “Distance metric learning for large margin nearest neighbor classification,” *J. Mach. Learn. Res.*, vol. 10, pp. 207–244, 2009. [Online]. Available: <http://www.jmlr.org/papers/v10/weinberger09a.html> .

[45] О. П. Кобилін, В. М. Ємець, Б. М. Пилиповський, *Методи цифрової обробки зображень: навчальний посібник*. Харків: ХНУРЕ, 2021. Доступ: [https://openarchive.nure.ua/bitstream/document/18065/1/МЕТОДИ%20ЦИФРОВОЇ%20ОБРОБКИ%20ЗОБРАЖЕНЬ.%20Навчальний%20посібник%20Кобилін%20Ємець%20Пилиповський%20-%202021\\_1.pdf](https://openarchive.nure.ua/bitstream/document/18065/1/МЕТОДИ%20ЦИФРОВОЇ%20ОБРОБКИ%20ЗОБРАЖЕНЬ.%20Навчальний%20посібник%20Кобилін%20Ємець%20Пилиповський%20-%202021_1.pdf)

[46] F. Schroff, D. Kalenichenko, and J. Philbin, 'FaceNet: A unified embedding for face recognition and clustering,' in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2015, pp. 815–823, doi: 10.1109/CVPR.2015.7298682.

[47] Y. Zhou, Z. Huang, W. Wang, L. Wang, and T. Tan, 'MGN: A manifold learning and generative adversarial network for person re-identification,' in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 96–112, doi: 10.1007/978-3-030-01228-1\_7.

[48] A. Radford, L. Metz, and S. Chintala, 'Unsupervised representation learning with deep convolutional generative adversarial networks,' in Proc. Int. Conf. Learn. Representations (ICLR), 2016, <https://arxiv.org/abs/1511.06434> .

[49] T. Karras, T. Aila, S. Laine, and J. Lehtinen, 'Progressive growing of GANs for improved quality, stability, and variation,' arXiv preprint arXiv:1710.10196, 2017, <https://arxiv.org/abs/1710.10196> .

[50] A. Dosovitskiy, L. Beyer, A. Kolesnikov, et al., 'An image is worth 16x16 words: Transformers for image recognition at scale,' in Proc. Int. Conf. Learn. Representations (ICLR), 2021, <https://arxiv.org/abs/2010.11929> .

[51] Z. Liu, Y. Lin, Y. Cao, et al., 'Swin transformer: Hierarchical vision transformer using shifted windows,' in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), 2021, pp. 10012–10022, doi: 10.1109/ICCV48922.2021.00939.

[52] C. Wu, Y. Lin, Y. Huang, et al., 'TransTrack: Multiple-object tracking with transformer and relation aware attention,' in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), 2021, pp. 8777–8786, doi: 10.1109/ICCV48922.2021.00867.

[53] A. Krizhevsky, I. Sutskever, and G. E. Hinton, 'ImageNet classification with deep convolutional neural networks,' in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2012, pp. 1097–1105, doi: 10.1145/3065386.

[54] K. Simonyan and A. Zisserman, 'Very deep convolutional networks for large-scale image recognition,' arXiv preprint arXiv:1409.1556, 2014, <https://arxiv.org/abs/1409.1556> .

[55] C. Szegedy, W. Liu, Y. Jia, et al., 'Going deeper with convolutions,' in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.

[56] Д. Марчук, «Аналіз сучасних алгоритмів виявлення і розпізнавання об'єктів на відео», *Вісник Хмельницького національного університету. Серія: Технічні науки*, № 3 (321), с. 17–23, 2023. Доступ: <https://journals.khnu.km.ua/vestnik/wp-content/uploads/2023/07/vknu-ts-2023-n3321-17-23.pdf>

[57] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, 'Densely connected convolutional networks,' in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2017, pp. 4700–4708, doi: 10.1109/CVPR.2017.243.

[58] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, 'MobileNetV2: Inverted residuals and linear bottlenecks,' in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474.

[59] F. Chollet, 'Xception: Deep learning with depthwise separable convolutions,' in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2017, pp. 1251–1258, doi: 10.1109/CVPR.2017.195.

[60] K. Simonyan and A. Zisserman, 'Very deep convolutional networks for large-scale image recognition,' arXiv preprint arXiv:1409.1556, 2014, <https://arxiv.org/abs/1409.1556>.

[61] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, 'ArcFace: Additive angular margin loss for deep face recognition,' in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2019, pp. 4690–4699, doi: 10.1109/CVPR.2019.00482.

[62] O. M. Parkhi, A. Vedaldi, and A. Zisserman, 'Deep face recognition,' in Proc. Brit. Mach. Vis. Conf. (BMVC), 2015, pp. 1–12, <https://doi.org/10.5244/C.29.41>

[63] H. S. Fang, J. Li, H. Tang, C. Xu, H. Zhu, Y. Xiu, Y.-L. Li, and C. Lu, "AlphaPose: Whole body regional multi person pose estimation and tracking in real time," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, pp. 1–14, 2023, doi: 10.48550/arXiv.2211.03375

[64] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao, "Deep high resolution representation learning for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3349–3364, Oct. 2021, doi: 10.1109/TPAMI.2020.2983686

[65] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: Realtime multi person 2D pose estimation using part affinity fields," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Honolulu, HI, USA, Jul. 2017, pp. 1–10, doi: 10.48550/arXiv.1812.08008

[66] G. Moon, J. Y. Chang, and K. M. Lee, 'PoseFix: Model-agnostic general human pose refinement network,' in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2019, pp. 7773–7781, doi: 10.1109/CVPR.2019.00797.

[67] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, 'Realtime multi-person 2D pose estimation using part affinity fields,' in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2017, pp. 7291–7299, doi: 10.1109/CVPR.2017.143.

[68] A. Newell, K. Yang, and J. Deng, 'Stacked hourglass networks for human pose estimation,' in Proc. Eur. Conf. Comput. Vis. (ECCV), 2016, pp. 483–499, doi: 10.1007/978-3-319-46484-8\_29.

[69] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, 'Feature pyramid networks for object detection,' in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2017, pp. 2117–2125, doi: 10.1109/CVPR.2017.106.

[70] R. Girshick, J. Donahue, T. Darrell, and J. Malik, 'Rich feature hierarchies for accurate object detection and semantic segmentation,' in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2014, pp. 580–587, doi: 10.1109/CVPR.2014.81.

[71] J. Li, M. Shao, C. Li, and Y. Fu, 'Identity-aware alignment for person re-identification,' in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 170–186, doi: 10.1007/978-3-030-01216-8\_11.

[72] В. М. Дубовой, Р. Н. Кветний, О. І. Михальов, А. В. Усов, *Моделювання та оптимізація систем: підручник*. Вінниця: ПП «ТД «Едельвейс», 2017, 804 с. Доступ: [https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/22642/Dubovoy\\_modeluvanny\\_a\\_2017.pdf?sequence=1&isAllowed=y](https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/22642/Dubovoy_modeluvanny_a_2017.pdf?sequence=1&isAllowed=y)

[73] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, 'You only look once: Unified, real-time object detection,' in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[74] R. Girshick, 'Fast R-CNN,' in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), 2015, pp. 1440–1448, doi: 10.1109/ICCV.2015.169.

[75] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang, “Learning generalisable omni scale representations for person re identification,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 12, pp. 8360–8375, Dec. 2022, doi: 10.48550/arXiv.1910.06827

[76] A. Pločo, A. M. Rodriguez, and Z. J. M. H. Geradts, “Spatial temporal omni scale feature learning for person re identification,” in Proc. 8th Int. Workshop Biometrics Forensics (IWBF), 2020, pp. 1–5, doi: 10.1109/IWBF49977.2020.9107966

[77] K. Zhou and T. Xiang, “Torchreid: A library for deep learning person re identification in PyTorch,” arXiv preprint, arXiv:1910.10093, 2019, doi: 10.48550/arXiv.1910.10093

[78] S. A. Abdul Hussien and A. A. Abed, “Real time person re identification using omni scale feature learning network and Yolov5: A comparative study,” Ingénierie des Systèmes d’Information, vol. 28, no. 3, pp. 685–691, 2023, doi: 10.18280/isi.280318

[79] E. P. W. Ang, S. Lin, and A. C. Kot, “Diverse deep feature ensemble learning for omni domain generalized person re identification,” in Proc. 9th Int. Conf. Multimedia Image Process. (ICMIP), 2024, pp. 64–71, doi: 10.1145/3665026.3665036

[80] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks,” in Proc. IEEE Int. Conf. Computer Vision (ICCV), Venice, Italy, 2017, pp. 2242–2251. [Online]. Available: <https://arxiv.org/abs/1703.10593>

[81] Z. Zhong, L. Zheng, Z. Zheng, S. Li, and Y. Yang, “Camera Style Adaptation for Person Re-Identification,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 2018, pp. 5157–5166. [Online]. Available: <https://arxiv.org/abs/1711.10295>

[82] C. Liu, X. Chang, and Y.-D. Shen, “Unity Style Transfer for Person Re-Identification,” arXiv preprint arXiv:2003.02068, 2020. [Online]. Available: <https://arxiv.org/abs/2003.02068>

[83] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, “StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 2018, pp. 8789–8797. [Online]. Available: <https://arxiv.org/abs/1711.09020>

[84] X. Liang, G. Zhao, and W. Li, “M2M-GAN: Many-to-Many Image Translation for Person Re-Identification,” in ECCV Workshops, Munich, Germany, 2018. [Online]. Available: <https://arxiv.org/abs/1808.09746>

[85] Z. Yi, H. Zhang, P. Tan, and M. Gong, “DualGAN: Unsupervised Dual Learning for Image-to-Image Translation,” in Proc. IEEE Int. Conf. Computer Vision (ICCV), Venice, Italy, 2017, pp. 2868–2876. [Online]. Available: <https://arxiv.org/abs/1704.02510>

[86] Z. Wei, S. Li, Z. Zheng, J. Wu, and X. Jing, “Person Transfer GAN to Bridge Domain Gap for Person Re-Identification,” in Proc. IEEE Int. Conf. Computer Vision (ICCV), Seoul, South Korea, 2019. [Online]. Available: <https://arxiv.org/abs/1811.07405>

[87] D. Deng, X. Chen, Z. Xue, and Y. Tian, “Similarity-Preserving Generative Adversarial Network for Unsupervised Person Re-Identification,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 2018, pp. 10797–10806. [Online]. Available: <https://arxiv.org/abs/1805.10804>

[88] W. Qian, Y. Li, L. Lin, and J. Chen, “Pose-Normalization GAN for Person Re-Identification,” in Proc. AAAI Conf. Artificial Intelligence (AAAI), New Orleans, LA, USA, 2018, pp. 8562–8569. [Online]. Available: <https://arxiv.org/abs/1803.05679>

[89] R. Zhang, Z. Cao, and H. Sun, 'Self-attention network for person re-identification,' Signal Process. Image Commun., vol. 96, p. 116263, 2021, doi: 10.1016/j.image.2021.116263.

[90] Rusyn. B., Lutsyk. O., Kosarevych. R., Rethinking Deep CNN Training: A Novel Approach for Quality-Aware Dataset Optimization. IEEE Access, vol.12, pp.137427-137438. doi.org/10.1109/ACCESS.2024.3414651.

[91] N. Parmar, A. Vaswani, J. Uszkoreit, et al., 'Image Transformer,' in Proc. Int. Conf. Mach. Learn. (ICML), 2018, pp. 4055–4064, <https://arxiv.org/abs/1802.05751>.

[92] H. Touvron, M. Cord, A. Sablayrolles, G. Bojanowski, and H. Jégou, "Training data efficient image transformers & distillation through attention," in Proc. 38th Int. Conf. Mach. Learn., 2021, pp. 10347–10357, doi: 10.48550/arXiv.2012.12877

[93] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin Transformer: Hierarchical vision transformer using shifted windows," arXiv preprint arXiv:2103.14030, 2021, doi: 10.48550/arXiv.2103.14030arxiv

[94] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z. Jiang, F. E. H. Tay, J. Feng, and S. Yan, "Tokens to Token ViT: Training vision transformers from scratch on ImageNet," arXiv preprint arXiv:2101.11986, 2021, doi: 10.48550/arXiv.2101.11986

[95] C. F. (Richard) Chen, Q. Fan, and R. Panda, "CrossViT: Cross attention multi scale vision transformer for image classification," in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), 2021, pp. 357–366, doi: 10.48550/arXiv.2103.14899

[96] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, "CvT: Introducing convolutions to vision transformers," arXiv preprint arXiv:2103.15808, 2021, doi: 10.48550/arXiv.2103.15808

[97] W. Liu, Y. Wen, Z. Yu, and M. Yang, 'SphereFace: Deep hypersphere embedding for face recognition,' in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2017, pp. 6738–6746, doi: 10.1109/CVPR.2017.714.

[98] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, 'CosFace: Large margin cosine loss for deep face recognition,' in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018, pp. 5265–5274, doi: 10.1109/CVPR.2018.00552.

[99] W. Liu, Y. Wen, Z. Yu, and M. Yang, 'Large-margin softmax loss for convolutional neural networks,' in Proc. Int. Conf. Mach. Learn. (ICML), 2016, pp. 507–516, <https://arxiv.org/abs/1602.03451>.

[100] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, 'A discriminative feature learning approach for deep face recognition,' in Proc. Eur. Conf. Comput. Vis. (ECCV), 2016, pp. 499–515, doi: 10.1007/978-3-319-46478-7\_31.

[101] K. Cao, Y. Shen, J. Xie, et al., 'SphereFace2: Make feature learning adaptive again,' IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 12, pp. 8766–8780, 2022, doi: 10.1109/TPAMI.2021.3124524.

[102] L. Zheng L. Shen L. Tian S. Wang J. Wang and Q. Tian "Scalable person re-identification: A benchmark" Proc. IEEE Int. Conf. Comput. Vis. pp. 1116-1124 Jun. 2015.

[103] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In ECCVW, 2016.

[104] Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. Deepreid: Deep filter pairing neural network for person reidentification. In CVPR, 2014.

[105] Evaluation Metrics. Cumulative Matching Characteristics. [Электронный ресурс] – Режим доступа: [https://cysu.github.io/open-reid/notes/evaluation\\_metrics.html](https://cysu.github.io/open-reid/notes/evaluation_metrics.html)

[106] D. Ha, A. Dai, and Q. V. Le, “HyperNetworks,” arXiv preprint arXiv:1609.09106, 2016, doi: 10.48550/arXiv.1609.09106a

[107] V. Dwaracherla, X. Lu, M. Ibrahimi, I. Osband, Z. Wen, and B. Van Roy, “Hypermodels for exploration,” arXiv preprint arXiv:2006.07464, 2020, doi: 10.48550/arXiv.2006.07464

[108] B. De Brabandere, X. Jia, T. Tuytelaars, and L. Van Gool, “Dynamic filter networks,” arXiv preprint arXiv:1605.09673, 2016, doi: 10.48550/arXiv.1605.09673

[109] J. von Oswald, C. Henning, B. F. Grewe, and J. Sacramento, “Continual learning with hypernetworks,” arXiv preprint arXiv:1906.00695, 2019, doi: 10.48550/arXiv.1906.00695

[110] J. Beck, M. T. Jackson, R. Vuorio, and S. Whiteson, “Hypernetworks in meta reinforcement learning,” arXiv preprint arXiv:2210.11348, 2022, doi: 10.48550/arXiv.2210.11348

- [111] W. Cho, K. Lee, D. Rim, and N. Park, “Hypernetwork based meta learning for low rank physics informed neural networks,” arXiv preprint arXiv:2310.09528, 2023, doi: 10.48550/arXiv.2310.09528
- [112] S. Schug, S. Kobayashi, Y. Akram, J. Sacramento, and R. Pascanu, “Attention as a hypernetwork,” arXiv preprint arXiv:2406.05816, 2024, doi: 10.48550/arXiv.2406.05816
- [113] K. Książek and P. Spurek, “HyperMask: Adaptive hypernetwork based masks for continual learning,” arXiv preprint arXiv:2310.00113, 2023, doi: 10.48550/arXiv.2310.00113
- [114] V. K. Chauhan, J. Zhou, P. Lu, S. Molaei, and D. A. Clifton, “A brief review of hypernetworks in deep learning,” Artificial Intelligence Review (accepted), 2023, doi: 10.48550/arXiv.2306.06955
- [115] D. P. Kingma and J. Ba, 'Adam: A method for stochastic optimization,' in Proc. Int. Conf. Learn. Representations (ICLR), 2015, <https://arxiv.org/abs/1412.6980>
- [116] A. Paszke, S. Gross, F. Massa, et al., “Automatic differentiation in PyTorch,” in Proc. 31st Conf. Neural Inf. Process. Syst. Workshops (NeurIPS Workshops), 2017. doi: 10.48550/arXiv.1706.02677
- [117] K. Zhou and T. Xiang, “Torchreid: A library for deep learning person re-identification in PyTorch,” arXiv preprint arXiv:1910.10093, 2019. doi: 10.48550/arXiv.1910.10093
- [118] OpenMMLab community, “MMReID: A comprehensive toolbox for deep learning-based person re-identification,” GitHub repository, 2021. [Online]. Available: <https://github.com/open-mmlab/mmreid>
- [119] G. Bradski and A. Kaehler, Learning OpenCV: Computer Vision with the OpenCV Library, Sebastopol, CA: O’Reilly Media, 2008.
- [120] C. R. Harris, K. J. Millman, S. J. van der Walt, et al., “Array programming with NumPy,” Nature, vol. 585, pp. 357–362, Sep. 2020. doi: 10.1038/s41586-020-2649-2
- [121] F. Pedregosa, G. Varoquaux, A. Gramfort, et al., “Scikit-learn: Machine learning in Python,” J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.

[122] W. McKinney, “Data structures for statistical computing in Python,” in Proc. 9th Python Sci. Conf. (SciPy), 2010, pp. 56–61.

[123] J. D. Hunter, “Matplotlib: A 2D graphics environment,” Comput. Sci. Eng., vol. 9, no. 3, pp. 90–95, May/Jun 2007. doi: 10.1109/MCSE.2007.55

[124] M. Abadi, P. Barham, J. Chen, et al., “TensorFlow: A system for large-scale machine learning,” in Proc. 12th USENIX Symp. Oper. Syst. Des. Implement. (OSDI), 2016, pp. 265–283.

[125] Amazon Web Services, “Amazon SageMaker Developer Guide,” AWS, 2017. [Online]. Available: <https://docs.aws.amazon.com/sagemaker/latest/dg/> (accessed Aug. 7, 2025).

[126] Google, “Google Colaboratory,” Google Research, 2017. [Online]. Available: <https://colab.research.google.com/> (accessed Aug. 7, 2025).

[127] Kaggle Inc., “Kaggle: Your Machine Learning and Data Science Community,” Kaggle, 2010. [Online]. Available: <https://www.kaggle.com/> (accessed Aug. 7, 2025).

[128] Microsoft Corporation, “Azure Notebooks,” Microsoft Azure, 2016. [Online]. Available: <https://notebooks.azure.com/> (accessed Aug. 7, 2025).

[129] JetBrains s.r.o., “PyCharm: Python IDE for Professional Developers,” JetBrains, 2010. [Online]. Available: <https://www.jetbrains.com/pycharm/> (accessed Aug. 7, 2025).

[130] FootfallCam Ltd., “FootfallCam 3D Pro2 — People Counting System,” FootfallCam. [Online]. Available: <https://www.footfallcam.com/en/Product/FootfallCam-3D-Pro2> . Accessed: Aug. 11, 2025.

[131] V-Count Inc., “Nano AI — Compact & Accurate People Counter,” V-Count. [Online]. Available: <https://v-count.com/nano/> . Accessed: Aug. 11, 2025.

[132] RetailNext Inc., “Aurora,” RetailNext. [Online]. Available: <https://retailnext.net/product/aurora> . Accessed: Aug. 11, 2025.

[133] Xovis AG, “People counting sensors,” Xovis. [Online]. Available: <https://www.xovis.com/technology/sensor> . Accessed: Aug. 11, 2025.

[134] Sensormatic Solutions (Johnson Controls), “People Counting: Know Your Store Traffic,” Sensormatic. [Online]. Available: <https://www.sensormatic.com/shopper-insights/people-counting> . Accessed: Aug. 11, 2025.

[135] Hikvision, “People Counting — Solutions by Function,” Hikvision. [Online]. Available: <https://www.hikvision.com/en/solutions/solutions-by-function/people-counting/> . Accessed: Aug. 11, 2025.

[136] Axis Communications, “AXIS People Counter,” Axis. [Online]. Available: <https://www.axis.com/products/axis-people-counter> . Accessed: Aug. 11, 2025.

## **ДОДАТКИ**

## ДОДАТОК А

## ТАБЛИЦЯ ОЦНОК МОДЕЛЕЙ ДЛЯ ВІДКРИТИХ ДАТАСЕТІВ

Run	Market_OSN et_mAP	Market_H N_mAP	Market_Δ _mAP	Market_OS Net_R1	Market_H N_R1	Market_ Δ R1
1	85.05	87.58	2.53	94.88	96.1	1.23
2	84.86	87.51	2.65	94.92	96.17	1.25
3	85.09	87.75	2.66	95.04	96.41	1.37
4	85.36	87.96	2.6	95.03	96.39	1.35
5	84.83	87.51	2.68	94.5	95.8	1.3
6	84.83	87.58	2.75	94.59	95.9	1.31
7	85.37	88.3	2.93	94.91	96.33	1.41
8	85.13	87.85	2.72	94.91	96.16	1.25
9	84.76	87.49	2.73	94.91	96.26	1.35
10	85.06	87.75	2.69	95.65	96.93	1.28
11	84.76	87.23	2.47	94.93	96.21	1.28
12	84.76	87.46	2.7	95.05	96.45	1.4
13	84.97	87.68	2.71	95.01	96.38	1.37
14	84.33	87.32	3.0	94.94	96.32	1.37
15	84.38	87.06	2.68	94.73	96.15	1.42
16	84.73	87.47	2.74	94.97	96.27	1.3
17	84.6	87.29	2.7	94.63	95.99	1.36
18	84.99	87.55	2.56	94.75	96.02	1.27
19	84.63	87.46	2.84	94.69	96.02	1.33
20	84.48	87.27	2.79	94.82	96.11	1.29
21	85.34	88.13	2.79	95.31	96.62	1.31
22	84.83	87.42	2.59	94.39	95.74	1.35
23	84.92	87.79	2.87	94.95	96.18	1.23
24	84.47	87.0	2.53	94.45	95.93	1.49
25	84.74	87.51	2.77	94.7	95.91	1.21
26	84.93	87.9	2.96	95.04	96.23	1.19
27	84.55	87.14	2.58	94.81	96.22	1.4
28	85.01	87.64	2.63	94.56	95.93	1.37
29	84.72	87.43	2.71	94.64	96.0	1.36
30	84.81	87.45	2.64	94.95	96.31	1.36
31	84.72	87.23	2.51	94.64	95.94	1.3
32	85.46	88.16	2.71	94.85	96.07	1.22
33	84.9	87.47	2.57	94.81	96.12	1.31
34	84.58	87.34	2.76	94.66	95.9	1.24
35	85.15	87.74	2.59	95.27	96.66	1.39
36	84.53	87.42	2.89	94.94	96.23	1.29
37	84.96	87.57	2.61	94.35	95.58	1.23

38	84.31	86.97	2.66	94.84	96.11	1.27
39	84.5	87.3	2.8	94.65	95.99	1.34
40	84.96	87.51	2.55	94.99	96.24	1.25
41	85.12	87.85	2.73	94.63	95.85	1.23
42	84.95	87.81	2.86	94.77	96.1	1.32
43	84.87	87.37	2.51	94.91	96.23	1.32
44	84.81	87.53	2.72	94.99	96.24	1.25
45	84.46	87.19	2.73	94.54	95.79	1.26
46	84.68	87.48	2.79	94.73	96.05	1.32
47	84.76	87.31	2.55	94.7	95.87	1.17
48	85.22	87.76	2.54	94.66	95.83	1.17
49	85.0	87.77	2.76	95.19	96.42	1.24
50	84.37	87.11	2.74	94.89	96.17	1.28
51	85.0	87.73	2.73	94.52	95.85	1.33
52	84.78	87.53	2.74	95.0	96.43	1.43
53	84.7	87.32	2.62	95.27	96.64	1.38
54	85.08	87.81	2.73	95.03	96.31	1.29
55	85.21	87.94	2.74	94.47	95.76	1.3
56	85.18	87.79	2.61	94.69	95.9	1.21
57	84.65	87.57	2.92	95.08	96.38	1.3
58	84.81	87.56	2.76	94.64	95.92	1.27
59	85.0	87.56	2.56	94.9	96.23	1.33
60	85.19	87.97	2.78	94.97	96.2	1.23
61	84.76	87.34	2.58	94.6	95.94	1.35
62	84.84	87.64	2.79	94.79	96.22	1.44
63	84.57	87.41	2.84	94.09	95.38	1.29
64	84.54	87.14	2.6	94.57	95.91	1.34
65	85.14	87.96	2.82	94.74	96.11	1.36
66	85.31	88.06	2.75	94.53	95.79	1.26
67	84.88	87.68	2.8	95.16	96.48	1.32
68	85.2	88.13	2.93	94.49	95.79	1.3
69	85.01	87.68	2.67	94.7	96.01	1.31
70	84.71	87.32	2.61	94.83	96.06	1.23
71	85.01	87.6	2.59	95.12	96.42	1.3
72	85.36	87.96	2.6	94.48	95.83	1.34
73	84.89	87.58	2.69	95.06	96.49	1.43
74	85.37	88.11	2.74	94.8	96.19	1.39
75	84.11	86.85	2.73	94.58	96.08	1.49
76	85.15	87.95	2.8	94.9	96.13	1.23
77	84.93	87.63	2.7	94.84	96.22	1.38
78	84.81	87.68	2.87	94.67	95.98	1.32
79	84.93	87.6	2.67	94.82	96.31	1.5
80	84.3	87.33	3.03	94.72	95.94	1.23

81	84.83	87.61	2.78	94.82	96.05	1.22
82	85.01	87.6	2.6	94.95	96.19	1.25
83	85.34	87.91	2.57	95.15	96.26	1.11
84	84.74	87.5	2.76	94.53	95.78	1.25
85	84.66	87.33	2.67	95.27	96.5	1.23
86	84.75	87.54	2.79	94.37	95.68	1.31
87	85.17	87.93	2.76	94.77	96.1	1.33
88	85.0	87.69	2.69	94.93	96.4	1.47
89	84.74	87.34	2.6	94.86	96.25	1.39
90	85.05	87.57	2.52	94.66	95.91	1.25
91	84.93	87.58	2.65	94.75	95.97	1.22
92	85.19	87.99	2.8	94.69	96.04	1.34
93	84.69	87.42	2.73	94.67	95.85	1.18
94	84.8	87.35	2.55	94.99	96.45	1.46
95	84.78	87.5	2.72	94.88	96.28	1.41
96	84.46	87.21	2.75	94.65	95.91	1.26
97	84.99	87.58	2.59	95.0	96.14	1.15
98	84.98	87.7	2.72	94.87	96.29	1.42
99	84.9	87.61	2.71	94.98	96.27	1.29
100	84.83	87.39	2.56	94.94	96.35	1.41

Run	Duke_OSN et_mAP	Duke_HN _mAP	Duke_Δ_ mAP	Duke_OS Net_R1	Duke_HN _R1	Duke_Δ_ R1
1	62.08	65.87	3.79	81.96	84.19	2.22
2	62.53	66.51	3.98	81.38	83.83	2.46
3	62.8	66.12	3.32	82.0	84.2	2.19
4	62.82	66.53	3.71	82.17	84.26	2.09
5	62.6	66.07	3.47	81.84	83.91	2.07
6	63.08	66.58	3.5	82.36	84.75	2.39
7	62.32	65.8	3.48	81.43	83.54	2.11
8	62.74	66.16	3.43	81.26	83.83	2.56
9	62.85	66.46	3.61	81.08	83.07	1.99
10	63.03	66.47	3.43	82.22	84.78	2.55
11	63.12	66.77	3.65	81.93	84.26	2.33
12	62.29	65.88	3.59	81.68	83.97	2.29
13	62.11	65.66	3.55	81.8	84.02	2.22
14	63.37	66.79	3.42	81.31	83.67	2.36
15	62.95	66.43	3.48	82.56	84.85	2.29
16	62.46	66.21	3.75	81.75	84.21	2.47
17	63.5	67.2	3.7	81.74	84.06	2.32
18	62.85	66.26	3.4	81.95	84.28	2.32

19	63.33	66.95	3.62	81.87	84.11	2.25
20	62.83	66.58	3.75	81.78	84.07	2.29
21	63.73	66.99	3.27	81.42	83.77	2.35
22	63.59	67.3	3.71	81.87	83.91	2.04
23	62.69	66.16	3.47	82.36	84.46	2.1
24	63.24	66.95	3.71	82.17	84.58	2.41
25	63.09	66.54	3.45	82.26	84.58	2.33
26	63.42	66.65	3.24	81.52	83.79	2.27
27	62.37	65.64	3.27	81.35	83.66	2.3
28	63.11	66.72	3.61	81.66	84.01	2.35
29	63.28	66.93	3.65	81.72	83.94	2.22
30	62.01	65.43	3.42	82.08	84.27	2.18
31	62.27	66.0	3.73	81.11	83.44	2.33
32	61.88	65.15	3.27	82.24	84.39	2.15
33	62.68	66.27	3.59	81.64	84.01	2.36
34	63.12	66.48	3.36	81.55	83.6	2.04
35	63.48	66.95	3.47	81.35	83.8	2.45
36	62.83	66.44	3.61	81.12	83.49	2.37
37	63.53	66.96	3.43	81.99	84.33	2.34
38	62.18	65.7	3.52	81.73	84.17	2.45
39	62.03	65.83	3.8	81.25	83.8	2.55
40	62.78	66.26	3.48	81.25	83.7	2.45
41	62.97	66.74	3.77	81.58	83.61	2.02
42	62.79	66.16	3.37	82.28	84.39	2.11
43	61.87	65.58	3.71	81.61	83.82	2.21
44	62.76	66.65	3.89	81.17	83.48	2.3
45	62.21	65.32	3.11	81.61	83.99	2.38
46	63.1	66.54	3.44	81.6	83.8	2.19
47	62.96	66.68	3.72	80.76	83.08	2.33
48	62.38	65.94	3.56	81.68	83.87	2.19
49	62.57	66.24	3.67	81.62	83.83	2.21
50	62.32	65.8	3.48	81.94	84.03	2.09
51	62.77	66.39	3.62	82.35	84.51	2.16
52	63.23	66.8	3.57	82.09	84.19	2.1
53	62.36	66.19	3.83	81.61	83.76	2.15
54	63.03	66.68	3.65	81.31	83.77	2.46
55	62.56	66.23	3.67	82.6	84.76	2.16
56	62.44	65.96	3.52	81.72	84.42	2.69
57	62.75	66.25	3.5	81.7	84.08	2.37
58	62.33	65.85	3.51	81.69	84.02	2.33
59	62.55	66.23	3.68	81.77	83.94	2.17
60	62.26	65.78	3.52	81.65	84.05	2.41
61	63.68	67.34	3.66	81.5	83.71	2.21

62	62.82	66.83	4.02	81.51	83.83	2.32
63	62.49	66.26	3.77	81.69	84.37	2.68
64	62.9	66.43	3.53	81.51	83.8	2.29
65	62.75	66.59	3.84	81.45	83.92	2.47
66	62.7	66.22	3.52	81.74	83.93	2.19
67	63.08	66.27	3.19	81.61	83.91	2.29
68	63.14	66.54	3.4	82.23	84.79	2.57
69	62.56	65.79	3.23	80.77	82.98	2.21
70	62.54	66.07	3.53	82.08	84.65	2.57
71	62.68	66.28	3.6	82.14	84.54	2.41
72	61.76	65.7	3.94	80.97	83.19	2.22
73	62.12	65.78	3.67	81.58	83.97	2.39
74	63.42	66.97	3.56	81.57	84.02	2.45
75	63.54	67.31	3.77	81.21	83.6	2.39
76	62.69	65.85	3.16	81.43	83.49	2.06
77	63.06	66.71	3.65	81.31	83.5	2.19
78	62.94	66.69	3.75	82.31	84.58	2.26
79	64.19	67.49	3.3	82.03	84.32	2.29
80	63.3	67.13	3.83	82.15	84.54	2.39
81	62.74	66.41	3.67	81.95	84.28	2.33
82	62.37	65.89	3.52	81.3	83.4	2.1
83	62.08	65.8	3.73	81.52	83.87	2.36
84	62.89	66.95	4.05	81.87	84.26	2.39
85	62.46	66.1	3.64	81.27	83.66	2.38
86	62.16	65.81	3.65	81.95	84.41	2.46
87	62.51	66.02	3.51	81.62	84.04	2.43
88	62.31	65.74	3.43	81.57	83.94	2.37
89	63.56	67.33	3.77	81.95	84.24	2.29
90	63.2	66.63	3.43	81.86	83.91	2.05
91	62.8	66.41	3.61	81.57	83.94	2.36
92	63.47	66.97	3.5	82.11	84.44	2.33
93	62.83	66.53	3.7	81.32	83.66	2.34
94	62.41	66.08	3.67	81.92	84.02	2.11
95	63.49	67.29	3.81	81.91	84.05	2.14
96	63.04	66.54	3.5	81.59	84.05	2.46
97	62.33	65.88	3.55	81.81	84.11	2.29
98	62.71	66.12	3.4	81.26	83.66	2.4
99	62.41	65.92	3.51	82.02	84.33	2.3
100	62.18	65.85	3.68	81.64	83.94	2.3

Эксперимент	mAP (пропозитивный метод), %	Rank-1 (пропозитивный метод), %	mAP (стандартный OSNet), %	Rank-1 (стандартный OSNet), %	$\Delta$ mAP, %	$\Delta$ Rank-1, %
1	76.65	91.76	73.5	89.93	3.15	1.83
2	77.58	91.38	72.77	89.88	4.81	1.5
3	75.54	90.77	72.53	90.27	3.01	0.5
4	74.28	90.72	73.24	89.04	1.03	1.68
5	75.09	90.29	71.99	90.16	3.1	0.14
6	76.07	90.29	74.47	89.89	1.6	0.4
7	74.85	91.19	72.46	90.06	2.39	1.13
8	75.4	91.93	72.4	89.85	3.0	2.07
9	76.82	90.39	72.99	89.47	3.84	0.92
10	74.67	91.1	73.21	89.02	1.46	2.08
11	75.88	90.85	73.74	90.09	2.15	0.76
12	75.54	91.53	71.52	89.64	4.02	1.89
13	76.32	90.81	73.34	89.12	2.98	1.69
14	77.03	91.47	72.32	90.31	4.71	1.16
15	76.33	91.49	72.16	89.85	4.17	1.64
16	74.89	90.4	72.52	89.91	2.37	0.49
17	75.93	91.5	73.81	90.68	2.12	0.82
18	76.36	91.77	73.36	89.68	3.0	2.09
19	73.38	91.41	72.96	90.78	0.42	0.63
20	76.09	90.01	73.09	89.85	3.0	0.16
21	77.48	90.74	72.78	90.18	4.7	0.56
22	76.92	91.16	72.19	89.75	4.72	1.42
23	76.1	91.48	72.47	90.26	3.63	1.23
24	75.61	90.27	72.3	89.84	3.31	0.43
25	76.01	90.88	73.3	90.13	2.71	0.75
26	75.66	90.6	71.58	89.79	4.07	0.81
27	77.89	91.09	72.84	90.2	5.05	0.89
28	74.08	90.99	73.26	89.96	0.82	1.02
29	75.81	91.15	73.06	91.23	2.75	-0.08
30	77.14	91.38	72.97	89.42	4.18	1.96
31	77.4	90.3	73.79	89.55	3.61	0.75
32	75.01	90.72	73.59	91.1	1.42	-0.38
33	74.45	91.03	73.1	89.75	1.35	1.29
34	75.08	91.77	71.94	90.24	3.14	1.54
35	76.81	90.38	72.22	89.84	4.6	0.55
36	74.39	91.09	73.23	90.65	1.17	0.44
37	74.76	90.34	73.26	90.39	1.5	-0.05
38	76.25	91.17	73.52	90.15	2.73	1.02

39	76.29	90.64	72.32	90.12	3.97	0.53
40	74.81	91.33	74.87	90.24	-0.06	1.09
41	77.16	90.59	72.03	90.39	5.13	0.2
42	76.82	91.95	73.96	90.21	2.86	1.74
43	75.11	90.59	72.75	89.62	2.36	0.97
44	76.28	91.41	72.92	90.17	3.35	1.24
45	75.74	92.36	73.01	90.73	2.72	1.63
46	74.93	91.24	73.63	89.57	1.3	1.67
47	76.47	90.96	72.78	90.36	3.7	0.61
48	75.55	91.43	72.15	89.24	3.4	2.19
49	76.17	91.19	73.21	89.38	2.96	1.82
50	76.06	90.43	72.12	90.08	3.94	0.35

Эксперимент	mAP (пропонованный метод), %	Rank-1 (пропонованный метод), %	mAP (стандартный OSNet), %	Rank-1 (стандартный OSNet), %	$\Delta mAP$ , %	$\Delta Rank-1$ , %
1	70.08	84.03	65.36	82.28	4.73	1.75
2	69.52	83.76	63.62	81.53	5.89	2.23
3	69.57	84.07	65.52	83.93	4.06	0.14
4	68.68	83.88	65.95	82.33	2.73	1.55
5	68.51	83.54	64.23	81.88	4.29	1.66
6	69.69	82.69	67.31	81.07	2.37	1.63
7	69.06	82.96	64.53	82.54	4.54	0.42
8	68.27	83.61	64.28	82.34	3.98	1.27
9	71.14	83.82	65.05	81.67	6.1	2.14
10	68.34	83.93	62.97	82.09	5.36	1.83
11	69.5	83.93	64.21	81.94	5.3	1.99
12	68.53	83.17	63.8	81.83	4.73	1.34
13	67.74	83.96	66.77	82.2	0.97	1.76
14	67.48	83.26	67.12	82.52	0.36	0.74
15	69.44	83.89	66.27	81.65	3.18	2.24
16	65.76	82.99	64.07	81.97	1.69	1.02
17	70.63	82.78	64.75	81.38	5.88	1.41
18	70.44	82.78	64.56	82.07	5.88	0.72
19	68.02	83.73	66.16	82.01	1.86	1.73
20	69.07	83.31	65.2	81.7	3.87	1.61
21	70.59	82.88	65.11	82.33	5.47	0.55
22	68.85	83.79	67.13	81.02	1.72	2.77
23	68.79	83.25	65.28	81.69	3.51	1.56
24	69.36	83.15	64.41	82.42	4.95	0.73

25	69.81	83.81	65.9	82.15	3.91	1.66
26	69.75	83.81	64.17	81.72	5.58	2.09
27	70.28	83.2	64.98	82.06	5.3	1.15
28	68.78	84.05	65.55	81.9	3.24	2.15
29	70.31	83.51	65.83	82.41	4.48	1.1
30	69.32	83.43	65.68	81.84	3.64	1.59
31	68.18	84.55	65.1	82.3	3.08	2.25
32	70.16	83.9	63.99	81.39	6.16	2.5
33	68.99	83.05	65.62	82.31	3.36	0.74
34	69.98	83.43	65.08	81.66	4.9	1.77
35	69.41	83.22	64.17	81.84	5.24	1.38
36	69.24	83.25	64.18	82.12	5.07	1.12
37	67.55	82.8	64.53	82.12	3.02	0.68
38	69.31	84.24	64.28	81.89	5.03	2.34
39	68.98	83.0	65.86	81.92	3.12	1.08
40	69.32	83.09	64.98	81.86	4.34	1.23
41	68.89	83.7	65.52	82.77	3.37	0.93
42	69.22	83.51	65.69	81.8	3.53	1.71
43	69.02	83.75	65.1	81.61	3.93	2.14
44	71.15	83.12	66.45	82.48	4.7	0.64
45	71.19	83.1	65.87	82.09	5.32	1.0
46	66.88	83.24	64.16	81.7	2.72	1.54
47	69.34	84.44	64.24	82.08	5.1	2.36
48	68.1	83.75	65.95	81.71	2.15	2.03
49	70.18	83.27	63.68	82.92	6.5	0.35
50	68.89	84.12	63.29	82.68	5.6	1.44

## ДОДАТОК Б

### СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ

**Наукові праці, в яких опубліковані основні наукові результати дисертації:**

[1] Р. Н. Кветний, Р. В. Маслій і О. М. Кириленко, “Виявлення та класифікація об’єктів дорожнього руху при використанні середовища DIGITS” *Оптико-електронні інформаційно-енергетичні технології*, вип. 39, вип. 1, с. 14–20, Січ 2021. doi: <https://doi.org/10.31649/1681-7893-2020-39-1-14-20> . (Index Copernicus, Наукове фахове видання України, категорія «Б» зі спеціальності 126)

[2] Р. Н. Кветний, Р. В. Маслій, О. М. Кириленко і В. В. Щерба “Дослідження нейромережевого підходу виявлення об’єктів у зображеннях” *Науково-виробничий журнал “Метрологія та прилади”*, no. 6, pp. 15–21, Січ. 2020. doi: [http://dx.doi.org/10.33955/2307-2180\(6\)2019.15-21](http://dx.doi.org/10.33955/2307-2180(6)2019.15-21). (Index Copernicus, Наукове фахове видання України, категорія «Б» зі спеціальності 126)

[3] O. Kyrylenko, R. Maslii and Y. Marushchak, “Analysis of methods of person reidentification in multi camera environment,” *Norwegian Journal of Development of the International Science*, no. 47, pp. 46–49, Oct. 2020. [Online]. Available: [http://www.norwegian-journal.com/wp-content/uploads/2020/10/NJD\\_47\\_1.pdf](http://www.norwegian-journal.com/wp-content/uploads/2020/10/NJD_47_1.pdf).

(Періодичне закордонне видання)

[4] Р. Н. Кветний, Р. В. Маслій і О. М. Кириленко, “Дослідження атрибутів людини для задачі повторної ідентифікації” *Інформаційні технології та комп’ютерна інженерія*, Вінниця, Україна, 2020. [Online]. Available: <https://itce.vntu.edu.ua/index.php/itce/article/view/764/507>. (Index Copernicus, Наукове фахове видання України, категорія «Б» зі спеціальності 126)

[5] R. Kvyetnyy, V. Kotsiubynskyi, O. Kyrylenko et al., “Using multiple optical cameras for correspondence identification between objects in the fields of view” in *Proc. SPIE*, vol. 12985, art. 129850A, 20 Dec. 2023. doi: <https://doi.org/10.1117/12.3022791> . (Входить до Scopus)

[6] R. Kvyetnyy, V. Kotsiubynskiy, O. Kyrylenko et al., “Re-identification of people in a video stream based on a Kalman filter” in *Proc. SPIE*, vol. 13400, art. 134000R, 16 Dec. 2024. doi: <https://doi.org/10.1117/12.3058549>. (Входить до Scopus)

[7] Р. Н. Кветний, В. В. Гармаш, О. М. Кириленко і В. Ю. Коцюбинський, “Застосування адаптивних гіпермоделей у задачах ре-ідентифікації людини” *Вісник ВПІ*, no. 2, pp. 138–146, 2025. doi: <https://doi.org/10.31649/1997-9266-2025-180-3>. (Index Copernicus, Наукове фахове видання України, категорія «Б» зі спеціальності 126)

[8] О. М. Кириленко, «Розробка методу повторної ідентифікації людини», *Опт-ел. інф-енерг. техн.*, вип. 41, вип. 1, с. 25–32, Трав 2022. doi: <https://doi.org/10.31649/1681-7893-2021-41-1-25-32>. (Index Copernicus, Наукове фахове видання України, категорія «Б» зі спеціальності 126)

### **Наукові праці, які засвідчують апробацію матеріалів дисертації:**

[9] О. М. Кириленко, “Сучасні методи виявлення та відстеження облич у складних умовах” *XLVI науково-технічна конференція ФКСА*, 2017, Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-017/paper/view/2003/2635>

[10] О. М. Кириленко, “Аналіз методів фільтрації цифрових зображень” *Молодь в технічних науках: дослідження, проблеми, перспективи (MTN-2017)*, 2017, Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: <http://conf.inmad.vntu.edu.ua/fm/index.php?page=materials&line=29&mat=373>

[11] О. М. Кириленко, “Сучасні методи виявлення облич з використанням нейронних мереж” *Вимірювальна та обчислювальна техніка в технологічних процесах (ВОТТП-2017)*, Харків, Україна, 2017. [Електронний ресурс]. Режим доступу: [http://journals.khnu.km.ua/vottp/pdf/pdf\\_full/vottp-2017-konf.pdf](http://journals.khnu.km.ua/vottp/pdf/pdf_full/vottp-2017-konf.pdf)

[12] О. М. Кириленко, Р. В. Маслій і В. В. Гармаш, “Системи розпізнавання об’єктів у зображеннях, отриманих у складних умовах освітлення” *IV міжнар.*

конференція «Вимірювання, контроль та діагностика в технічних системах (ВКДТС-2017)», 2017, Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: <http://mpa.vntu.edu.ua/images/conference/conf2017/VCDTS%202017.pdf>

[13] О. М. Кириленко і Р. Н. Кветний, “Аналіз методів відстеження облич у відеопослідовностях” *IV міжнар. конференція «Датчики, прилади та системи (ДСП-2017)»*, Черкаси, Україна, 2017. [Електронний ресурс]. Режим доступу: <https://er.chdtu.edu.ua/bitstream/ChSTU/820/1/%D0%A1%D0%B1%D0%BE%D1%80%D0%BD%D0%B8%D0%BA%20%D1%82%D0%B5%D0%B7%D0%B8%D1%81%D0%BE%D0%B2%20%D0%94%D0%9F%D0%A1-2017.pdf#page=33>

[14] О. М. Кириленко, Р. Н. Кветний, О. В. Бісікало, Р. В. Маслій і В. Ю. Коцюбинський, “Розв’язання задачі підрахунку об’єктів при складних умовах отримання зображень” *ІТСМ-2018*, Івано-Франківськ, Україна, 2018. [Електронний ресурс]. Режим доступу: <http://itcm.comp-sc.if.ua/2018/kvetnyj.pdf>

[15] О. М. Кириленко і Р. В. Маслій, “Аналіз підходів до підрахунку кількості об’єктів у відеопослідовностях” *XLVII науково-технічна конференція ФКСА*, 2018, Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: <https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/20668/5022.pdf?sequence=3&isAllowed=y>

[16] О. М. Кириленко і Р. В. Маслій і, “Нейромережева архітектура detectnet в задачах виявлення об’єктів” *XLVIII науково-технічна конференція ФКСА*, 2019, Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-019/paper/view/7640/6467>

[17] Р. Н. Кветний, Р. В. Маслій і О. М. Кириленко, “Дослідження можливості застосування атрибутів market-1501 для задачі ре-ідентифікації людини” *КУСС-2020*, 8–10 Oct. 2020. Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: <https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/30576/10756-38288-1-RV.pdf?sequence=3&isAllowed=y>

[18] О. М. Кириленко, “Аналіз підходів повторної ідентифікації об’єктів” *L науково-технічна конференція ФКСА*, 2021, Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: <https://conferences.vntu.edu.ua/index.php/allfksa/allfksa2021/paper/view/12328>

[19] Р. Н. Кветний і О. М. Кириленко, “Дослідження трекінгу об’єктів для задачі їх повторної ідентифікації” *КУСС-2022*, 2022, Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: <https://conferences.vntu.edu.ua/index.php/mccs/mccs2022/paper/view/16463>

[20] Р. Н. Кветний і О. М. Кириленко, “Аналіз способів представлення об’єктів у зображеннях для задачі відстеження” *ЛІІ науково-технічна конференція ФІІТА*, 2023, Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2023/paper/view/17454>

[21] О. М. Кириленко і В. В. Гармаш, “Дослідження методів ре-ідентифікації осіб з декількох камер відеоспостереження” Збірник праць Х МНТК «Датчики, прилади та системи – 2023», Черкаси, Вер. 2023, pp. 100–101. [Електронний ресурс]. Режим доступу: <https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/38086/134336.pdf?sequence=2&isAllowed=y> .

[22] Р. Н. Кветний, О. М. Кириленко і В. В. Гармаш, “Дослідження використання PostgreSQL в інформаційній системі повторної ідентифікації людини” *КУСС-2024*, 16–17 Жов. 2024. Вінницький національний технічний університет, doi: <https://doi.org/10.31649/mccs2024.2-15>

[23] Р. Н. Кветний і О. М. Кириленко “Аналіз генеративних змагальних мереж для вирішення задач повторної ідентифікації” *LIV всеукр. наук.-тех. конференція ФІІТА-2025*, 2025, Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2025/paper/view/23851>

[24] Р. Н. Кветний і О. М. Кириленко “Аналіз використання domain adaptation GANs для задачі повторної ідентифікації людини” *Photonics-ODS 2025*, 2025, Вінницький національний технічний університет. [Електронний

ресурс]. Режим доступа: [https://conferences.vntu.edu.ua/index.php/ods/ods\\_2025/paper/view/23850](https://conferences.vntu.edu.ua/index.php/ods/ods_2025/paper/view/23850)

**ДОДАТОК В**  
**АКТ ПРО ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЙНОЇ**  
**РОБОТИ**



Товариство з обмеженою відповідальністю  
**“КВАРК КОНСАЛТИНГ”**

Україна, 21000, м. Вінниця, вул. 1 Травня, 36/2, тел. 067 722-78-48  
код ЄДРПОУ 38135120, ПІН 381351202284, свідоцтво платника ПДВ № 200140635,  
Р/р № 26005000003225 в АТ «Укресімбанк» у м. Київ, МФО 322313

---

**ЗАТВЕРДЖУЮ**  
Директор  
ТОВ «КВАРК КОНСАЛТИНГ»  
Алла СТУК  
*Алла СТУК*  
08 серпня 2025 р.

**АКТ**

про впровадження результатів дисертаційної роботи  
**«Інформаційна технологія повторної ідентифікації людей»**  
Кириленка Олександра Михайловича

Комісія у складі головного ІТ спеціаліста Житанського Сергія Ігоровича, старшого спеціаліста, розробника відділу моніторингу Дусанюка Сергія Вікторовича та інженера Кватернюка Максима Олеговича цим актом свідчить про впровадження на підприємстві результатів, отриманих у дисертаційній роботі Олександра КИРИЛЕНКА, а саме інформаційної технології повторної ідентифікації людей.

Під час апробації у 2025 році науково-технічної продукції на ТОВ «КВАРК КОНСАЛТИНГ» було проведено ряд експериментальних досліджень, які дозволяють стверджувати наступне:

1. Запропонована інформаційна технологія повторної ідентифікації людей, яка використовує інтеграцію гіпермоделей та генеративно-змагальних мереж із

OSNet архітектурою, що забезпечує динамічну адаптацію архітектури та параметрів нейронних мереж до варіації даних.

2. Розроблене програмне забезпечення, для використання реалізації запропонованої інформаційної технології, може бути застосоване на локальних серверах із обмеженою кількістю ресурсів, що підтверджує практичну цінність розробки.

3. Проведено експеримент з використанням розробленого програмного забезпечення для зональної аналітики у торговельному центрі, що охоплювала 12 попередньо визначених зон протягом однієї годинної сесії кожного дня. У першому експерименті відносна похибка коливалася від 87.2 % до 97.7 %, із середнім значенням 92.4 %; у другому — від 88.3 % до 93.4 %, із середнім показником 90.9 %. Також було проведено експерименти на окремих відеозаписах, таких як записані годинні ролики з низки IP-камер. Відношення ручного підрахунку до автоматичного склало від 89.7 % до 97,5 % у різних сесіях, із середнім значенням 93.6 %. Це демонструє стабільну роботу запропонованої інформаційної технології в умовах різних сценаріїв освітлення та ракурсів камер.

Розроблена інформаційна технологія повторної ідентифікації людей на основі інтеграції гіпермоделей та генеративно-змагальних мереж із OSNet архітектурою впроваджено у проект «Molengo» (замовник Transfotech Ltd Israel) для побудови детальних теплових карт, кластерного аналізу та поведінкових патернів відвідувачів. Практична цінність отриманих наукових результатів не викликає сумнівів.

Голова комісії

Сергій ЖИТАНСЬКИЙ

Члени комісії

Сергій ДУСАНЮК

Максим КВАТЕРНЮК



# ДОДАТОК Г

## АКТ ПРО ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ У НАВЧАЛЬНИЙ ПРОЦЕС ВНТУ



**ЗАТВЕРДЖУЮ**

Проректор з науково-педагогічної роботи та організації освітнього процесу  
Національного технічного університету

доц. Олександр ПЕТРОВ

«16» серпня 2025 р.

### АКТ

про впровадження результатів дисертаційної роботи  
Кириленка Олександра Михайловича

**«Інформаційна технологія повторної ідентифікації людей», представлена до захисту  
на здобуття наукового ступеня доктора філософії, в навчальному процесі**

Комісія Вінницького національного технічного університету у складі:

голова комісії – декан Факультету інтелектуальних інформаційних технологій та автоматизації (ФІТА), к.т.н, доцент Севастьянов В. М. та члени комісії – заступник декана з навчально-методичної роботи ФІТА, ст. викладач каф. САІТ Присяжнюк В. В., заступник завідувача кафедри Автоматизації та інтелектуальних інформаційних технологій (АІТ) ФІТА, к.т.н., професор Папінов В. М., склала цей акт про підтвердження того, що дійсно у Вінницькому національному технічному університеті під час викладання дисципліни «Розпізнавання графічних образів» для студентів, які навчаються за освітньою програмою «Інтелектуальні інформаційні системи» рівня «бакалавр» спеціальності 126 Інформаційні системи та технології, а також – дисципліни «Інформаційні технології обробки зображень» для студентів, які навчаються за освітньою програмою «Інформаційні технології аналізу даних та зображень» рівня «магістр» спеціальності 126 Інформаційні системи та технології впроваджено результати дисертаційної роботи на здобуття доктора філософії здобувача кафедри АІТ Кириленка Олександра Михайловича у ряді лекцій та лабораторних робіт з питань інформаційних технологій.

При викладанні цих дисциплін використовуються такі результати досліджень, отримані у дисертаційній роботі Кириленка О.М.:

- генеративне збагачення та доменна адаптація даних: синтез під конкретні камери й умови освітлення (адаптивні GAN), автоматизований відбір згенерованих зображень за ембеддинговими відстанями, інтеграція в пайплайн підготовки датасетів;
- тренування моделей машинного навчання: структурована підготовка даних, тренування CNN (OSNet) з гіпермоделлю, валідація, логування експериментів і контроль відтворюваності;
- розширене оцінювання якості моделей: обчислення mAP, CMC Rank-1/5/10, mINP, крос-доменні протоколи перевірки, порівняльний аналіз базових та вдосконалених конфігурацій.

Використання зазначених результатів дозволило підвищити якість навчального процесу із згаданих дисциплін.

Голова комісії

Володимир СЕВАСТ'ЯНОВ

Члени комісії

Василь ПРИСЯЖНЮК

Володимир ПАПІНОВ

## ДОДАТОК Г

## ЛІСТИНГ ПРОГРАМНИХ МОДУЛІВ ДЛЯ ТРЕНУВАННЯ МОДЕЛЕЙ

```

import os
import glob
import random
import logging
import time
import numpy as np
import torch
import torch.nn as nn
import torch.nn.functional as F
from torch.utils.data import Dataset, DataLoader
from torch.utils.tensorboard import SummaryWriter
from torchvision import transforms
from torchvision.utils import make_grid
from PIL import Image

from torch.optim.lr_scheduler import (
    CosineAnnealingLR,
    StepLR,
    OneCycleLR,
    CosineAnnealingWarmRestarts
)
from torchreid.utils.rerank import re_ranking
from torchreid.models import build_model

# ===== Utility: Log Retrieval Examples =====
def log_retrieval_examples(writer, query_imgs, gallery_imgs, topk_indices, step,
    num_examples=3):
    """
    Log grids combining one query image with its top-K retrieved gallery images.
    Each grid shows [query, top1, top2, ..., topK].

```

```

"""
for i in range(min(num_examples, topk_indices.shape[0])):
    # Prepare query and retrieved images
    query_img = query_imgs[i].unsqueeze(0) # shape [1, C, H, W]
    retrieved = gallery_imgs[topk_indices[i]] # shape [K, C, H, W]
    # Move query to same device
    retrieved_device = retrieved.device if isinstance(retrieved, torch.Tensor) else None
    if retrieved_device is not None:
        query_img = query_img.to(retrieved_device)
    # Combine: query first, then retrieved
    combined = torch.cat([query_img, retrieved], dim=0)
    # Create grid: single row of K+1 images
    grid = make_grid(combined, nrow=combined.size(0), normalize=True, scale_each=True)
    # Log the grid
    writer.add_image(f'Retrieval/Example_{i}', grid, step)

# ===== Reproducibility & Performance =====

seed = 42
random.seed(seed)
np.random.seed(seed)
torch.manual_seed(seed)
torch.cuda.manual_seed_all(seed)
torch.backends.cudnn.benchmark = True

# ===== Logging =====
logger = logging.getLogger("ReID_Trainer")
logger.setLevel(logging.INFO)
file_handler = logging.FileHandler("training.log", mode='w')
file_formatter = logging.Formatter("%(asctime)s - %(levelname)s - %(message)s")
file_handler.setFormatter(file_formatter)
logger.addHandler(file_handler)

```

```

# ===== WarmupScheduler =====
class WarmupScheduler(torch.optim.lr_scheduler._LRScheduler):
    def __init__(self, optimizer, warmup_epochs, base_lr, final_lr, last_epoch=-1):
        self.warmup_epochs = warmup_epochs
        self.base_lr = base_lr
        self.final_lr = final_lr
        super().__init__(optimizer, last_epoch)

    def get_lr(self):
        if self.last_epoch >= self.warmup_epochs:
            return [self.final_lr for _ in self.base_lrs]
        alpha = self.last_epoch / float(self.warmup_epochs)
        new_lr = self.base_lr + alpha * (self.final_lr - self.base_lr)
        return [new_lr for _ in self.base_lrs]

# ===== Losses =====
def batch_hard_triplet(embeddings, labels, margin=0.6):
    emb = F.normalize(embeddings, p=2, dim=1)
    dist = 1 - torch.mm(emb, emb.t())
    mask_pos = labels.unsqueeze(1).eq(labels.unsqueeze(0))
    dist_ap = torch.max(dist * mask_pos.float(), dim=1)[0]
    dist_an = torch.min(dist + mask_pos.float()*1e9, dim=1)[0]
    loss = F.relu(dist_ap + margin - dist_an).mean()
    return loss, dist_ap, dist_an

class ArcFaceLoss(nn.Module):
    def __init__(self, in_features, out_features, scale=15.0, margin=0.2):
        super().__init__()
        self.scale = scale
        self.margin = margin
        self.weight = nn.Parameter(torch.FloatTensor(out_features, in_features))
        nn.init.kaiming_uniform_(self.weight, a=5 ** 0.5)

```

```

def forward(self, embeddings, labels):
    W = F.normalize(self.weight, p=2, dim=1).to(embeddings.device)
    X = F.normalize(embeddings, p=2, dim=1)
    cos_theta = F.linear(X, W).clamp(-1,1)
    theta = torch.acos(cos_theta)
    theta_m = theta + self.margin
    one_hot = F.one_hot(labels, num_classes=W.size(0)).float().to(embeddings.device)
    logits = torch.cos(torch.where(one_hot.bool(), theta_m, theta)) * self.scale
    return F.cross_entropy(logits, labels), logits

```

```

class CenterLoss(nn.Module):

```

```

    def __init__(self, num_classes, feat_dim, device='cuda'):
        super().__init__()
        self.centers = nn.Parameter(torch.randn(num_classes, feat_dim).to(device))

    def forward(self, features, labels):
        dist_mat = torch.cdist(features, self.centers)
        target = dist_mat.gather(1, labels.unsqueeze(1))
        return target.mean()

```

```

# ===== Attention & BNNeck =====

```

```

class AttentionLayer(nn.Module):

```

```

    def __init__(self, dim):
        super().__init__()
        self.attn = nn.Sequential(
            nn.Conv2d(dim, dim, 1), nn.ReLU(),
            nn.Conv2d(dim, 1, 1), nn.Sigmoid()
        )

    def forward(self, x):
        return x * self.attn(x)

```

```

class BNNeck(nn.Module):

```

```

    def __init__(self, feat_dim):
        super().__init__()

```

```

self.bn = nn.BatchNorm1d(feat_dim)
self.bn.bias.requires_grad_(False)
def forward(self, x):
    return self.bn(x)

```

```
# ===== Model Definitions =====
```

```

class CustomOSNetWithSpatialFeatures(nn.Module):
    def __init__(self, backbone, emb_dim, num_classes, use_bnneck=False):
        super().__init__()
        backbone.classifier = nn.Identity()
        self.backbone = backbone
        self.attn = AttentionLayer(backbone.feature_dim)
        self.pool_embed = nn.Sequential(
            nn.AdaptiveAvgPool2d((1,1)), nn.Flatten(),
            nn.Linear(backbone.feature_dim, emb_dim), nn.ReLU()
        )
        self.use_bnneck = use_bnneck
        if use_bnneck:
            self.bnneck = BNNeck(emb_dim)
        self.classifier = nn.Linear(emb_dim, num_classes)

    def forward(self, x, return_spatial=False):
        f = self.backbone.conv1(x)
        f = self.backbone.maxpool(f)
        f = self.backbone.conv2(f)
        if hasattr(self.backbone, 'pool2'): f = self.backbone.pool2(f)
        if hasattr(self.backbone, 'conv3'): f = self.backbone.conv3(f)
        if hasattr(self.backbone, 'pool3'): f = self.backbone.pool3(f)
        if hasattr(self.backbone, 'conv4'): f = self.backbone.conv4(f)
        if hasattr(self.backbone, 'conv5'): f = self.backbone.conv5(f)
        sf = self.attn(f)
        embed = self.pool_embed(sf)
        if self.use_bnneck:

```

```

    embed = self.bnneck(embed)
    logits = self.classifier(embed)
    return (logits, embed, sf) if return_spatial else (logits, embed)

```

```
class EnhancedHyperNetworkReID(nn.Module):
```

```

    def __init__(self, in_dim, out_dim, hidden=(512,256), norm='ln'):
        super().__init__()
        layers=[]
        prev=in_dim
        for h in hidden:
            layers+= [nn.Linear(prev,h), nn.LayerNorm(h) if norm=='ln' else nn.BatchNorm1d(h),
nn.ReLU()]
            prev=h
        layers.append(nn.Linear(prev,out_dim))
        self.net=nn.Sequential(*layers)
    def forward(self,x):
        x=F.normalize(x,p=2,dim=1)
        x=F.dropout(x,p=0.1,training=self.training)
        return self.net(x)

```

```
# ===== Dataset & Sampler =====
```

```
class PKSampler(torch.utils.data.Sampler):
```

```

    def __init__(self, dataset, p=8, k=4):
        self.labels=np.array(dataset.labels)
        self.p,self.k=p,k
        self.idx_by_label={ }
        for i,l in enumerate(self.labels): self.idx_by_label.setdefault(l,[]).append(i)
        self.unique=list(self.idx_by_label)
        self.num_samples=p*k
        self.all_indices=list(range(len(dataset)))
    def __iter__(self):
        random.shuffle(self.unique)
        batch=[]

```

```

for lbl in self.unique:
    inds=self.idx_by_label[lbl]
    chosen= random.sample(inds,self.k) if len(inds)>=self.k else random.choices(inds,k=self.k)
    batch+=chosen
    if len(batch)==self.num_samples:
        yield batch; batch=[]
    if batch:
        needed=self.num_samples-len(batch)
        batch+=random.choices(self.all_indices,k=needed)
        yield batch
def __len__(self): return len(self.unique)//self.p

class MarketDataset(Dataset):
    def __init__(self,root,mode='train',transform=None):
        folder={'train':'bounding_box_train','query':'query','gallery':'bounding_box_test'}[mode]
        path=os.path.join(root,folder)
        self.paths=glob.glob(os.path.join(path,'*.jpg'))
        self.labels=[int(os.path.basename(p).split('_')[0]) for p in self.paths]
        if mode=='train':
            uniq=sorted(set(self.labels));mapping={pid:i for i,pid in enumerate(uniq)}
            self.labels=[mapping[l] for l in self.labels]
        self.transform=transform
    def __len__(self): return len(self.paths)
    def __getitem__(self,i):
        img=Image.open(self.paths[i]).convert('RGB')
        if self.transform: img=self.transform(img)
        return img,self.labels[i]

# ===== Utils =====
def compute_dist_matrix(x,y):
    xx=(x**2).sum(1,keepdim=True)
    yy=(y**2).sum(1,keepdim=True).t()
    dist=xx+yy-2*x.mm(y.t())

```

```

return dist.clamp(min=1e-12).sqrt().cpu().numpy()

def compute_map_cmc_minp(dist,ql,gl):
    nq,ng=dist.shape;cmc=np.zeros(ng);APs=[];INPs=[]
    for i in range(nq):
        d=dist[i];idx=np.argsort(d);m=(gl[idx]==ql[i]).astype(int)
        if m.sum()==0:continue
        cmc+=(np.cumsum(m)>0).astype(int)
        prec=(np.cumsum(m)/(np.arange(ng)+1));APs.append((prec*m).sum()/m.sum())
        rp=np.where(m==1)[0].max()+1;INPs.append(m.sum()/rp)
    return (np.mean(APs) if APs else 0,cmc/nq,np.mean(INPs) if INPs else 0)

def evaluate_model(qf,ql,gf,gl):
    qf_t=torch.from_numpy(qf) if isinstance(qf,np.ndarray) else qf
    gf_t=torch.from_numpy(gf) if isinstance(gf,np.ndarray) else gf

    qg=compute_dist_matrix(qf_t,gf_t);qq=compute_dist_matrix(qf_t,qf_t);gg=compute_dist_matrix(gf_t,gf_t)
    rerank=re_ranking(qg,qq,gg,k1=20,k2=6,lambda_value=0.3)
    return compute_map_cmc_minp(rerank,ql,gl)

# ===== Training & Evaluation =====
def train_and_evaluate(config):
    writer=SummaryWriter(log_dir=os.path.join(config.get('tb_log_dir','runs'),config['experiment_name']))
    device='cuda' if torch.cuda.is_available() else 'cpu'

    # Data transforms with RandAugment
    aug_list=[transforms.Resize(config['img_size']),transforms.RandomHorizontalFlip()]
    if config.get('use_randaugment',False):
        from torchvision.transforms.autoaugment import RandAugment

    aug_list.append(RandAugment(num_ops=config['randaug_num_ops'],magnitude=config['randaug_magnitude']))

```

```

aug_list+=[
    transforms.ColorJitter(brightness=config.get('colorjitter_brightness',0.4),
                           contrast=config.get('colorjitter_contrast',0.4),
                           saturation=config.get('colorjitter_saturation',0.4)),

transforms.RandomApply([transforms.Grayscale(num_output_channels=3)],p=config.get('grayscale
_p',0.1)),
    transforms.ToTensor(),transforms.Normalize([0.5]*3,[0.5]*3),
    transforms.RandomErasing(p=config.get('random_erasing_p',0.5))
]
train_t=transforms.Compose(aug_list)

test_t=transforms.Compose([transforms.Resize(config['img_size']),transforms.ToTensor(),transfor
ms.Normalize([0.5]*3,[0.5]*3)])

# Datasets & loaders
train_ds=MarketDataset(config['data_path'],'train',train_t)
qry_ds=MarketDataset(config['data_path'],'query',test_t)
gal_ds=MarketDataset(config['data_path'],'gallery',test_t)

train_loader=DataLoader(train_ds,batch_sampler=PKSampler(train_ds,p=config.get('p',8),k=config
.get('k',4)),num_workers=config.get('num_workers',4))

qry_loader=DataLoader(qry_ds,batch_size=config.get('eval_batch_size',64),shuffle=False,num_wo
rkers=4)

gal_loader=DataLoader(gal_ds,batch_size=config.get('eval_batch_size',64),shuffle=False,num_wor
kers=4)

# Models

backbone=build_model(name=config.get('backbone_name','osnet_ain_x1_0'),num_classes=1000,pr
etrained=True)

model=CustomOSNetWithSpatialFeatures(backbone,config['embedding_dim'],len(set(train_ds.labe
ls)),use_bnneck=config.get('use_bnneck',False)).to(device)

hyper=EnhancedHyperNetworkReID(in_dim=backbone.feature_dim,out_dim=config['embedding_

```

```

dim'],hidden=config.get('hyper_hidden_dims',(512,256)),norm=config.get('hyper_norm_type','ln')).to(device)

    if torch.cuda.device_count()>1:
        model=nn.DataParallel(model);hyper=nn.DataParallel(hyper)

arcfn=ArcFaceLoss(config['embedding_dim'],len(set(train_ds.labels)),scale=config.get('arcface_scale',15.0),margin=config.get('arcface_margin',0.2)).to(device)

cenfn=CenterLoss(len(set(train_ds.labels)),config['embedding_dim'],device)

opt=torch.optim.AdamW(list(model.parameters())+list(hyper.parameters()),lr=config['lr'],weight_decay=config.get('weight_decay',5e-4))

# schedulers
we=config.get('warmup_epochs',0)
warmup_sched=WarmupScheduler(opt,we,config.get('base_lr',1e-5),config['lr']) if we>0 else None
st=config.get('scheduler_type','cosine')
if
st=='onecycle':sched=OneCycleLR(opt,max_lr=config['lr'],epochs=config['epochs'],steps_per_epoch=len(train_loader))
elif st=='cawr':sched=CosineAnnealingWarmRestarts(opt,T_0=config.get('T_0',50),T_mult=1)
elif
st=='step':sched=StepLR(opt,step_size=config.get('step_size',40),gamma=config.get('step_gamma',0.1))
else:sched=CosineAnnealingLR(opt,T_max=max(1,config['epochs']-we))
scaler=torch.cuda.amp.GradScaler() if config.get('use_amp',False) else None

best_map,best_ep=0.0,0
global_step=0
for epoch in range(config['epochs']):
    start=time.time()
    model.train();hyper.train();epoch_loss=0
    for imgs,labels in train_loader:
        imgs,labels=imgs.to(device),labels.to(device)
        opt.zero_grad()
        logits,emb,feat=model(imgs,return_spatial=True)

```

```

# classification accuracy
preds=logits.argmax(dim=1);acc=(preds==labels).float().mean()
# distances
loss_tri,dist_ap,dist_an=batch_hard_triplet(emb,labels,config['triplet_margin'])
loss_arc,_=arcfn(emb,labels)
loss_ctr=cnfn(hyper(F.adaptive_avg_pool2d(feet,(1,1)).view(feet.size(0),-
1)),labels)*config['center_loss_coef']
loss=loss_arc+loss_tri+loss_ctr
if scaler:
    with torch.cuda.amp.autocast():pass
loss.backward()
# gradient norm
total_norm=torch.norm(torch.stack([p.grad.norm() for p in model.parameters() if p.grad is
not None]))
opt.step()
if st=='onecycle' and global_step<sched.total_steps: sched.step()
# log scalars
writer.add_scalar('Loss/total',loss.item(),global_step)
writer.add_scalar('Loss/arcface',loss_arc.item(),global_step)
writer.add_scalar('Loss/triplet',loss_tri.item(),global_step)
writer.add_scalar('Loss/center',loss_ctr.item(),global_step)
writer.add_scalar('Train/cls_acc',acc.item(),global_step)
writer.add_scalar('Dist/mean_pos',dist_ap.mean().item(),global_step)
writer.add_scalar('Dist/mean_neg',dist_an.mean().item(),global_step)
writer.add_scalar('Grad/total_norm',total_norm.item(),global_step)
writer.add_scalar('LR/batch',opt.param_groups[0]['lr'],global_step)
epoch_loss+=loss.item();global_step+=1
# epoch-level scheduling
if st!='onecycle':
    if warmup_sched and epoch<we: warmup_sched.step()
    else: sched.step()
epoch_time=time.time()-start
writer.add_scalar('Time/epoch_sec',epoch_time,epoch)
writer.add_scalar('Time/imgs_per_sec',len(train_ds)/epoch_time,epoch)

```

```

avg_loss=epoch_loss/len(train_loader)
writer.add_scalar('EpochLoss/train',avg_loss,epoch)
writer.add_scalar('LR/epoch',opt.param_groups[0]['lr'],epoch)
embed_norm=emb.norm(p=2,dim=1).detach().cpu()
writer.add_histogram('Embedding/norm',embed_norm,epoch)
for name,param in model.named_parameters():
    writer.add_histogram(f'Weights/{name}',param,epoch)
    if param.grad is not None: writer.add_histogram(f'Grads/{name}',param.grad,epoch)
# evaluation
if (epoch+1)%config.get('eval_freq',10)==0:
    model.eval();hyper.eval()
    qf,ql,gf,gl=[],[],[],[]
    with torch.no_grad():
        for imgs,labels in qry_loader:
            imgs=imgs.to(device)
            _,_,f=model(imgs,return_spatial=True)
            p=F.adaptive_avg_pool2d(f,(1,1)).view(f.size(0),-1)
            qf.append(hyper(p).cpu());ql.append(labels.numpy())
        for imgs,labels in gal_loader:
            imgs=imgs.to(device)
            _,_,f=model(imgs,return_spatial=True)
            p=F.adaptive_avg_pool2d(f,(1,1)).view(f.size(0),-1)
            gf.append(hyper(p).cpu());gl.append(labels.numpy())
    qf=np.concatenate([x.numpy() for x in qf]);ql=np.concatenate(ql)
    gf=np.concatenate([x.numpy() for x in gf]);gl=np.concatenate(gl)
    mAP,cmc,minp=evaluate_model(qf,ql,gf,gl)
    writer.add_scalar('Eval/mAP',mAP,epoch)
    writer.add_scalar('Eval/Rank1',cmc[0],epoch)
    writer.add_scalar('Eval/Rank5',cmc[4],epoch)
    writer.add_scalar('Eval/Rank10',cmc[9],epoch)
    writer.add_scalar('Eval/mINP',minp,epoch)
# embedding projector

```

```

        writer.add_embedding(torch.from_numpy(qf[:500]),metadata=[str(l) for l in
ql[:500]],global_step=epoch,tag='QueryProj')
        # log retrieval images
        try:
            query_batch,_=next(iter(qry_loader)); gallery_batch,_=next(iter(gal_loader))

topk_idx=np.argsort(compute_dist_matrix(torch.from_numpy(qf),torch.from_numpy(gf)),axis=1)

log_retrieval_examples(writer,query_batch,gallery_batch,topk_idx,epoch,num_examples=3)
        except Exception as e:
            logger.warning(f"Could not log retrieval images: {e}")
            if mAP>best_map:
                best_map, best_ep=mAP, epoch+1

torch.save(model.state_dict(),'best_osnet.pth');torch.save(hyper.state_dict(),'best_hyper.pth')
        # end eval
        # end epoch
        logger.info(f"Training complete: best_mAP={best_map:.4f} at epoch={best_ep}")
        writer.close();torch.cuda.empty_cache()
        return {'best_epoch':best_ep,'best_map':best_map}

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

#-----
# Fallback evaluation metrics
#-----

```

```

def eval_cmc_map(q_feats, g_feats, q_labels, g_labels):
    all_cmc, all_AP = [], []
    for i, qf in enumerate(q_feats):
        d = np.linalg.norm(g_feats - qf, axis=1)
        order = np.argsort(d)
        matches = (np.array(g_labels) == q_labels[i])[order]
        cmc = matches.cumsum(); cmc[cmc>1] = 1
        all_cmc.append(cmc)
        num_rel = matches.sum()
        if num_rel == 0:
            all_AP.append(0.0)
        else:
            precision = matches.cumsum() / (np.arange(len(matches)) + 1)
            all_AP.append((precision * matches).sum() / num_rel)
    cmc_scores = np.mean(np.stack(all_cmc), axis=0)
    mAP = float(np.mean(all_AP))
    return mAP, cmc_scores

#-----
# Paths and hyperparameters
#-----
MARKET1501_PATH = "/home/ubuntu/test_jupyter/re-id/osnet-hypernet/datasets/market-1501/Market-1501-v15.09.15/"
device = 'cuda' if torch.cuda.is_available() else 'cpu'
device_ids = [0,1]
batch_size = 32
lr = 2e-4
beta1, beta2 = 0.5, 0.999
num_epochs = 50
noise_dim = 100
# Loss weights
default_lambda_adv = 1.0
default_lambda_cls = 1.0

```



```

    pids = sorted({pid for _, pid in self.samples})
    self.pid2label = {pid:i for i,pid in enumerate(pids)}
def __len__(self):
    return len(self.samples)
def __getitem__(self, idx):
    fname, pid = self.samples[idx]
    img = Image.open(os.path.join(self.root_dir, fname)).convert('RGB')
    if self.transform:
        img = self.transform(img)
    return img, self.pid2label[pid]

train_loader = DataLoader(
    MarketDataset(os.path.join(MARKET1501_PATH, 'bounding_box_train'), transform),
    batch_size, shuffle=True, num_workers=4
)
query_loader = DataLoader(
    MarketDataset(os.path.join(MARKET1501_PATH, 'query'), transform),
    batch_size, shuffle=False, num_workers=4
)
gallery_loader = DataLoader(
    MarketDataset(os.path.join(MARKET1501_PATH, 'bounding_box_test'), transform),
    batch_size, shuffle=False, num_workers=4
)

#-----
# Models
#-----

class AdaIN(nn.Module):
    def __init__(self, channels):
        super().__init__()
        self.norm = nn.InstanceNorm2d(channels, affine=False)
        self.fc = nn.Linear(channels, channels * 2)
    def forward(self, x, style):

```

```

gamma_beta = self.fc(style)
gamma, beta = gamma_beta.chunk(2, 1)
gamma = gamma.view(-1, x.size(1), 1, 1)
beta = beta.view(-1, x.size(1), 1, 1)
out = self.norm(x)
return gamma * out + beta

```

```

class ResAdaINBlock(nn.Module):
    def __init__(self, channels):
        super().__init__()
        self.conv1 = nn.Conv2d(channels, channels, 3, padding=1)
        self.adain1 = AdaIN(channels)
        self.relu = nn.ReLU(True)
        self.conv2 = nn.Conv2d(channels, channels, 3, padding=1)
        self.adain2 = AdaIN(channels)
    def forward(self, x, style):
        out = self.conv1(x)
        out = self.adain1(out, style)
        out = self.relu(out)
        out = self.conv2(out)
        out = self.adain2(out, style)
        return x + out

```

```

class Generator(nn.Module):
    def __init__(self, style_dim=512):
        super().__init__()
        c0 = 64
        layers = [
            nn.ReflectionPad2d(3),
            nn.Conv2d(3, c0, 7),
            nn.InstanceNorm2d(c0),
            nn.ReLU(True)
        ]

```

```

c = c0
for _ in range(2):
    layers += [
        nn.Conv2d(c, c*2, 4, 2, 1),
        nn.InstanceNorm2d(c*2),
        nn.ReLU(True)
    ]
    c *= 2
self.init = nn.Sequential(*layers)
self.style_enc = nn.Sequential(
    nn.AdaptiveAvgPool2d((1,1)),
    nn.Flatten()
)
self.res_blocks = nn.ModuleList([ResAdaINBlock(c) for _ in range(6)])
up = []
for _ in range(2):
    up += [
        nn.Upsample(scale_factor=2, mode='nearest'),
        nn.Conv2d(c, c//2, 5, 1, 2),
        nn.InstanceNorm2d(c//2),
        nn.ReLU(True)
    ]
    c //= 2
self.up = nn.Sequential(*up)
self.final = nn.Sequential(
    nn.ReflectionPad2d(3),
    nn.Conv2d(c0, 3, 7),
    nn.Tanh()
)
def forward(self, x, z=None):
    feat = self.init(x)
    style = self.style_enc(feat)
    out = feat

```

```

for blk in self.res_blocks:
    out = blk(out, style)
out = self.up(out)
return self.final(out)

```

```

class Discriminator(nn.Module):
    def __init__(self):
        super().__init__()
        self.model = nn.Sequential(
            nn.Conv2d(3, 64, 4, 2, 1), nn.LeakyReLU(0.2, True),
            nn.Conv2d(64, 128, 4, 2, 1), nn.InstanceNorm2d(128), nn.LeakyReLU(0.2, True),
            nn.Conv2d(128, 256, 4, 2, 1), nn.InstanceNorm2d(256), nn.LeakyReLU(0.2, True),
            nn.Conv2d(256, 512, 4, 1, 1), nn.InstanceNorm2d(512), nn.LeakyReLU(0.2, True),
            nn.Conv2d(512, 1, 4, 1, 1)
        )
    def forward(self, x):
        return self.model(x).view(x.size(0), -1)

```

```

from torchreid.models import build_model
num_classes = len(train_loader.dataset.pid2label)

gen = nn.DataParallel(Generator(), device_ids).to(device)
disc = nn.DataParallel(Discriminator(), device_ids).to(device)
cls = nn.DataParallel(build_model(name='osnet_x1_0', num_classes=num_classes,
pretrained=True), device_ids).to(device)

#-----
# Losses and optimizers
#-----
criterion_adv = nn.BCEWithLogitsLoss()
criterion_cls = nn.CrossEntropyLoss()
criterion_id = nn.L1Loss()

```

```

opt_g = optim.Adam(gen.parameters(), lr, betas=(beta1, beta2))
opt_d = optim.Adam(disc.parameters(), lr, betas=(beta1, beta2))
opt_c = optim.Adam(cls.parameters(), lr, betas=(beta1, beta2))

#-----
# Training loop with R1 regularization
#-----
global_step = 0
for epoch in range(1, num_epochs+1):
    gen.train(); disc.train(); cls.train()
    for imgs, labels in train_loader:
        imgs, labels = imgs.to(device), labels.to(device)
        bs = imgs.size(0)
        fake = gen(imgs, torch.randn((bs, noise_dim), device=device))
        # Discriminator update
        if epoch > freeze_disc_epochs:
            imgs.requires_grad_()
            d_real = disc(imgs)
            d_fake = disc(fake.detach())
            loss_d = 0.5 * (
                criterion_adv(d_real, torch.ones_like(d_real)) +
                criterion_adv(d_fake, torch.zeros_like(d_fake))
            )
            # R1 regularization
            grad_real = torch.autograd.grad(outputs=d_real.sum(), inputs=imgs, create_graph=False,
retain_graph=False, only_inputs=True)[0]
            r1_loss = grad_real.view(bs, -1).pow(2).sum(1).mean()
            loss_d += r1_gamma * 0.5 * r1_loss
            opt_d.zero_grad(); loss_d.backward(); opt_d.step()
            imgs.requires_grad_(False)
        # Generator update
        d_fake_g = disc(fake)
        loss_g_adv = criterion_adv(d_fake_g, torch.ones_like(d_fake_g))

```

```

logits_fake = cls(fake)
loss_g_cls = criterion_cls(logits_fake, labels)
loss_g_id = criterion_id(fake, imgs)
loss_g = default_lambda_adv * loss_g_adv + default_lambda_cls * loss_g_cls +
default_lambda_id * loss_g_id
opt_g.zero_grad(); loss_g.backward(); opt_g.step()
# Classifier update on real
logits_real = cls(imgs)
loss_c = criterion_cls(logits_real, labels)
opt_c.zero_grad(); loss_c.backward(); opt_c.step()
# Logging
if global_step % 100 == 0:
    writer.add_images('Input',    unnormalize(imgs[:4]), global_step)
    writer.add_images('Generated', unnormalize(fake[:4]), global_step)
if epoch > freeze_disc_epochs:
    writer.add_scalar('Loss/Discriminator', loss_d.item(), global_step)
writer.add_scalar('Loss/G_adv', loss_g_adv.item(), global_step)
writer.add_scalar('Loss/G_cls', loss_g_cls.item(), global_step)
writer.add_scalar('Loss/G_id', loss_g_id.item(), global_step)
writer.add_scalar('Loss/C',    loss_c.item(),    global_step)
global_step += 1
# Evaluation
gen.eval(); cls.eval()
with torch.no_grad():
    gal_feats, gal_lbls = [], []
    for ig, lg in gallery_loader:
        ig = ig.to(device)
        fg = cls(gen(ig, torch.randn((ig.size(0), noise_dim), device=device)))
        gal_feats.append(fg.cpu().numpy()); gal_lbls.extend(lg.tolist())
    gal_feats = np.vstack(gal_feats)
    qry_feats, qry_lbls = [], []
    for iq, lq in query_loader:
        iq = iq.to(device)

```

```
fq = cls(iq)
qry_feats.append(fq.cpu().numpy()); qry_lbls.extend(lq.tolist())
qry_feats = np.vstack(qry_feats)
mAP, cmc_scores = eval_cmc_map(qry_feats, gal_feats, qry_lbls, gal_lbls)
writer.add_scalar('Metric/mAP', mAP*100, epoch)
writer.add_scalar('Metric/R1', cmc_scores[0]*100, epoch)
print(f'Epoch {epoch}/{num_epochs} mAP: {mAP*100:.2f}% Rank-1:
{cmc_scores[0]*100:.2f}%")
writer.close()
```

## ДОДАТОК Д

### ПРИКЛАД РОБОТИ ОТРИМАНИХ МОДЕЛЕЙ ПОВТОРНОЇ ІДЕНТИФІКАЦІЇ

