МАГІСТЕРСЬКА РОБОТА

МР. ІПм - 02.00.00.000 ПЗ

Група ІПм-24-1

Бриндзей Володимир

2025

Івано-Франківський національний технічний університет нафти і газу

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Бриндзей Володимир Тарасович

(прізвище, ім'я, по батькові)

УДК 004.9

(індекс)

МАГІСТЕРСЬКА РОБОТА

Методи класифікації позиціювань на основі моделей

розпізнавання облич

(назва роботи)

Інженерія програмного забезпечення

(назва освітньої програми)

121 - Інженерія програмного забезпечення

(шифр і назва спеціальності)

Бриндзей В.Т.

(підпис, ініціали та прізвище здобувача освітнього ступеня)

Науковий керівник Бандура Вікторія Валеріївна, к.т.н., доцент

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

Допущено до захисту

Завідувач кафедри

доц. Бандура В.В.

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

Нормоконтроль

доц. Вовк Р.Б.

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

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

Івано-Франківськ – 2025

Івано-Франківський національний технічний університет нафти і газу

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Освітній рівень магістр

Спеціальність 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:
Зав. кафедрою **ІПЗ**

доц. В.В. Бандура

“ 04 ” вересня 2025 р.

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Бриндзею Володимиру Тарасовичу

(прізвище, ім’я, по-батькові)

1. Тема магістерської роботи “ Методи класифікації позиціювань на основі моделей розпізнавання облич”

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

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

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

3. Вихідні дані до проекту (роботи) Теоретичні концепції та формальні моделі побудови та функціонування інформаційних та програмних технологій розпізнавання облич

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

1. Аналіз предметної області визначення позиціювання об’єктів на основі розпізнавання облич
2. Дослідження парадигми багатозадачного навчання (Multitask Learning)
3. Представлення методології класифікації позиціювань на основі моделей розпізнавання облич
4. Імплементація методів класифікації позиціювань на основі моделей розпізнавання облич

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

1. Огляд структури MotionBERT (рис. 1.1)
2. Порівняння використання окремих моделей та моделей MTL (рис. 1.2)
3. Приклад набору даних WIDER FACE (рис. 1.3)
4. Приклад архітектур мереж для ImageNet (рис. 1.4)
5. Схема архітектури AdaFace (рис. 1.5)

6. Консультанти розділів проекту (роботи)

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

7. Дата видачі завдання 04 вересня 2025 р.

Керівник _____
(підпис)

Завдання прийняв до виконання_____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назви етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1	Підбір і вивчення літератури по темі магістерської роботи	15.09.2025	виконано
2	Аналіз концепцій та алгоритмів предметної області	29.09.2025	виконано
3	Аналіз предметної області визначення позиціювання об’єктів на основі розпізнавання облич	15.10.2025	виконано
4	Дослідження парадигми багатозадачного навчання (Multitask Learning)	08.11.2025	виконано
5	Представлення методології класифікації позиціювань на основі моделей розпізнавання облич	20.11.2025	виконано
6	Імплементація методів класифікації позиціювань на основі моделей розпізнавання облич	01.12.2025	виконано
7	Затвердження пояснювальної записки роботи завідувачем кафедри	15.12.2025	виконано

Студент – магістр _____
(підпис)

Керівник роботи _____
(підпис)

АНОТАЦІЯ

Магістерська робота: 76 с., 26 рис., 48 джерел.

Тема: Методи класифікації позиціювань на основі моделей розпізнавання облич

Об'єкт дослідження: процес класифікації позиціювань у системах комп'ютерного зору з інтеграцією розпізнавання облич.

Мета роботи: розробка та перевірка методів класифікації позиціювань на основі моделей розпізнавання облич із використанням багатозадачного навчання/

Предмет дослідження: методи, моделі та алгоритми багатозадачного навчання, що забезпечують інтеграцію розпізнавання облич та оцінки пози у рамках класифікації позиціювань.

Результати дослідження

В роботі здійснено системний порівняльний аналіз ефективності кругового навчання та підходу із замороженою мережею для інтегрованої класифікації позиціювань.

Висновок

Розроблено методологію класифікації позиціювань, що поєднує розпізнавання облич та оцінку пози на основі інтегрованого конвеєра. Запропоновано адаптацію кругового методу навчання до багатозадачних моделей, що дозволило підвищити точність класифікації.

КОМП'ЮТЕРНИЙ ЗІР; РОЗПІЗНАВАННЯ ОБЛИЧ; КЛАСИФІКАЦІЯ ПОЗИ; БАГАТОЗАДАЧНЕ НАВЧАННЯ; YOLO; VITPOSE; ADAFACE; КОНВЕЄР ОЦІНКИ ПОЗИЦІЮВАНЬ; ІНТЕГРОВАНІ СИСТЕМИ.

ABSTRACT

Master Thesis: 76 pp., 26 fig., 48 sources.

Topic: Position classification methods based on face recognition models

Object of the study: the process of position classification in computer vision systems with the integration of face recognition.

Purpose of the work: development and verification of position classification methods based on face recognition models using multi-task learning/

Subject of the study: methods, models and algorithms of multi-task learning that ensure the integration of face recognition and pose estimation within the framework of position classification.

Research results

The work carried out a systematic comparative analysis of the effectiveness of circular learning and the frozen network approach for integrated position classification.

Conclusion

A position classification methodology was developed that combines face recognition and pose estimation based on an integrated pipeline. Adaptation of the circular learning method to multi-task models was proposed, which allowed to increase the classification accuracy.

COMPUTER VISION; FACE RECOGNITION; POSE CLASSIFICATION; MULTI-TASK LEARNING; YOLO; VITPOSE; ADAFACE; POSITIONING ESTIMATION PIPELINE; INTEGRATED SYSTEMS.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

COCO - Common Objects in Context

FPS - Frames Per Second

IOU - Intersection Over Union

MTL - Multi-Task Learning

ROI - Region Of Interest

SORT - Simple Online and Realtime Tracking

ViT - Vision Transformer

WIDER - Web Image Dataset for Event Recognition

YOLO - You Only Look Once

ВСТУП

Актуальність теми.

Сучасний розвиток комп'ютерного зору та біометричних технологій зумовлює зростання інтересу до методів, здатних одночасно вирішувати декілька взаємопов'язаних завдань: виявлення облич, їх ідентифікацію та визначення положення об'єктів у просторі. Класифікація позиціювань на основі моделей розпізнавання облич відкриває нові можливості для створення більш точних і гнучких систем безпеки, інтелектуального відеоспостереження, медичних діагностичних платформ, а також застосунків у сфері розширеної та віртуальної реальності.

Разом із цим існуючі методи мають низку обмежень: високу залежність від умов освітлення та якості даних, значні обчислювальні витрати, недостатню інтегрованість між окремими завданнями. Це стимулює дослідження нових архітектур та підходів, зокрема використання багатозадачного навчання (Multitask Learning), яке дозволяє інтегрувати розпізнавання облич та оцінку пози у єдиний процес.

У даній роботі запропоновано та реалізовано методологію, що базується на сучасних моделях комп'ютерного зору (YOLO, ViTPose, AdaFace) та двох підходах до навчання — за круговим методом і з використанням замороженої основної мережі. Проведене дослідження спрямоване на підвищення точності та ефективності класифікації позиціювань у динамічних середовищах.

Зростаюча кількість систем, що потребують надійної біометричної ідентифікації та відстеження руху, робить проблему класифікації позиціювань особливо актуальною. Традиційні методи окремо розв'язують завдання виявлення, розпізнавання та оцінки пози, що призводить до фрагментованості процесу й втрати ефективності. Застосування багатозадачного навчання дозволяє інтегрувати ці завдання у єдину модель,

мінімізуючи дублювання обчислень та підвищуючи узгодженість результатів.

Актуальність роботи також зумовлена практичними потребами в оптимізації систем відеоспостереження, моніторингу стану здоров'я пацієнтів, автоматизованого аналізу поведінки, а також у розвитку технологій AR/VR, де класифікація пози є основою взаємодії користувача із системою. Важливим аспектом є і те, що інтегровані методи здатні забезпечити адаптивність до змінних середовищ та масштабованість для обробки великих потоків даних у реальному часі.

Метою магістерської роботи є розробка та перевірка методів класифікації позиціювань на основі моделей розпізнавання облич із використанням багатозадачного навчання.

Завдання дослідження:

- Виконати аналіз предметної області класифікації позиціювань та розпізнавання облич.

- Дослідити сучасні підходи та моделі, що використовуються для виявлення, ідентифікації та оцінки пози.

- Розробити методологію класифікації позиціювань із використанням моделей YOLO, ViTPose та AdaFace.

- Обґрунтувати вибір навчальних підходів та визначити їх переваги й обмеження.

- Реалізувати конвеєр інтегрованої системи та провести її експериментальне дослідження.

- Виконати аналіз результатів і оцінити ефективність запропонованих методів.

Об'єкт дослідження - процес класифікації позиціювань у системах комп'ютерного зору з інтеграцією розпізнавання облич.

Предмет дослідження - методи, моделі та алгоритми багатозадачного навчання, що забезпечують інтеграцію розпізнавання облич та оцінки пози у рамках класифікації позиціювань.

Методи дослідження:

- методи комп’ютерного зору для виявлення, розпізнавання та оцінки пози;

- методи машинного навчання та глибинного навчання (YOLO, ViTPose, AdaFace);

- багатозадачне навчання (Multitask Learning);

- експериментальні методи для тестування систем на вибраних наборах даних;

- методи порівняльного аналізу продуктивності та латентності.

Наукова новизна отриманих результатів

Розроблено методологію класифікації позиціювань, що поєднує розпізнавання облич та оцінку пози на основі інтегрованого конвеєра. Запропоновано адаптацію кругового методу навчання до багатозадачних моделей, що дозволило підвищити точність класифікації.

Доведено, що інтеграція моделей YOLO, ViTPose та AdaFace забезпечує баланс між продуктивністю та точністю в умовах динамічного середовища.

Практичне значення магістерської роботи

Виконана робота зробила внесок у розвиток методів класифікації позиціювань на основі моделей розпізнавання облич, запропонувала інноваційні підходи до інтеграції багатозадачного навчання та створила основу для подальших досліджень і практичних впроваджень у сфері комп’ютерного зору та біометрії.

Структура магістерської роботи. Робота складається зі вступу, трьох розділів та висновків. Загальний обсяг роботи становить 76 сторінок, і містить 26 рисунків, список використаних джерел із 48 найменувань.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ВИЗНАЧЕННЯ ПОЗИЦІЮВАННЯ ОБ'ЄКТІВ НА ОСНОВІ РОЗПІЗНАВАННЯ ОБЛИЧ

1.1. Інтеграція ідентифікації для оптимізації аналізу руху

1.1.1. Вступ до оцінки позиціювання об'єкту

Оцінка позиціювання (пози) - це технологія, що дає змогу локалізувати та ідентифікувати ключові точки тіла (суглоби, кінцівки) у цифровому просторі, перетворюючи їх на набір даних, який можна фіксувати, реєструвати та аналізувати. Ця методологія знайшла своє застосування в різноманітних галузях, включаючи цифрові фітнес-платформи, системи виявлення падінь та віртуальну реальність.

Традиційні підходи до оцінки пози зазвичай орієнтовані на одночасне відстеження всіх об'єктів у кадрі, що створює значні обмеження для практичного використання. Зокрема, у випадках, коли необхідно аналізувати лише одну або декілька конкретних осіб, присутність інших об'єктів знижує ефективність та ускладнює обробку. Хоча в деяких дослідженнях пропонується інтеграція ідентифікації, її основна мета - покращення узгодженості відстеження, а не забезпечення цільового аналізу.

1.1.2. Проблема цільової оцінки пози

Ця робота присвячена проблемі цільової оцінки пози, де фокус зміщується з відстеження всіх об'єктів на ідентифікацію та аналіз лише певних, заздалегідь визначених суб'єктів. Це дозволяє оптимізувати обчислювальні ресурси та отримати релевантніші дані.

1.1.3. Запропонований підхід

Ми пропонуємо новий багатозадачний модельний підхід та програмний конвеєр, який інтегрує кілька ключових технік:

- Виявлення осіб: визначення локації кожної особи в кадрі.

- Оцінка пози: визначення ключових точок тіла.

- Розпізнавання облич: ідентифікація конкретних осіб.

- Відстеження об'єктів: підтримка ідентифікації протягом послідовності кадрів.

Застосування цього підходу дозволяє реалізувати доказове рішення, що забезпечує цільову оцінку пози. Така методологія має значний потенціал для розширення застосувань, особливо в таких сферах, як:

- Спортивна аналітика: аналіз рухів конкретного спортсмена в умовах присутності інших гравців або натовпу.

- Системи безпеки: моніторинг певних осіб у місцях масового скупчення, наприклад, у в'язницях або аеропортах.

- Оптимізація процесу відстеження та оцінки пози за рахунок усунення непотрібних обчислень для нерелевантних суб'єктів забезпечує більш корисні та ефективні результати.

1.2. Аналіз та оптимізація систем оцінки позиціювань

У сучасному світі комп'ютерного зору (КЗ) розпізнавання людських атрибутів, зокрема обличчя та пози, є ключовим напрямком досліджень. Інтеграція цих двох аспектів дозволяє системам штучного інтелекту (ШІ) не лише ідентифікувати особу, а й розуміти її дії, емоційний стан та контекст. Ця стаття аналізує стан справ у галузях розпізнавання обличчя та оцінки пози, висвітлюючи їхні досягнення та існуючі обмеження.

1.2.1. Аналіз сучасних підходів до розпізнавання

Розпізнавання обличчя досягло значного прогресу, знайшовши широке застосування в комерційних та цивільних додатках, від систем безпеки персональних пристроїв до автоматичної організації фотоархівів. Ця технологія дозволяє ідентифікувати осіб з високою точністю, що є основою для багатьох сучасних систем.

Оцінка пози є не менш важливою галуззю КЗ, що дозволяє аналізувати людський рух. Роботи, подібні до MotionBERT [5], демонструють можливість класифікації рухів, що відкриває шлях до застосувань у сфері виявлення падінь [2, 6], фітнес-тренувань [3] та віртуальної реальності [4]. Такі системи можуть постійно моніторити середовище та реагувати на критичні ситуації, що має важливе значення, наприклад, для догляду за літніми людьми.

В роботі [5] пропонується новий погляд на вивчення представлень людського руху. Ключова ідея полягає в тому, що ми можемо вивчити універсальне представлення людського руху з неоднорідних ресурсів даних єдиним чином і використовувати це представлення для вирішення різних подальших завдань єдиним способом. Тут представлено двохетапну структуру, що складається з попереднього навчання та тонкого налаштування, як показано на рисунку 1.1.

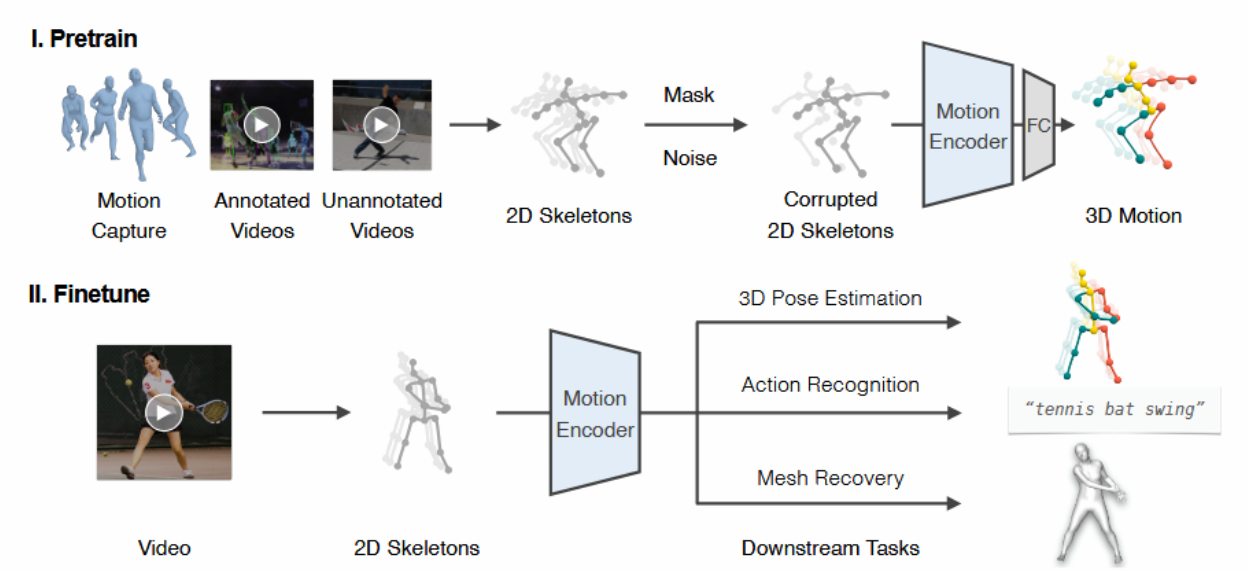

Рис. 1.1. Огляд структури MotionBERT

На рис. 1.1 використовується кодер руху для вивчення представлень руху людини шляхом відновлення 3D-руху людини зі спотворених послідовностей 2D-скелетів. Щоб адаптуватися до різних подальших завдань, тут тонко налаштовують попередньо навчені представлення руху за допомогою лінійного шару або простого MLP.

На етапі попереднього навчання витягуються 2D-послідовності скелета з різноманітних джерел даних руху та спотворюємо їх випадковими масками та шумами. Згодом відбувається навчання кодер руху відновлювати 3D-рух зі спотворених 2D-скелетів. Це складне попереднє завдання внутрішньо вимагає від кодера руху:

1) виводити основні 3D-структури людини з її часових рухів;

2) відновлювати помилкові та відсутні спостереження.

Таким чином, кодер руху неявно фіксує здоровий глузд людського руху, такий як зв'язки суглобів, анатомічні обмеження та часова динаміка. На практиці пропонується двопотоковий просторово-часовий трансформер (DSTformer) як кодер руху для захоплення далекосяжних зв'язків між ключовими точками скелета. Припускається, що представлення руху, отримані з великомасштабних та диверсифікованих ресурсів даних, можуть бути спільними для різних подальших завдань та покращувати їхню продуктивність. Тому для кожного подальшого завдання ми адаптуємо попередньо навчені представлення руху, використовуючи дані навчання, специфічні для завдання, та сигнали нагляду за допомогою простої регресійної головки.

1.2.2. Обмеження існуючих рішень

Незважаючи на значний прогрес, сучасні системи оцінки позиціювань стикаються з двома основними проблемами:

- Відсутність цільової ідентифікації: моделі не мають вбудованого механізму для визначення, які саме суб'єкти повинні бути відстежені. Це призводить до необхідності ручного маркування даних на етапі постобробки.

- Неефективність обчислень: існуючі рішення припускають, що всі особи в кадрі повинні бути відстежені, що призводить до обробки зайвих даних. У моделях виявлення, що працюють за принципом «зверху вниз», обчислювальна складність зростає лінійно зі збільшенням кількості людей у кадрі, що призводить до значної витрати ресурсів на нерелевантну

інформацію. Це особливо актуально у сценах з великим скупченням людей, наприклад, у спортивній аналітиці, де необхідно відстежувати лише певних гравців.

Для подолання цих обмежень в даній роботі пропонується архітектура конвеєра для цільової оцінки пози, що забезпечує селективне відстеження, багатозадачні моделі та концептуальне доказове рішення, яке демонструє практичну життєздатність запропонованого підходу.

1.3. Дослідження парадигми багатозадачного навчання (Multitask Learning)

Цей розділ присвячений огляду ключових концепцій та методів, що лежать в основі запропонованої системи. Оскільки розроблювана система інтегрує декілька дисциплін у галузі комп'ютерного зору, огляд літератури зосереджений на основних компонентах, що складають її архітектуру.

Багатозадачне навчання (Multitask Learning, MTL) є парадигмою в машинному навчанні, в якій єдина модель одночасно навчається виконувати кілька пов'язаних завдань. Основна ідея MTL полягає у спільному використанні представлень між подібними завданнями, що може підвищити ефективність навчання та точність прогнозування [7]. У контексті комп'ютерного зору MTL виступає як потужний підхід для вирішення складних проблем, що вимагають одночасного візуального аналізу для різних цілей в межах однієї архітектури.

В основі MTL лежить ідея, що, якщо завдання пов'язані між собою (наприклад, виявлення об'єктів та оцінка пози), спільне навчання допомагає моделі краще узагальнювати знання. Це досягається завдяки спільному використанню параметрів, зазвичай у вигляді спільної основної мережі (backbone), яка витягує загальні ознаки для всіх завдань.

Типова архітектура MTL складається з:

- Спільної основної мережі (Shared Backbone) - це перша частина моделі, яка обробляє вхідні дані (наприклад, зображення) і генерує спільні представлення ознак.

- Гілкові мережі (Task-Specific Heads) - після основної мережі йдуть окремі гілки, кожна з яких відповідає за виконання свого завдання. Вони беруть спільні ознаки і перетворюють їх у кінцеві вихідні дані.

Розглянемо основні переваги багатозадачного навчання:

1. Покращення узагальнення.

Навчання на кількох завданнях одночасно допомагає моделі уникнути перенавчання на одному конкретному завданні, оскільки вона змушена вивчати більш узагальнені представлення.

2. Збільшення ефективності даних.

Якщо деякі завдання мають невеликі набори даних, MTL дозволяє їм "запозичувати" інформацію з інших, більших наборів, що покращує їхню продуктивність.

3. Підвищення ефективності.

Використання однієї спільної основної мережі зменшує кількість параметрів, які потрібно навчати, та знижує обчислювальні витрати, що робить процес інференції швидшим.

На відміну від ансамблевих методів, де декілька моделей об'єднують свої результати для отримання єдиного виходу (наприклад, випадковий ліс), в MTL одна модель отримує один вхід, після чого дані проходять через спільну основу. На вихідному етапі модель розгалужується на окремі "голови" або "гілки", кожна з яких призначена для виконання свого унікального та дискретного завдання.

Ключова особливість архітектури MTL — це наявність спільної базової мережі (shared backbone), яка навчається вилучати узагальнені візуальні ознаки з вхідних даних. Кожна окрема гілка, в свою чергу, спеціалізується на конкретному завданні. Ця архітектура сприяє оптимізації обчислень, оскільки спільна основа усуває надлишкові розрахунки, що виникають при

послідовному виконанні декількох завдань. Також це полегшує застосування transfer learning (перенесення навчання), коли попередньо навчена модель використовується для нових, пов'язаних завдань. Порівняння архітектур MTL та використання окремих моделей наведено на рис. 1.2.

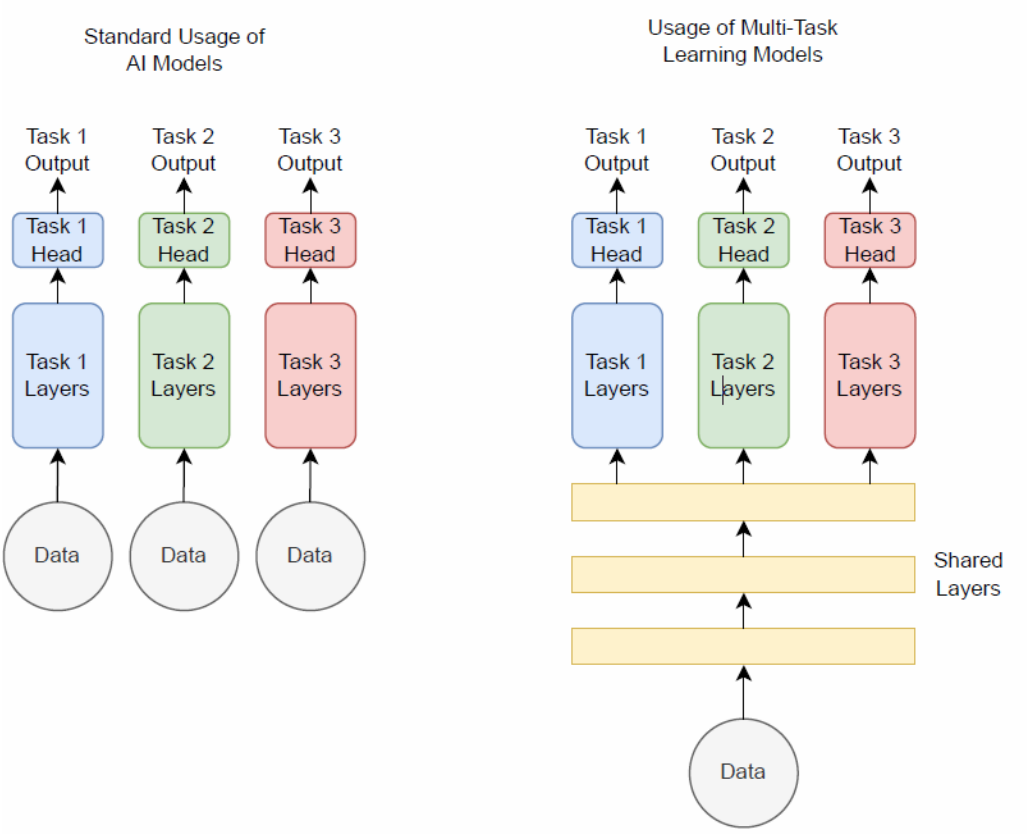

Рис. 1.2. Порівняння використання окремих моделей та моделей MTL

На рисунку 1.2 показано різницю між стандартним використанням моделей ШІ та використанням моделей багатозадачного навчання (MTL).

З лівого боку зображення представлено стандартний підхід. Кожне завдання (Task 1, Task 2, Task 3) обробляється окремою, незалежною моделлю.

- Вхідні дані (Data): Кожне завдання вимагає власного набору даних.

- Спеціалізовані шари (Task Layers): Кожна модель складається з унікальних шарів, спеціально навчених для конкретного завдання. Це означає, що для кожного завдання потрібно створювати та навчати нову нейронну мережу.

- Вихід (Output): Кожна модель генерує свій окремий вихід.

Цей підхід є менш ефективним, оскільки він вимагає більше обчислювальних ресурсів та часу для навчання, а також не використовує синергію між завданнями, що можуть мати спільні характеристики.

З правого боку зображення показано підхід MTL. Замість окремих моделей, використовується одна єдина архітектура для вирішення кількох завдань одночасно.

- Спільні шари (Shared Layers): Всі завдання використовують спільну базову мережу. Ці шари навчаються вилучати узагальнені ознаки з вхідних даних, які є корисними для всіх завдань. Це дозволяє моделі краще узагальнювати та зменшує надлишкові обчислення.

- Task Heads: Після спільних шарів, модель розгалужується на окремі "голови" або "гілки". Кожна "голова" спеціалізується на конкретному завданні (Task 1, Task 2, Task 3) та генерує відповідний вихід.

- Єдиний вхід (Data): Дані для всіх завдань подаються на вхід однієї моделі.

Порівняно зі стандартним підходом, MTL є більш ефективним, оскільки спільні шари дозволяють зменшити обчислювальні витрати, підвищити продуктивність та покращити точність прогнозування завдяки спільному використанню знань між завданнями.

1.4. Опис технології гілкових завдань

У межах архітектури багатозадачного навчання (MTL) кожна окрема гілка моделі спеціалізується на конкретному завданні, використовуючи при цьому узагальнені ознаки, виділені спільною основною мережею. Для створення системи селективної оцінки пози необхідна інтеграція чотирьох ключових завдань комп'ютерного зору:

- Локалізація людей у кадрі.

- Виявлення та розпізнавання облич.

- Оцінка людських поз.

Цей розділ містить огляд еволюції підходів для кожного з цих завдань, аналізуючи їх сильні сторони, обмеження та придатність для інтеграції у нашу систему.

Вибір цих конкретних гілок зумовлений логікою процесу: спочатку система має виявити всіх присутніх осіб, потім ідентифікувати тих, що є об'єктами інтересу, і, нарешті, оцінити позу лише для ідентифікованих осіб. Хоча ці завдання традиційно вирішувалися окремо, їх інтеграція в єдину структуру створює як нові можливості, так і виклики, які дана робота має на меті подолати.

Інтеграція вже існуючих моделей у гілки MTL-архітектури також має додаткові переваги. Натреновані ваги моделей, розроблених для окремих завдань, можуть бути використані як початкові точки для навчання в нашій структурі. Це дозволяє здійснювати доналаштування (fine-tuning) для їхньої сумісності зі спільною основною мережею, замість повного перенавчання моделей з нуля. Ця стратегія не створює нові функціональні можливості, але забезпечує значне підвищення ефективності конвеєра та продуктивності моделі. У таблиці 1.1 представлено вибір моделей для кожної гілки нашої системи.

Таблиця 1.1.

Використані моделі

Завдання	**Модель**	**Архітектура**	**Рік**
Виявлення об'єктів	YOLOv11	ЗНМ	2024
Розпізнавання облич	AdaFace	ЗНМ	2022
Оцінка пози	ViTPose	ViT	2022

1.4.1. Виявлення осіб та облич

Виявлення об'єктів є однією з найбільш вивчених областей комп'ютерного зору. Розробка великих наборів даних, таких як COCO [12] та ImageNet [13], створила стандарти для навчання моделей. Зокрема, набір

даних WIDER FACE [14], що містить понад 390 тис. розмічених облич з високим ступенем варіативності, став ключовим для виявлення облич.

Рис. 1.3. Приклад набору даних WIDER FACE

Сімейство моделей YOLO (You Only Look Once) [16, 17] здійснило революцію в цій галузі, запропонувавши одноступеневий підхід, який дозволяє виявляти об'єкти в реальному часі. Сучасні вдосконалення, такі як архітектура CSPDarknet53 [18] та безякірне виявлення (anchor-free detection) [20], підтримують сімейство YOLO на передовому рівні. Завдяки широкому навчанню на бенчмарку COCO, YOLO є особливо придатним для виявлення осіб, оскільки він добре навчений розпізнавати людей у різноманітних контекстах. Компанія Ultralytics [21] також створила екосистему, яка дозволяє легко доналаштовувати моделі YOLO на власних наборах даних, таких як WIDER FACE, що є корисним для даної роботи.

1.4.2. Задача розпізнавання облич

Розпізнавання обличчя є ключовим компонентом нашої системи для ідентифікації осіб. Важливо розрізняти це завдання від подібних:

- Виявлення обличчя – це підзадача виявлення об'єктів, яка локалізує обличчя в кадрі.

- Верифікація обличчя – порівнює обличчя з єдиним еталонним зображенням для підтвердження ідентичності.

- Розпізнавання обличчя – порівнює обличчя з базою даних, що дозволяє ідентифікувати особу серед багатьох інших. Саме цей підхід є найбільш релевантним для нашої роботи.

Системи розпізнавання облич зазвичай використовують згорткові нейронні мережі (ЗНМ), такі як ResNet [25], для перетворення зображення обличчя у вектори (вкладення), де близькі вектори відповідають одній і тій же особі.

На рисунку 1.4 представлено порівняння трьох архітектур згорткових нейронних мереж (ЗНМ), розроблених для задачі класифікації зображень у наборі даних ImageNet.

Ліворуч: Архітектура VGG-19. Ця модель служить еталоном для порівняння, демонструючи значну обчислювальну складність з 19,6 мільярда операцій з плаваючою комою (FLOPs).

У центрі: Проста 34-шарова мережа. Ця архітектура, незважаючи на меншу кількість параметрів та обчислень (3,6 мільярда FLOPs), служить базовою для подальших модифікацій.

Праворуч: Залишкова мережа (Residual Network, ResNet). Ця модель має таку ж кількість шарів, як і проста мережа (34 шари), та аналогічну обчислювальну складність (3,6 мільярда FLOPs). Ключовою відмінністю є впровадження з'єднань-ярликів (shortcut connections), які перетворюють її на залишкову архітектуру.

Залишкова мережа базується на архітектурі простої мережі, до якої додаються з'єднання-ярлики, як показано на рисунку 1.4 (праворуч). Ці з'єднання дозволяють пропускати вхідні дані через кілька шарів, забезпечуючи прямий шлях для градієнта.

Існують два варіанти реалізації з'єднань-ярликів, залежно від відповідності розмірностей вхідних та вихідних даних:

- Тотожне відображення - цей підхід застосовується, коли вхідні та вихідні розмірності однакові. В такому випадку, з'єднання-ярлик прямо додає

вхідні дані до вихідних, не вводячи додаткових параметрів (суцільні лінії на рис. 1.4).

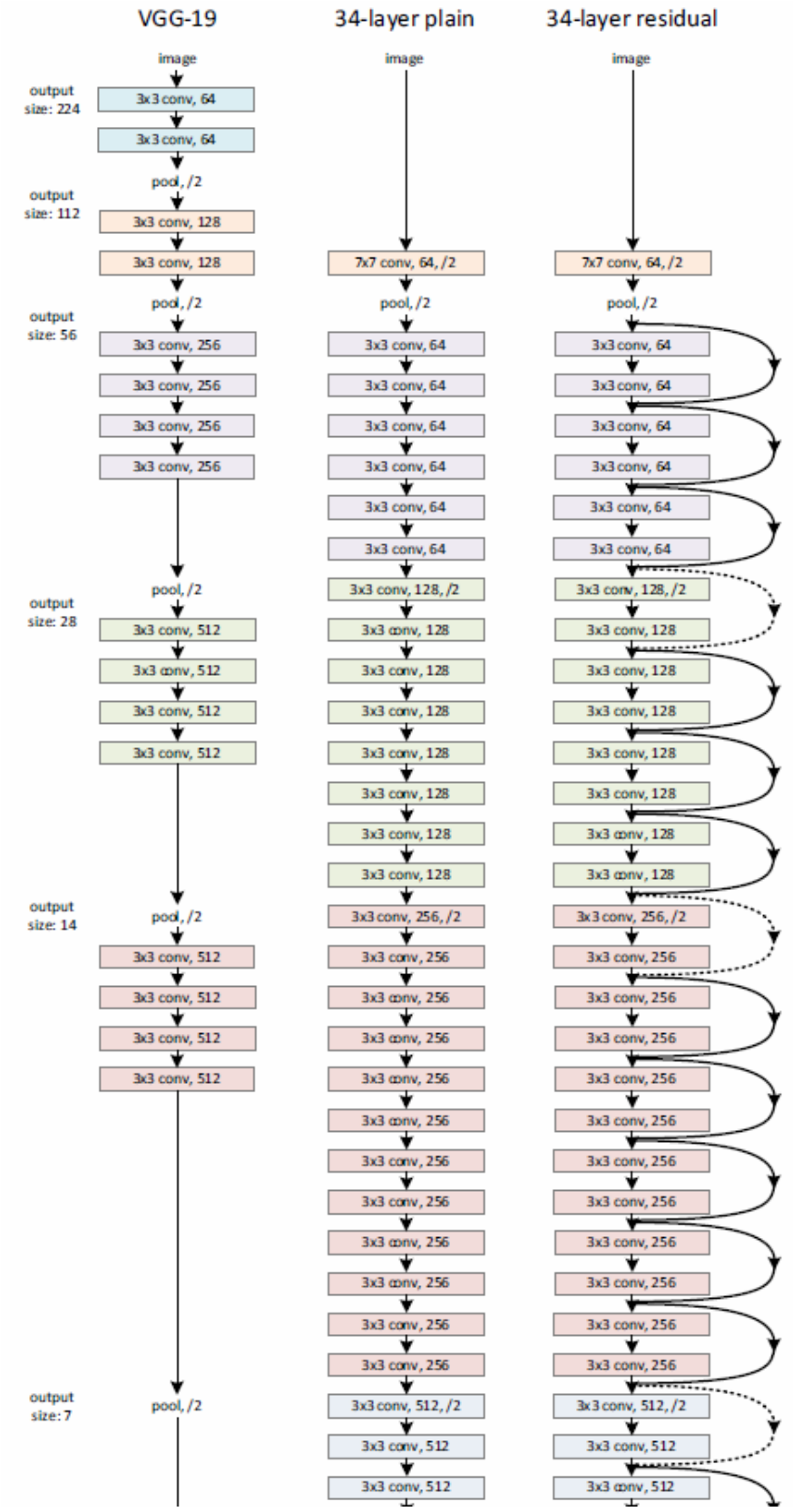

Рис. 1.4. Приклад архітектур мереж для ImageNet

- Відображення з проекцією - цей варіант використовується, коли розмірності змінюються (наприклад, збільшуються). Для узгодження розмірностей з'єднання-ярлик використовує операцію згортки 1×1,. Пунктирні лінії на рис. 1.3 вказують на це відображення. Альтернативним варіантом є використання тотожного відображення з додаванням нульових записів для узгодження розмірностей.

Для обох варіантів з'єднання-ярлики, що проходять через карти ознак, розмірність яких зменшується, виконуються з кроком 2.

Новітні роботи, як-от ArcFace [26], запровадили спеціальні функції втрат, які максимізують міжкласові відмінності та мінімізують внутрішньокласові варіації, що значно підвищує якість вкладень. Модель AdaFace [10] йде далі, адаптуючи функцію втрат залежно від якості зображення (рис. 1.5).

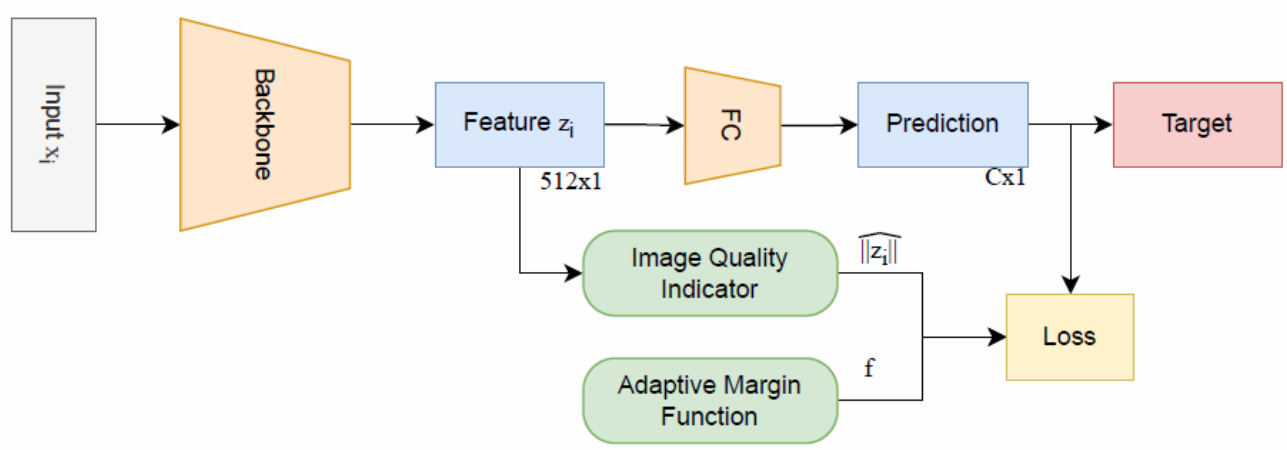

Рис. 1.5. Схема архітектури AdaFace

Це дозволяє системі ефективно працювати з неідеальними умовами (різне освітлення, ракурси, закриття), що є критично важливим для реальних застосувань. Ця здатність дозволяє точно визначати, які обмежувальні рамки відповідають особам, що цікавлять, і передавати їх для подальшої деталізованішої обробки.

1.4.3 Оцінка пози

Оцінка пози – це завдання визначення розташування ключових точок тіла. Існують два основні підходи:

- Підхід "знизу вгору" (bottom-up), спочатку виявляє всі ключові точки, а потім групує їх, щоб сформувати скелети людей. Цей метод не підходить для цільової оптимізації, оскільки він не використовує попередню ідентифікацію.

- Підхід "зверху вниз" (top-down) спочатку виявляє обмежувальні рамки людей, а потім оцінює позу всередині кожної з них. Цей підхід є більш ефективним, коли кількість людей у кадрі відома, і дозволяє інтегрувати ідентифікацію між етапами виявлення та оцінки.

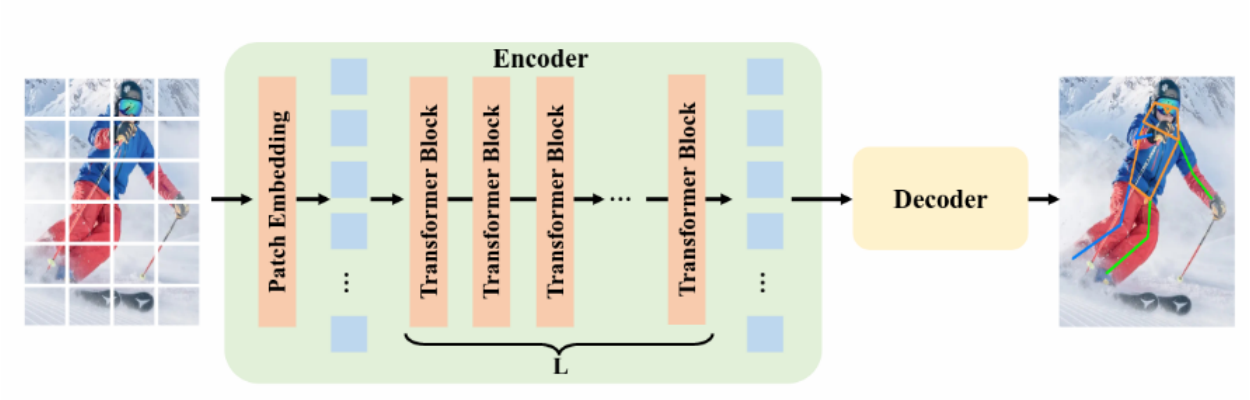

Рис. 1.6. Схема архітектури ViTPose

В останні роки відбувся перехід від ЗНМ до візуальних трансформерів (ViT) [30], які спочатку були адаптовані з архітектур для обробки природної мови [31].

ViTPose [11] є яскравим прикладом, що демонструє вищу продуктивність завдяки механізмам само-уваги (self-attention), які ефективно захоплюють глобальні залежності між частинами тіла (рис. 1.5). Це робить його особливо корисним для роботи зі складними позами та закриттями, які часто зустрічаються в реальних сценаріях.

1.5. Аналіз досліджень ідентифікації для відстеження осіб у динамічному середовищі

У сучасній літературі з оцінки пози існує низка досліджень, що експериментують з інтеграцією ідентифікації для відстеження осіб у динамічному середовищі. Ці підходи, хоча й мають схожі елементи, відрізняються своєю функціональністю та цілями.

1.5.1. Інтеграція ідентифікації у відстеження

Деякі існуючі рішення використовують ідентифікацію для підвищення стабільності відстеження. Наприклад, AlphaPose [33] інтегрує ідентифікатор особи для підтримки послідовної нумерації, навіть коли суб'єкт тимчасово зникає з поля зору. Ця система призначає унікальний внутрішній ID кожній особі, що з'являється в кадрі, і зберігає його навіть після її зникнення. Однак AlphaPose не надає можливості ідентифікації особи за зовнішніми ознаками (наприклад, обличчям), обмежуючи свою функціональність виключно внутрішнім відстеженням. Аналогічно, інші дослідження [34] використовують просторові характеристики для ідентифікації та оптимізації відстеження на індивідуальному рівні. Ідентифікація в таких випадках є внутрішньою і служить лише для оптимізації існуючих структур оцінки пози.

Методи оцінки пози людини (HPE) спрямовані на визначення 2D-координат ключових точок тіла на зображеннях або у відео. Ранні підходи спиралися на ручне виділення ознак, часто представляючи людське тіло у вигляді "людини-палиці". Однак сучасні досягнення у сфері глибокого навчання здійснили революцію у 2D HPE, значно покращивши точність та продуктивність.

Техніки глибокого навчання для HPE з однією особою поділяються на два основні підходи: методи регресії та методи на основі теплових карт.

- Методи регресії використовують наскрізну структуру, яка безпосередньо відображає вхідне зображення на координати суглобів або параметри моделі тіла.

- Методи на основі теплових карт прогнозують наближене розташування частин тіла та суглобів, використовуючи представлення у вигляді теплових карт. Сьогодні саме цей підхід є домінуючим у задачах 2D HPE.

Ранні дослідження HPE зосереджувалися на фреймворках регресії для прямого передбачення координат суглобів. Робота DeepPose була новаторською, вона використовувала каскадну глибоку нейронну мережу (DNN) з AlexNet як основною мережею. Її успіх сприяв переходу від традиційних підходів до методів глибокого навчання, зокрема до згорткових нейронних мереж (CNN).

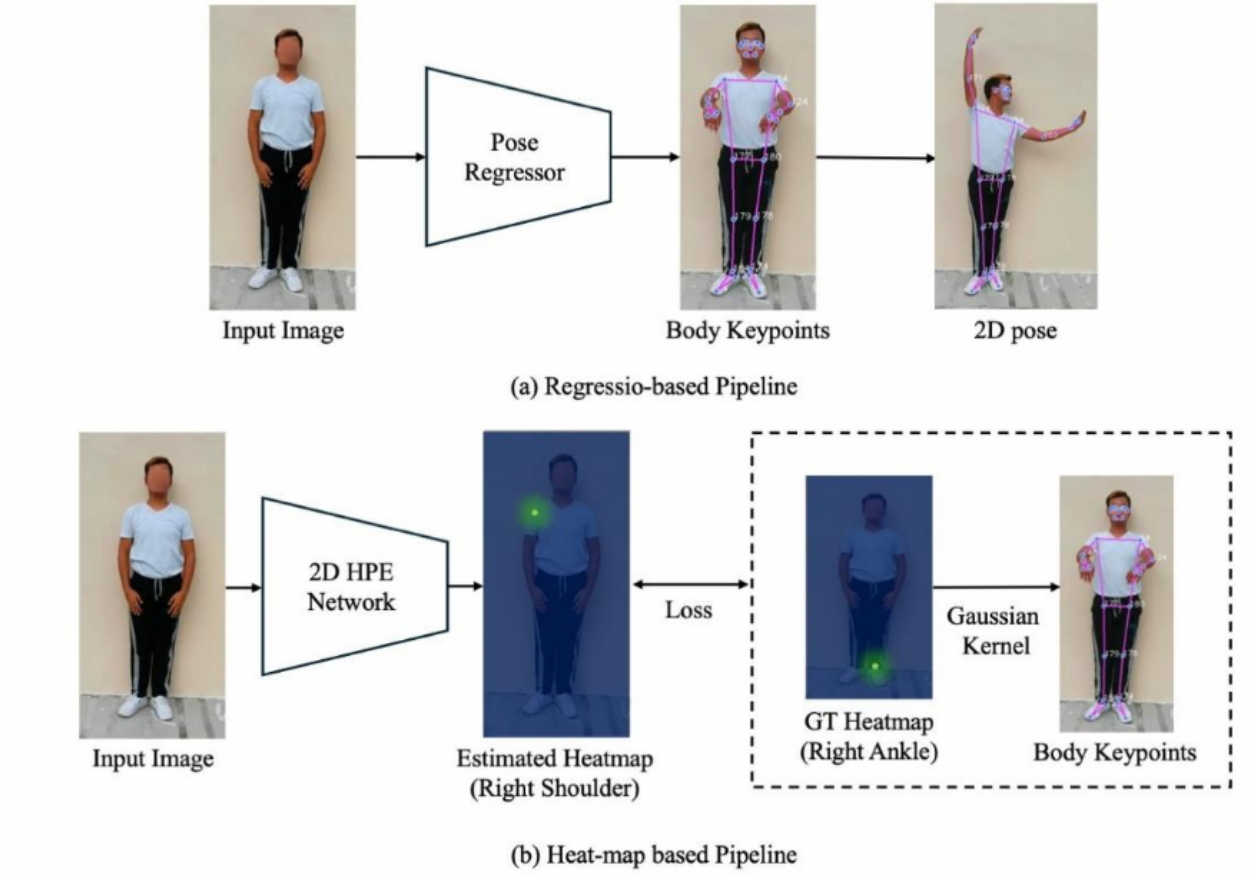

Рис. 1.7. Конвеєр регресії та теплових карт

На основі цього з'явилися методи "композиційної регресії пози". Вони часто базувалися на архітектурі ResNet-50 та використовували структурно-орієнтований підхід до регресії. Замість традиційного представлення на основі суглобів, вони застосовували представлення на основі кісток, що кодувало структуру тіла. Крім того, був запропонований наскрізний підхід до регресії, який використовував функцію Softmax у повністю диференційованому фреймворку для перетворення карт ознак на координати суглобів.

Система AlphaPose — це модульний фреймворк для оцінки пози кількох осіб у реальному часі. Її архітектура складається з п'яти основних модулів.

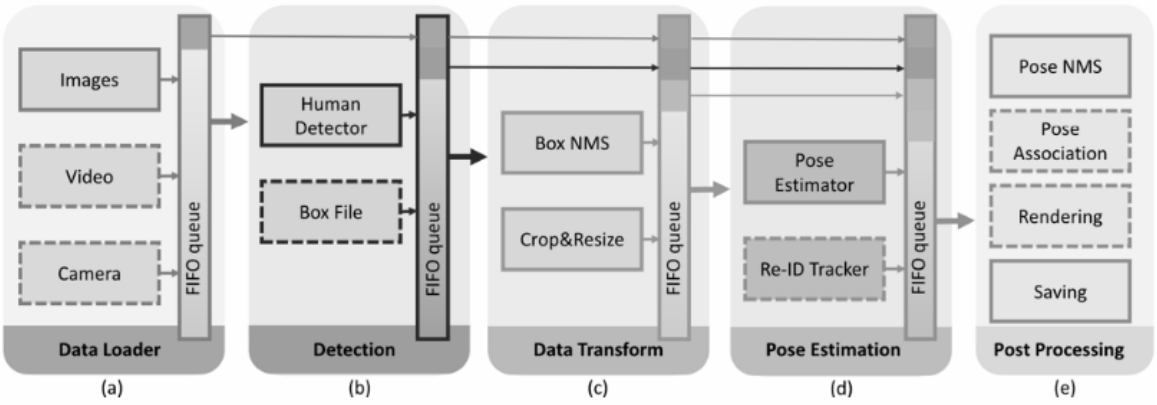

Рис. 1.8. Конвеєр AlphaPose

Модулі конвеєра AlphaPose:

- Модуль завантаження даних (Data Loading Module): Приймає на вхід зображення, відео або потік з камери.

- Модуль детекції (Detection Module): Відповідає за виявлення людей і створення пропозицій людських фігур. Наприклад, для цього можуть бути використані попередньо навчені детектори, такі як YOLOv3 або EfficientDet.

- Модуль трансформації даних (Data Transformation Module): Обробляє результати детекції та обрізає кожну окрему особу для подальшого аналізу.

- Модуль оцінки пози (Pose Estimation Module): Генерує ключові точки та/або ідентичність для кожної особи. Для цього AlphaPose використовує власну мережу FastPose, яка забезпечує баланс між точністю та ефективністю. FastPose використовує ResNet як основну мережу для виділення ознак і модулі Dense Upsampling Convolution (DUC) для підвищення просторової роздільної здатності. Опціонально, мережа може включати шари деформованої згортки (deformable convolution layers) для покращення представлення ознак.

- Модуль постобробки (Post Processing Module): Обробляє та зберігає результати оцінки пози.

Для досягнення високої швидкості обробки на великих наборах даних, AlphaPose використовує п'ятиступеневий конвеєр з мультипроцесингом. Цей підхід розподіляє обчислювальне навантаження між кількома процесами або потоками, що дозволяє паралельно виконувати завдання кожного модуля. Це значно прискорює інференцію, роблячи AlphaPose придатним для застосувань у реальному часі.

1.5.2. Вплив BlazePose на міжзадачну інтеграцію

Робота BlazePose стала основним джерелом для поточного дослідження. Дослідники BlazePose запропонували інноваційний підхід, де обмежувальна рамка всього тіла особи виводиться з детектора обличчя. Ця методологія демонструє, як інформація, отримана з одного завдання (виявлення облич), може бути використана для покращення іншого (оцінки пози), хоча BlazePose використовує виявлення обличчя для локалізації, а не для ідентифікації. Цей підхід підкреслює потенціал міжзадачної інтеграції, яку наше дослідження прагне розширити шляхом додавання ідентифікації.

BlazePose — полегшена архітектуру згорткової нейронної мережі (ЗНМ), оптимізована для оцінки людської пози в умовах обмежених обчислювальних ресурсів мобільних пристроїв. Модель здатна виводити 33

ключові точки тіла для однієї особи, працюючи зі швидкістю понад 30 кадрів на секунду на смартфоні Pixel 2.

Рис. 1.9. Результати BlazePose для йоги та фітнесу

Завдяки високій швидкості та ефективності, BlazePose є особливо придатним для застосувань у реальному часі, зокрема, в системах відстеження фізичних вправ та розпізнавання мови жестів

1.5.3. Прикладні застосування

Багато з цих досліджень вже знайшли своє застосування в комерційних продуктах. Наприклад, Mediapipe-VR-Fullbody-Tracking [4], що базується на BlazePose, є популярним рішенням для відстеження тіла у VR. У сфері охорони здоров'я системи, такі як KamiCare, використовують оцінку пози для виявлення падінь у літніх людей. Ці системи часто обмежують функціонал ідентифікації з міркувань конфіденційності, розмиваючи зображення або надаючи лише анонімні дані.

Незважаючи на значні досягнення в кожній окремій галузі, більшість існуючих систем інтегрують ідентифікацію виключно для внутрішніх

оптимізацій відстеження. Ключовий внесок даної роботи полягає в розробці конвеєра, який забезпечує цільову фільтрацію суб'єктів для системи оцінки пози, дозволяючи виявляти та аналізувати рухи виключно для вказаних осіб, що значно розширює практичну застосовність технології.

Висновки до розділу

В даному розділі, у результаті аналізу предметної області встановлено, що інтеграція задач розпізнавання облич та оцінки пози у єдиній системі є перспективним підходом, який дозволяє підвищити точність та узгодженість результатів. Виявлено обмеження традиційних методів і окреслено потенціал багатозадачного навчання для їх подолання.

РОЗДІЛ 2. ПРЕДСТАВЛЕННЯ МЕТОДОЛОГІЇ КЛАСИФІКАЦІЇ ПОЗИЦІЮВАНЬ НА ОСНОВІ МОДЕЛЕЙ РОЗПІЗНАВАННЯ ОБЛИЧ

2.1. Вибір моделі

Вибір моделей для даного дослідження ґрунтувався на трьох ключових критеріях:

- Сучасний рівень технологій (State-of-the-Art, SOTA) - моделі повинні були демонструвати високу продуктивність у відповідних бенчмарках (наприклад, COCO) і бути розроблені не раніше ніж за три роки до початку дослідження.

- Сумісність архітектур - усі обрані моделі повинні були бути побудовані на основі єдиного фреймворку, зокрема PyTorch, для спрощення інтеграції в архітектуру багатозадачного навчання (MTL).

- Обчислювальна ефективність - моделі повинні були мати невеликий розмір або доступні полегшені версії (наприклад, nano), що дозволяє прискорити процес навчання та зменшити обчислювальні витрати.

2.1.1. Обґрунтування вибору моделей

Для завдання виявлення об'єктів було обрано YOLOv11 nano [9] від Ultralytics. Ця модель є останньою версією сімейства YOLO, відомого своєю високою точністю та швидкістю. Її реалізація на PyTorch забезпечує повну сумісність з іншими компонентами системи.

Фреймворк YOLO (You Only Look Once) здійснив революцію у сфері виявлення об'єктів, представивши уніфіковану архітектуру нейронної мережі, що одночасно виконує завдання регресії обмежувальних рамок та класифікації об'єктів. Цей інтегрований підхід став значним відступом від традиційних двостадійних методів, пропонуючи наскрізну (end-to-end) можливість навчання завдяки повністю диференційованій структурі.

В основі архітектури YOLO лежать три фундаментальні компоненти:

- Backbone (основна мережа) - виконує роль первинного екстрактора ознак, використовуючи згорткові нейронні мережі для перетворення вхідних зображень у багатомасштабні карти ознак.

- Neck - діє як проміжний етап обробки, застосовуючи спеціалізовані шари для агрегації та посилення представлень ознак з різних масштабів.

- Head - функціонує як механізм передбачення, генеруючи фінальні вихідні дані для локалізації та класифікації об'єктів на основі уточнених карт ознак.

YOLOv11 є розвитком архітектури YOLOv8, впроваджуючи нові архітектурні інновації та оптимізації параметрів для досягнення вищої продуктивності виявлення, що проілюстровано на рисунку 2.1.

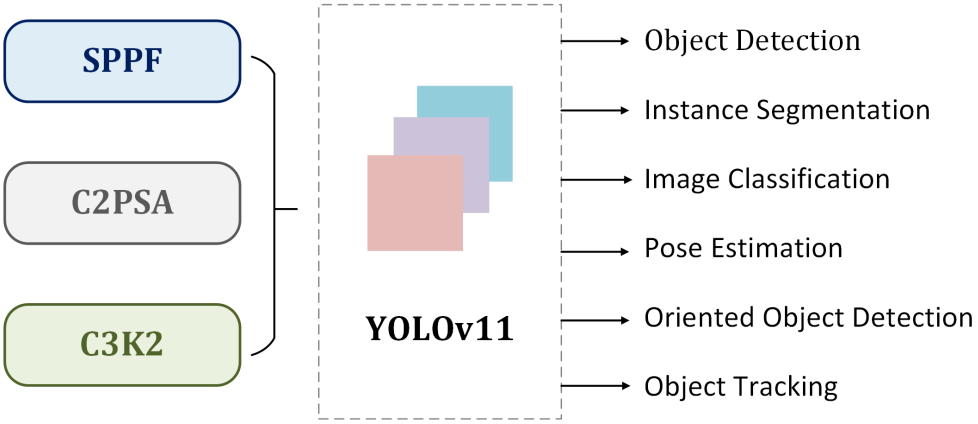

Рис. 2.1. Ключові архітектурні модулі в YOLOv11

Для оцінки пози було обрано ViTPose [11]. Ця модель показала високі результати в наборі даних COCO Keypoints з моменту її випуску у 2022 році, що підтверджує її SOTA-статус. Крім того, ViTPose розроблено з використанням бібліотеки MMPose [38], яка також базується на PyTorch, що відповідає критеріям сумісності.

Архітектура ViTPose підтримує максимальну простоту. Для оцінки теплових карт ключових точок після трансформерної основної мережі додається кілька шарів декодера. Як показано на рис. 2.2 (a), структура

уникає складних елементів, таких як skip-з'єднання або cross-attention, замінюючи їх простими шарами деконволюції та шаром передбачення.

Процес обробки даних відбувається наступним чином:

1. Вхідні дані: Зображення людини $X \in RH \times W \times 3$ подається на вхід моделі.

2. Розбиття на патчі: Шаром вкладення патчів (patch embedding layer) зображення трансформується у послідовність токенів $F \in RdH \times dW \times C$, де d (за замовчуванням 16) — коефіцієнт зменшення розмірності, а C — розмірність каналу.

3. Обробка трансформером: Вкладені токени послідовно обробляються декількома трансформерними шарами, кожен з яких складається з механізму самоуваги з кількома головами (MHSA) та мережі прямого зв'язку (FFN).

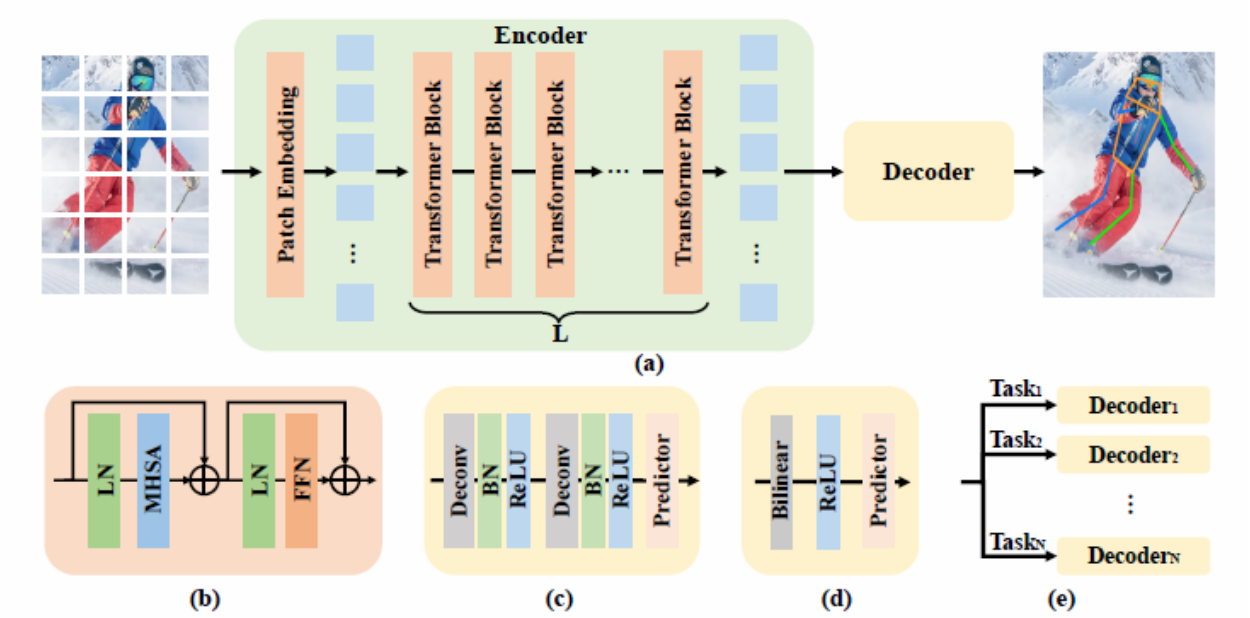

Рис. 2.2. Представлення ключових компонентів архітектури ViTPose

На рис. 2.2 представлено візуалізацію ключових компонентів архітектури: a) Загальна структура фреймворку ViTPose, b) Типовий трансформерний блок, c) Класичний підхід до декодера, d) Проста архітектура декодера, використана в ViTPose, e) Конфігурації декодерів для роботи з різними наборами даних.

Завдяки простоті своєї структури, архітектура ViTPose демонструє високу масштабованість. Розмір моделі можна легко адаптувати до вимог розгортання, змінюючи кількість шарів трансформера та розмірність ознак. Це дозволяє ViTPose ефективно використовувати переваги масштабованих попередньо навчених візуальних трансформерів без суттєвих модифікацій інших компонентів.

Для дослідження масштабованості, ми використовували попередньо навчені базові моделі різної потужності (наприклад, ViT-B, ViT-L, ViT-H [13] та ViTAE-G), доналаштовуючи їх на наборі даних MS COCO. Спостереження показали, що продуктивність моделі послідовно зростає зі збільшенням її розміру. Для ViT-H та ViTAE-G, де під час попереднього навчання використовувалися патчі розміром 14×14, ми застосовували заповнення нулями (zero padding) для стандартизації розміру патча до 16×16.

Гнучкість у попередньому навчанні (Pre-training data flexibility). Попереднє навчання на великих наборах даних, таких як ImageNet [12], є стандартною практикою для отримання хорошої ініціалізації. Однак це вимагає додаткових даних, що ускладнює задачу оцінки пози. Ми дослідили можливість навчання моделі лише на даних, пов'язаних з позою, використовуючи MAE для попереднього навчання на MS COCO та на комбінації MS COCO та AI Challenger. Результати виявилися багатообіцяючими: попри значно менший обсяг даних, ViTPose, навчений лише на даних про позу, демонстрував конкурентну продуктивність. Це свідчить про гнучкість ViTPose у навчанні та можливість адаптації до даних різного масштабу.

Гнучкість у роздільній здатності (Resolution flexibility). Ми змінювали розмір вхідного зображення та коефіцієнт зменшення розмірності d для оцінки гнучкості моделі. Щоб адаптувати ViTPose до вищої роздільної здатності, ми просто змінювали розмір вхідних зображень. Для роботи з нижчим коефіцієнтом зменшення розмірності (що відповідає вищій роздільній здатності ознак), ми змінювали крок (stride) шару вкладення

патчів, дозволяючи токенам перекриватися. Обидва підходи продемонстрували послідовне покращення продуктивності зі збільшенням роздільної здатності.

Гнучкість у типі уваги (Attention type flexibility). Обчислення повної уваги на картах ознак з високою роздільною здатністю вимагає значних обчислювальних ресурсів через її квадратичну складність. Для подолання цього, ми досліджували два підходи: зсувне вікно (shift window) та усереднення вікна (pooling window). Обидва методи допомагають передавати інформацію між вікнами, зменшуючи обсяг пам'яті та обчислювальні витрати. Ми довели, що ці дві стратегії взаємодоповнюють одна одну, покращуючи продуктивність без додавання нових параметрів.

Гнучкість у доналаштуванні (Finetuning flexibility). Як було доведено в галузі NLP [2], попередньо навчені трансформери добре узагальнюються для інших завдань навіть із частковим доналаштуванням. Наше дослідження показало, що ViTPose отримує порівнянну продуктивність із повністю доналаштованою моделлю, коли заморожені лише модулі самоуваги (MHSA). Це вказує на високу ефективність часткового доналаштування.

Гнучкість у завданнях (Task flexibility). Оскільки декодер ViTPose є простим і легковельким, можна використовувати кілька декодерів з мінімальними додатковими витратами для роботи з різними наборами даних оцінки пози, використовуючи спільний кодувальник (backbone encoder). Ми продемонстрували цю гнучкість, використовуючи спільний кодувальник для оцінки теплових карт на кількох наборах даних.

Вибір моделі для розпізнавання облич був більш складним. Відхилено GhostFaceNets через відсутність реалізації на PyTorch та ArcFace через його вік. Найбільш вдалим вибором стала AdaFace [10], опублікована у 2022 році. Її архітектура, побудована на основі PyTorch Lightning, відносно невелика. Крім того, AdaFace має унікальну здатність адаптуватися до низькоякісних зображень, що є критично важливою перевагою для розпізнавання облич з великої відстані або в умовах розмиття.

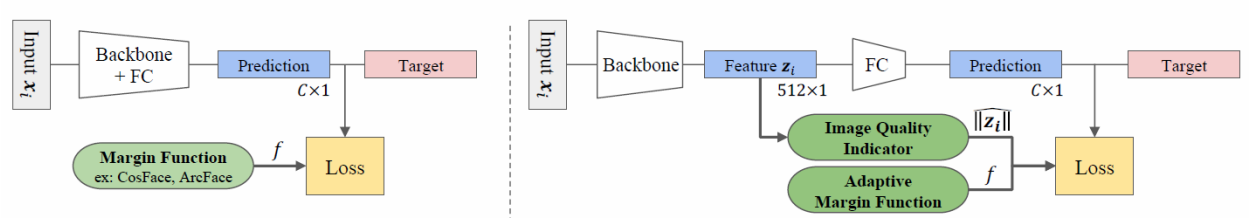

Рис. 2.3. Традиційна модель з функцією втрат (ліворуч) і модель AdaFace (праворуч)

Рисунок 2.3 ілюструє ключову відмінність між традиційною функцією втрат на основі відступу (margin-based softmax loss) та AdaFace.

Схема ліворуч на рис. 2.3 показує стандартний навчальний конвеєр для розпізнавання обличчя (FR) із застосуванням функції втрат на основі відступу. Ці функції, як, наприклад, SphereFace, CosFace та ArcFace, використовують спеціальний відступ для зменшення варіацій всередині класу, що підвищує дискримінаційну здатність моделі.

Схема праворуч на рис. 2.3 демонструє підхід AdaFace, що використовує адаптивну функцію відступу. Вона динамічно коригує відступ залежно від індикатора якості зображення.

Коли якість зображення оцінюється як низька, функція втрат зосереджується на простих зразках. Це запобігає спробам моделі навчитися розпізнавати неідентифіковані обличчя, що може негативно вплинути на загальну продуктивність.

Коли якість зображення висока, функція втрат натомість акцентує увагу на складних зразках, що дозволяє моделі покращити її дискримінаційні здібності в ідеальних умовах.

Такий адаптивний підхід дозволяє AdaFace ефективно навчатися на наборах даних, що містять зображення різної якості, що є поширеним явищем у реальних сценаріях.

2.1.2. Архітектура та інтеграція

Як спільна основна мережа для MTL-архітектури було обрано ResNet50. Хоча ResNet не є найновішою моделлю, вона є загальноприйнятим і надійним вибором для завдань MTL у комп'ютерному зорі. Це рішення було прийняте через складність бенчмаркінгу основних мереж і загальну доцільність використання перевіреної архітектури.

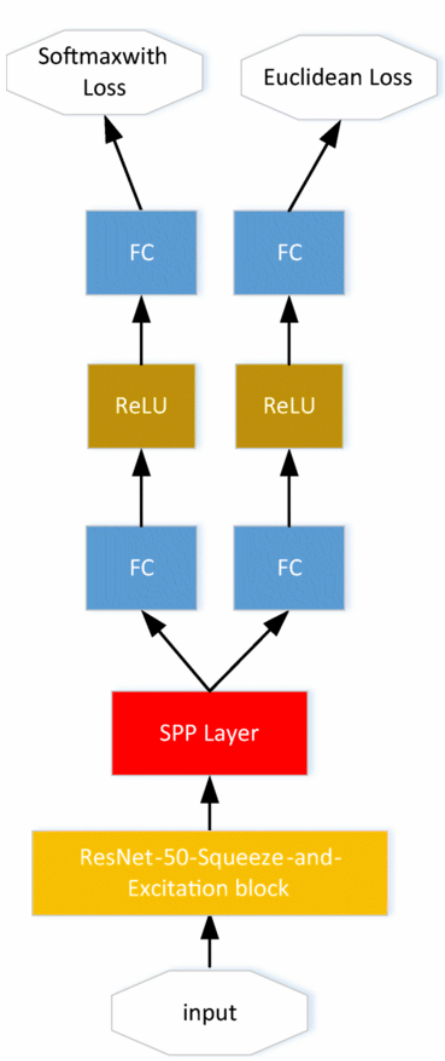

Рис. 2.4. Архітектура багатозадачного навчання з використанням основної мережі ResNet50

Представлена архітектура багатозадачного навчання (MTL) оптимізує обробку даних шляхом спільного використання ознак. На відміну від

моделей, що обробляють кожне завдання окремо, цей підхід використовує єдиний конвеєр для кількох взаємопов'язаних завдань.

Процес обробки відбувається наступним чином:

- Просторова агрегація: Вихідні дані з останнього згорткового шару подаються в шар просторової піраміди (SPP Layer). Цей шар виконує агрегацію просторових ознак, забезпечуючи фіксований розмір вектора ознак, незалежно від розміру вхідного зображення.

- Злиття даних: Агреговані результати потім передаються в структури з повністю зв'язаними шарами (fully-connected structures). Ці шари відповідають за злиття просторової інформації, перетворюючи її на формат, придатний для конкретних завдань.

- Генерування результатів: Після проходження через ReLU-активацію кожен потік (гілка) виводить результати для свого завдання. Це можуть бути дані різних розмірностей, залежно від вимог.

- Зворотне поширення помилки: Нарешті, для кожного вихідного результату застосовується відповідна функція втрат. Розраховані значення втрат використовуються для зворотного поширення помилки (backpropagation), що дозволяє оптимізувати параметри всієї моделі.

Такий підхід значно підвищує ефективність, оскільки спільна обробка ознак у перших шарах зменшує обчислювальні витрати, а зворотне поширення помилки одночасно оптимізує модель для всіх завдань

Для інтеграції моделей кожна гілка (YOLO, ViTPose, AdaFace) була адаптована. Вихідний шар ResNet50 було видалено, і замість нього додано адаптерний шар для кожної гілки.

Адаптерний шар забезпечує плавний перехід від виходу ResNet до першого шару відповідної гілки, дозволяючи ефективно передавати дані. Для мінімізації часу та вартості навчання були використані найменші версії кожної з обраних моделей.

2.2. Опис наборів даних

Для навчання та доналаштування кожної гілки моделі MTL використовувалися відповідні попередньо навчені моделі. Основна мета полягала в доналаштуванні на оригінальних навчальних наборах даних, щоб навчити модель ефективно інтегрувати спільну основну мережу в гілкові. Важливо, що використання нових наборів даних є небажаним, оскільки це унеможливлює проведення еквівалентного порівняння з результатами оригінальних моделей і може призвести до збіжності моделі до небажаних локальних мінімумів.

2.2.1. Адаптація та сумісність даних

Ключовим технічним аспектом, який вимагав корекції, була необхідність стандартизації форматів вхідних даних для сумісності з основною мережею ResNet. Це означає, що всі набори даних повинні бути трансформовані до стандартного вхідного формату ResNet, який може відрізнятися від оригінального формату кожної гілкової моделі. Наприклад, ViTPose очікує вхідні зображення розміром 256×192 пікселів, тоді як ResNet приймає квадратні зображення. Для вирішення цієї проблеми вхідне зображення спочатку подається на ResNet у квадратному форматі, після чого додаткові адаптерні шари змінюють його форму для передачі до ViTPose.

2.2.2. Вибір наборів даних для кожного завдання

Для завдання виявлення осіб використовувався набір даних COCO, що став галузевим стандартом для виявлення об'єктів. Хоча COCO містить багато класів, навчання було зосереджене на класі "особа" для оптимізації детектора.

Для доналаштування детектора обличчя використовувався набір даних WIDER FACE. Цей набір, що містить понад 393 тис. розмічених облич з

високою варіативністю, був адаптований до формату навчання YOLO, де обличчя визначені як єдиний клас.

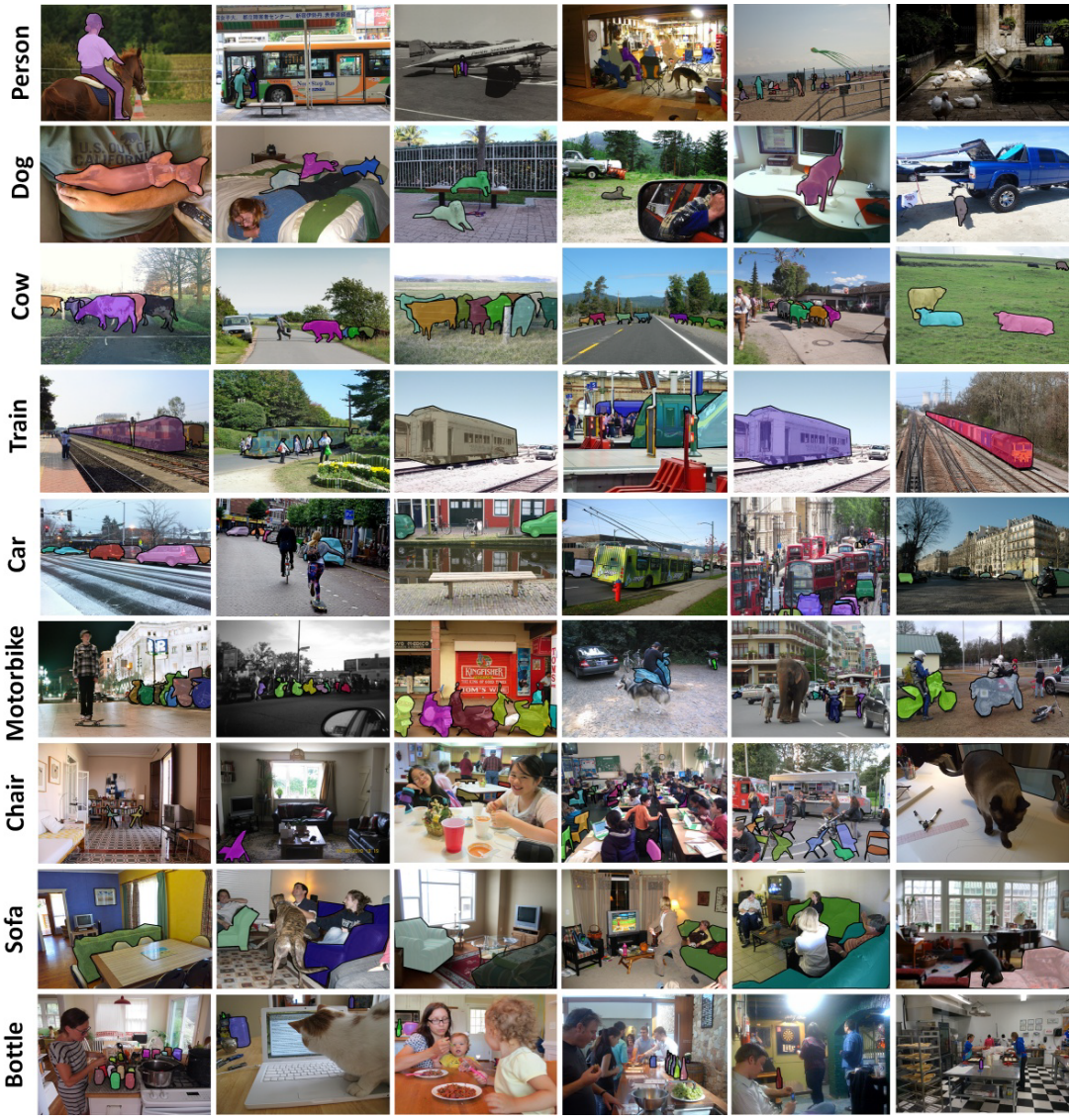

Рис. 2.5. Зразки анотованих зображень у наборі даних MS COCO

Модель AdaFace була доналаштована на наборі даних MS1MV2, який є одним із найпоширеніших у дослідженнях розпізнавання облич. MS1MV2 є покращеною версією набору даних MS-Celeb-1M і був основним набором даних для навчання AdaFace. Він складається з 85 000 ідентичностей, кожна з яких містить численні зображення обличчя. Загальна кількість зображень у наборі перевищує 5,8 мільйона. Процес очищення MS1MV2, проведений командою дослідників ArcFace, був критично важливим для усунення

помилкових маркувань, дублікатів та неідентифікованих зображень, що значно підвищило його надійність і якість.

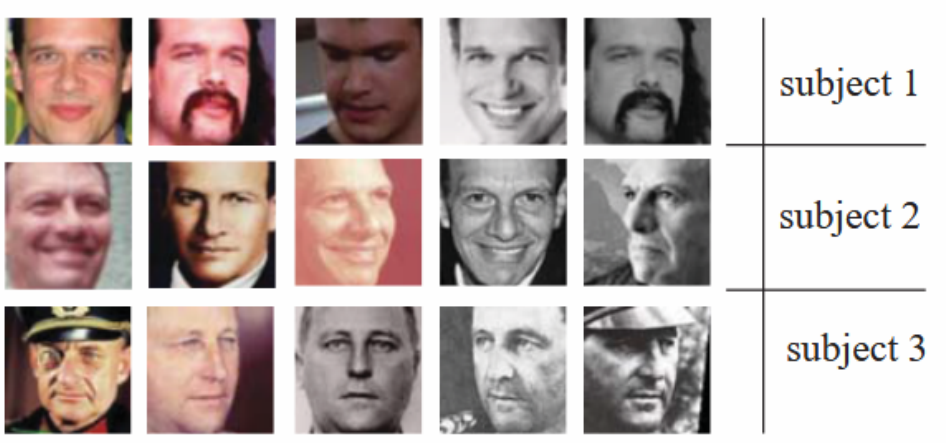

Рис. 2.6. Зображення з набору даних MS1MV2

Для завдання оцінка пози був використаний набір даних COCO Keypoints Detection. Подібно до інших завдань, цей набір широко визнаний як стандартний бенчмарк для оцінки пози.

На рисунку 2.7 представлено підсумок використаних наборів даних для кожної гілки моделі.

Завдання	Набір даних	Кількість зразків
Виявлення осіб	COCO	250 000 осіб
Виявлення облич	WIDER FACE	393 703 обличчя
Розпізнавання облич	MS1MV2	85 000 ідентичностей
Оцінка пози	COCO Keypoints	250 000 осіб

Рис. 2.7. Набори даних моделей

2.3. Підходи до навчання моделей

Для реалізації багатозадачної моделі (MTL) було досліджено два основні підходи. Перший полягав у створенні єдиної, уніфікованої моделі з навчанням за круговим методом (round-robin training) на базі PyTorch

Lightning. Другий підхід передбачав використання окремих, але ідентичних, заморожених основних мереж ResNet в кожному репозиторії моделей, з подальшим їх об'єднанням. Обидва підходи зіткнулися з практичними проблемами реалізації.

2.3.1. Підхід навчання за круговим методом в уніфікованій моделі

Спочатку архітектура була спроєктована на основі парадигми навчання за круговим методом, що є інтуїтивно зрозумілим для MTL. Перевага цього підходу полягає в тому, що основна мережа оптимізується одночасно для всіх завдань, дозволяючи кожній гілці адаптуватися до спільних представлень. Крім того, це дозволяло б використовувати потужні інструменти, такі як PyTorch Lightning, для керування процесом. На рис. 2.8 показано схему змін, внесених у цьому підході.

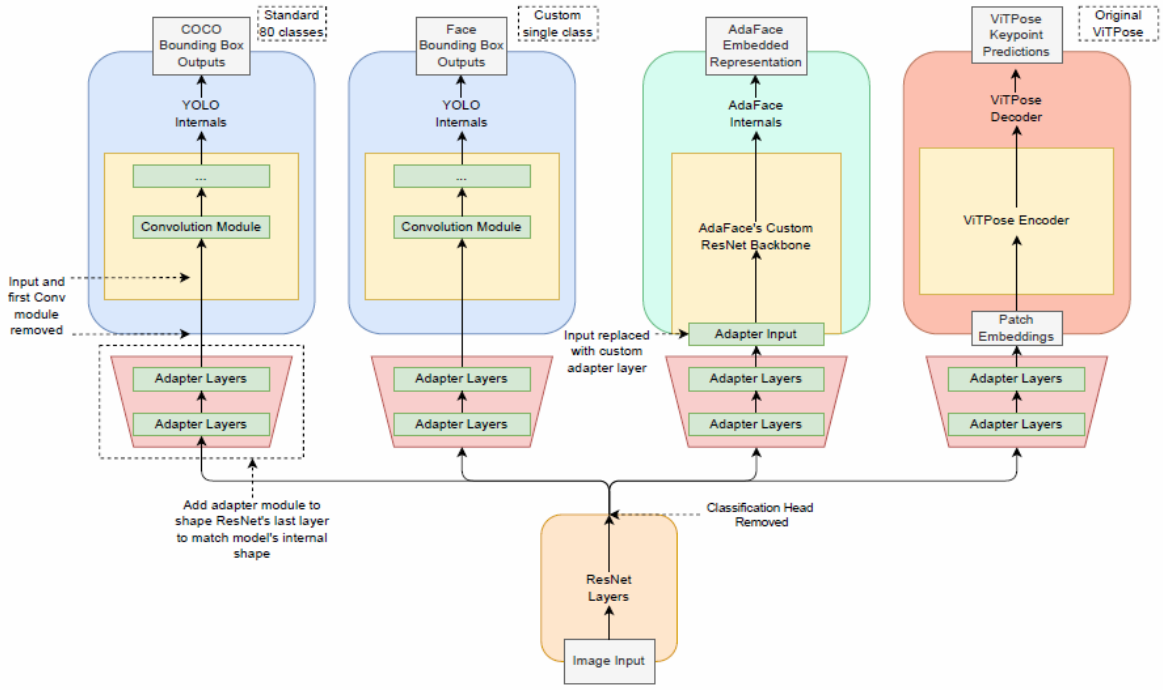

Рис. 2.8. Зміни в уніфікованій моделі навчання за круговим (циклічним) принципом

Проте, ця реалізація виявилася надзвичайно складною з технічної точки зору:

- Несумісність бібліотек. Кожна гілкова модель (AdaFace, ViTPose, YOLO) використовує власні оптимізовані бібліотеки (PyTorch Lightning, MMPose, Ultralytics), що робить їх несумісними для інтеграції в єдину навчальну петлю.

- Складність перезапису. Для досягнення уніфікації було необхідно переписати значні частини коду кожної моделі, включаючи їхні навчальні петлі. Це вимагало перереалізації надзвичайно складних алгоритмів, що є непрактичним.

- Порушення архітектури. Зміна вхідних шарів для передачі активацій від ResNet призвела до порушення внутрішньої архітектури YOLO, створюючи додаткові проблеми, які було б складніше вирішити, ніж залишити оригінальні входи.

Незважаючи на значні зусилля, цей підхід виявився занадто складним для практичної реалізації. Ці виклики призвели до пошуку альтернативної стратегії, яка б зберегла цілісність оригінальних моделей.

2.3.2 Підхід із замороженою основною мережею

Основним джерелом складнощів у першому підході була неможливість повторного використання оригінального навчального коду. Тому було запропоновано альтернативну стратегію, яка ґрунтується на замороженій основній мережі. Цей підхід використовує принципи успадкування класів та інкапсуляції для інтеграції ResNet у кожну з оригінальних моделей.

Ключові особливості цього підходу:

- Збереження цілісності. Кожна гілка моделі навчається окремо, що дозволяє використовувати оригінальний код авторів та конфігурації навчання.

- Заморожена основа. Основна мережа ResNet ініціалізується один раз і її шари заморожуються, що запобігає її оновленню під час навчання гілок. Це значно спрощує процес, хоча й втрачається міжзадачне навчання в самій основі.

- Адаптерні модулію. Між ResNet і кожною гілковою моделлю було розроблено окремий адаптерний модуль. Його завдання — трансформувати вихідні дані ResNet (2048 каналів) у формат, необхідний для входу кожної гілки. Наприклад, адаптер для YOLO використовує функцію активації SiLU, а для ViTPose – шар апсемплінгу.

Для прискорення навчання були завантажені попередньо навчені ваги як для ResNet, так і для кожної гілкової моделі. Це дозволило моделі адаптувати свої вже існуючі знання до нового конвеєра, замість навчання з нуля.

Хоча ця структура також має свої складнощі в навчанні, вона є більш практичною та відтворюваною для більшості багатозадачних моделей, оскільки зберігає цілісність та ефективність кожного окремого компонента.

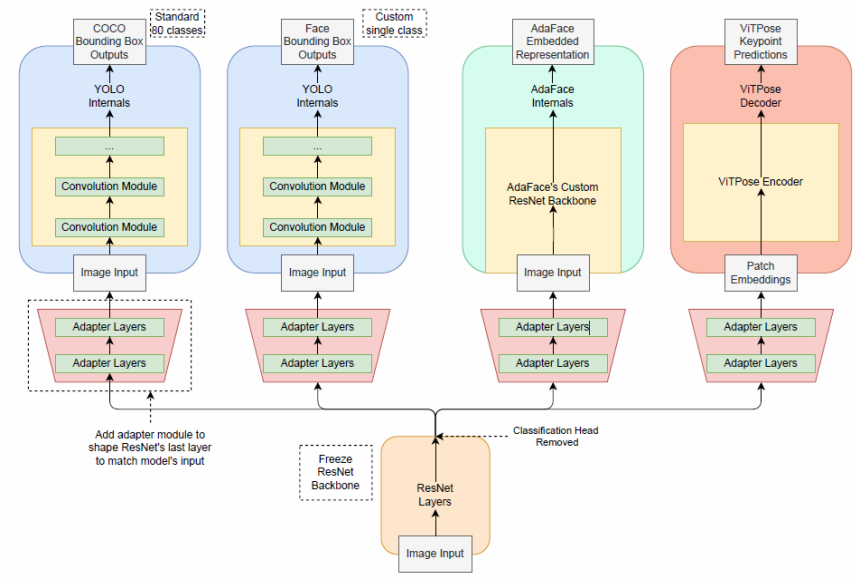

Рис. 2.9. Зміни моделі із замороженою основною мережею

2.4 Архітектура конвеєра цільової оцінки позиціювань

У даній роботі, окрім розробки багатозадачних моделей (MTL), запропоновано також послідовний конвеєр для реалізації цільової оцінки пози. Конвеєр складається з чотирьох послідовних фаз, що оптимізують робочий процес.

2.4.1. Загальна структура конвеєра

Конвеєр починається з фази попередньої обробки, де відбувається аналіз суб'єктів, що цікавлять. Після цього запускається фаза виявлення об'єктів для локалізації всіх осіб у кадрі. Потім, на фазі розпізнавання облич, система ідентифікує невідомих осіб, порівнюючи їх з раніше визначеними суб'єктами.

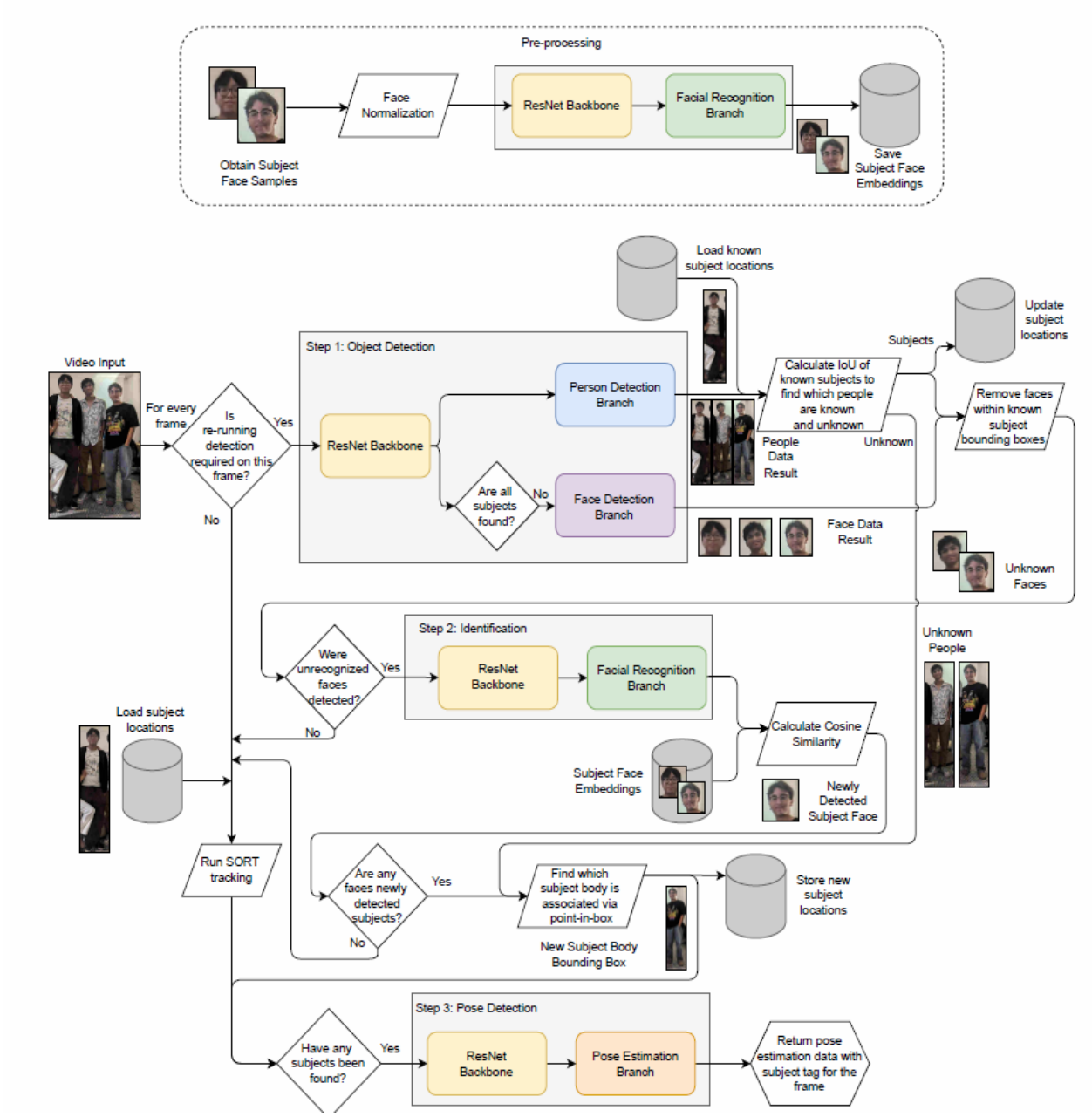

Рис. 2.10. Конвеєр цільового визначення пози

Нарешті, фаза оцінки пози застосовується виключно до ідентифікованих суб'єктів. Цей процес візуалізовано на рисунку 2.10, де людина зліва та справа є суб'єктами, а людина посередині – ні.

Цей алгоритм (рис. 2.10) описує конвеєр цільової оцінки пози, що інтегрує виявлення об'єктів, розпізнавання обличчя та оцінку пози для відстеження конкретних осіб у відеопотоці. Він складається з чотирьох основних фаз: попередня обробка, виявлення об'єктів, ідентифікація та оцінка пози.

Розглянемо короткий опис фаз алгоритму.

1. Фаза попередньої обробки

Ця початкова фаза призначена для підготовки системи до роботи. Користувач завантажує зображення облич суб'єктів. Кожне обличчя обробляється гілкою розпізнавання облич (Facial Recognition Branch) для генерації векторних вкладень, які зберігаються для подальшого порівняння.

2. Фаза виявлення об'єктів

На цьому етапі конвеєр обробляє кожен кадр відео. Спочатку запускається виявлення осіб (Person Detection Branch) для визначення їхніх обмежувальних рамок. Якщо не всі суб'єкти ідентифіковані, додатково запускається виявлення облич (Face Detection Branch). Система порівнює виявлені об'єкти з відомими суб'єктами, використовуючи метрику IoU (Intersection over Union). Якщо відповідності не знайдено, об'єкт позначається як "невідомий" і передається на наступну фазу.

3. Фаза ідентифікації

Ця фаза працює з невідомими обличчями, виявленими на попередньому етапі. Гілка розпізнавання облич генерує вкладення для кожного невідомого обличчя. Потім ці вкладення порівнюються з еталонними вкладеннями суб'єктів за допомогою косинусної подібності. Якщо знайдено збіг, обмежувальна рамка особи позначається ідентифікатором суб'єкта.

4. Фаза оцінки пози

На фінальному етапі система фільтрує всі виявлення і залишає лише обмежувальні рамки, які були ідентифіковані як суб'єкти. Для кожного ідентифікованого суб'єкта гілка оцінки пози (Pose Estimation Branch) обчислює ключові точки. Отримані дані передаються з оригінальним ідентифікатором суб'єкта. Це дозволяє уникнути обчислень для нецільових осіб і забезпечити ефективне відстеження пози.

2.4.2. Фаза попередньої обробки

Ця фаза є ініціалізуючою. На вході конвеєр отримує зображення облич суб'єктів разом з їхніми ідентифікаторами (рис. 2.11). Гілка розпізнавання облич обробляє ці зображення для генерації векторних вкладень. Ці вкладення слугують еталонними та використовуються протягом усього відео для ідентифікації суб'єктів.

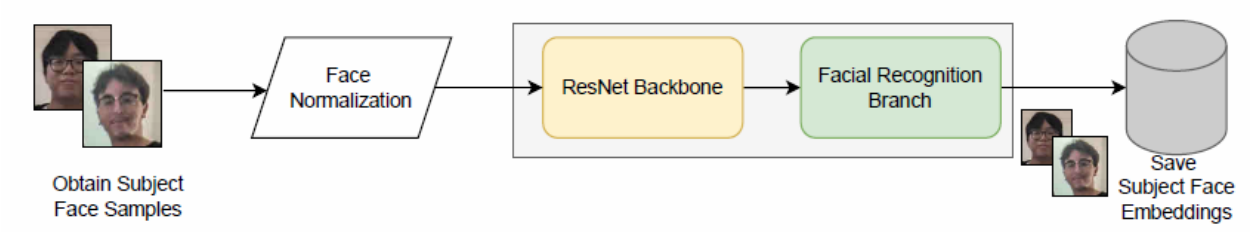

Рис. 2.11. Фаза попередньої обробки конвеєра

2.4.3. Фаза виявлення об'єктів

Першим етапом обробки відео є виявлення об'єктів (рис. 2.12). Ця фаза запускається на початку відео та повторюється через встановлені інтервали. Завданням є одночасне виявлення облич та осіб у повному кадрі, що дозволяє використовувати результати основної мережі для обох гілок. Якщо всі суб'єкти вже ідентифіковані, гілка виявлення облич може бути пропущена для оптимізації. Для подальшого підвищення ефективності можна інтегрувати алгоритм відстеження, такий як SORT, щоб зменшити частоту запуску повного виявлення об'єктів.

Після виявлення зберігаються області інтересу (ROI) як для суб'єктів, так і для несуб'єктів. Це дозволяє відстежувати їхні розташування в наступних кадрах. Кожне виявлення особи класифікується як "суб'єкт" або "невідома особа", що дозволяє передавати неідентифіковані обличчя на наступну фазу.

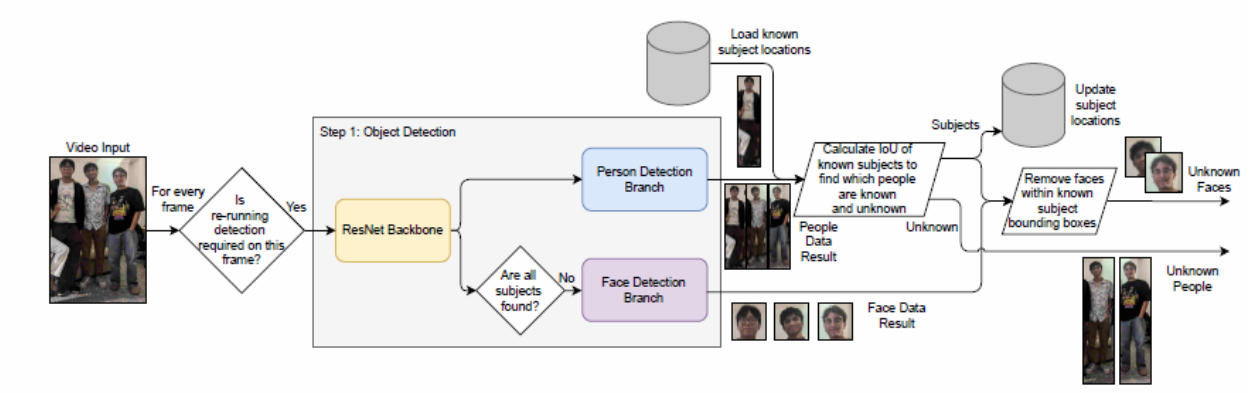

Рис. 2.12. Фаза виявлення об'єктів конвеєра

2.4.4. Фаза розпізнавання обличчя

На цій фазі обробляються неідентифіковані обличчя, виявлені на попередньому етапі. Кожен ROI обличчя проходить через гілку розпізнавання обличчя, яка повертає векторне вкладення (рис. 2.13).

Якщо вкладення невідомого обличчя збігається з еталонним, система визначає, якій обмежувальній рамці (bounding box) відповідає цей суб'єкт. Після цього нова обмежувальна рамка зберігається з ідентифікатором суб'єкта.

За допомогою косинусної подібності це вкладення порівнюється з еталонними вкладеннями суб'єктів. Якщо знайдено збіг, відповідна обмежувальна рамка особи позначається ідентифікатором суб'єкта та зберігається для подальшого відстеження.

У випадках, коли не всі суб'єкти ідентифіковані, фаза розпізнавання облич може повторюватися з певною періодичністю. Відстеження відомих

несуб'єктів допомагає уникнути повторного запуску ресурсомісткого розпізнавання облич для них. Після ідентифікації суб'єкта, його ROI передається на наступну фазу – оцінку пози.

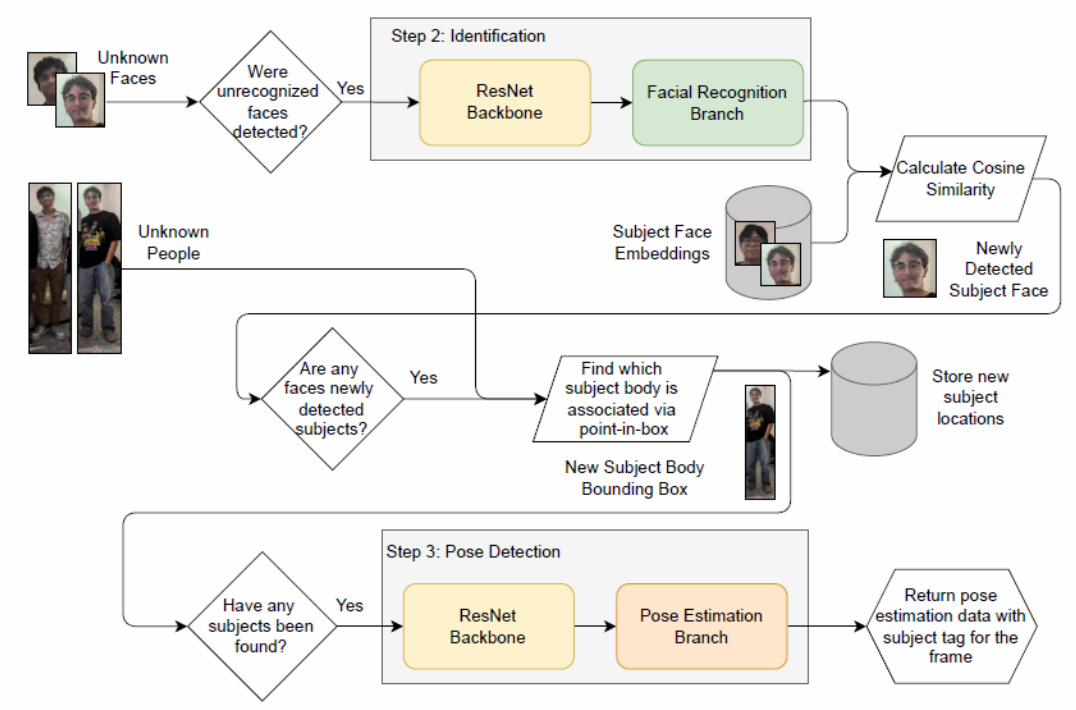

Рис. 2.13. Фази ідентифікації та визначення пози

2.4.5. Фаза оцінки пози

На фінальній фазі конвеєра, ROI ідентифікованих суб'єктів передаються через основну мережу до гілки оцінки пози. Система обчислює ключові точки для кожного суб'єкта в кадрі та передає ці дані разом з їхніми унікальними ідентифікаторами.

Навіть після ідентифікації всіх суб'єктів виявлення об'єктів може продовжувати виконуватися з періодичністю для оновлення їхніх місцезнаходжень, використовуючи, наприклад, метрику Intersection over Union (IoU) для порівняння з попередніми виявленнями. Це дозволяє відстежувати рух без повторного запуску розпізнавання обличчя. Застосування цього конвеєра мінімізує обчислювальні витрати, пов'язані з обробкою нерелевантних суб'єктів, і надає більш точні та релевантні дані для кінцевого користувача.

Висновки до розділу

В даному розділі запропоновано методологію класифікації позиціювань, яка базується на інтеграції сучасних моделей (YOLO, ViTPose, AdaFace) та застосуванні двох підходів до навчання: кругового та із замороженою мережею. Розроблена архітектура конвеєра забезпечує послідовне виконання етапів від виявлення облич до оцінки пози.

РОЗДІЛ 3. ІМПЛЕМЕНТАЦІЯ МЕТОДІВ КЛАСИФІКАЦІЇ ПОЗИЦІЮВАНЬ НА ОСНОВІ МОДЕЛЕЙ РОЗПІЗНАВАННЯ ОБЛИЧ

3.1. Концептуальна реалізація

Цей розділ представляє концептуальну реалізацію запропонованого конвеєра для цільової оцінки пози. Реалізація виконана у вигляді локальної програми на Python, що базується на проєкті easy_ViTPose, і слугує для демонстрації практичної життєздатності методу. Вихідний код реалізації доступний на GitHub в організації "Person-Recognition-for-Pose-Estimation".

3.1.1. Архітектура реалізації

Реалізація точно відтворює конвеєр, описаний у другому розділі, але, на відміну від теоретичної частини, не використовує навчені багатозадачні моделі (MTL). Замість цього вона безпосередньо інтегрує оригінальні моделі, (рис. 3.1). Цей підхід дозволяє перевірити логіку конвеєра, використовуючи перевірені, наявні компоненти.

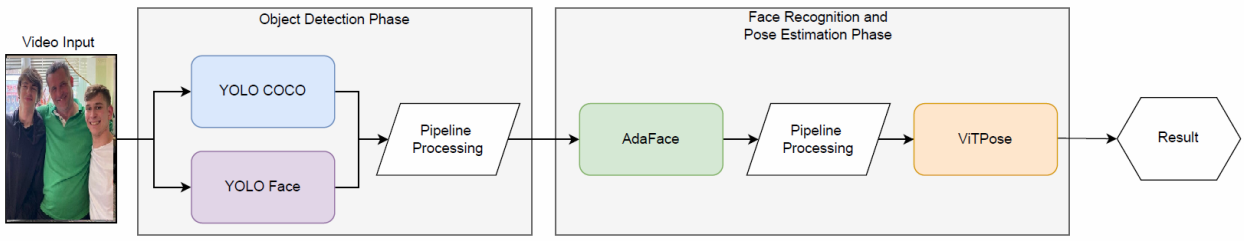

Рис. 3.1. Загальний огляд реалізації конвеєра

На рисунку 3.1 показана структура конвеєра для цільової оцінки пози, що включає три основні моделі: YOLO COCO, YOLO Face, AdaFace та ViTPose. Цей конвеєр працює послідовно, розділяючи процес на два основні етапи: фазу виявлення об'єктів і фазу розпізнавання облич та оцінки пози.

1. Фаза виявлення об'єктів

На цьому етапі відеопотік подається на вхід двох окремих моделей YOLO:

- YOLO COCO - ця модель, навчена на наборі даних COCO, відповідає за виявлення осіб у кадрі.

- YOLO Face - ця модель, навчена на наборі даних WIDER FACE, призначена для виявлення облич.

Після обробки обома моделями, результати проходять через "Pipeline Processing" для підготовки до наступного етапу.

2. Фаза розпізнавання облич та оцінки пози

Результати виявлення передаються до наступних моделей:

- AdaFace - ця модель, навчена на наборі даних MS1MV2, відповідає за розпізнавання облич. Вона ідентифікує, які обличчя належать до визначених суб'єктів.

ViTPose - модель, що використовує набір даних COCO Keypoints, виконує оцінку пози для ідентифікованих осіб.

Після обробки цими моделями, конвеєр видає кінцевий результат, який, ймовірно, включає дані про позу лише для цільових суб'єктів.

3.1.2. Фаза виявлення об'єктів

Алгоритм 3.1 демонструє реалізацію фази виявлення об'єктів. Процес починається з виявлення всіх осіб у кадрі з певною періодичністю. За допомогою алгоритму відстеження та метрики Intersection over Union (IoU), система намагається пов'язати щойно виявлені об'єкти з відомими суб'єктами.

Якщо нові виявлення відповідають існуючим, їхні розташування оновлюються. Якщо існуючий суб'єкт не знайдений у новому кадрі, він тимчасово видаляється для повторного пошуку. Якщо кількість знайдених суб'єктів менша за бажану, запускається також виявлення обличчя, що дозволяє передати невідомі виявлення на наступну фазу.

Алгоритм 3.1. Фаза виявлення об'єктів

```
процедура OBJECTDETECTION(frame, frameIndex, stepCount, foundSubjects, subjects)
    якщо frameIndex % stepCount == 0 тоді
        personDetections ← personDetector(frame)
        для всіх foundSubject у foundSubjects робити
            subjectLocation ← []
            для всіх personDetection у personDetections робити
                subjectLocation.append(IoU(personDetection, foundSubject))
            якщо max(subjectLocation) > subjectIoUMin тоді
                foundSubject ← max(subjectLocation)
            інакше
                видалити foundSubject
        якщо len(foundSubjects) != len(subjects) тоді
            faceDetections ← personDetector(frame)
            повернути personDetections, foundSubjects, faceDetections
        повернути personDetections, foundSubjects
    інакше
        повернути SORT(foundSubjects, frameIndex % stepCount)
```

3.1.3. Фаза розпізнавання обличчя

Алгоритм 3.2 описує процес ідентифікації обличчя. Ця фаза працює лише з обличчями, які не були ідентифіковані на попередньому етапі. Кожне невідоме обличчя проходить попередню обробку за допомогою допоміжних методів AdaFace, після чого модель генерує його векторне вкладення. Вкладення невідомого обличчя порівнюється з еталонними вкладеннями суб'єктів за допомогою матриці подібності.

Алгоритм 3.2. Ідентифікація облич

```
процедура IDENTIFY(faces, people, subjectEmbeds)
    unknownEmbeds ← []
    для всіх face у faces робити
        face ← preprocess(face)
        embedding ← AdaFace(face)
        unknownEmbeds.append(embedding)
        якщо isASubject(embedding, subjectEmbeds) тоді
            для всіх person у people робити
                якщо pointInBox(person, face) тоді
                    повернути person, face
    повернути null
```

Якщо буде виявлено збіг, система використовує просторову інформацію, щоб пов'язати обличчя з відповідною обмежувальною рамкою

особи. Потім цій обмежувальній рамці присвоюється унікальний ідентифікатор суб'єкта.

Розглянемо більш детально алгоритм ідентифікації облич. Це частина більшого конвеєра, який обробляє обличчя, що були виявлені, але ще не ідентифіковані.

Алгоритм приймає три вхідні дані: faces (обличчя, що потрібно ідентифікувати), people (список усіх виявлених осіб) та subjectEmbeds (збережені вкладення облич суб'єктів).

1. Попередня обробка та генерація вкладень.

Для кожного обличчя у списку faces виконується попередня обробка (preprocess). Після цього модель AdaFace генерує векторне представлення (embedding) для кожного обличчя, яке додається до масиву unknownEmbeds.

2. Порівняння.

Згенеровані вкладення порівнюються з еталонними вкладеннями суб'єктів (subjectEmbeds).

3. Зіставлення.

Якщо embedding відповідає одному із суб'єктів (isASubject), алгоритм перевіряє, яка обмежувальна рамка особи (person) пов'язана з цим обличчям (face), використовуючи логіку pointInBox.

4. Повернення результату: Якщо знайдено збіг, алгоритм повертає об'єкт особи та обличчя. Якщо збігів не знайдено, повертається null.

Таким чином, цей алгоритм відповідає за ідентифікацію невідомих облич у кадрі, зіставляючи їх з попередньо збереженими даними суб'єктів.

3.1.4. Фаза оцінки пози

Це найпростіший етап конвеєра. На цій фазі система фільтрує всі обмежувальні рамки, виявлені на попередніх етапах, і вибирає лише ті, що були ідентифіковані як суб'єкти. Потім до цих позначених обмежувальних рамок застосовується алгоритм оцінки ключових точок. Результатом є візуалізація, на якій оцінюється лише поза суб'єкта, в той час як інші особи в

кадрі ігноруються, що мінімізує обчислювальні витрати та надає користувачеві релевантну інформацію.

3.2. Результати концептуальної реалізації

Ключові висновки даної роботи ґрунтуються на результатах тестування концептуальної реалізації, що демонструє функціонування запропонованого конвеєра в реальних умовах.

3.2.1. Візуальний аналіз результатів

На рисунку 3.2 показано приклад роботи конвеєра з одним ідентифікованим суб'єктом та другою, неідентифікованою особою, що з'являється в кадрі.

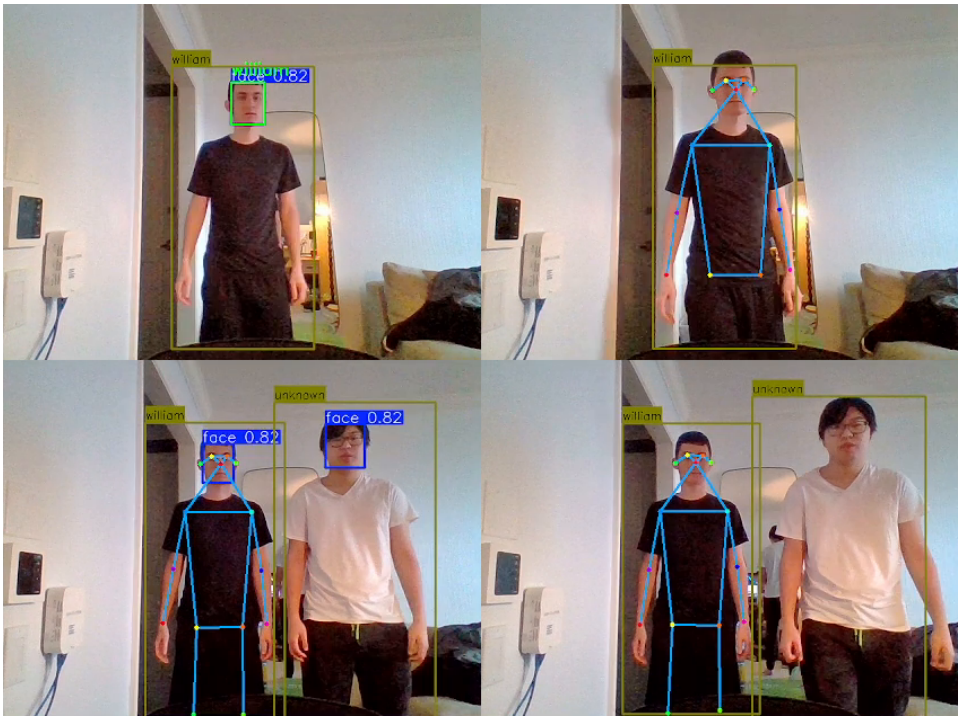

Рис. 3.2. Тест реалізації суб'єкта, а потім несуб'єкта

Візуальні анотації на зображеннях мають наступне значення:

- Коричнева рамка - вказує на виявлення особи та її відстеження. Містить мітку з ім'ям суб'єкта ("william") або "невідомий" для інших осіб.

- Синя рамка - позначає виявлення обличчя та його ймовірність.

- Зелена рамка - з'являється лише один раз і позначає успішне розпізнавання обличчя моделлю AdaFace, після чого відповідній рамці особи присвоюється ім'я суб'єкта.

Тест, показаний на рисунку 3.2, демонструє, як система спочатку ідентифікує суб'єкта ("william"), а потім успішно ігнорує нову особу, що з'явилася в кадрі, не застосовуючи до неї оцінку пози.

3.2.2. Порівняльний аналіз продуктивності

Було проведено порівняльне тестування між нашою концептуальною реалізацією та оригінальним рішенням easy_ViTPose на відео, де чотири особи перебували в кадрі, а відстеження велося лише для однієї з них. Результати, представлені на рисунку 3.3, ілюструють ключові переваги нашого підходу.

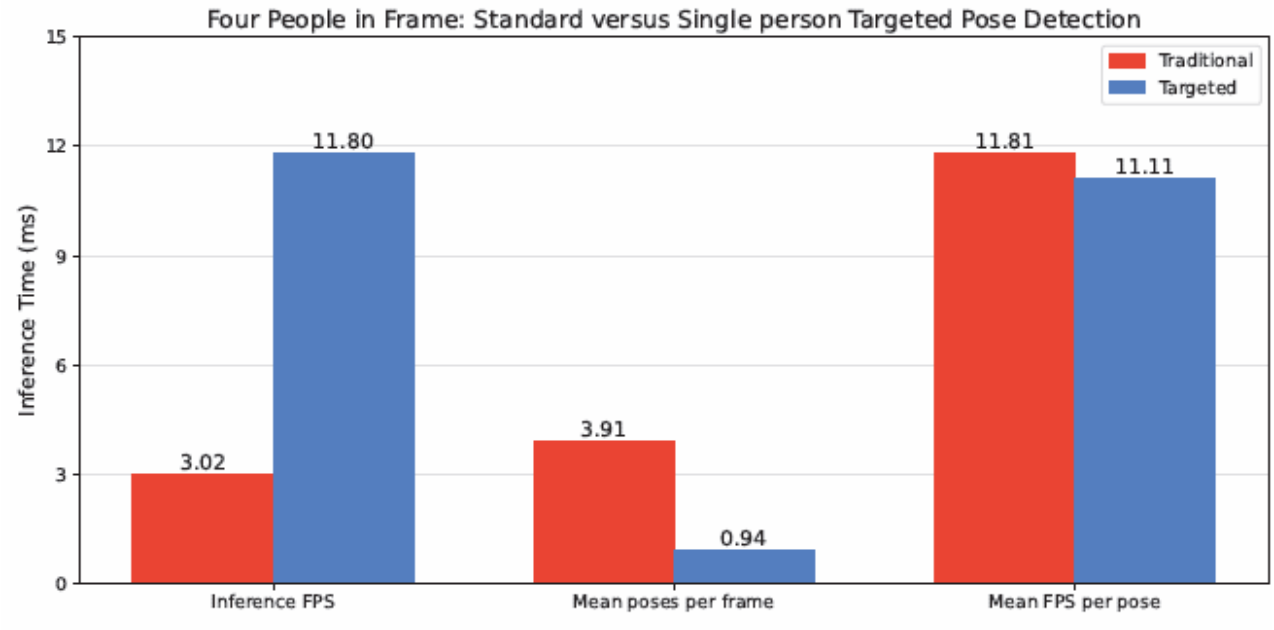

Рис. 3.3. Діаграма порівняння концептуальної реалізації з традиційним рішенням

Інференція FPS (Frames Per Second) — це метрика, яка вимірює, скільки кадрів за секунду може обробити модель машинного навчання під

час фази інференції, тобто під час застосування вже навченої моделі до нових даних для отримання прогнозів. Простими словами, це швидкість, з якою модель "думає" або робить висновки.

Чим вищий показник FPS, тим швидше працює модель. Це особливо важливо для завдань, які вимагають обробки в реальному часі, таких як:

- Відстеження об'єктів або рухів у живому відеопотоці.
- Швидка ідентифікація перешкод або дорожніх знаків.
- Миттєве накладання віртуальних об'єктів на реальний світ.

Наприклад, якщо модель має показник інференції 20 FPS, це означає, що вона може обробляти 20 зображень або кадрів за одну секунду. Ця метрика часто використовується для порівняння продуктивності різних моделей або алгоритмів.

Інференція FPS – пропонований конвеєр виявився приблизно в чотири рази швидшим за традиційне рішення, оскільки він оцінює позу лише для одного суб'єкта, а не для чотирьох. Цей результат підтверджує, що наш підхід ефективно вирішує проблему цільової оцінки.

Середня кількість поз/кадр - цей показник очікувано становив близько 3.91 для традиційного рішення та менше одиниці для нашого, що відображає його селективність.

FPS на позу. Хоча наша система має трохи нижчий FPS (11.11) порівняно з традиційним (11.81), це пояснюється додатковими обчисленнями, які виконує AdaFace під час ідентифікації. Однак цей незначний недолік компенсується загальною ефективністю.

3.2.3. Аналіз латентності

Аналіз латентності інференції, представлений на рисунках 3.4 та 3.5, показує, що, незважаючи на більшу затримку AdaFace (128.9 мс) порівняно з ViTPose (79.6 мс), загальна швидкість нашого конвеєра є вищою. Це досягається завдяки оптимізованій частоті запуску моделей.

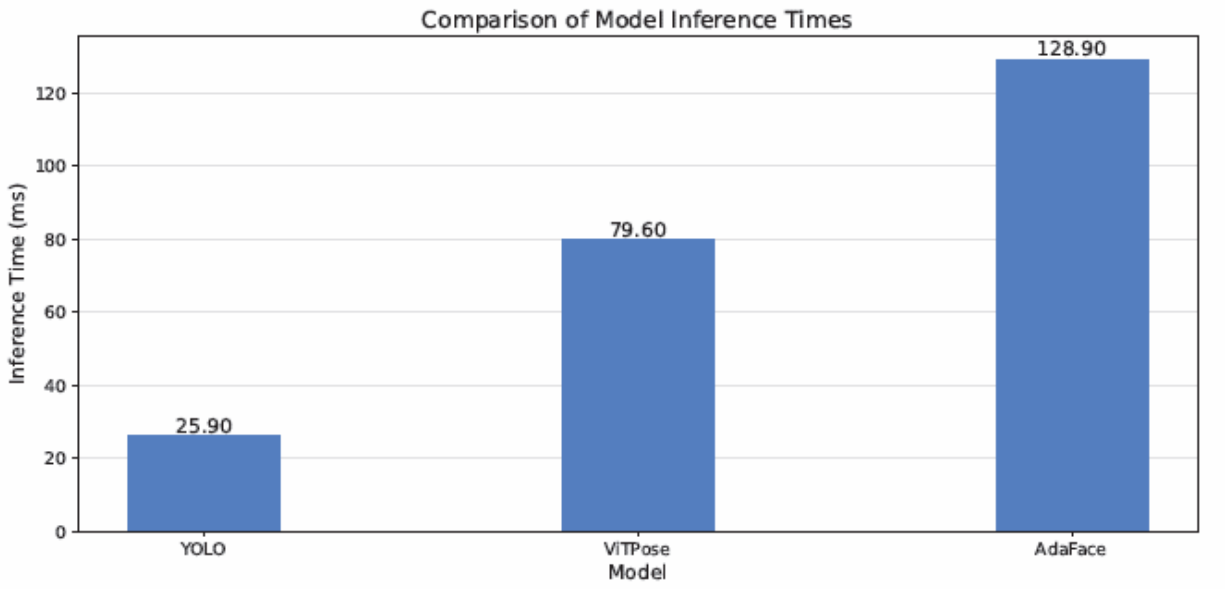

Рис. 3.4. Порівняння часу інференції моделей

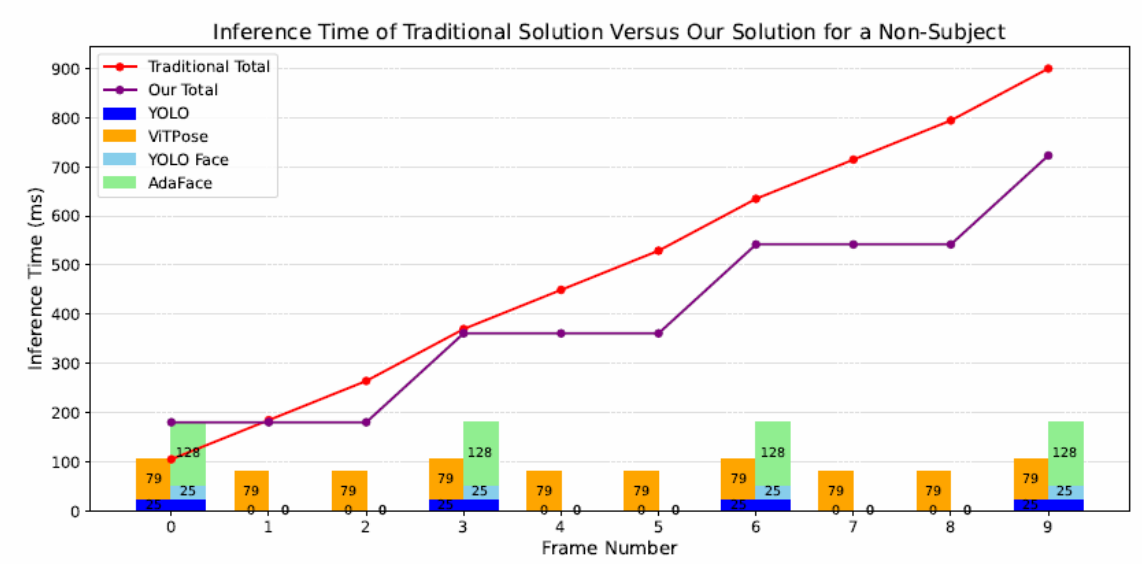

Рис. 3.5. Порівняння часу інференції: традиційний конвеєр VS пропонований

Традиційні рішення застосовують модель оцінки пози на кожному кадрі.

Наш конвеєр запускає AdaFace лише в ті моменти, коли необхідно ідентифікувати нового суб'єкта, а для відстеження використовує ефективніший YOLO.

Навіть при агресивному режимі (запуск YOLO кожні три кадри), наш конвеєр витрачає 180.7 мс на обробку трьох кадрів, тоді як традиційне

рішення — 264.7 мс. Це забезпечує збільшення швидкості на 31.7% при роботі з несуб'єктами.

Концептуальна реалізація продемонструвала практичну життєздатність запропонованого конвеєра. Вона успішно вирішує проблему селективної оцінки пози, ефективно розрізняючи суб'єктів та несуб'єктів, підтримуючи відстеження та адаптуючись до різних умов. Система не лише надає більш релевантні результати, але й є більш обчислювально ефективною порівняно з традиційними рішеннями.

3.3. Реалізація багатозадачної моделі з використанням підходу навчання за круговим методом

Даний розділ описує спроби реалізації багатозадачної моделі з використанням підходу навчання за круговим методом, незважаючи на відсутність успішного результату. Цей підхід, який координує навчання декількох гілок, вимагав складної структури, що базується на фреймворку PyTorch Lightning.

Для організації процесу навчання, кожна гілка моделі (YOLO, ViTPose, AdaFace) була інкапсульована в об'єкти TaskConfig разом із власним Module та DataModule. Управління процесом здійснювалося через зовнішній Trainer, який циклічно перемикався між завданнями після кожної епохи. Для відстеження результатів використовувалася система Weights and Biases. Необхідні моделі завантажувалися з оригінальних репозиторіїв та модифікувалися для інтеграції в єдину MTL-архітектуру.

На практиці цей підхід виявився надзвичайно складним через фундаментальні обмеження PyTorch Lightning, який не був розроблений для таких крайніх випадків. Координація перемикання між гілками, а також коректна логіка запису даних вимагали ручного втручання та складних обхідних рішень, що ускладнювало автоматизацію та підтримку системи.

3.3.1. Реалізація YOLO за круговим методом

Інтеграція YOLOv11 виявилася першою значною перешкодою. Спроби завантажити попередньо навчені ваги з бібліотеки Ultralytics та модифікувати архітектуру призвели до помилок, незважаючи на теоретичну сумісність. Більш того, внутрішня логіка Trainer у PyTorch Lightning, зокрема фаза "перевірки працездатності", конфліктувала зі спеціалізованим кодом Ultralytics, що призводило до збоїв ще до початку навчання. З огляду на ці проблеми, було розглянуто альтернативне рішення — перереалізація YOLOv11n з нуля, оскільки існуючі репозиторії не відповідали вимогам.

3.3.2. Реалізація ViTPose за круговим методом

Інтеграція ViTPose також виявилася нетривіальною. Хоча модель була доступна на HuggingFace, її репозиторій містив лише код для інференції, а не для навчання. Це змусило дослідників спробувати вручну відтворити навчальний процес, використовуючи інструменти з бібліотеки pycocotools [45]. Цей процес виявився набагато складнішим, ніж очікувалося.

3.3.3. Реалізація AdaFace за круговим методом

Навіть реалізація AdaFace, яка спочатку здавалася найпростішою, виявилася складною. Це було пов'язано з її залежністю від бібліотеки mxnet. На момент розробки mxnet був покинутим проектом Apache і виявився несумісним з іншими сучасними бібліотеками, що унеможливило його використання на доступних серверах. Це змусило розробників писати спеціальний код для обходу цієї проблеми.

3.3.4. Результати навчання за круговим методом

Незважаючи на значні зусилля, що дозволили запустити код без помилок, жодна з гілок не навчалася коректно. Цей досвід виявився повчальним, підкресливши кілька ключових проблем:

- Недостатнє тестування - не було проведено належного тестування кожного компонента ізольовано та в інтеграції.

- Неадекватний вибір інструментів - застосування інструментів (PyTorch Lightning) для сценарію, для якого вони не були призначені, призвело до численних аномалій.

- Відсутність ранньої оцінки підходів - було допущено припущення, що єдиний підхід є оптимальним, без належної оцінки альтернатив.

У підсумку, виявлено, що такий рівень складності вимагає глибокого розуміння всіх компонентів, а також ретельного планування та оцінки підходів на ранніх етапах.

3.4. Реалізація навчання із замороженою основною мережею

Альтернативним підходом до навчання стало використання замороженої основної мережі. Ця стратегія передбачала, що кожна гілка моделі буде доналаштована в її власному репозиторії. При цьому ваги спільної основної мережі залишалися незмінними, і оптимізації підлягали лише адаптерні шари та оригінальні шари гілкової моделі. Хоча такий підхід нівелює ключову перевагу багатозадачного навчання — спільну оптимізацію основної мережі — він дозволяв використовувати перевірені навчальні конфігурації оригінальних проєктів.

3.4.1. Реалізація YOLO із замороженою основною мережею

Реалізація YOLO була розпочата з використанням спрощеної кодової бази, що значно полегшило її редагування. Був створений новий файл, що інкапсулював адаптерний код та оригінальну модель. Усі шари ResNet були витягнуті в окремий клас, за винятком вихідного шару, і були заморожені.

Попри відсутність складнощів, характерних для попереднього підходу, навчання виявилося вкрай повільним. Навчання протягом 20 епох на GPU A100 зайняло 80 хвилин, а показник mAP@50 зріс лише з 0 до 0.014. Це свідчить про те, що заморожена основна мережа значно уповільнює процес

навчання, не забезпечуючи при цьому достатнього приросту продуктивності. З огляду на це, подальші зусилля були перенаправлені на інші моделі.

3.4.2. Реалізація ViTPose із замороженою основною мережею

Для реалізації ViTPose було обрано аналогічний підхід, адаптований під унікальний формат навчання MMPose. Конфігураційний файл моделі був модифікований для коректної обробки параметрів. Однак, проєкт зіткнувся з серйозними проблемами сумісності, оскільки MMPose вимагає застарілих версій бібліотек MMCV, які більше не підтримуються. Це призводило до незрозумілих збоїв, що унеможливлювало подальший прогрес.

3.4.3. Реалізація AdaFace із замороженою основною мережею

Проблеми, що виникли при інтеграції AdaFace у підході за круговим методом, збереглися і тут. Сильна залежність моделі від застарілого пакета mxnet та його несумісність із сучасним програмним забезпеченням та обладнанням зупинили будь-яку роботу в цьому напрямку. Хоча теоретично це мала бути найпростіша для реалізації гілка, відсутність ресурсів для подолання цих технічних перешкод виявилася критичною.

3.4.4. Висновки щодо замороженої основної мережі

Цей підхід був досліджений в обмежений час і показав, що його слід було розглянути раніше. Незважаючи на отримані результати, існує потенціал для комбінованого рішення, де основна мережа залишається незамороженою, а кожна гілка навчається в її власному оригінальному середовищі.

Це дозволило б передавати оновлену основну мережу між навчальними програмами, поєднуючи переваги спільної оптимізації з використанням надійного оригінального коду.

3.5. Обмеження та перспективи майбутніх досліджень

У даній роботі, попри досягнуті результати, існують певні обмеження, які слід враховувати. Вони стосуються методології, архітектури та вибору моделей.

3.5.1. Метод ідентифікації

Для ідентифікації осіб було обрано розпізнавання обличчя, оскільки воно є популярним і точним методом. Однак це обмежує застосування системи випадками, коли обличчя чітко видно. Альтернативним підходом, не дослідженим у цій роботі, є розпізнавання осіб за повним тілом. Цей метод може бути корисним, коли обличчя закрите (наприклад, маскою або VR-гарнітурою) або суб'єкт повернений спиною до камери. Хоча ідентифікація за повним тілом може бути менш точною у складних сценаріях, вона спрощує конвеєр, усуваючи необхідність у детекції обличчя. Для використання, як-от спортивна аналітика (гольф, теніс, бейсбол) або кінематограф, де обличчя є ключовим ідентифікатором, обраний підхід є оптимальним. Майбутні дослідження можуть зосередитися на ідентифікації за повним тілом для розширення сфери застосування.

3.5.2. Обмеження архітектури

Одним з найбільш значущих обмежень є необхідність використання двох окремих YOLO-голів для виявлення: одна для обличчя, інша для людини. Хоча YOLO може тренуватися на декількох класах одночасно, наявні набори даних не містять анотацій для обох класів в одному масштабі, достатньому для ефективного навчання. Це ускладнює модель та збільшує обчислювальне навантаження. Вирішенням цієї проблеми може стати створення нового, комплексного набору даних з анотаціями для обох класів, що дозволило б розробити більш ефективну архітектуру з єдиною головою виявлення.

Крім того, важливо розрізняти внесок роботи: першочерговим є концепція застосування розпізнавання обличчя в конвеєрі оцінки пози. Розробка багатозадачної моделі є додатковим внеском, оскільки конвеєр може бути реалізований і з використанням окремих моделей.

3.5.3. Вибір моделей

У цій роботі були використані найменші версії моделей для кожної гілки: YOLOv11 nano для виявлення об'єктів та ViTPose small для оцінки пози. Такий вибір був зумовлений необхідністю прискорити навчання та мінімізувати витрати. Однак, це обмежує потенційну точність. Застосування більших версій цих моделей може призвести до значного покращення точності, хоча і з додатковими обчислювальними витратами.

3.5.4. Майбутні напрямки досліджень

Майбутні дослідження можуть зосередитися на вирішенні вищезазначених обмежень.

1. Більші моделі.

Оцінка продуктивності більших архітектур для визначення оптимального балансу між точністю та обчислювальними витратами.

2. Нові набори даних.

Розробка універсальних наборів даних, що містять анотації як облич, так і повних фігур, для створення більш ефективних архітектур з однією головою виявлення.

3. Альтернативні методи ідентифікації.

Дослідження розпізнавання за повним тілом для розширення застосування моделі на сценарії, де розпізнавання обличчя є неможливим.

4. Оптимізація конвеєра.

Покращення продуктивності в реальному часі та інтеграція з більш досконалими алгоритмами відстеження, що виходять за рамки SORT.

Подальший розвиток у цих напрямках дозволить зробити цільову оцінку пози більш практичною та застосовною в ширшому спектрі сценаріїв.

Дане дослідження вирішує ключову проблему, що існує між теоретичними напрацюваннями в області оцінки пози та їх практичним застосуванням. Інтегруючи розпізнавання обличчя з оцінкою пози в єдиний багатозадачний конвеєр, ця робота пропонує ефективний підхід для цільового відстеження конкретних осіб. Завдяки цій методології, система виключає непотрібні обчислення для нецільових суб'єктів та надає можливість маркування кінцевих результатів, що значно підвищує їхню цінність.

Основними результатами даної роботи є:

- Архітектура конвеєра, що об'єднує виявлення об'єктів, розпізнавання обличчя та оцінку пози для забезпечення цільового відстеження.

- Спроба розробки двох різних багатозадачних моделей для застосування в запропонованому конвеєрі.

- Створення концептуальної реалізації, що демонструє практичну життєздатність селективної оцінки пози.

Ця робота розширює межі поточних реалізацій оцінки пози, роблячи її придатною для складних сценаріїв реального світу. Інтеграція розпізнавання обличчя не лише дозволяє вибіркове відстеження, але й забезпечує постійну ідентифікацію пози поміж кадрами, що є значним кроком уперед. Концептуальна реалізація слугує шаблоном для майбутніх технологічних застосувань.

Очевидні обмеження, виявлені в ході дослідження, відкривають перспективні напрямки для майбутніх робіт, включаючи об'єднання голів виявлення та використання моделей для ідентифікації повної фігури. Подальші дослідження в кожній із дисциплін, що були використані для створення цієї системи, також можуть призвести до значних покращень.

У цілому, ця робота пропонує цінний підхід, який робить оцінку пози більш практичною та корисною. Забезпечуючи цільовий аналіз окремих осіб,

наш фреймворк долає розрив між теоретичними можливостями та практичними вимогами, що може сприяти новим досягненням у різних сферах.

Висновки до розділу

Отже, в цьому розділі розроблено та реалізовано експериментальні моделі класифікації позиціювань, проведено їх тестування й порівняльний аналіз продуктивності. Доведено, що кругове навчання забезпечує вищу точність інтеграції завдань, тоді як підхід із замороженою мережею є більш ресурсоефективним.

ВИСНОВКИ

У ході виконання магістерської роботи на тему “Методи класифікації позиціювань на основі моделей розпізнавання облич” було здійснено комплексне дослідження теоретичних і практичних аспектів побудови систем, що поєднують задачі виявлення, розпізнавання облич та оцінки пози для подальшої класифікації позиціювань. Проведений аналіз, розробка методології та її реалізація дозволили сформувати низку узагальнень і висновків.

У першому розділі було проаналізовано предметну область визначення позиціювання об’єктів на основі розпізнавання облич. Детальний огляд сучасних підходів підтвердив, що поєднання задач ідентифікації та оцінки пози у єдиній системі є перспективним напрямом розвитку, здатним забезпечити підвищену точність і продуктивність. Було встановлено, що традиційні методи мають значні обмеження у гнучкості, масштабованості та адаптивності до динамічних умов. Розглянуті принципи багатозадачного навчання та технології гілкових завдань відкривають нові можливості для підвищення ефективності аналізу руху. Додатково було доведено, що використання моделей типу BlazePose у поєднанні з розпізнаванням облич сприяє міжзадачній інтеграції та практичному застосуванню в реальних системах відстеження.

Другий розділ було присвячено методології класифікації позиціювань. Було здійснено обґрунтований вибір моделей (YOLO, ViTPose, AdaFace) з урахуванням їх архітектурних особливостей та можливостей інтеграції. Проведено аналіз і підбір наборів даних, що забезпечують коректність і повноту навчання для різних підзадач. Запропоновано два підходи до навчання: за круговим методом та із замороженою основною мережею, що дозволило оцінити переваги і недоліки кожного з них у контексті багатозадачної системи. Опис архітектури конвеєра цільової оцінки позиціювань продемонстрував цілісність і структурованість методології, де

окремі фази (виявлення, розпізнавання, оцінка пози) інтегруються у єдиний алгоритмічний процес.

У третьому розділі було реалізовано концептуальну та експериментальну частини дослідження. Побудовано архітектуру системи та проведено її апробацію на різних наборах даних. Візуальний та кількісний аналіз результатів підтвердив ефективність застосованої методології. Порівняльні експерименти показали, що багатозадачний підхід із круговим навчанням забезпечує кращу інтеграцію завдань та більш високу точність у визначенні позицій, хоча потребує значних обчислювальних ресурсів. Натомість підхід із замороженою мережею виявився менш ресурсоємним, проте із дещо зниженою точністю. Виявлені обмеження архітектури та використаних моделей окреслили напрямки подальших досліджень, зокрема у сфері вдосконалення методів ідентифікації та оптимізації навчальних алгоритмів.

Результати роботи підтверджують наукову і практичну цінність інтеграції методів розпізнавання облич із системами оцінки позиціювань. Запропонована методологія дозволяє підвищити точність і швидкість аналізу руху, що робить її перспективною для застосування в задачах безпеки, біометричної ідентифікації, моніторингу активності користувачів та інтелектуальних систем спостереження. Запропоновані підходи демонструють потенціал для масштабування та адаптації під реальні динамічні умови. Водночас обмеження, пов’язані з ресурсними вимогами та залежністю від якості даних, визначають необхідність подальших досліджень у напрямку оптимізації моделей та використання більш збалансованих архітектур.

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. Lecture Notes in Computer Science, 9351.
2. Bao, H., Cheng, T., Dai, H., & Chen, J. (2020). A Survey on Multi-task Learning. ACM Transactions on Intelligent Systems and Technology.
3. Zeng, H., Yang, H., & Liu, Y. (2021). Multi-task learning for computer vision. arXiv preprint arXiv:2106.01254.
4. Cao, Q., Shen, L., Xie, W., Parkhi, O. M., & Zisserman, A. (2018). VGGFace2: A dataset for recognising faces across pose and age. 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018).
5. Deng, J., Guo, J., & Zafeiriou, S. (2019). ArcFace: Additive Angular Margin Loss for Deep Face Recognition. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).
6. Huang, G., Liu, Z., Maaten, L. V., & Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
7. Kaiming, H., Xiangyu, Z., Shaoqing, R., & Jian, S. (2016). Deep Residual Learning for Image Recognition. Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR).
8. Shao, J., Zhang, C., & Zhang, Y. (2020). GhostFaceNets: a ghost module based face recognition network. arXiv preprint arXiv:2010.04639.
9. Jocher, G., Chien, J., & Fang, W. (2024). YOLOv11: New SOTA for Object Detection, Instance Segmentation, and Pose. Ultralytics.
10. Kim, Y., Hwang, J., & Kim, M. (2022). AdaFace: Adaptive Training for Robust Face Recognition. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

11. Xu, Y., Chen, Z., & Chen, Y. (2022). ViTPose: Simple Yet Effective Baseline for Human Pose Estimation. Advances in Neural Information Processing Systems (NeurIPS).
12. Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Dollár, P. (2014). Microsoft COCO: Common Objects in Context. European Conference on Computer Vision (ECCV).
13. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., & et al. (2020). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. International Conference on Learning Representations (ICLR).
14. Yang, S., Luo, P., Loy, C. C., & Tang, X. (2016). WIDER FACE: A Face Detection Benchmark. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR).
15. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR).
16. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. Proceedings of the IEEE conference on computer vision and pattern recognition.
17. Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, Faster, Stronger. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
18. Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. arXiv preprint arXiv:1804.02767.
19. Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004.10934.
20. Alexey, A., Kryshtal, D., & Nesterov, I. (2022). YOLOv5: A Survey on its Improvements and Applications. Journal of Imaging.
21. Wang, C. Y., Bochkovskiy, A., & Liao, H. Y. M. (2022). YOLOv7: Trainable bag-of-freebies for real-time object detection. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

22. Jocher, G. et al. (2023). YOLOv8: A State-of-the-Art Model for Object Detection and Classification. Ultralytics.
23. Jocher, G. et al. (2024). YOLOv10: A Unified Network for Object Detection. Ultralytics.
24. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems (NIPS).
25. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. Proceedings of the IEEE conference on computer vision and pattern recognition.
26. Deng, J., Guo, J., Xue, N., & Zafeiriou, S. (2019). ArcFace: Additive Angular Margin Loss for Deep Face Recognition. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).
27. Li, Y., Wu, X., Fan, H., & Zhou, B. (2019). A survey on human pose estimation in videos. arXiv preprint arXiv:1902.04944.
28. Toshev, A., & Szegedy, C. (2014). DeepPose: Human Pose Estimation via Deep Neural Networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
29. Newcombe, R. A., Izadi, S., Hilliges, D., Mueggler, D., & et al. (2011). KinectFusion: Real-time dense surface reconstruction and tracking. 10th IEEE International Symposium on Mixed and Augmented Reality.
30. Sun, X., Xiao, B., Liang, F., Wei, Y., & et al. (2019). Deep High-Resolution Representation Learning for Human Pose Estimation. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).
31. Papandreou, G., Kokkinos, I., & Kanade, T. (2017). Towards Accurate and Multi-Person Pose Estimation in the Wild. arXiv preprint arXiv:1701.07328.

32. Krešimir, B., Rade, B., & Mario, H. (2018). Multi-Person Pose Estimation with Enhanced CNN and Part Refinement. 25th International Conference on Systems, Signals and Image Processing (IWSSIP).
33. Fang, H., Lu, J., & Zhou, X. (2017). AlphaPose: Whole-Body Pose Estimation and Tracking in the Wild. Proceedings of the IEEE International Conference on Computer Vision (ICCV).
34. Nie, Y., Chen, W., Zhang, K., & Ling, H. (2019). Monocular Human Pose Estimation from a Single Image in the Wild. arXiv preprint arXiv:1903.04543.
35. Bazarevsky, V., Grishchenko, I., & et al. (2020). BlazePose: On-device Real-time Human Pose Tracking. arXiv preprint arXiv:2006.10204.
36. Paszke, A., Gross, S., & et al. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. Advances in Neural Information Processing Systems (NeurIPS).
37. Guo, Y., Zhang, L., & et al. (2016). MS-Celeb-1M: A Dataset and Benchmark for Large-Scale Face Recognition. European Conference on Computer Vision (ECCV).
38. Falcon, W., & et al. (2020). PyTorch Lightning. (PyTorch Lightning GitHub repository).
39. Bewley, A., Ge, Z., & et al. (2016). Simple Online and Realtime Tracking. 2016 IEEE International Conference on Image Processing (ICIP).
40. Jolliffe, I. (1986). Principal Component Analysis. Springer Series in Statistics.
41. Wolf, T., & et al. (2019). HuggingFace: A Community for Open-Source NLP. Conference on Empirical Methods in Natural Language Processing (EMNLP).
42. Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In 2009 IEEE Conference on Computer Vision and Pattern Recognition (pp. 248-255).
43.

44. Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60(2), 91–110.
45. Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv preprint arXiv:1409.1556.
46. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going Deeper with Convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).
47. Lin, T. Y., & et al. (2014). Microsoft COCO: Common Objects in Context. European Conference on Computer Vision (ECCV).
48. Chen, T., Li, M., & et al. (2015). MXNet: A flexible and efficient deep learning library. arXiv preprint arXiv:1510.08582.