КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет інформаційних технологій
Кафедра інтелектуальних технологій

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня «магістр»

НА ТЕМУ:

«Дослідження та оцінювання аномалій в даних пози людини»

Галузь знань: 12 «Інформаційні технології»
Спеціальність: 122 «Комп'ютерні науки»
Освітньо-наукова програма «Технології штучного інтелекту»

Виконала:
студентка 2 курсу магістратури, групи ТШІ-21
__Стрюкова К.Є.___
(ПІБ)

Науковий керівник: __Снитюк В.Є._____
(ПІБ)

Доктор технічних наук, професор _____
(науковий ступінь, вчене звання)

Засвідчую, що в цій кваліфікаційній роботі
немає запозичень з праць інших
авторів без відповідних посилань

Студент(ка) _____
підпис

Кваліфікаційна робота допущена до захисту
рішенням кафедри *інтелектуальних технологій*

Протокол № _8_ від «_5_» травня 2022 р.

Зав. кафедри _____ доц. Іларіонов О.Є.
підпис

Київ 2022

ЗМІСТ

ВСТУП

Оцінювання пози людини є активним сферою дослідження в області комп’ютерного зору. Можна знайти сотні наукових робіт пов’язаних з цією темою. Причиною такого інтересу є широкий спектр застосування даної технології. В даній кваліфікаційній роботі буде розглянуто як можна застосувати даний підхід до задачі виявлення аномалії в позі людини.

Метою кваліфікаційної роботи є розробка технології оцінювання аномалій в даних пози людини, на основі результатів якої можна визначити аномальні пози на зображеннях та відповідним чином на них відреагувати.

Об’єктом дослідження у кваліфікаційній роботі є процеси оцінювання аномальної поведінки людини в повсякденному житті, яка свідчить про необхідність надання першої медичної допомоги.

Предметом дослідження випускної кваліфікаційної роботи роботі є моделі і методи оцінювання аномалій у даних пози людини з ціллю запобігання надзвичайним ситуаціям.

Актуальність роботи полягає в тому, що можливість виявлення аномалії в позах людей дасть змогу запобігти можливим надзвичайним ситуаціям пов’язаних зі здоров’ям людини.

РОЗДІЛ 1 ПРОБЛЕМИ ВИЯВЛЕННЯ ТА ОЦІНЮВАННЯ АНОМАЛІЙ В ДАНИХ ПОЗИ ЛЮДИНИ

Сьогодні камери відеоспостереження розташовані майже всюди, у кожному куточку світу. Установлення камер відеоспостереження в комерційних приміщеннях є ключовим заходом безпеки бізнесу для запобігання крадіжок і вандалізму протягом року. Зазвичай такі системи відеоспостереження, доповнені службами віддаленого моніторингу для забезпечення контролю над ситуацією в місті та в громадських закладах.

Зазвичай, моніторинг за камерами відеоспостереження здійснюється тільки у разі необхідності: коли потрібно переглянути відео для встановлення, що саме відбулося, або в місцях підвищеного контролю безпеки. Записи зберігаються в базі даних і найчастіше слугують інструментом для збору інформації.

Створення програмного забезпечення, яке дозволяє автоматично аналізувати та виявляти аномалії дозволило б покращити безпеку життів людей в великих містах. Коли людині стало погано та вона впала, коли людина поранена в результаті нападу і її життю загрожує небезпека – такі ситуації є цільовим об’єктом даної роботи. Така система може бути особливо ефективною вночі, коли на вулицях міста дуже мало людей, та у місцях з високим рівнем злочинності.

Тому, оцінювання аномалій в позах людини має важливе значення для удосконалення роботи правоохоронних органів та охорони здоров’я.

1.1 Роль оцінювання аномалій даних пози людини

Оцінка пози людини є однією з ключових задач комп’ютерного зору, яка вивчається вже більше 15 років. Однією з причин важливості оцінки пози людини є велика кількість додатків, які можуть отримати користь від такої

технології. Оцінювання пози людини дає можливість міркування вищого рівня в контексті взаємодії людини та; це також є одним із основних будівельних блоків для технології захоплення руху. Технологія оцінки пози людини корисна для вирішення широкого спектру задач: починаючи від анімації персонажів і закінчуючи клінічним аналізом патологій ходи.

Загалом, виявлення аномальних даних про позу людини має вирішальне значення для багатьох систем штучного інтелекту. Наприклад, системи моніторингу поведінки пацієнта можуть аналізувати поведінку пацієнта на основі руху пацієнта та прогнозування її пози [1]; системи допомоги водінню здійснюють прийняття рішень на основі прогнозування пози водія або пішохода [5, 6]; Системи взаємодії людини з комп'ютером використовують передбачення пози користувачів для проведення реабілітації [7] на основі Kinect технології [8]. Хоча методи відстеження пози вдосконалювалися з роками, аномальні оцінки поз, навіть якщо вони трапляються не часто, можуть призвести до катастрофічних подій, на що і буде звернено увагу у даній роботі.

Виявлення аномалії – це метод пошуку незвичайної точки або шаблону в заданому наборі даних. Термін «аномалія» також називають викидом. Раніше дослідники аналізу даних були зосереджені на інших техніках, таких як класифікація та кластеризація. Викиди відносяться до частини процесу очищення даних. Однак у 2000-му році такий погляд зазнав змін, коли дослідники виявили, що виявлення «ненормальних» речей може допомогти вирішити реальні проблеми, пов'язані з виявленням пошкоджень, виявленням шахрайства, виявленням аномального стану здоров'я тощо. Існує три види аномалій, які називаються точковими аномаліями, контекстуальними аномаліями та колективними аномаліями.

Алгоритми виявлення аномалій даних низької розмірності не підходять для даних великої розмірності. Загалом, викидом або аномалію можна знайти за допомогою алгоритмів на основі відстані або щільності. Ці алгоритми вимірюють відстань між екземплярами даних і виходячи з припущення, що аномальна точка даних буде віддалена від інших точок даних, де виявлені

аномалії. Однак у випадку високої розмірності дані стають розрідженими, і всі точки даних виглядають нормально. На основі даних алгоритми можна поділити на контрольовані, напівкеровані та некеровані. Якщо відомі мітки даних як для нормальних, так і для аномальних, вони класифікуються як контрольовані. Якщо відома лише мітка даних нормальної, її називають напівкерованими алгоритмами. Якщо мітки даних як нормальних, так і аномальних невідомі, це неконтрольований алгоритм. Багато програм реального світу не містять міток даних. Розмітка даних вручну є дорогим завданням. Ці проблеми викликають потребу в дослідженнях у цій галузі. Наша головна мета — дослідити та запропонувати ефективну структуру для неконтрольованого виявлення аномалій для даних великої розмірності.

Незважаючи на значний прогрес у цій галузі, оцінка пози залишається складним і досі значною мірою невирішеним завданням. Було досягнуто прогресу в оцінці конфігурацій переважно незакритих та ізольованих суб'єктів. Відкриті проблеми включають спілкування з кількома потенційно взаємодіючими людьми (наприклад, [7]) і терпимість до несподіваних оклюзій. Майбутні дослідження також, ймовірно, розширять типи поз і умов зображення, з якими можуть працювати поточні алгоритми.

1.2 Класифікація методів виявлення даних поз людини

Оцінка пози людини зазвичай формується ймовірнісним шляхом врахування неоднозначностей, які можуть існувати у висновку (хоча є помітні винятки, наприклад, [11]). У таких випадках нас цікавить оцінка апостеріорного розподілу $p(x|z)$, де x — поза тіла, а z — набір ознак, отриманий із зображення. Основні варіанти моделювання, які впливають на висновок:

- Представлення пози – x;
- Характер і кодування ознак зображення – z;
- Система висновків, необхідна для оцінки апостеріорного розподілу – $p(x|z)$.

Далі розглядаються основні напрямки досліджень оцінки пози щодо цих варіантів моделювання. Варто зазначити, що ці три варіанти моделювання не завжди є незалежними. Наприклад, деякі рамки висновків спеціально розроблені для використання заданого представлення пози.

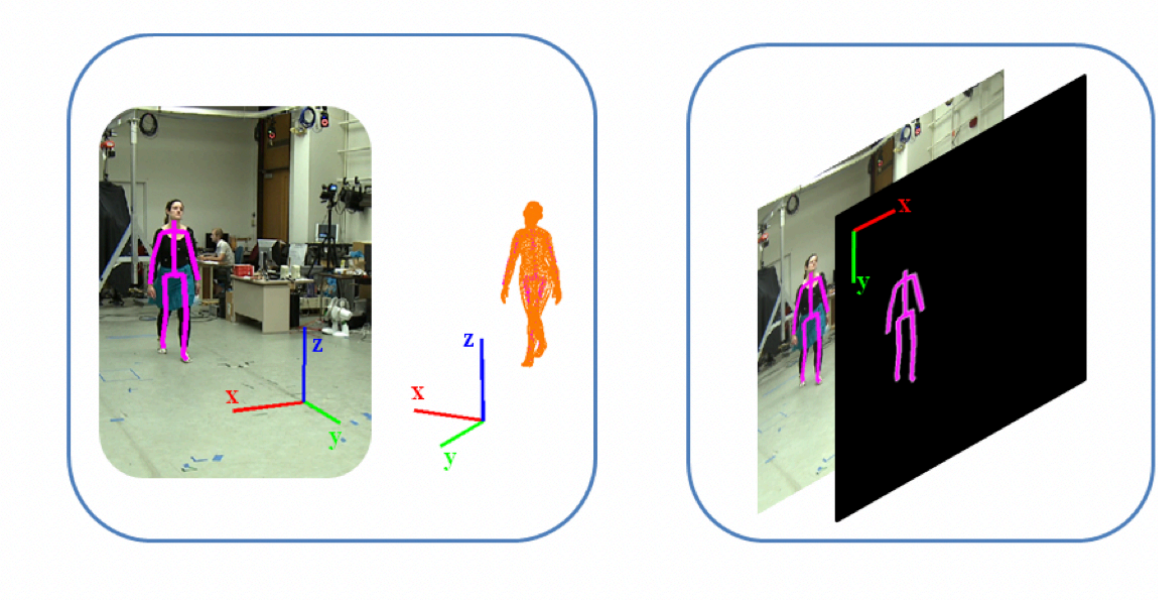

Рисунок 1.1 Відображення скелета: ілюстрація 3d та 2d кінематичного зображення скелета дерева зліва та справа відповідно.

Конфігурацію людського тіла можна представити різними способами. Найбільш пряме і поширене уявлення отримують шляхом параметризації тіла як кінематичного дерева (див. рисунок 2.1), $x = \{\tau, \theta_\tau, \theta_1, \theta_2, ..., \theta_N\}$, де поза кодується за допомогою положення кореневий сегмент (щоб кінематичне дерево було якомога коротшим, таз зазвичай використовується як кореневий сегмент), τ, орієнтація кореневого сегмента у світі, θ_τ, і набір відносних кутів суглоба, $\{\theta i\}Ni = 1$, які представляють орієнтацію частин тіла відносно їхніх батьків уздовж дерева (наприклад, орієнтація стегна відносно тазу, гомілки відносно стегна тощо).

Кінематичне представлення дерева можна отримати для 2d, 2,5d і 3d моделей тіла. У 3d $\tau \in R^3$ і $\theta_\tau \in SO(3)$; $\theta i \in SO(3)$ для сферичних суглобів (наприклад, шиї), $\theta_i \in R^2$ для сідловидних суглобів (наприклад, зап'ястя) і $\theta i \in R1$ для шарнірних суглобів (наприклад, коліна) і представляє позу тіла у реальному вимірі. У 2d $\tau \in R^2$ і $\theta_\tau \in R^1$; $\theta_i \in R^1$ відповідає позі картонної людини в площині зображення.

2.5d уявлення є найменш поширеними і є розширеннями 2d представлення таким чином, що поза, x, доповнюється (як правило, дискретними) змінними, що кодують відносну глибину (шаровість) частин тіла відносно один одного в 2d картонній моделі. У всіх випадках, будь то у 2d чи 3d, це подання призводить до високорозмірного вектора пози x, у R^{30} – R^{70}, залежно від вірності та точності параметризації скелета та суглобів.

Крім того, можна задати позу тіла за 2d або 3d розташуванням основних суглобів [6]. Наприклад, $x = \{p_1,p_2,...,p_N\}$, де p_i — спільне місце у світі, $p_i \in R^3$, або на зображенні, pi ∈ R2. Однак це останнє представлення менш поширене, оскільки воно не є інваріантним до морфології (довжини сегментів тіла) даної особини.

Типовою альтернативою моделям кінематичного дерева є моделювання тіла як набору частин, $x = \{x1, x2, \ldots, xM\}$, кожна зі своїм власним положенням і орієнтацією в просторі, $xi = \{\tau i, \theta i\}$, які пов'язані набором статистичних або фізичних обмежень, які забезпечують узгодженість скелета (а іноді й зображення). Оскільки параметризація на основі частин є надлишковою, вона призводить до ще більшого представлення. Однак це робиться таким чином, що робить його ефективним висновком про позу, як буде розглянуто в наступному розділі. Методи, які використовують таку параметризацію, часто називають частково-базованими. Як і в моделях кінематичного дерева, частини можуть бути визначені у 2d або в 3d [10], причому 2d параметризації є значно більш поширеними. У 2d представлення кожної частини часто доповнюється додатковою змінною s_i, яка враховує рівномірне масштабування частини тіла на зображенні, тобто $xi = \{\tau i, \theta i, si\}$ з $\tau i \in R2, \theta i \in R1$ і $si \in R1$.

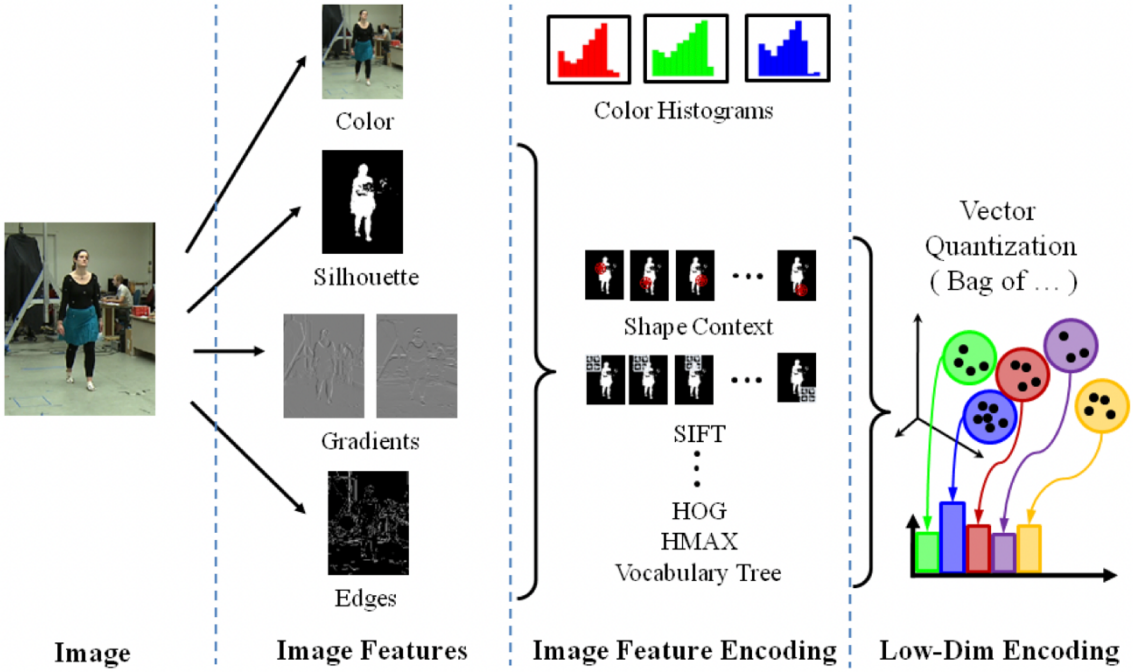

Рисунок 1.2 Особливості зображення: Ілюстрація загальних ознак зображення та методів кодування, які використовуються в літературі.

Ефективність будь-якого підходу до оцінки пози значною мірою залежить від спостережень або особливостей зображення, які вибираються для представлення помітних частин зображення щодо пози людини. Пов'язане і не менш важливе питання - це один із способів кодування цих функцій. На додаток до використання різних кодувань, деякі підходи пропонують зменшити розмірність результуючих векторів ознак. Ці більш грубі уявлення спрощують узгодження ознак, але за рахунок втрати просторової структури зображення. Загальні особливості та методи кодування показані на малюнку 2.2.

Протягом багатьох років різні автори запропонували багато функцій. До найпоширеніших ознак належать: силуети зображень [1] для ефективного відокремлення людини від фону в статичних сценах; колір [16], для моделювання незакритої шкіри або одягу; ребра [6], для моделювання зовнішніх і внутрішніх контурів тіла; і градієнти [5], для моделювання текстури над частинами тіла. Менш поширені ознаки включають затінення та фокусування [14]. Щоб зменшити розмірність і підвищити стійкість до шуму, ці необроблені функції часто інкапсулюються в дескрипторах зображень, таких як контекст форми [6],

SIFT [7] та гістограма орієнтованих градієнтів [5]. В якості альтернативи можна використовувати ієрархічні багаторівневі кодування зображень, такі як HMAX [12], просторові піраміди та словникові дерева. Ефективність різних типів ознак щодо оцінки пози досліджувалася в контексті кількох архітектур висновків.

1.3 Огляд існуючих методичних підходів виявлення аномалій в даних поз людини

1.3.1 Регресійні моделі оцінки пози людини

Висновок (регресійні моделі): характеристику апостеріорного розподілу, $p(x|z)$, можна зробити кількома способами. Можливо, найбільш інтуїтивно зрозумілим способом є визначення параметричної [1] або непараметричної [15] форми для умовного розподілу $p(x|z)$ і вивчення параметрів цього розподілу з набір навчальних зразків. Цей клас моделей більш відомий як дискримінаційні методи, і було показано, що вони дуже ефективні для оцінки пози. Такі методи безпосередньо вивчають $p(x|z)$ з міченого набору поз і відповідних зображень, $D = \{(xi, zi)\}Ni = 1$, які можна створити штучно за допомогою програмних пакетів комп'ютерної графіки (наприклад, Poser). Висновок приймає форму ймовірнісної регресії. Після вивчення функції регресії під час тестування зазвичай використовується підхід до вікна сканування для виявлення частини зображення (обмежувальної рамки), де перебуває людина; $p(x|z)$ потім використовується для характеристики конфігурації особи в цьому цільовому вікні.

Найпростішим методом у цій категорії є метод лінійної регресії [1], де конфігурація тіла, x, вважається лінійною комбінацією ознак зображення, z, з адитивним Гаусовським шумом:

$$x = A[z - \mu_z] + \mu_x + v; \qquad v \sim \mathcal{N}(0, \Sigma);$$

$\mu_x = \frac{1}{N}\sum_{i=1}^{N} x_i$ та $\mu_z = \frac{1}{N}\sum_{i=1}^{N} z_i$ – це середні, обчислені протягом навчання зразки для центрування даних. Як варіант, це можна записати так:

$$p(x|z) = \mathcal{N}A[z - \mu_z] + \mu_x, \Sigma). \qquad (1.1)$$

Коефіцієнти регресії A можна легко дізнатися з парних навчальних вибірок $\mathcal{D} = \{(x_i, z_i)\}_{i=1}^{N}$, використовуючи формулювання найменших квадратів.

Параметричні проти непараметричних методів. Параметричні дискримінаційні методи [6,12] є привабливими, оскільки представлення моделі фіксоване щодо розміру навчального набору даних D. Однак прості параметричні моделі, такі як лінійна регресія [1] або Relevance Vector Machine [1], не в змозі працювати зі складними нелінійними зв'язками між ознаками зображення та позами. Непараметричні методи, такі як регресія найближчого сусіда [8] або регресія ядра [8], здатні моделювати довільні складні відносини між вхідними ознаками та вихідними позами. Недоліком цих непараметричних методів є те, що і модель, і складність висновку є функціями розміру навчального набору. Наприклад, у регресії ядра (Kernel Regression):

$$p(x|z) = \sum_{i=1}^{N} K_x(x, x_i) \frac{K_z(z,z_i)}{\sum_{k=1}^{N} K_z(z,z_k)} \quad (1.2)$$

де $K_x(\cdot,\cdot)$ і $K_z(\cdot,\cdot)$ — функції ядра, що вимірюють подібність аргументів (наприклад, ядра Гаусса), складність висновку O(N) (де N — розмір навчального набору даних). Більш складні непараметричні методи, такі як моделі латентної змінної гауссового процесу (GPLVM), можуть мати ще більшу складність; GPLVM мають $O(N^3)$ навчання та $O(N^2)$ складність висновку. На практиці непараметричні методи, як правило, працюють краще, але повільніше.

Робота з неоднозначностями: якщо припустити, що p(x|z) є унімодальним [1], умовне очікування можна використовувати для характеристики правдоподібної конфігурації людини на зображенні з урахуванням вивченої моделі. Наприклад, для лінійної регресії в рівнянні (1), $E[x|z] = A[z - \mu_z] + \mu_x$; для регресії ядра в рівнянні (3):

$$E[x|z] = \sum_{i=1}^{N} x_i \frac{K_z(z,z_i)}{\sum_{k=1}^{N} K_z(z,z_k)}. \quad (1.3)$$

На практиці, однак, більшість функцій у стандартних умовах зображення є неоднозначними, що призводить до мультимодальних розподілів. Неоднозначність природним чином виникає в проекціях зображень, де кілька поз можуть призвести до схожих, якщо не ідентичних, рис зображення (наприклад,

поза, звернена вперед і ззаду, дають майже ідентичні риси силуету). Щоб пояснити ці неоднозначності, параметричні моделі сумішей були введені у вигляді суміші експертів [6,12]. Непараметричні альтернативи, такі як моделі латентної змінної локального процесу Гаусса (LGPLVM) [5], об'єднують дані в опуклі локальні набори і роблять передбачення на кожному кластері або шукають помітні режими в p(x|z).) [15].

Навчання: отримати великі набори даних, необхідні для вивчення дискримінаційних моделей, які можуть узагальнювати рухи та умови зображення, є складним завданням. Синтетичні набори даних часто не демонструють характеристик зображення, наявних у реальних зображеннях, а реальні повністю позначені набори даних є мізерними. Крім того, навіть якщо можна було б отримати великі набори даних, навчання з величезних обсягів даних не є тривіальним завданням [6]. Для вирішення цієї проблеми було запропоновано два рішення: навчання з малих наборів даних шляхом виявлення проміжного низьковимірного латентного простору для регуляризації [15] і навчання в умовах напівнагляду, де відносно невеликий набір парних вибірок супроводжується великою кількістю немаркованих даних [12].

Обмеження: незважаючи на популярність і багато успіхів, дискримінаційні методи мають обмеження. По-перше, вони здатні відновити лише відносну тривимірну конфігурацію тіла, а не його положення в тривимірному просторі. Причина цього практична, оскільки міркування про положення в тривимірному просторі вимагало б надмірно великих наборів навчальних даних, які охоплюють весь тривимірний об’єм простору, видимого з камери. По-друге, їх продуктивність має тенденцію погіршуватися, оскільки розподіл тестових і навчальних даних починає розходитися; іншими словами, узагальнення залишається одним із ключових питань. Нарешті, ефективне вивчення дискримінаційних моделей із великих наборів даних, які охоплюють широкий спектр реалістичних видів діяльності та пози, залишається складним завданням.

1.3.2 Генеративна модель оцінки пози людини

В якості альтернативи можна застосувати генеративний підхід і виразити бажаний апостеріор, p(x|z), як добуток ймовірності та апріорного:

$$p(x|z) \propto p(z|x)\ p(x). \qquad (1.4)$$

Охарактеризувати цей задній розподіл високого виміру зазвичай важко; отже, більшість підходів покладаються на апостеріорні (MAP) рішення, які шукають найбільш імовірні конфігурації, які є як типовими (мають високу попередню ймовірність), так і можуть добре пояснити дані зображення (мають високу ймовірність):

$$x_{MAP} = \arg\max p(x|z)\,. \qquad (1.5)$$

Однак пошук таких конфігурацій у високовимірному (40+) артикуляційному просторі є дуже складним, і більшість підходів часто застряють у локальному оптимумі. Глобальні ієрархічні методи пошуку, такі як фільтр відпалених частинок [10], показали деякі багатообіцяючі результати для простих скелетних конфігурацій, коли тіло переважно вертикальне, і коли доступні спостереження з кількох камер. Для більш загальних артикуляцій і монокулярних спостережень, які часто є в центрі уваги алгоритмів оцінки пози, цей клас методів на сьогоднішній день не був дуже успішним.

1.3.3 Моделі оцінки пози людини на основі часток

Для боротьби зі складністю генеративних моделей були введені моделі на основі часток. Ці методи виникли в спільноті розпізнавання об’єктів з формулюванням Фішлера і Ельшлагера (1973) і припускають, що тіло можна представити як сукупність частин, які з’єднані обмеженнями, накладеними суглобами всередині скелетної структури (а іноді і структурою скелета). обмеження зображення, накладені проекціями на площину зображення, які враховують оклюзії). Ця формулювання зменшує складність висновку, оскільки ймовірні місця розташування частин тіла можна шукати незалежно, лише враховуючи сусідні частини тіла, які їх обмежують, що значно скорочує загальний простір пошуку.

Серед найбільш ранніх успіхів у цьому напрямку досліджень є робота Лі та Коена [13]. Їхній підхід був зосереджений на отриманні карт пропозицій щодо

розташування окремих суглобів на зображенні. Ці карти пропозицій були отримані на основі низки ознак, які були щільно обчислені над зображенням. Наприклад, для отримання гіпотез щодо розташування голови використовували розпізнавання обличчя; узгодження контурів голови та плеча, отримане з використанням деформованої контурної моделі та градієнтного спуску, було використано як доказ розташування плечового суглоба; еліптичні ділянки шкіри, отримані за допомогою сегментації за кольором шкіри, використовували для визначення розташування гомілок і гомілок. Крім того, спостереження другого похідного (хребта) використовувалися як докази інших кінцівок тіла. Враховуючи пропозиції щодо різних з'єднань, зважені на впевненість відповідних детекторів, для відновлення 3D-конфігурацій скелета використовувався підхід Монте-Карло з ланцюгом Маркова (MCMC), керований даними. Цей висновок спирався на пряму зворотну кінематику (IK), отриману з 2d карт пропозиції. Для подальшого покращення результатів був також введений процес пропозиції кінематичного стрибка. Процес пропозиції кінематичного стрибка включає перевертання частини тіла або набору частин (тобто голови, кисті або всієї руки) у напрямку глибини навколо його шарнірного суглоба.

Інші підходи, засновані на частинах, намагаються об'єднати області зображення в частини тіла і послідовно побудувати ці частини в тіло. Яскраві приклади таких методів представлені Mori et al. [11] та Ren et al. [12]. У [11] суперпікселі були вперше зібрані в частини тіла на основі оцінки сигналів зображення низького рівня, включаючи контур, форму, затінення та фокус. Потім пропозиції частин були обрізані та зібрані разом, використовуючи довжину, примикання частин тіла та симетрію одягу. Подібний підхід було використано в [12], але замість складання суперпікселів використовувалися сегменти. Паралельні лінії були зібрані в частини-кандидати за допомогою набору заздалегідь визначених правил, а частини-кандидати, у свою чергу, були зібрані в тіло з набором обмежень щодо з'єднання, масштабу, зовнішнього вигляду та орієнтації. На відміну від [14], пошук найбільш вірогідних конфігурацій тіла був

сформульований як рішення задачі цілочисельного квадратичного програмування (IQP).

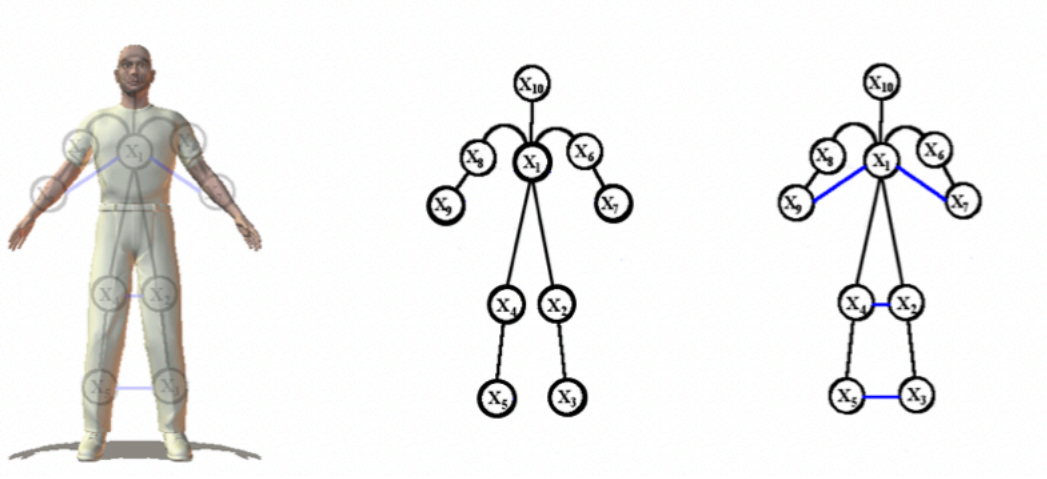

Рисунок 2.3 Модель графічних структур

На ілюстрації зображено 10-частину деревоподібну модель графічних структур (в середині) і модель графічних структур без деревоподібної структури (петлі) (праворуч). У моделі без деревоподібної структури додаткові обмеження, що кодують оклюзії, ілюстровані синім кольором.

Найбільш традиційним і успішним підходом, однак, є представлення тіла за допомогою випадкового поля Маркова (MRF) з частинами тіла, що відповідають вузлам, і обмеженнями між частинами, закодованими потенційними функціями, які враховують фізичні та статистичні залежності (див. малюнок 2.3). Формально апостеріор, $p(x|z)$, можна виразити як:

$$p(x|z) \propto p(z|x)p(x) = p(z|\{x_1, x_2, \dots, x_M\})p(\{x_1, x_2, \dots, x_M\}) \approx$$

$$\prod_{i=1}^{M} p(z|x_i)p(x_1) \qquad \prod_{(i,j)\in E} p(x_i, x_j). \qquad (1.6)$$

У цьому випадку оцінка пози приймає форму висновку в загальній мережі MRF. Висновок можна ефективно вирішити за допомогою алгоритмів передачі повідомлень, таких як поширення віри (BP). BP складається з двох окремих фаз: (1) виконується набір ітерацій для передачі повідомлень для поширення узгоджених оцінок частин у межах графіка, і (1.2) граничні задні розподіли оцінюються для кожної частини тіла [2,8]. Типова формулювання розглядає конфігурацію тіла в площині 2d зображення і передбачає дискретизацію пози для кожної окремої частини, наприклад, $x_i = \{\tau_i, \theta_i, s_i\}$ де $\tau_i \in R^2$ — місце розташування,

а $\theta_i \in R^1$ і $s_i \in R^1$ — орієнтація та масштаб частини і (представленої у вигляді прямокутної ділянки) у площині зображення. Як результат, висновок здійснюється за набором конфігурацій дискретних частин $l_i \in Z$ (для частини і), де Z — перерахування поз для частини зображення (l_i — дискретна версія x_i). З додатковим припущенням про парні потенціали, які враховують кінематичні обмеження, модель формує дерево-структурований графік, відомий як модель деревоподібних зображувальних структур (PS). Також можливий наближений висновок з безперервними змінними.

Висновок у моделі PS з деревоподібною структурою спочатку відбувається шляхом надсилання рекурсивно визначених повідомлень у вигляді:

$$m_{i\to j}(l_j) = \sum_{l_i} p(l_i, l_j)p(z|l_i)\ ,\ \prod_{k\in A(i)\setminus j} m_{k\to i}(l_i), \qquad (1.7)$$

де $m_i\to j$ – повідомлення від частини і до частини j, причому $p(l_i, l_j)$ вимірює сумісність поз для двох частин, а $p(z|l_i)$ – ймовірність, а A(i)\j – набір частин у графі, суміжних з і, крім j. Сумісність, $p(l_i,l_j)$, часто вимірюється фізичною консистенцією двох частин у з'єднанні або їх статистичною (наприклад, кутовою) спільною зустрічністю один щодо одного. У деревоподібному графіку PS ці повідомлення надсилаються від крайніх кінцівок всередину, а потім назад назовні.

Після завершення всіх оновлень повідомлень граничні задні частини для всіх частин можна оцінити як:

$$p(l_i|z) \propto p(z|l_i)\ ,\ \prod_{j\in A(i)} m_{j\to i}(l_i). \qquad (1.8)$$

Аналогічно, найбільш ймовірну конфігурацію можна отримати як оцінку MAP:

$$l_{i,MAP} = \arg\max_{l_i} p(l_i|z). \qquad (1.9)$$

Однією з ключових переваг парадигми зображувальних структур (PS) є її простота та ефективність. У PS точний висновок можливий за час, лінійний до кількості дискретних конфігурацій, які може прийняти дана частина. Завдяки цій властивості останні реалізації [2] можуть обробляти конфігурації частин із щільністю пікселів, що призводить до мільйонів потенційних дискретних станів для кожної частини тіла. Лінійна складність походить із спостереження, що

загалом складний негауссовий аператор над сусідніми частинами, $p(xi, xj)$, може бути виражений як гауссовий попередній над перетвореними місцями, що відповідають стикам, головним чином $p(xi, xj) = N(Tij(xi); Tji(xj), \Sigma ij)$. Це робиться шляхом визначення перетворення $Tij(xi)$, яке відображає загальне з'єднання між частинами i і j, визначене в системі координат частини i, на його розташування в просторі зображення. Аналогічно, $Tji(xj)$ визначає перетворення від того самого загального з'єднання, визначеного в системі координат j, до розташування в площині зображення. Це перетворення дозволяє у висновку використовувати ефективне рішення, яке включає згортку (докладніше див. [8]).

Продуктивність. Нещодавно було показано, що ефективність моделі PS тісно пов'язана з якістю ймовірності частини [2]. Дискримінаційно навчені моделі [16] та більш складні моделі зовнішнього вигляду [2], як правило, перевершують моделі, визначені вручну [8]. Методи, які вивчають каскади імовірності, що відповідають кращим і кращим функціям, налаштованим на конкретне зображення, також були досліджені як для високої швидкості, так і для продуктивності [16]. Найновіша дискримінаційна формулювання моделі PS дозволяє спільно вивчати зовнішній вигляд деталей і структуру моделі [26] за допомогою структурної машини опорних векторів (SVM).

Швидкість. Каскади детекторів деталей служать не тільки для покращення продуктивності, але й для прискорення висновку (наприклад, [23]). Швидкі ймовірності можуть бути використані для вилучення великих частин простору пошуку перед застосуванням більш складних і дорогих у обчисленні моделей правдоподібності. Інші підходи до прискорення продуктивності включають методи, керовані даними (наприклад, поширення віри на основі даних). Ці методи спочатку шукають частини на зображенні, а потім збирають невеликий набір частин-кандидатів у тіло (схоже до методів [14]). Проблема з такими підходами полягає в тому, що будь-які закриті частини взагалі пропускаються, оскільки вони не можуть бути виявлені початковими детекторами частин. Висновок також можна прискорити за допомогою методів уточнення прогресивного пошуку [9]. Наприклад, деякі методи використовують детектори

верхньої частини тіла, щоб обмежити пошук перспективними частинами зображення замість пошуку всього зображення.

Розширення без деревоподібної структури. Хоча моделі PS з деревоподібною структурою є обчислювально ефективними та точними, їх, як правило, недостатньо для моделювання всіх необхідних обмежень, накладених тілом. Більш складні відносини між частинами, які виходять за межі сфери цих моделей, включають обмеження непроникнення та обмеження оклюзії [20]. Включення таких зв'язків у модель додає петлі, що відповідають далеким залежностям між частинами тіла. Ці цикли ускладнюють висновки, оскільки: (1) не можна ефективно знайти оптимальні рішення (алгоритми передачі повідомлень, такі як BP, не гарантують збіжність у циклічних графах) і (2) навіть наближений висновок, як правило, є дорогим з точки зору обчислень. Незважаючи на ці проблеми, стверджується, що додавання таких обмежень необхідно для підвищення продуктивності [4]. Щоб полегшити деякі складності висновку з цими недеревоподібними моделями, було введено ряд конкуруючих методів. Ранні спроби використовували методи вибірки з задньої частини з деревоподібною структурою як пропозиції для оцінки більш складної недеревоподібної моделі [7,8].

1.4 Актуальність запропонованого підходу в оцінюванні аномалій в даних пози людини

На даний момент не існує чітких методологічних підходів для виявлення аномалій в оцінці пози людини. Існуючі підходи більше направлені на вдосконалення виявлення пози людини.

Дана кваліфікаційна робота має на меті зробити основний акцент не на навчанні моделі розпізнавати позу людини, а виявленні аномалій в даних пози та правильній оцінці цієї пози.

Таким чином буде створена модель аномалій пози людини. Зробити оцінку пози для кількох осіб складніше, ніж у випадку однієї людини, оскільки місце розташування та кількість людей на зображенні невідомі. Для цього буде

застосовано підхід оцінки пози людини знизу-вгору. Він полягає у виявленні всіх частин зображення (тобто частин кожної людини), а потім асоціювання/групування частин, що належать окремим особам.

Підхід, який буде використаний в даній кваліфікаційній роботі, оснований на глибинному навчанні. Існує декілька можливих підходів, які можна застосувати для вирішення даної задачі. Розглянемо декілька з них.

- OpenPose;
- DeepCut;
- Mask R-CNN;
- HR Net.

Архітектура OpenPose глибинного навчання спочатку виявляє частини (ключові точки), що належать кожній людині на зображенні, а потім призначає частини окремим особам. Нижче показано архітектуру даної моделі.

Мережа спочатку витягує ознаки із зображення за допомогою перших кількох шарів (VGG-19 у наведеній вище блок-схемі). Потім об’єкти подають у дві паралельні гілки згорткових шарів. Перша гілка передбачає набір з 18 карт довіри, кожна з яких представляє певну частину скелета пози людини. Друга гілка передбачає набір з 38 полів спорідненості частин (PAF), які представляють ступінь асоціації між частинами.

Послідовні етапи використовуються для уточнення прогнозів, зроблених кожною галуззю. Використовуючи карти впевненості частин, дводольні графіки формуються між парами частин (як показано на зображенні вище). Використовуючи значення PAF[15], слабші ланки в дводольних графіках обрізаються. За допомогою наведених вище кроків можна оцінити скелети пози людини та призначити кожній людині на зображенні.

Далі, для виявлення аномалій в моделі буде використана оцінка щільності ядра — це процес оцінки невідомої функції щільності ймовірності за допомогою функції ядра. Оцінка щільності ядра (KDE) для моделювання розподілу амплітуди, кута та послідовності виявлення аномалії для кожного зацікавленого з’єднання, як зазначено в рівняннях:

$$\widehat{p_n}(x) = \frac{1}{nh}\sum_{i=1}^{n} K(\frac{X_i - x}{h}), \quad (1.10)$$

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}} . \quad (1.11)$$

K(x) є функцією ядра, у нашому випадку є гаусівською. h - це пропускна здатність. X_i — точка даних, n - номер точки даних. Використовуючи KDE, ми можемо вибирати дані про нескінченні аномалії щодо амплітуди, кута та числа послідовних кадрів із реальних даних про аномалії, які ми маємо.

DeepCut був запропонований Леонідом Піщуліним та ін. у 2016 році з метою спільного вирішення завдань виявлення та оцінювання пози одночасно.

Це підхід знизу вгору до оцінки пози людини. Ідея полягала в тому, щоб виявити всі можливі частини тіла на даному зображенні, а потім позначити їх, такі як голова, руки, ноги тощо, з подальшим процесом відокремлення частин тіла, що належать кожній людині.

Мережа використовує моделювання інтегрального лінійного програмування (ILP) для неявного групування всіх виявлених ключових точок у заданому вхідному наборі даних, щоб результат нагадував скелетне представлення людини.

Mask R-CNN — дуже популярний алгоритм для сегментації об’єктів на зображенні.

Модель має можливість одночасно локалізувати та класифікувати об’єкти, створюючи обмежувальну рамку навколо об’єкта, а також створюючи маску сегментації.

Базову архітектуру можна легко розширити для завдань оцінки пози людини. Швидкий R-CNN використовує CNN для вилучення функцій і представлення з заданого введення. Вилучені ознаки потім використовуються, щоб запропонувати, де може бути присутнім об’єкт через мережу регіональних пропозицій (RPN).

Оскільки обмежувальна рамка може мати різні розміри, як на зображенні вище, шар під назвою RoIAlign використовується для нормалізації вилучених об’єктів, щоб усі вони були однакового розміру.

Вилучені ознаки передаються в паралельні гілки мережі, щоб уточнити запропоновану область інтересу (RoI), щоб створити обмежувальні рамки та маски сегментації.

HRNet представляє собою мережу високої роздільної здатності, що відноситься до високої роздільної здатності зображень, що обробляються. «Надійні зображення з високою роздільною здатністю відіграють важливу роль у проблемах позначення пікселів і регіонів, наприклад, семантична сегментація, оцінка пози людини, виявлення орієнтирів обличчя та виявлення об'єктів». [1] Зі збільшенням кількості пікселів і проблемами, пов'язаними з відео, висока роздільна здатність може відігравати все більшу роль у майбутньому.

Мережа, яка стоїть за HRNet, називається HRNetV1, і вона «підтримує представлення з високою роздільною здатністю, паралельно з'єднуючи згортки високої та низької роздільної здатності, де відбуваються повторювані багатомасштабні злиття паралельних згорток».[1]

Узагальнення інформації щодо підходів можна побачити в порівняльній таблиці 1.1.

Таблиця 1.1

Назва підходу	Основа архітектури	Спосіб оцінювання	AP, %
OpenPose	CNN	Підхід знизу-вгору	69.6
DeepCut	ILP	Підхід знизу-вгору	61.8
Mask R-CNN	CNN	Підхід зверху-вниз	63.1
HRNet	CNN	Підхід зверху-вниз	71.2

Як видно з порівняльної таблиці дані методологічні підходи дуже схожі між собою. Архітектура даних методів заснована на згортковій нейронній мережі.

Два підходи використовують спосіб зверху-вниз: локалізувати людей на зображенні або відео, потім оцінити частини, а потім оцінити позу. Інші два - знизу вгору: оцінити частини людського тіла на зображенні, а потім оцінити позу.

Стандартна метрика оцінки заснована на схожості об'єктної ключової точки (OKS):

$$OKS = \frac{\sum_i \exp\left(-\frac{d_i^2}{2s^2k_i^2}\right)\delta(v_i>0)}{\sum_i \delta(v_i>0)} \quad (1.12)$$

Тут d_i – евклідова відстань між виявленою ключовою точкою та відповідною основною істиною, v_i – маркер видимості основної істини, s – масштабом об'єкта, а k_i – константою основної точки, яка контролює спад.

Можна побачити з таблиці 1.2, що найкращі показники спостерігаються в OpenPose та HRNet.

Таблиця 1.2

Назва підходу	Вхідний розмір зображення	OKS
OpenPose	353×257	73.7
DeepCut	356×256	69.4
Mask R-CNN	384×288	71.4
HRNet	384×288	79.0

Для оцінки якості роботи різних підходів було порівняно показник точності та повноти – AP. З даної таблиці можна побачити, що найкращі показники мають OpenPose та HRNet. Ці два методологічні підходи базуються на різних способах оцінювання поз, в OpenPose спочатку визначаються люди присутні на зображенні, а потім оцінюються їхні пози, а в HRNet спочатку оцінюються частини тіла на зображенні, а потім визначається поза людини. Доцільно перевірити два методологічних підходи, щоб визначити який з методів підходить краще для заданого набору даних. Тому дані підходи і було обрано для подальшої роботи. Буде перевірено, який із способів оцінювання працює краще для вирішення задачі виявлення аномалій.

РОЗДІЛ 2 ПРОЕКТУВАННЯ ТЕХНОЛОГІЇ ОЦІНЮВАННЯ АНОМАЛІЙ В ДАНИХ ПОЗИ ЛЮДИНИ

2.1Постановка задачі моделювання оцінювання аномалій в даних пози людини

Класифікація – це процес категоризації даного набору даних на класи. Її можна виконувати як для структурованих, так і для неструктурованих даних. Процес починається з прогнозування класу заданих точок даних. Класи часто називають цільовими, мітками або категоріями.

Прогностичне моделювання класифікації є завданням апроксимації функції відображення від вхідних змінних до дискретних вихідних змінних. Основна мета – визначити, до якого класу/категорії потраплять нові дані.

Саме тому задача виявлення аномалій в даних пози людини зводиться до задачі класифікації [16].

Маємо множину поз $S = \{A_1, A_2, \dots, A_n, N_1, N_2, \dots, N_k\}$, де S – це множина визначених даних поз людини на зображеннях. $A, N = \{x_1, x_2, \dots, x_{17}\}$ – це множина точок пози кожної людини визначеної на зображенні.

В даному контексті основною метою розроблюваного програмного забезпечення є визначення до якого класу належить та чи інша поза.

Маємо дві підмножини S_1 та S_2 множини S.

$S_1 = \{A_1, A_2, \dots, A_n\}$ – множина аномальних даних поз людини.

$S_2 = \{N_1, N_2, \dots, N_k\}$ – множина неаномальних даних поз людини.

Для визначення до якої множини (до якого класу) відноситься та чи інша поза буде застосовуватись пошук різниці між очікуваними даними точок пози людини та отриманими.

Нехай, Q це новий образ з множиною точок $\{k_1, k_2, \dots, k_{17}\}$.

Маємо знайти:

$$\min_{\substack{A_i \in S_1 \\ N_i \in S_2}} \{\sum |Q - A_i|, \sum |Q - N_i|\} \quad (2.1)$$

Таким чином, щоб визначити, до якого класу відноситься поза людини – чи є вона аномальною чи ні потрібно визначити мінімальну відстань між отриманими точками та очікуваними [17][18].

Перед тим як приступати до модулювання алгоритму вирішення даної задачі потрібно визначити, що в даному контексті буде сприйматися системою як аномалія, а що ні. Для цього потрібно виявити область визначення аномалії та спосіб її розрахунку.

2.1.1 Область визначення аномалії в даних пози людини

Область визначення аномалії має важливе значення. Дана робота зосереджена на виявлення аномалії в даних пози людини в ситуаціях, які пов’язанні зі здоров’ям та зосереджена на виявлення аномалій при незвичайній поведінці пов’язаній з фізичним станом людини.

Виявлення аномальних подій є складним завданням через його контекстно-залежний характер. Це означає, що подія, яка вважається ненормальною в одному сценарії, може вважатися нормальною в іншому сценарії. У системах автоматичного спостереження виявлення ненормальних подій виконується шляхом моделювання очікуваних шаблонів у заданому наборі даних і пошуку моделей, які не відповідають очікуваній поведінці. Очікувана поведінка моделюється за допомогою звичайних зразків.

Камери, які будуть відповідати за відправку сигналів мають знаходитися в на вулицях міста або в торгово-розважальних центрах. Це має важливе значення, так як інтерпретація результатів часто залежить від локації, звідки були взяті дані.

Тож перейдемо до визначення аномалії в даному контексті роботи, так як саме визначалася аномальна поза в даній задачі.

У цій роботі буде використано SVM одного класу для побудови нормальних моделей. Основними причинами переваги методів однокласної класифікації при виявленні аномальних подій є те, що існує широкий спектр аномальних подій, але є труднощі зі збором зразків цих випадків. З цією метою було запропоновано метод, який адаптує класичну методологію SVM до

проблеми класифікації одного класу. [19] В однокласному SVM спочатку визначається розподіл нормальних даних. Класифікація здійснюється відповідно до наявності або відсутності даних тесту в цьому розподілі. Нехай $x_1, x_2, \dots, x_n$— навчальні приклади, що належать одному класу X і $\Phi: X \to H$. H це простір ознак і Φ — це карта ядра, яка перетворює навчальні вибірки в інший простір. Процес відокремлення звичайних вибірок від інших за допомогою ядра досягається шляхом вирішення наступної задачі квадратичного програмування:

$$\min \frac{1}{2}\|w\|^2 + \frac{1}{vl}\sum_{i=1}^{l}\xi_i - p \qquad (2.2)$$

За умови:

$$\big(w.\Phi(x_i)\big) \geq p - \xi_i, \qquad i = 1,2,\dots,l \quad \xi_i \geq 0 \qquad (2.3)$$

де w — вектор, що визначає гіперплощину, v — параметр регуляризації, l — кількість навчальних вибірок, ξ_i — змінна слабини, ρ — відстань до початку координат у просторі ознак.

Функція рішення визначається як:

$$f(x) = sign(\big(w.\Phi(x)\big) - p) \qquad (2.4)$$

Функція (2.4) даватиме додатне значення для вибірок у навчальній множині.

Тому розробка алгоритму виявлення аномалій має велике значення. Зростає попит на автоматичні методи для аналізу величезної кількості відеоданих відеоспостереження. Цифрова обробка зображень використовує алгоритм для обробки зображення, після обробки ми можемо отримати інформацію, яка відповідає нашим потребам.

2.2 Вибір теоретичних та експериментальних методів дослідження поставлених задач та аналіз адекватності розроблених моделей

В минулому розділі було проведено огляд існуючих підходів до оцінки пози людини та визначено основний підхід для проектування майбутньої системи. Було обрано 2 найефективніші підходи на основі глибинного навчання. Далі, буде детально описано методологічні підходи даних архітектур.

Оскільки оцінка пози є легко застосовною технікою комп’ютерного зору, можна реалізувати оцінку пози, використовуючи існуючі архітектури глибокого навчання. Для дослідження було обрано 2 існуючих архітектури оцінки пози людини:

- OpenPose
- High-Resolution Net (HRNet).

OpenPose — це один з найпопулярніших підходів знизу-вгору для оцінки пози людини багатьох осіб на зображенні. Ця архітектура передбачає оцінку пози кількох осіб у реальному часі. OpenPose має відкритий вихідний код для виявлення кількох осіб у реальному часі з високою точністю виявлення ключових точок тіла, стопи, руки та обличчя. Перевага OpenPose полягає в тому, що це API, який дає користувачам гнучкість вибору вихідних зображень із полів камери, веб-камер та інших, що ще важливіше для вбудованих системних програм (наприклад, інтеграція з камерами та системами відеоспостереження). Він підтримує різні апаратні архітектури, такі як графічні процесори CUDA, графічні процесори OpenCL або пристрої, що працюють лише на ЦП. Полегшена версія достатньо ефективна для додатків Edge з обробкою на пристрої в режимі реального часу з граничними пристроями.

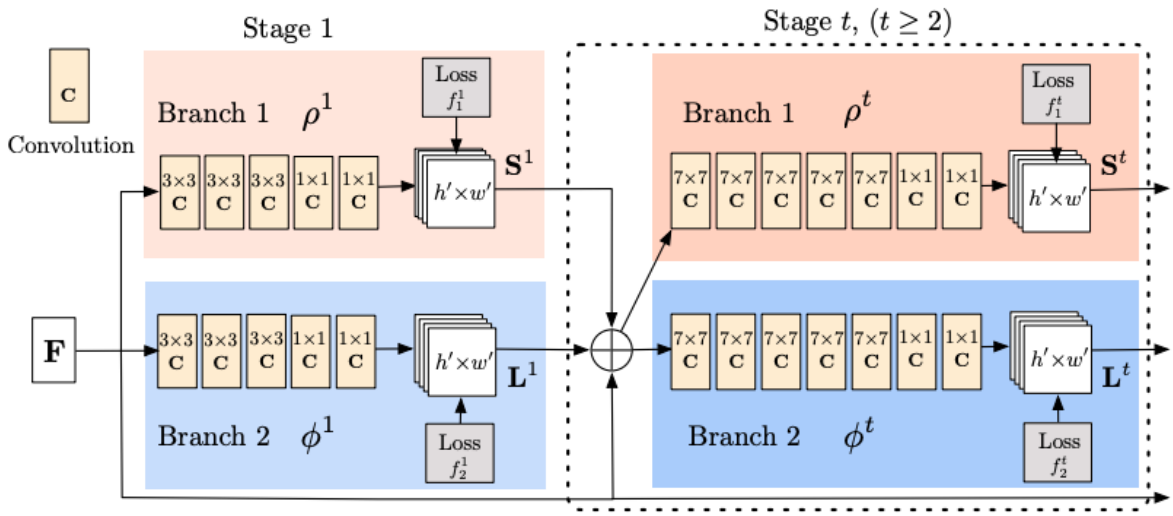

Рисунок 2.1 Блок-схема архітектури OpenPose.

На першому кроці зображення передається через базову мережу CNN для вилучення карт ознак вхідних даних у статті. У цій роботі автори використали перші 10 шарів мережі VGG-19 [21].

Потім карта об’єктів обробляється в багатоступінчатому конвеєрі CNN для створення частини карт довіри і частини поля спорідненості.

На останньому кроці згенеровані вище карти впевненості та поля спорідненості частин обробляються жадібним дводольним алгоритмом відповідності, щоб отримати пози для кожної людини на зображенні [13][27].

Карта впевненості — це двовимірне відображення переконання, що певна частина тіла може бути розташована в будь-якому даному пікселі. Карти впевненості описуються таким рівнянням:

$$S = (S_1, S_2, \dots, S_J), \text{де } S_j \in R^{w*h}, j \in 1 \dots J \quad (2.5)$$

де J – кількість місць розташування частин тіла.

Поле спорідненості — це набір двовимірних векторних полів, які кодують розташування та орієнтацію кінцівок різних людей на зображенні. Він кодує дані у вигляді парних зв’язків між частинами тіла.

$$L = (L_1, L_2, \dots, L_C), \text{де } L_c \in R^{w*h*c}, c \in 1 \dots C \quad (2.6)$$

де J – кількість місць розташування частин тіла.

Для того, щоб мережа навчилася генерувати найкращі набори S і L, застосовуються дві функції втрат наприкінці кожного етапу, по одній на кожній гілці відповідно. У статті використовується стандартна втрата L2 між очікуваними прогнозами, картами реальних даних та полями. Більше того, у даному методі додано певну вагу функціям втрати, щоб вирішити практичну проблему, що деякі набори даних не повністю маркують усіх людей. Функції втрат на певному етапі t задаються таким чином.

$$f_S^t = \sum_{j=1}^{J} \sum W(p) \cdot \left\| S_j^t(p) - S_j^*(p) \right\|_2^2 \quad (2.7)$$

$$f_L^t = \sum_{c=1}^{c} \sum W(p) \cdot \| L_c^t(p) - L_c^*(p) \|_2^2 \quad (2.8)$$

Позначення p представляє розташування одного пікселя на зображенні w x h. Позначення $*$ біля множини S і L означає, що це основна істина. Результатом

$S(p)$ є одновимірний вектор, який складається з оцінки довіри для цієї конкретної частини тіла j у місці зображення p.

Вихід $L(p)$ є двовимірним вектором, який складається з вектора спрямованості для цієї конкретної кінцівки c у місці зображення p. У статті OpenPose J загальна кількість частин тіла дорівнює 17. Крім того, C, загальна кількість «кінцевих» або з'єднань між тілом – 17. $W(p)$ являє собою функцію зважування, як згадувалося раніше. $W(p) = 0$, коли анотація відсутня на зображенні p. Маска використовується, щоб уникнути штраф за справжні позитивні прогнози під час тренування [5][34].

High-Resolution Net (HRNet) — це ще одна нейронна мережа для оцінки пози людини. Дана архітектура часто використовується в задачах обробки зображень, щоб знайти те, що ми знаємо як ключові точки (стики) щодо конкретного об'єкта або людини на зображенні. Однією з переваг цієї архітектури перед іншими архітектурами є те, що більшість існуючих методів узгоджують представлення пози з високою роздільною здатністю від уявлень з низькою роздільною здатністю щодо використання мереж з високою роздільною здатністю. Замість цього упередження нейронна мережа підтримує представлення з високою роздільною здатністю під час оцінки поз.

Підхід до оцінки пози людини, або виявлення ключових точок, має на меті виявити розташування K ключових точок або частин (наприклад, лікоть, зап'ястя тощо) на зображенні I розміром $W \times H \times 3$. Найсучасніші методи трансформують цю проблему до оцінки K теплових карт розміром $W' \times H', \{H_1, H_2, \dots, H_K\}$, де кожна теплова карта H_K вказує впевненість розташування k-ї ключової точки.

Алгоритм базується на дотриманні широко прийнятого методологічного підходу для прогнозування ключових точок людини за допомогою згорткової нейронної мережі, яка складається з основи, що складається з двох стрімких зсувів, що зменшують роздільну здатність. Основна частина виводить карти об'єктів з такою ж роздільною здатністю, що і його вхідні карти об'єктів, а регресор, що оцінює теплові карти, де позиції ключових точок вибираються і

перетворюються - на високу роздільну здатність. Мережа HRNet зображена на малюнку 2.2.

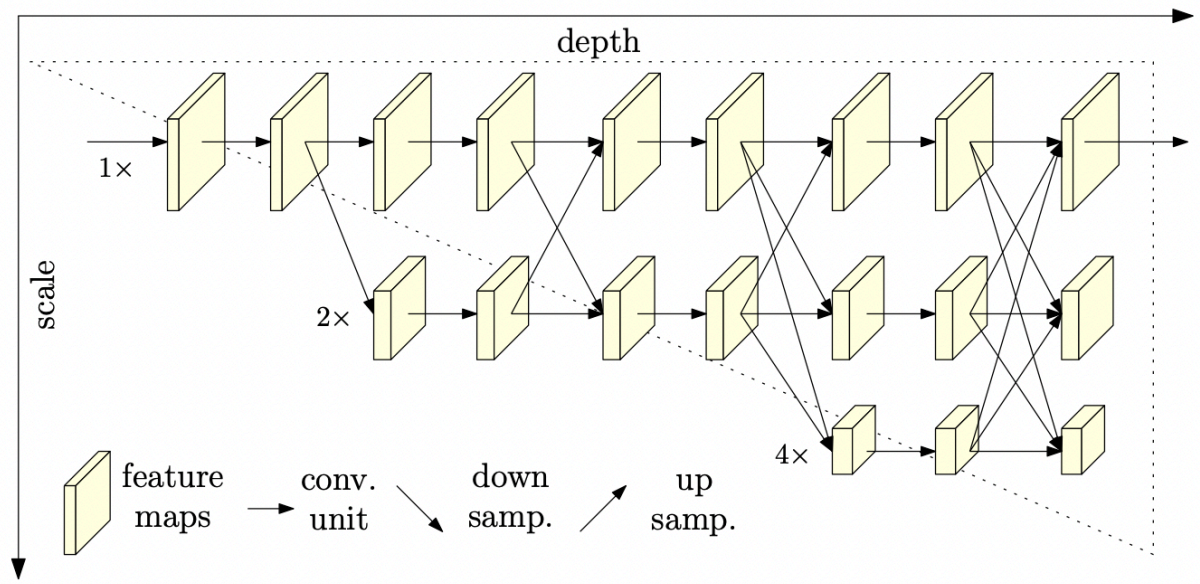

Рисунок 2.2 Ілюстрація архітектури HRNet

Архітектура складається з паралельних підмереж з високою і низькою роздільною здатністю з повторюваним обміном інформацією між підмережами з різною роздільною здатністю (багато масштабне злиття). Горизонтальний та вертикальний напрямки відповідають глибині мережі та масштабу карт об'єктів відповідно.

Існуючі мережі для оцінки пози будуються шляхом послідовного з'єднання підмереж з високою і низькою роздільною здатністю, де кожна підмережа, яка формує етап, складається з послідовності згорток і є шар нижньої вибірки через сусідні підмережі, щоб вдвічі зменшити роздільну здатність.

Нехай $\mathcal{N}_{sr}$ — підмережа на s-му етапі, а r — індекс роздільної здатності (її роздільна здатність дорівнює $\frac{1}{2^{r-1}}$ роздільної здатності першої підмережі). Мережа від високого до низького з S етапи можна позначити так:

$$\mathcal{N}_{11} \to \mathcal{N}_{22} \to \mathcal{N}_{33} \to \mathcal{N}_{44} \qquad (2.9)$$

Ми починаємо з підмережі з високою роздільною здатністю як на першому етапі, поступово додаємо підмережі з високою роздільною здатністю одну за одною, утворюючи нові етапи, і підключаємо підмережі з різною роздільною здатністю паралельно. В результаті роздільна здатність для паралельних підмереж

наступного етапу складається з роздільної здатності попереднього етапу та додаткового нижчого.

На рисунку 2.3 ілюструється блок обміну і представляється формулювання в наступному. Ми скидаємо нижній індекс s і верхній індекс b для зручності обговорення. Вхідними є s карти відповідей, де $\{X_1, X_2, \dots, X_S\}$. Вихідними є s карти відповідей: $\{Y_1, Y_2, \dots, Y_s\}$, роздільна здатність і ширина яких однакові для входу. Кожен вихід є сукупністю вхідних карт $Y_k = \sum_{i=1}^{s} a(X_i, k)$. Блок обміну між етапами має додаткову вихідну карту Y_{s+1}: $Y_{s+1} = a(Y_s, s+1)$ [27].
Функція $a(X_i, k)$ складається з підвищення або зниження дискретизації X_i від роздільної здатності i до роздільної здатності k. Ми використовуємо стрибкові згортки 3×3 для зниження дискретизації. Наприклад, один крок згортки 3×3 з кроком 2 для 2-кратного зниження дискретизації і два послідовних кроки згортки 3×3 з кроком 2 для 4-кратного зниження дискретизації. Для підвищення дискретизації приймається проста вибірка найближчого сусіда після згортки 1×1 для вирівнювання кількості каналів. Якщо $i = k, a(\cdot,\cdot)$ є просто ідентифікаційним зв'язком: $a(X_i, k) = X_i$ [25].

На рисунку 2.3 показано, як блок обміну агрегує інформацію для високої, середньої та низької роздільної здатності зліва направо відповідно.

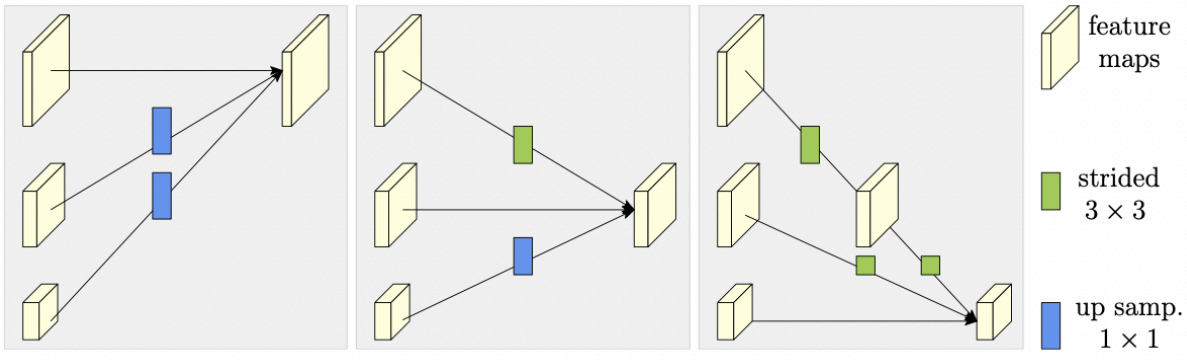

Рисунок 2.3 Ілюстрація блоку обміну

Основний архітектура, тобто HRNet, містить чотири ступені з чотирма паралельними підмережами, роздільна здатність яких поступово зменшується вдвічі і відповідно ширина (кількість каналів) збільшується вдвічі.

В даній роботі буде використано дві архітектури для оцінки пози людини. В ході реалізації буде проведено експеримент для визначення, яку саме архітектуру доцільніше використовувати у даній роботі [29].

2.3 Алгоритм та моделювання технології оцінювання аномалій в даних пози людини

В попередніх дослідженнях, видно, що функція оптичного потоку, яка представляє рух натовпу, часто використовується для визначення аномалій. Таким чином можна виявити нерівності в напрямку руху та швидкості. Як відомо, у таких програмах, як розпізнавання об'єктів, відстеження об'єктів, розпізнавання дій, виявлення ненормальних подій, успішний результат досягається шляхом поєднання сигналів руху та. У цьому дослідженні обидва підходи дотримуються створення двох різних форм коваріаційної матриці. У першій формі коваріаційної матриці функції на основі оптичного потоку та на основі градієнта використовуються разом. У другій формі використовуються лише функції на основі оптичного потоку. Для оцінки оптичного потоку використовується метод Горна-Шунка (Horn and Schunck, 1981).

Коваріаційна матриця визначається як:

$$C_t = \frac{1}{N-1}\sum_{k=1}^{N}(f_k - \mu)(f_k - \mu)^T \qquad (2.9)$$

де N — розмір набору ознак, а μ — середнє значення векторів ознак. Використання коваріаційної матриці має багато переваг:

- Це простий та ефективний спосіб інтеграції різних функцій.
- Це дескриптор низької розмірності. Розмір коваріаційної ознаки не залежить від розміру області, де вона обчислюється.

Хоча переваги, перераховані вище, роблять підходи, засновані на коваріаційній матриці, привабливими, важливою проблемою є те, що коваріаційні матриці визначені в рімановому різноманітті і не підходять для евклідових операцій. Щоб зробити коваріаційні матриці придатними для евклідових операцій, пропонується логарифмічна евклідова метрика (Arsigny et al., 2007) [13].

Коваріаційна матриця є симетричною позитивно визначеною (SPD) матрицею, і SPD-матриці не лежать у векторному просторі. Щоб зробити коваріаційні матриці придатними для використання в евклідовому просторі, використовується логарифмічна евклідова структура. Відповідно до цієї структури, коваріаційні матриці відображаються в евклідовий простір за допомогою матричної логарифмічної операції. Оцінка лог-коваріаційної матриці виконується наступним чином.

Нехай $SPD(n)$ і $S(n)$ позначають простір $n \times n$ дійсних SPD матриць і $n \times n$ дійсних симетричних матриць відповідно. Власне розкладання коваріаційної матриці $S \in S(n)$ дорівнює $S = U\Lambda U^T$, де U - нормальна матриця і $\Lambda = Diag(\lambda_1, \ldots, \lambda_n)$ є діагональною матрицею, яка містить власні значення λ_i від S. Якщо S є позитивно визначеною матрицею, $S \in SPD(n)$, то $\lambda_i > 0$ для $i = 1, \ldots, n$. [31] Використовуючи власне розкладання, експонента $S \in S(n)$ можна обчислити таким чином:

$$\exp(S) = U.Diag(\exp(\lambda_1), \ldots, \exp(\lambda_n)).U^T \quad (2.10)$$

Логарифм $S \in SPD(n)$ має такий вигляд:

$$\log(S) = U.Diag(\log(\lambda_1), \ldots, \log(\lambda_n)).U^T \quad (2.11)$$

Оскільки коваріаційна матриця є симетричною матрицею, виконується напіввекторизація, а остаточне представлення містить $n(n+1)/2$ значень.

Виявлення аномалії в переповнених сценах з використанням логарифмічної евклідової коваріаційної матриці. Ці дві функції доповнюють одна одну і дають інформацію про зовнішній вигляд і рух відповідно. Якщо їх використовувати разом, вони дають хороші результати. Вектор ознак $f_1(x, y, t)$, який витягується з позиції пікселя (x, y, t), має такий вигляд:

$$f_1(x, y, t) = [x, y, t, g, o]^T \quad (2.12)$$

Де:

$$g = [|I_x|, |I_y|, |I_{xx}|, |I_{yy}|, \sqrt{I_x^2 + I_y^2}] \quad (2.13)$$

$$o = \left[u, v, \frac{\partial u}{\partial t}, \frac{\partial v}{\partial t}, \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right), \left(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}\right)\right] \quad (2.14)$$

g і o представляють зовнішній вигляд і сигнали руху. Перші чотири ознаки на основі градієнта в (2.13) позначають градієнти інтенсивності першого та другого порядку в розташуванні пікселя (x, y, t), а останнє значення — це величина градієнта. [32][33] Характеристики на основі оптичного потоку в (2.14) позначають горизонтальну та вертикальну складові вектора потоку (u, v), похідні першого порядку горизонтальної та вертикальної складових оптичного потоку за часом $(\partial u/\ \partial t, \partial v/\partial t)$. Останні два значення, основані на оптичному потоці, — це просторова розбіжність і завихренність поля потоку (Алі та Шах, 2010)[19].

Друга форма коваріаційної матриці створюється з використанням тільки оптичних функцій, основаних на потоках. Вектор ознак $f_2(x, y, t)$, який витягується з положення пікселя *(x,y,t)*, має такий вигляд:

$$f_2(x, y, t) = [x, y, t, o]^T \tag{2.15}$$

Де:

$$o = [u, v, \frac{\partial u}{\partial t}, \frac{\partial v}{\partial t}, \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right), \left(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}\right), Gten, Sten] \tag{2.16}$$

Характеристики на основі оптичного потоку в (2.16) позначають горизонтальну та вертикальну складові вектора потоку (u, v), похідні першого порядку горизонтальної та вертикальної складових оптичного потоку за часом $(\partial u/\ \partial t\ , \partial v/\partial t\ )$ та просторової розбіжності та завихренності поля потоку. $Gten, Sten$ є тензорними інваріантами, які залишаються незмінними незалежно від того, в якій системі координат вони посилаються (Алі та Шах, 2010). $Gten, Sten$ є похідними від тензора градієнта оптичного потоку та тензора швидкості деформації. Тензор градієнта оптичного потоку $\nabla u(x, y, t)$ є 2×2 вимірною матрицею і визначається як:

$$\nabla u(x, y, t) = \begin{pmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{pmatrix} \tag{2.17}$$

Швидкість тензора деформації $S(x, y, t\ )$ дорівнює:

$$S(x, y, t) = \frac{1}{2}((\nabla u(x, y, t) + \nabla^2 u(x, y, t)) \tag{2.18}$$

$Gten$ та $Sten$ визначаються за допомогою $\nabla u(x,y,t)$ і $S(x,y,t)$:

$$Gten(x,y,t) = \frac{1}{2}(tr^2(\nabla u(x,y,t) + (\nabla^2 u(x,y,t))) \quad (2.19)$$

$$Sten(x,y,t) = \frac{1}{2}(tr^2(\mathrm{S}(x,y,t) - tr(S^2(x,y,t))) \quad (2.20)$$

де $tr(.)$ представляє операцію трасування

2.4 Архітектура технології оцінювання аномалій в даних пози людини

Перед створенням архітектури розроблюваної системи буде представлена use-case діаграма. Дана діаграма дозволить краще зрозуміти принцип роботи майбутньої системи та її взаємодію з користувачем. Таким чином буде представлено опис способів взаємодії користувача із системою. Для кращого представлення візуалізуємо діаграму за допомогою інструменту моделі варіантів використання.

Залежно від цільової аудиторії та системи, що обговорюється, варіант використання може бути настільки детальним або базовим, наскільки це необхідно. У діаграмі з варіантом використання буде визначено кілька ключових компонентів, а саме:

Система: система — в даному випадку програмне забезпечення, що обговорюється.

Актори: актор — це користувач або будь-що інше, що демонструє поведінку під час взаємодії з системою. Актором може бути інша система – в даній системі це модуль оцінювання аномалій.

Сценарій: «сценарій — це конкретна послідовність дій і взаємодій між акторами та системою, що обговорюється; його також називають екземпляром варіантів використання».

Випадок використання: сценарій використання описує сценарії успіху які будуть виникати, коли учасник(и) взаємодіє із системою. У даній діаграмі було встановлено основний сценарій успіху, тобто найбільш бажаний результат між актором і системою [37].

Суть даної діаграми полягає у тому, що проектована система представляється як сутності чи актори, які взаємодіють із системою за

допомогою варіантів використання. Варіанти використання застосовують для опису дії, які система представляє актору. Нижче описано сценарії успіху використання діаграми:

- Основна діюча особа: ініціатор запиту – користувач.
- Мета в контексті: система надає користувачу дані про наявність аномальних поз на зображенні.
- Область дії: соціальна сфера.
- Рівень: узагальнений.
- Учасники та інтереси: користувачу потрібно визначити чи присутні аномальні пози на зображенні чи відео матеріалі.
- Передумова: Інформація про наявні пози на зображенні.
- Мінімальні гарантії: робота системи зупиниться та на екрані з’явиться вікно про помилку.
- Гарантії успіху: система виведе результат аналізу оцінки аномалій в даних поз людей.
- Тригер: користувач завантажує зображення чи відео.

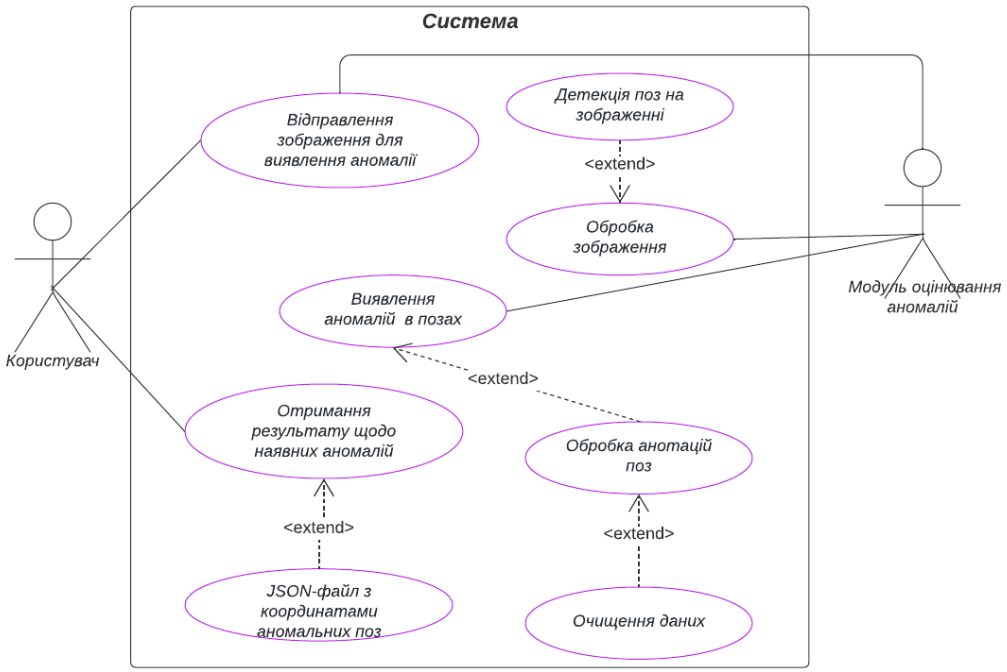

Рисунок 2.4 Діаграма варіантів використання

1. Система отримує дані про зображення:

    1а Дані завантажено некоректно;

    1а1 На екрані з’являється вікно про помилку;

    1а2 Користувач повертається на етап завантаження даних.

2. Система обробляє отримане зображення:

   – Детекція поз на зображенні.

3. Виявлення аномалій в позах людини:

   – Обробка анотацій в позах:

      - Очищення даних.

4. Отримання результату щодо наявних аномалій:

   – Отримання файлу з координатами аномальних поз.

UseCase: *«Оцінка аномалій в даних пози людини на зображенні.»*

   – Мета в контексті: користувач хоче виявити чи присутні аномалії в позах людей на зображенні.

   – Область дії: Система.

   – Рівень: Результуючий (Summary)

   – Передумова: Зображення на анатовано, а отже не виявлені пози людей.

- Критерій невдалого завершення: Виведення повідомлення про помилку, аналіз не виконано.
- Критерій вдалого завершення: Виконано детекцію поз людей та виявлені/невиявлені (якщо неприсутні) аномалії в даних на зображенні.
- Основна діюча особа: Користувач системи
- Основний вдалий сценарій:

1. Користувач обирає зображення, яке хоче проаналізувати.
2. Користувач відкриває систему.
3. Користувач запускає систему для виявлення аномалій в даних пози людини.
4. На екран виводиться результат аналізу та інформація про виявленні пози.

Тепер перейдемо до моделювання архітектури майбутньої системи.

На рисунку 2.5 представлена загальна блок-схема системи. Спочатку буде анотовано набір даних «Поза людини». Потім буде проведено відбір даних для нашого проекту, щоб переконатися в справедливості між різними суб’єктами. Потім ми проаналізуємо оцінки аномалії рамки PersonPose з попереднього дослідження. На додачу, буде виконано моделювання аномалій для різних суглобів на основі різних факторів для досягнення контрольованого/порівнянного дослідження для кожного суглоба/об'єкта. Буде відібрано вибірку з моделі аномалії до вибраного набору даних пози людини, щоб сформувати набір даних для виявлення аномалій. Довгі послідовності поз можуть мати досить високі розміри. Для цього має сенс навчити модель на основі VAE виконувати загальне зменшення розмірів однієї пози, а також корекцію аномалії однієї пози. На основі техніки зменшення розміру, буде виконано навчання моделі для виявлення аномалій послідовних поз для кожного суб’єкта [40].

Загальна система глибокого навчання побудована з двох підсистем. Перший етап - це система моделювання загальної пози на основі VAE для аналізу неспецифічних поз і надання необхідної інфраструктури для системи другого етапу. Система першого ступеня також може використовуватися для виправлення аномалій на суглобах, таких як плечовий. Другим етапом є система

моделювання пози для конкретного предмета на основі VAE для аналізу та виявлення тимчасових послідовних аномальних поз. Система другого етапу допомагає фіксувати суглоби аномалій за допомогою рухів у часовій послідовності.

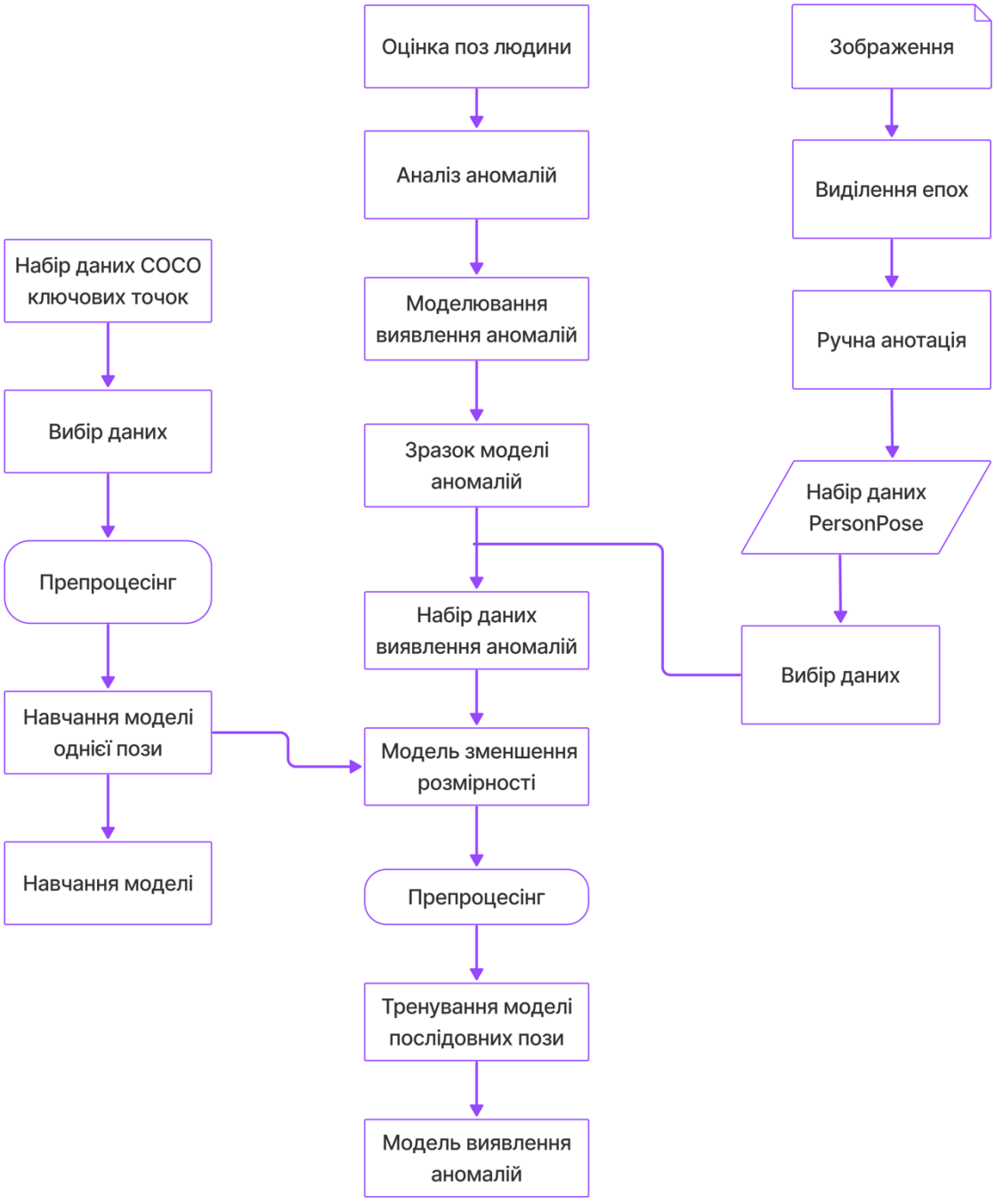

Рисунок 2.5 Блок-схема системи

На першому етапі ми використовували варіаційний автокодер (VAE) [2] на основі DNN (нейронної мережі глибокого навчання), щоб змоделювати загальну

позу верхньої частини тіла людини [41]. Отже, коли ви вводите один кадр суглобів верхньої частини тіла, він виконає нормалізацію розташування та масштабу, а потім подає в наш перший етап VAE для реконструкції суглобів верхньої частини тіла. На основі подібності реконструйованих суглобів верхньої частини тіла та вихідного входу ми можемо оцінити, чи є вхідний суглоб аномалій чи ні. Систему першого етапу можна побачити на малюнку 2.9.

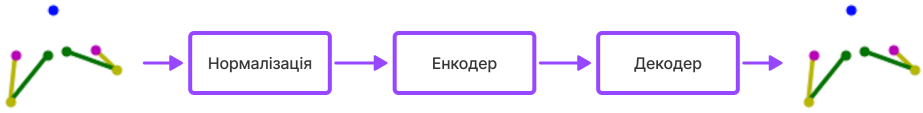

Рисунок 2.6 Аналіз першого етапу

Вхідними даними для аналізу першого етапу є 7 двовимірних суглобів верхньої частини тіла з x, y кожного суглоба, загалом дані 17 вимірів. Ідеальний вихід декодера VAE першого етапу повинен бути таким же, як і нормований вхід. Для другого етапу ми використали інший варіаційний автокодер (VAE) [22] на основі DNN (нейронної мережі глибокого навчання), щоб моделювати безперервні послідовності пози людини, що стосуються конкретного предмета. Для введення VAE другого етапу ми використовуємо ту саму техніку нормалізації та кодер, що й на першому етапі VAE, щоб виконати зменшення розмірів і віднімання ключової інформації для кожного з даних про безперервну позицію нашого суб'єкта [44].

Потім відбираємо безперервні пози в невеликі послідовності поз, кожна послідовність є вхідною одиницею для другого етапу VAE. Отже, можемо оцінити, чи є вхідна послідовність аномалією чи ні, на основі подібності відновленої послідовності та вхідної послідовності. Систему другого етапу можна побачити на малюнку 2.7.

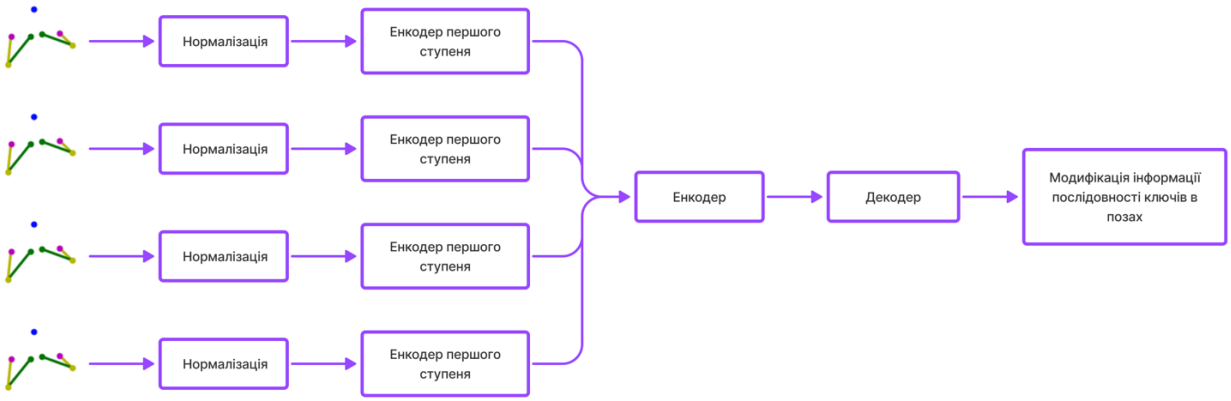

Рисунок 2.7 Аналіз другого етапу

Вхідні дані другого етапу аналізу - це послідовність поз предмета, це може бути будь-яка кількість послідовних кадрів.

РОЗДІЛ 3 СТРУКТУРА ІНФОРМАЦІЙНО-АНАЛІТИЧНОГО ЗАБЕЗПЕЧЕННЯ ТЕХНОЛОГІЇ ОЦІНЮВАННЯ АНОМАЛІЙ В ДАНИХ ПОЗИ ЛЮДИНИ ТА ПЕРЕВІРКА ЇЇ РОБОТИ

3.1 Модуль програмного забезпечення та необхідних бібліотек для модулювання

Програмне забезпечення узагальнює технологію оцінювання аномалій. Для того, щоб дана технологія була застосованою серед звичайних користувачів було створено сервіс обробки зображень, для аналізу на аномалії [46].

Щоб краще зрозуміти роботу системи та всі необхідні бібліотеки та налаштування в роботі сервісу було створено блок-схему роботи даної системи.

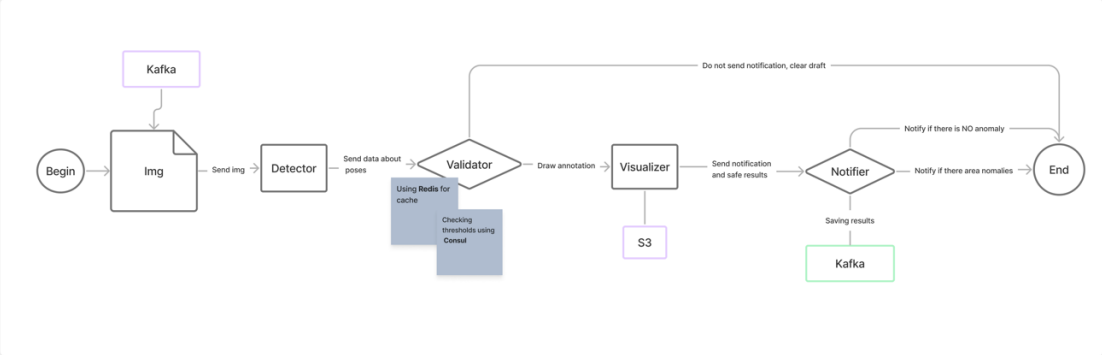

Рисунок 3.1 Блок-схема сервісу

Цей моніторинг буде здійснюватися на знімках, а не на потоці. Повідомлення буде надіслано відповідно до такої конфігурації та порогових значень.

Сервіс буде використовувати об’єкт знімка через Kafka. Об’єкт знімка містить знімок і мітку часу, camera_id тощо.

Apache Kafka — це платформа з відкритим вихідним кодом, розроблена Apache Software Foundation, яка використовується для обробки потоків. Він написаний на Java та Scala. Мета проекту — запропонувати високопродуктивну, уніфіковану платформу з низькими затримками для обробки даних у реальному часі. Рівень сховища Apache Kafka, по суті, є «масштабованою чергою повідомлень публічних/підрядних повідомлень, розробленою як розподілений журнал транзакцій», що робить його надзвичайно цінним для інфраструктури

підприємства для обробки потокових даних. Крім того, Kafka підключається до зовнішніх систем (для імпорту та експорту даних) через Kafka Connect і пропонує Kafka Streams, бібліотеку для обробки потоків Java [48].

Сервіс отримає свою конфігурацію через Consul. Consul — це рішення для сервісної сітки, що надає повнофункціональний рівень керування з функціями виявлення, конфігурації та сегментації послуг. Кожну з цих функцій можна використовувати окремо, якщо потрібно, або їх можна використовувати разом для створення повної сітки послуг. Consul потребує площині даних і підтримує як проксі, так і власну модель інтеграції. Consul постачається із простим вбудованим проксі-сервером, щоб усе працювало «з коробки», але також підтримує інтеграцію сторонніх проксі-серверів, наприклад Envoy [48].

Служба зберігає зображення X (з точками) у S3 (сховище об’єктів AWS) для кожного зображення (ідентифікатор зображення має бути назвою сегмента).

Amazon Simple Storage Service (Amazon S3) — це служба зберігання об’єктів, яка пропонує провідну в галузі масштабованість, доступність даних, безпеку та продуктивність. Клієнти всіх розмірів і галузей можуть зберігати та захищати будь-яку кількість даних практично для будь-якого випадку використання, наприклад, озера даних, хмарні програми та мобільні додатки. Завдяки економічно ефективним класам зберігання та простим у використанні функціям керування ви можете оптимізувати витрати, упорядкувати дані та налаштувати точні засоби контролю доступу, щоб відповідати конкретним вимогам бізнесу, організації та відповідності вимогам[50].

Сервіс ігнорує певні області у кадрах, полігони, які потрібно ігнорувати, та зберігає кеш у Consul.

Сервіс має на меті сповіщати, коли виявлені аномалії на зображенні і коли їх немає.

Для кожного отриманого знімка ми виконаємо:

1. Передача знімку до детектора.
2. Перевірка, що точки, які були виявленні детектором не були виявлені раніше.

3. Якщо були виявлені аномалії, перевірка, що кожна виявлена поза задовольняє пороговим значенням.
4. Візуалізація аномальної пози на зображенні та збереження зображення на S3.
5. Надіслання сповіщення про виявлену аномалію в окремий простір Kafka з усіма зображеннями, збереженими в S3.
6. Збереження сповіщення в кеші, щоб служба не надсилала одне й те саме сповіщення знову і знову про одне й те саме зображення.

3.2 Дослідження програмного забезпечення технології оцінювання аномалій в даних пози людини

Для реалізації модулю навчання було використано мову програмування Python. Для реалізації навчання було використано такі бібліотеки та надбудови:

- PIL;
- NumPy;
- OpenCV;
- PyTorch.

PIL є бібліотекою зображень Python додає можливості обробки зображень до інтерпретатора Python.

Ця бібліотека забезпечує широку підтримку форматів файлів, ефективне внутрішнє представлення та досить потужні можливості обробки зображень.

Основна бібліотека зображень розроблена для швидкого доступу до даних, що зберігаються в кількох основних форматах пікселів. Він повинен забезпечити міцну основу для загального інструменту обробки зображень [52].

NumPy — це керований спільнотою проект із відкритим кодом, розроблений різноманітною групою учасників. Керівництво NumPy взяло на себе тверде зобов’язання створити відкриту, інклюзивну та позитивну спільноту. Будь ласка, прочитайте Кодекс поведінки NumPy, щоб отримати вказівки щодо того, як взаємодіяти з іншими, щоб наша спільнота процвітала[53].

OpenCV (Open Source Computer Vision Library) — це бібліотека програмних методів, в основному спрямованих на комп’ютерний зір у реальному часі. Спочатку розроблений Intel, пізніше він був підтриманий Willow Garage, а потім Itseez (який пізніше був придбаний Intel). Бібліотека є кросплатформною та безкоштовною для використання за ліцензією Apache 2 з відкритим кодом. Починаючи з 2011 року, OpenCV має прискорення графічного процесора для операцій у реальному часі [54].

PyTorch — це фреймворк машинного навчання з відкритим вихідним кодом, заснований на бібліотеці Torch, який використовується для таких додатків, як комп’ютерний зір та обробка природної мови, в першу чергу розроблений Meta AI. Це безкоштовне програмне забезпечення з відкритим вихідним кодом, випущене під ліцензією Modified BSD. Хоча інтерфейс Python є більш відшліфованим і основним напрямом розробки, PyTorch також має інтерфейс C++.

PyTorch визначає клас під назвою Tensor (torch.Tensor) для зберігання й роботи з однорідними багатовимірними прямокутними масивами чисел. Тензори PyTorch подібні до масивів NumPy, але також можуть працювати на графічному процесорі Nvidia з підтримкою CUDA. PyTorch підтримує різні підтипи тензорів. Зауважте, що термін «тензор» тут не має такого значення, як у математиці чи фізиці. Значення слова в цих областях лише дотично пов’язане зі значенням в машинному навчанні. У математиці тензор - це певний вид об'єкта в лінійній алгебрі, тоді як у фізиці термін "тензор" зазвичай відноситься до того, що математики називають тензорним полем. [55]

3.2.1 Характеристика вхідних та вихідних даних програмного забезпечення

Кожній позі відповідає 17 певних точок. Це координати розташування пози людини на зображенні. Кожній точці відповідає певна частина тіла. Тому для виявлення аномалій було застосовано евклідову відстань. Було визначено різні рівні аномалії для набору даних з якими проводилась робота. Були задані порогові значення евклідової відстані між з’єднаннями, позначеними вручну та передбаченими з’єднаннями. Було визначено амплітуду аномалії, як з’єднання з

найбільшою евклідовою відстанню між міченими вручну з’єднаннями та передбаченими з’єднаннями в позі.

На основі даних спостережень та отриманих результатів було виявлено два найпоширеніші типи аномалій. широкі типи аномалій. Ці два типи аномалій можна розділити на дві групи: аномалія жорсткого збою та аномалія м’якого збою.

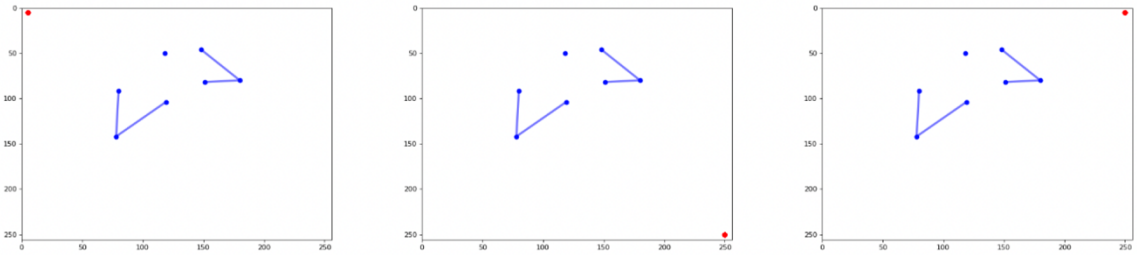

Рисунок 3.2 Візуалізація аномалії жорсткого збою. Блакитний колір – це базова поза правди, червоний – передбачувана поза. Передбачувані суглоби завжди згруповані в кутку прицiла.

Аномалія жорсткого збою визначається як оцінка пози людини без видимої форми людини, наприклад, екстремальне прогнозне значення за межами об’єму або всі з’єднання, згруповані в одній точці, як показано на малюнку 3.2 У нашому наборі даних тестування є 11 із 3000 кадрів, які є серйозними аномаліями, ми відфільтровуємо їх перед подальшим аналізом аномалій.

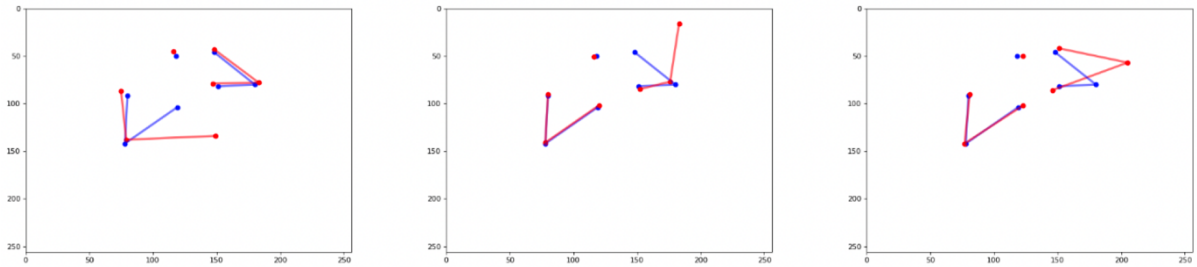

Рисунок 3.3 Аномалія м'якого збою. Візуалізація аномалії м’якого збою. Блакитний колір – це базова поза правди, червоний – передбачувана поза. Більшість аномалій м’якої недостатності трапляються на зап’ястях, плечах і ліктях. Прогнозований суглоб відходить від основної істини з помітною амплітудою.

Аномалія м’якого збою визначається як оцінена поза, яка все ще має форму людини, але з певними неточностями, наприклад невеликі зсуви на кількох суглобах або великий зсув на одному суглобі, як показано на малюнку 3.3. Іншими словами, аномалії, що не є жорсткими, є аномаліями м’якого збою.

Як і в оригінальній статті PersonPose, найменший суглоб, зап’ястя, має ширину близько 15 пікселів у наших записах, ми визначаємо людську помилку під час маркування як до 15 пікселів евклідової відстані між з’єднання, позначені вручну, і передбачені з’єднання. Отже, будь-які оцінки суглобів з похибкою або вище 15 пікселів можна прийняти як випадок збою або аномалію [23].

Для набору даних, запропонованого в даних PersonPose з порогом 15 пікселів розподіл виявлених аномалії для різних суглобів можна побачити на малюнку 3.4. Більшість аномалій зустрічається на зап'ястях, у цілому 731 зап'ястя з 3000 кадрів як аномалії, що становить більше половини всіх аномалій. Секунда аномальними суглобами є плечі, із загальною аномалією 263 суглоби, що становить чверть усіх аномалій. Лікті становлять меншу частину загальної кількості аномалій у наборі даних.

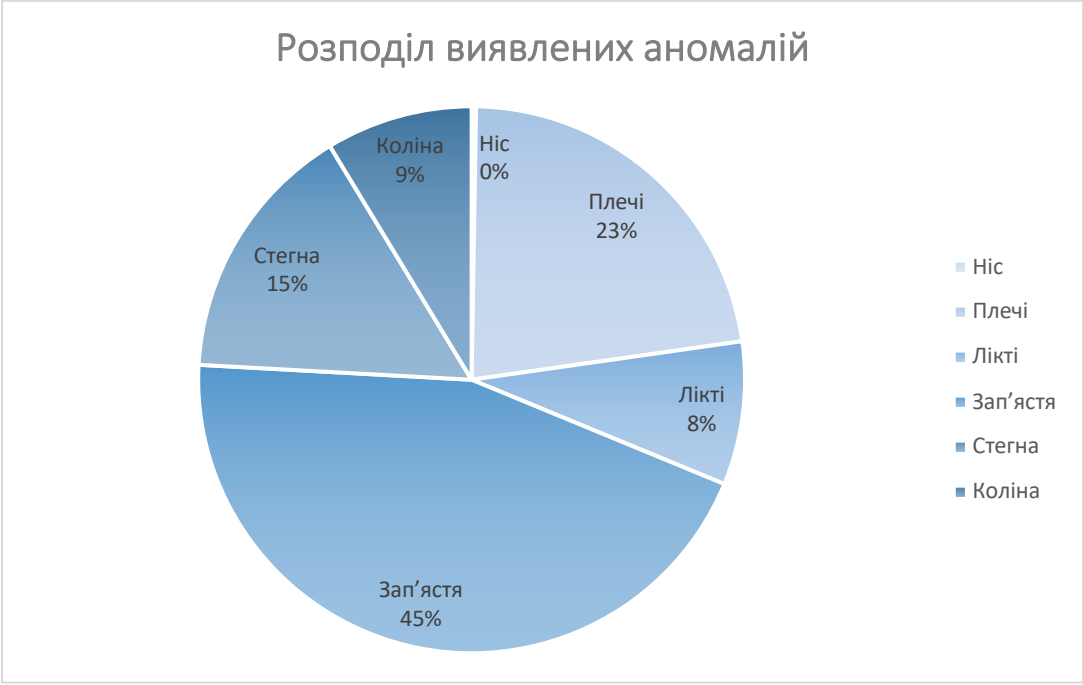

Рисунок 3.4 Розподіл виявлених аномалій

Ми моделюємо амплітуду, кут і тимчасову безперервність (або «послідовність» виявлених аномалій для кожного цікавить суглоба, як показано на малюнку 3.5.

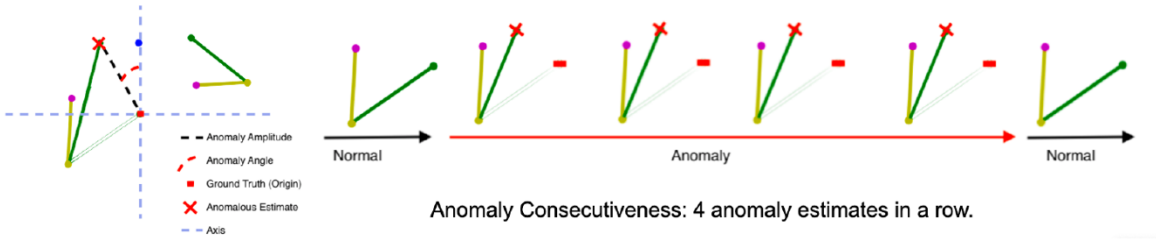

Рисунок 3.5 Візуалізація факторів аномалій

Амплітуда визначається евклідовою відстанню між оцінками положення суглоба та основною істиною в одиницях пікселів.

Кут визначається встановленням початку координат як основної істини та кута за годинниковою стрілкою, утвореного від додатної осі Y до сегмента виявлення аномалії та основної істини [42].

Послідовність визначається як кількість послідовних кадрів, які є аномаліями для кожного суглоба. При спостереженні послідовних кадрів аномалій послідовні кадри аномалій мають подібні амплітуду та кут аномалії.

Було використано оцінку щільності ядра (KDE) для моделювання розподілу амплітуди, кута та послідовності виявлення аномалії для кожного зацікавленого з'єднання, як зазначено в рівняннях [34].

$$\widehat{p_n}(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right), \qquad (11)$$

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}} . \qquad (12)$$

$K(x)$ — функція ядра, у нашому випадку — гауссова. h - це пропускна здатність, щоб вказати плавність кривої PDF, яку ми збираємося отримати. X_i — точка даних, n - номер точки даних.

Використовуючи KDE, ми можемо вибирати дані про нескінченні аномалії щодо амплітуди, кута та числа послідовних кадрів із наявних даних про реальні аномалії.

Таблиця 3.1

Поріг (Pixel)	Кадри з аномаліями (з 3000 кадрів)	Частота аномалій
15	903	0.301
16	624	0.208
17	477	0.159
18	294	0.098
19	258	0.086
20	228	0.076
21	198	0.066
22	150	0.05
23	120	0.04
24	108	0.036
25	102	0.034
26	99	0.033
27	93	0.031
28	90	0.03
29	84	0.028
30	80	0.027

Єдиним гіперпараметром є пропускна здатність. Щоб отримати оптимальну пропускну здатність, ми дотримуємося припущення, що принаймні 50 або більше точок даних можуть узагальнити аномальний розподіл даних. Менше 50 точок даних можуть не узагальнити розподіл даних про аномалії. Отже, опускаємо голову (ніс) за комбінезон.

Процес виявлення аномалії PatientPose. 20-кратна перехресна перевірка була застосована для виконання пошуку гіперпараметрів пропускної здатності.

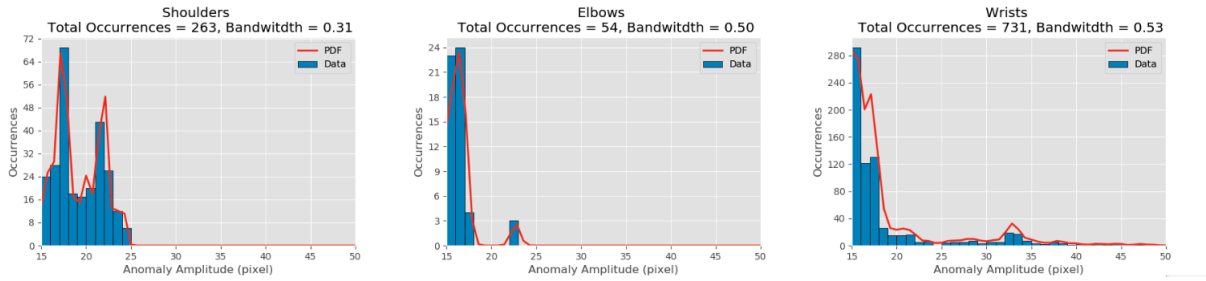

Рисунок 3.6 Амплітудне моделювання аномалій

Результат моделювання амплітуди аномалії KDE можна побачити на малюнку 3.7. Зап’ястя мають найбільший спектр з точки зору діапазону амплітуди аномалії від 15 пікселів до 50 пікселів [25].

Результат моделювання кута аномалії KDE можна побачити на малюнку 3.7. Кути аномалій сильно розподілені між 45 і 270 градусами за годинниковою стрілкою.

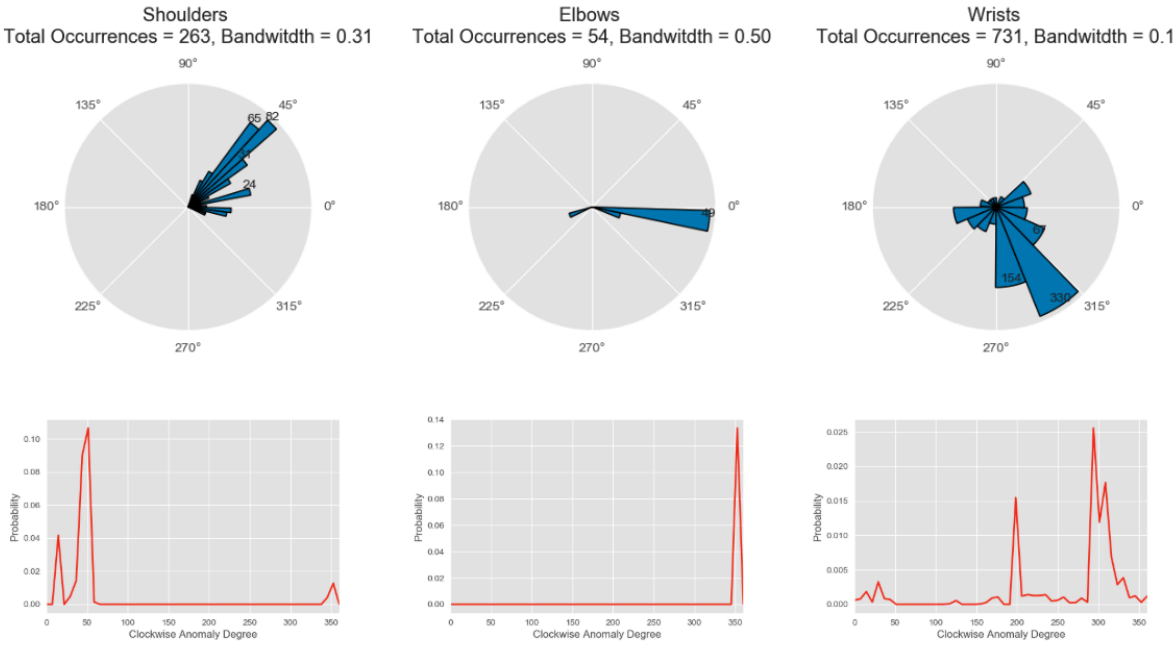

Рисунок 3.7 Моделювання кута аномалії

Тепер ми можемо випробувати аномалію для кожного з’єднання, відбираючи зразки з цих моделей аномалій. Ми можемо згенерувати однакову кількість

даних про аномалії для кожного з’єднання, щоб порівняти, як працює наша структура між різними суглобами.

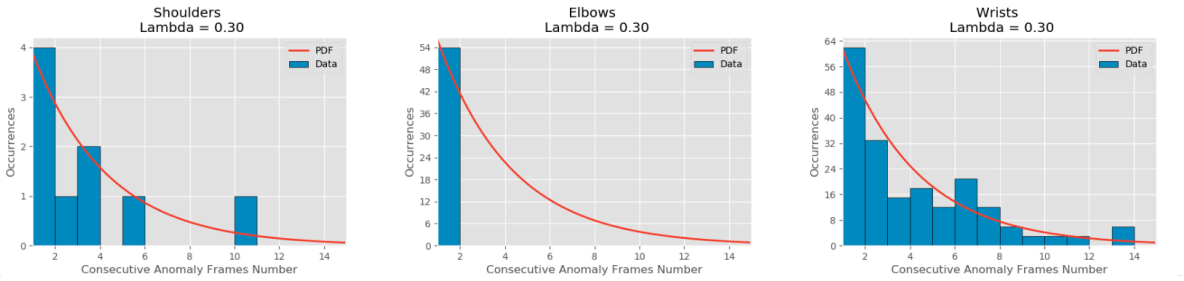

Рисунок 3.8 Моделювання послідовності аномалій

Для моделювання пози було використано набір ключових точок COCO [13]. Набір даних ключових точок COCO [13] спочатку був запропонований для створення 2D фреймворків виявлення пози. Набір даних складається з наборів підготовки, перевірки та тестування, які містять понад 200 000 зображень і 250 000 екземплярів різних масштабів, позначених ключовими точками.

Маємо 17 ключових точок:

- ніс,
- праве плече,
- правий лікоть,
- праве зап’ястя,
- ліве плече,
- лівий лікоть,
- ліве зап’ястя,
- праве стегно,
- праве коліно,
- права щиколотка,
- ліве стегно,
- ліве коліно,
- ліва щиколотка,
- ліве око,
- праве око,

- ліве вухо,
- праве вухо.

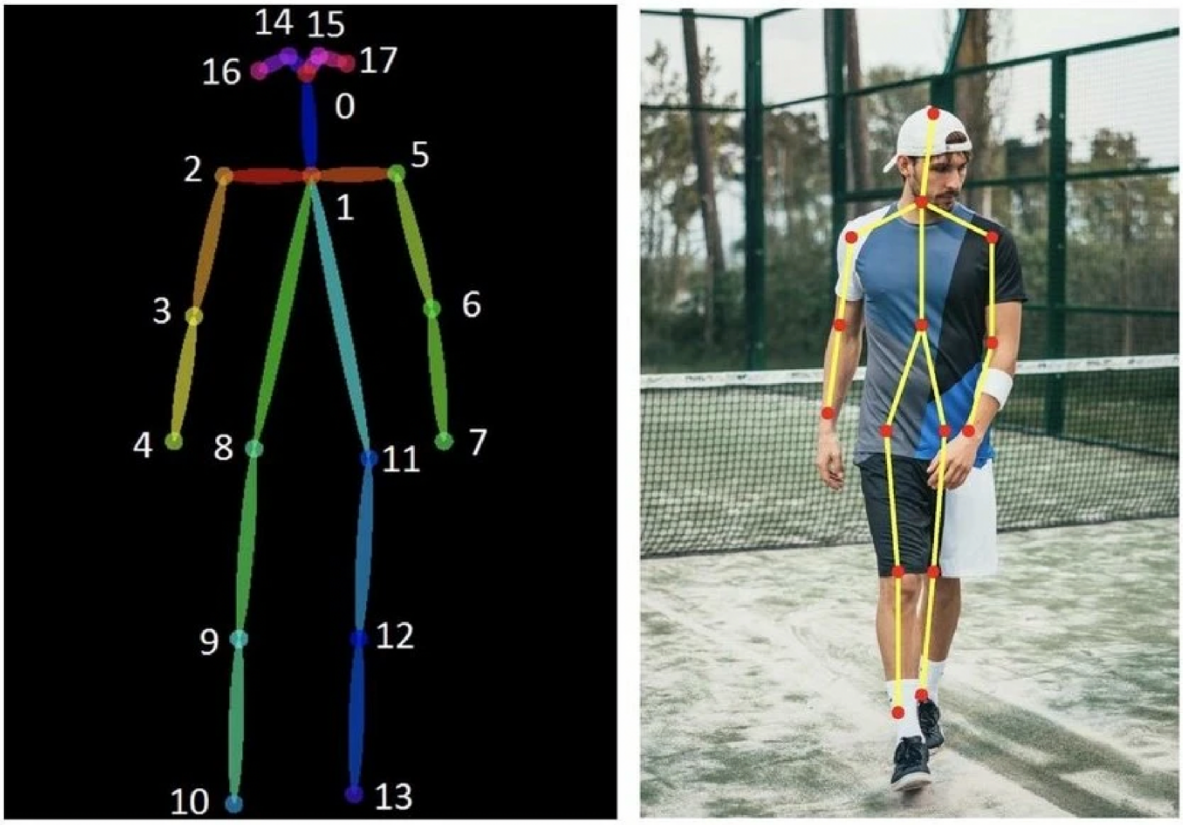

Рисунок 3.9 Ключові точки у наборі даних COCO.

На одному зображенні може бути кілька людей, і не всі суглоби мають бути повністю позначені. Визначення суглобів верхньої частини тіла схоже на наш набір даних PatientPose 3D. У ньому понад 150 000 людей і загалом 1,7 мільйона позначених ключових точок. Ми помічаємо, що деякі з'єднання позначені погана якість, наприклад, суглоби оклюзії, невидимі суглоби або просто помилкові мітки. Анотації набору даних COCO були позначені за допомогою краудсорсингу Amazon Mechanical Turk. У середньому Amazon Mechanical Turk має нижчу точність порівняно з експертами з маркування [13]. Незважаючи на людську помилку в анотаціях, ми вважаємо, що анотації COCO є основною правдою.

Було використано набір даних Humaneva I [14] для подальшої перевірки універсальності нашої системи виявлення клінічних аномалій. Набір даних Humaneva I [14] спочатку запропоновано для побудови та оцінки тривимірної

системи відстеження поза людини. Це професійний набір даних на основі системи захоплення руху. Він містить 4 відтінки сірого та 3 відкалібровані за кольором відеопослідовності. Відеозаписи синхронізовані з 3D позами тіла. За замовчуванням набір даних містить набори для навчання, перевірки та тестування.

Ключові точки визначаються аналогічним чином, як набір даних COCO keypoints та набір даних PatientPose 3D, але є також деяка незначна відмінність, наприклад, Humaneva визначає ключову точку голови як верхню частину голови, де набір даних COCO ключових точок визначає її як ніс. Крім того, на відміну від набору даних PatientPose 3D, набір даних Humaneva I знаходиться в лабораторному контексті з жорсткими рухами всього тіла, де PatientPose [43].

Для моделювання однієї пози та аналізу аномалій ми використовували набір даних формату COCO ключових точок. Як ми згадували раніше, набір ключових точок COCO не є ідеальним набором даних. Всього в наборі даних близько 273 469 поз, лише 39 714 поз задовольняють нашим вимогам для навчання VAE першого етапу. Детальні вимоги і процес фільтрації можна побачити в таблиці 3.1.

Таблиця 3.2

	Навчальний набір	Валідаційний набір	Тестовий набір	Всього
К-сть поз	4656	1886	1886	7428

Для моделювання однієї пози та аналізу аномалій ми використовували набір даних ключових точок COCO. Як ми згадували раніше, набір ключових точок COCO не є ідеальним набором даних. Всього в наборі даних 273 469 поз, лише 39 714 поз задовольняють нашим вимогам для навчання VAE першого етапу. Детальні вимоги та процес фільтрації можна побачити в таблиці 3.2.

Таблиця 3.3

Вимоги	Пози, що залишилися
Дійсна поза	7469
Повністю позначені суглоби верхньої частини тіла	1910
Позначенні видимі з'єднання	2289
Дані про з’єднання в допустимому діапазоні	9714

3.2.2 Аналіз отриманих результатів роботи технології оцінювання виявлення аномалій в даних пози людини

Як і говорилось у 2-му розділі для навчання були обрані дві архітектури основані на глибинному навчанні:

- OpenPose
- High-Resolution Net (HRNet).

Навчання проводилось на однакових даних. Та на основі навчання було проведено експеримент та обрано найефективнішу архітектуру для визначення поз (36).

Нижче буде показано результати навчання. У задачах комп’ютерного зору виявлення об’єктів — це проблема визначення місця розташування одного або кількох об’єктів на зображенні. Окрім традиційних методів виявлення об’єктів, передові моделі глибокого навчання, які можуть виявляти різних типів об’єктів. Ці моделі приймають зображення як вхідні дані і повертають координати обмежувальної рамки навколо кожного виявленого об’єкта.

Для валідації роботи двох архітектур застосовані параметри Precision та Recall. І точність (Precision), і повнота (Recall).

Точність (Precision) — це відношення істинно позитивних результатів до загальної кількості істинно позитивних і хибно позитивних результатів. Precision використовується, щоб побачити, скільки небажаних позитивних результатів було класифіковано як виброси. Якщо немає хибно позитивних результатів (ті FP), то модель має 100% точність. Чим більше FP потрапить у виброси, тим гірше буде виглядати точність.

$$Precision = TP/(TP + FP) \qquad (3.1)$$

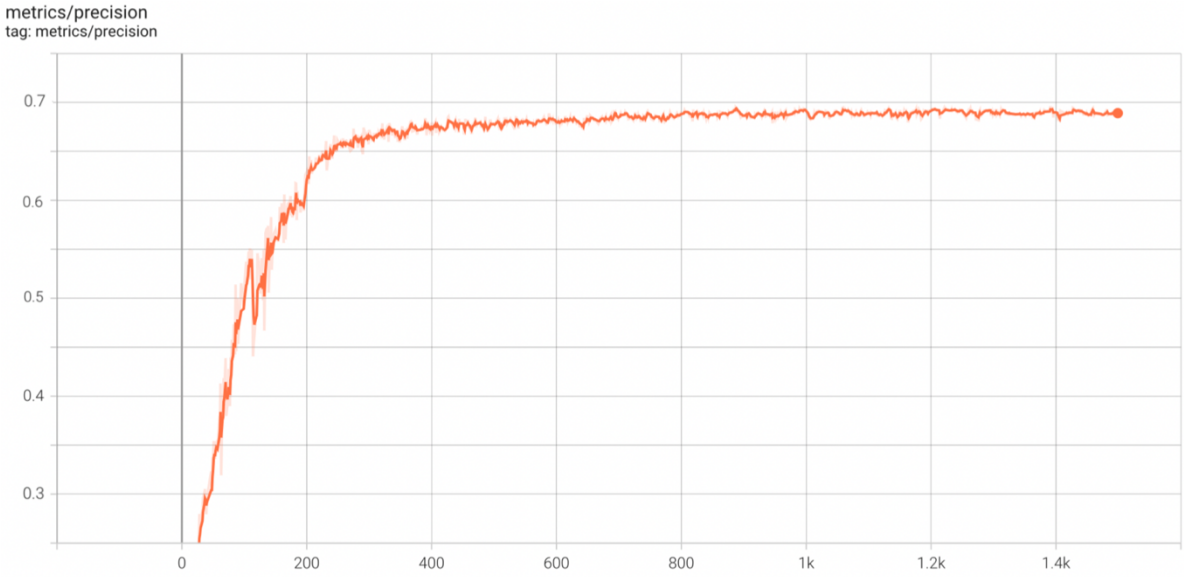

Рисунок 3.11 Показник Precision моделі OpenPose

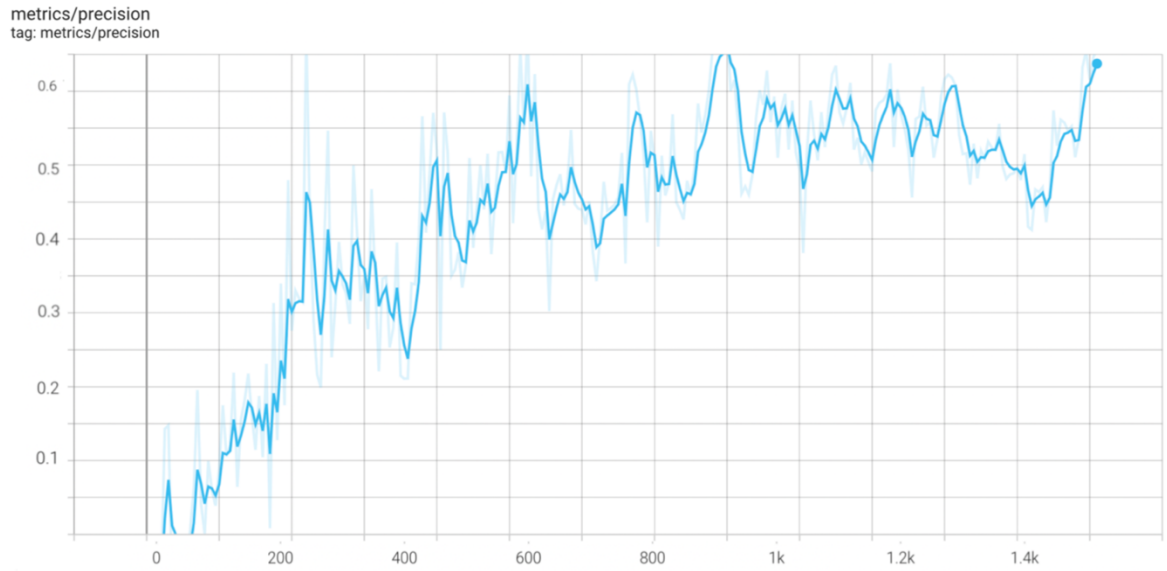

Рисунок 3.12 Показник Precision моделі HRNet

Як можна побачити з графіків результат архітектури OpenPose дає кращі результати на даному наборі даних, його найкращий результат становить 0.6974, в той час як HRNet дає результат в 0.6125.

Повнота (Recall) оцінюється іншим шляхом шляхом. Замість того, щоб дивитися на кількість хибно позитивних результатів, передбачуваних моделлю, тут перевіряється кількість хибно негативних результатів, які були додані до передбачення.

$$Recall = TP/(TP + FN) \qquad (3.2)$$

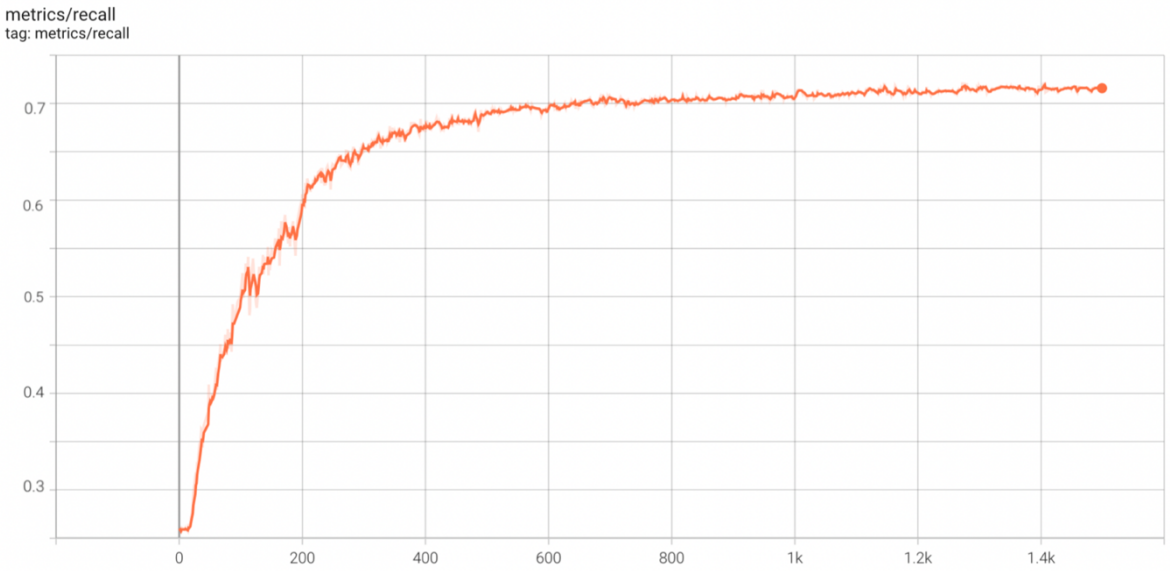

Рисунок 3.13 Показник Precision моделі OpenPose

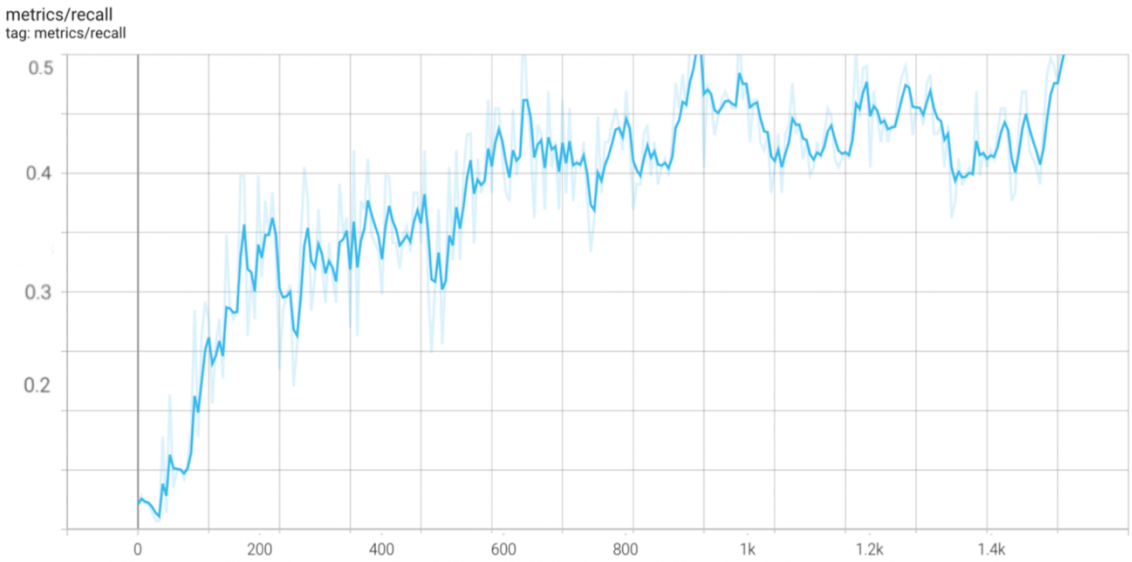

Рисунок 3.14 Показник Precision моделі HRNet

З двох графіків можна побачити, що результати роботи з застосуванням архітектури OpenPose дали результат в 0.7365, в той час як HRNet має найкращий показник в 0.5309.

З проаналізованих результатів можна зробити висновок, що застосування OpenPose архітектури значно краще вплинуло на кінцевий результат в роботі програмного забезпечення. Саме тому було вирішено зупинитися на архітектурі OpenPose для подальшрого вдосконалення роботи моделі.

3.3 Аналіз та узагальнення отриманих результатів роботи програмного забезпечення

Перейдемо до узагальнення роботи програмної системи. В результаті навчання моделі було отримано результат в 0.8042.

Функція втрат являє собою суму втрат (середньоквадратична помилка) і втрати від дивергенції KL, зважених на b. Детальну функцію втрат можна побачити в рівняннях нижче [32].

$$Loss = \frac{1}{N}\sum_{n=1}^{N}(x_i - x_t) \quad (3.3)$$

$$Loss_{KL} = KL(q(z|x)||p(z)), \quad (3.4)$$

$$Loss = \ Loss + \ \beta * Loss_{KL} \quad (3.5)$$

Втрата реконструкції вимірює різницю між вхідними позами та відновленими позами. Отже, якщо відновлені пози знаходяться далеко від вхідних поз, втрати при реконструкції будуть високими, і навпаки. Втрата KL буде високою, якщо розподіл $q(z|x)$, створений VAE, далекий від реального розподілу вхідних даних $p(z)$ у латентному просторі, який ми вважаємо нормальним розподілом.

Подібно до гіперпараметрів моделювання загальної пози, ми розробили набір VAE з одним проміжним щільним шаром як в кодері, так і в декодері. Отже, решта гіперпараметрів, які підлягають пошуку, - це b втрати KL, розміри проміжного щільного шару, прихований розмір і число послідовних кадрів у блоці вхідних даних.

В якості експерименту було проаналізовано роботу моделі з анотованими даними та без, для оцінки якості використаємо показник AP, який дає середній результат точності (precision) та повноти (recall).

На рис. 3.11 можна дві криві, які відображають процес навчання моделі. Крива позначена сірим кольором визначається моделю з додаванням власного набору даних, а рожевим – без.

```
AutoAnchor: 4.45 anchors/target, 0.994 Best Possible Recall (BPR). Current anchors are a good fit to dataset ✅
Image sizes 640 train, 640 val
Using 4 dataloader workers
Logging results to runs/train/exp52
Starting training for 1000 epochs...

      Epoch   gpu_mem       box       obj       cls    labels  img_size
      0/999     3.44G    0.1229    0.0787         0       211       640: 100% 8/8 [00:04<00:00,  1.89it/s]
                Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  4.61it/s]
                  all          5        106    0.00692     0.0283    0.00116   0.000189

      Epoch   gpu_mem       box       obj       cls    labels  img_size
      1/999     3.44G    0.1211   0.07355         0       286       640: 100% 8/8 [00:01<00:00,  4.37it/s]
                Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  4.71it/s]
                  all          5        106    0.00872     0.0566    0.00316   0.000467

      Epoch   gpu_mem       box       obj       cls    labels  img_size
      2/999     3.44G    0.1194   0.07801         0       194       640: 100% 8/8 [00:02<00:00,  3.03it/s]
                Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  4.41it/s]
                  all          5        106     0.0122     0.0377    0.00283   0.000395

      Epoch   gpu_mem       box       obj       cls    labels  img_size
      3/999     3.44G    0.1177   0.07784         0       129       640: 100% 8/8 [00:03<00:00,  2.60it/s]
                Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  5.32it/s]
                  all          5        106    0.00919     0.0283    0.00194   0.000298

      Epoch   gpu_mem       box       obj       cls    labels  img_size
      4/999     3.44G    0.1154   0.07587         0       216       640: 100% 8/8 [00:03<00:00,  2.46it/s]
                Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  5.68it/s]
                  all          5        106    0.00647     0.0189    0.00219   0.000634

      Epoch   gpu_mem       box       obj       cls    labels  img_size
      5/999     3.44G    0.1147   0.07905         0       161       640: 100% 8/8 [00:02<00:00,  2.69it/s]
                Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  5.16it/s]
                  all          5        106    0.00637     0.0189    0.00255   0.000642
```

Рисунок 3.15 Скріншот роботи ПЗ

Найкращий результат було досягнуто з додаванням власного набору даних – 0.7965. В той час як модель без додавання даних дає показник в 0.7342.

Хоча результат не відрізняється суттєво, це значно впливає на здатність точно оцінити наявність аномалій в роботі моделі.

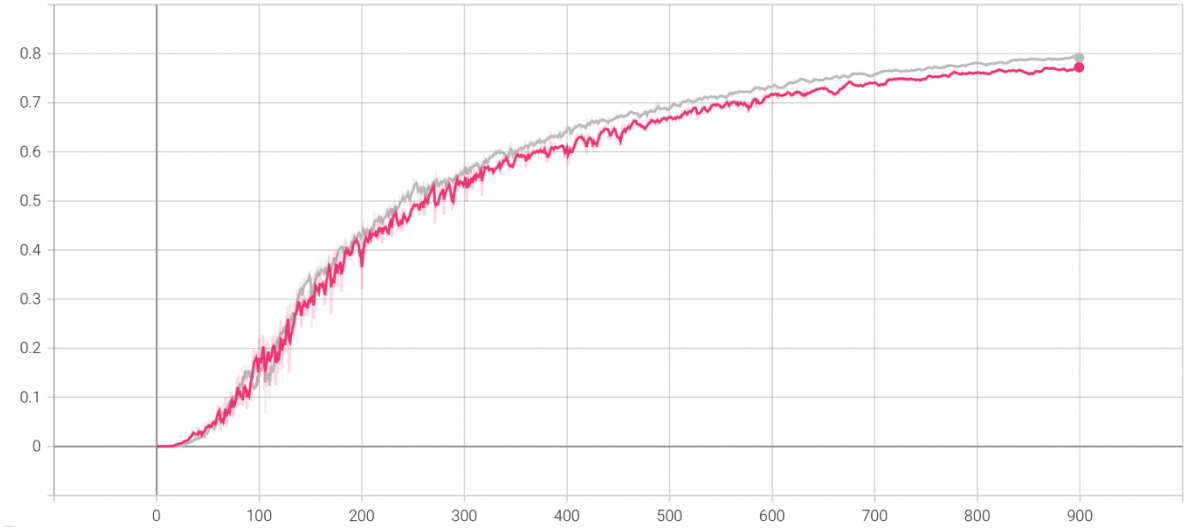

Рисунок 3.16 Перевірка роботи моделей

Проведемо тестування роботи інформаційно-аналітичної системи на основі 20-х тестових зображень.

Нижче буде показано декілька зображень, які використовувалися при тестуванні.

Рисунок 3.17 Приклади зображень

Зображення для проведення тестування були взяті з відкритого набору даних з камер відеоспостереження, розташованих ну вулицях міст з різних куточків світу.

Дані отримані в результаті роботи сервісу показані нижче. Отримано JSON з повною інформацією про позу (якщо така виявлена). Де спочатку маємо інформаціє про ID людини, поза якої є аномальною, потім координати з'єднувальних ліній між точками, а потім координати точок людини.


```
{
  "status": "Anomaly",
  "predictions": [
    {
      "human_id": 1,
      "pose_lines": [
        {
          "line": [
            505,
            233,
            499,
            233
          ]
        },
        {
          "line": [
            505,
            233,
            514,
            233
          ]
        },
        {
          "line": [
            499,
            233,
            499,
            258
          ]
        }
```

```
      "body_parts": [
        {
          "part_id": 0,
          "part_name": "Nose",
          "score": "0.27673",
          "x": 497,
          "y": 222
        },
        {
          "part_id": 1,
          "part_name": "Neck",
          "score": "0.51888",
          "x": 505,
          "y": 233
        },
        {
          "part_id": 2,
          "part_name": "RShoulder",
          "score": "0.48903",
          "x": 499,
          "y": 233
        },
        {
          "part_id": 3,
          "part_name": "RElbow",
          "score": "0.14957",
          "x": 499,
          "y": 258
        },
```


Рисунок 3.18 Результати роботи сервісу

Таблиця 3.4

Номер зображення	Результат моделі	Реальний результат
1	1	Правильно
2	1	Правильно
3	1	Правильно
4	0	Правильно
5	0	Правильно
6	0	Хибно
7	1	Правильно
8	0	Правильно
9	0	Правильно
10	0	Правильно
11	0	Хибно
12	0	Правильно
13	0	Правильно
14	1	Правильно
15	0	Хибно
16	0	Правильно
17	1	Правильно
18	0	Правильно
19	0	Правильно
20	0	Хибно

Робота моделі відображає результат інформаційно-аналітичної системи. 1 – означає, що аномалію виявлено, 0 – аномалії немає. Реальний результат показує чи правильно виявлено аномалію, чи ні. Як показано в таблиці 3.1 з 20 зображень правильно визначено 16.

Результат тестування показує, що програма здатна визначати аномалії та ефективна в застосуванні. Дану систему можна покращувати, збільшуючи набір даних анотацій поз людей.

ВИСНОВОК

Розроблена інформаційна-аналітична система відповідає меті кваліфікаційної роботи – на основі результатів роботи системи можна визначати аномальні пози на зображеннях та відповідним чином реагувати на них.

Дана інтелектуальна-система може бути застосована для камер відеоспостереження на вулицях міста та в приміщеннях громадського користування (торгові центри, музеї, приміщення залізничного та авто вокзалах тощо) в реальному часі. Таке програмне забезпечення може покращити безпеку в великих містах, де це є вкрай актуальним.

Завдяки великій кількості відкритих даних з камер відеоспостереження модель оцінювання аномалій можна постійно покращувати, розширивши дата сет та донавчити модель.

Модель, використана для навчання, реалізована на основі архітектури згорткової нейронної мережі. Аномалії в позі людини визначаються з використанням розрахунку евклідової відстані між точками. Дана інформаційно-аналітична система може виявляти аномалії з точністю 80 %. Найкращий результат роботи моделі складає 0.8042.

Запропонована технологія є ефективною для розв’язання подібного типу задач та завдяки тому, що вже існуючий набір даних про пози людини вдалось розширити (були використані як звичайні, так і аномальні дані), дає точніші результати у порівнянні з іншими моделями.

ЛІТЕРАТУРА

[1] - A. Agarwal and B. Triggs. Recovering 3d human pose from monocular images. IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(1):44–58, 2006.

[2] - M. Andriluka, S. Roth, and B. Schiele. Monocular 3d pose estimation and tracking by detection. In IEEE Conference on Computer Vision and Pattern Recognition, 2010.

[3] - L. Sigal, R. Memisevic, and D. J. Fleet. Shared kernel information embedding for discriminative inference. In IEEE Conference on Computer Vision and Pattern Recognition, 2009.

[4] - Yao-Jen Chang, Shu-Fang Chen, and Jun-Da Huang. A kinect-based system for physical rehabilitation: A pilot study for young adults with motor disabilities. Research in developmental disabilities, 32(6):2566–2570, 2011.

[5] - Sigal L. Human pose estimation / Leonid Sigal - https://www.cs.ubc.ca/~lsigal/Publications/SigalEncyclopediaCVdraft.pdf.

[6] - Kenny Chen, Paolo Gabriel, Abdulwahab Alasfour, Chenghao Gong, Werner K Doyle, Orrin Devinsky, Daniel Friedman, Patricia Dugan, Lucia Melloni, and Thomas Thesen. Patient-specific pose estimation in clinical environments. IEEE Journal of Translational Engineering in Health and Medicine, 6:1–11, 2018.

[7] - Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. Journal of machine learning research, 9(Nov):2579–2605, 2008.

[8] - Yunchen Pu, Zhe Gan, Ricardo Henao, Xin Yuan, Chunyuan Li, Andrew Stevens, and Lawrence Carin. Variational autoencoder for deep learning of images, labels and captions. In Advances in neural information processing systems, pages 2352–2360, 2016.

[9] - Daniel Im Jiwoong Im, Sungjin Ahn, Roland Memisevic, and Yoshua Bengio. Denoising criterion for variational auto-encoding framework. In Thirty-First AAAI Conference on Artificial Intelligence, 2017.

[10] - J. Redmon and A. Farhadi, “Yolo9000: better, faster, stronger,” in CVPR, 2017.

[11] - S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards realtime object detection with region proposal networks,” in NIPS, 2015.

[12] - X. Nie, J. Feng, J. Xing, and S. Yan, “Pose partition networks for multi-person pose estimation,” in ECCV, 2018.

[13] - T. Simon, H. Joo, I. Matthews, and Y. Sheikh, “Hand keypoint detection in single images using multiview bootstrapping,” in CVPR, 2017.

[14] - K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in ICLR, 2015.

[15] - J. Redmon and A. Farhadi, “Yolo9000: better, faster, stronger,” in CVPR, 2017.

[16] – M. Andriluka, U. Iqbal, A. Milan, E. Insafutdinov, L. Pishchulin, J. Gall, and B. Schiele. Posetrack: A bench- mark for human pose estimation and tracking. In CVPR, pages 5167–5176, 2018

[17] – M.Andriluka,L.Pishchulin,P.V.Gehler,andB.Schiele.2d human pose estimation: New benchmark and state of the art analysis. In CVPR, pages 3686–3693, 2014

[18] – L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Trans. Pattern Anal. Mach. Intell., 40(4):834– 848, 2018.

[19] – A. Doering, U. Iqbal, and J. Gall. Joint flow: Temporal flow fields for multi person tracking, 2018.

[20] – X. Fan, K. Zheng, Y. Lin, and S. Wang. Combining local appearance and holistic view: Dual-source deep neural net- works for human pose estimation. In CVPR, pages 1347– 1355, 2015.

[21] – K. He, G. Gkioxari, P. Dolla´r, and R. B. Girshick. Mask R-CNN. In ICCV, pages 2980–2988, 2017.

[22] – G.Ning,Z.Zhang,andZ.He.Knowledge-guideddeepfrac- tal neural networks for human pose estimation. IEEE Trans. Multimedia, 20(5):1246–1259, 2018.

[23] – F. Xia, P. Wang, X. Chen, and A. L. Yuille. Joint multi- person pose estimation and semantic part segmentation. In CVPR, pages 6080–6089, 2017.

[24] – L. Zhao, M. Li, D. Meng, X. Li, Z. Zhang, Y. Zhuang, Z. Tu, and J. Wang. Deep convolutional neural networks with merge-and-run mappings. In IJCAI, pages 3170–3176, 2018.

[25] – F. Xia, P. Wang, X. Chen, and A. L. Yuille. Joint multi- person pose estimation and semantic part segmentation. In CVPR, pages 6080–6089, 2017.

[26] – X. Zhu, Y. Jiang, and Z. Luo. Multi-person pose estimation for posetrack with enhanced part affinity fields. In ICCV PoseTrack Workshop, 2017.

[27] – T. Zhang, G. Qi, B. Xiao, and J. Wang. Interleaved group convolutions. In ICCV, pages 4383–4392, 2017.

[28] – L. Zhao, M. Li, D. Meng, X. Li, Z. Zhang, Y. Zhuang, Z. Tu, and J. Wang. Deep convolutional neural networks with merge-and-run mappings. In IJCAI, pages 3170–3176, 2018.

[29] – Xiao B. Deep High-Resolution Representation Learning for Human Pose Estimation [Електронний ресурс] / Bin Xiao. – 2019. – Режим доступу до ресурсу: https://arxiv.org/pdf/1902.09212.pdf.

[30] – Human Pose Estimation Model HRNet Breaks Three COCO Records; CVPR Accepts Paper. medium.com. URL: https://medium.com/syncedreview/human-pose-estimation-model-hrnet-breaks-three-coco-records-cvpr-accepts-paper-74e57fabdeb6.

[31] – Human Pose Estimation: Deep Learning Approach [2022 Guide]. V7 - AI Data Platform for Computer Vision Annotation. URL: https://www.v7labs.com/blog/human-pose-estimation-guide

[32] – Human Pose Estimation Model HRNet Breaks Three COCO Records; CVPR Accepts Paper. medium.com. URL: https://medium.com/syncedreview/human-pose-estimation-model-hrnet-breaks-three-coco-records-cvpr-accepts-paper-74e57fabdeb6.

[33] – Human Pose Estimation Using Machine Learning in Python. Analytics Vidhya. URL: https://www.analyticsvidhya.com/blog/2021/10/human-pose-estimation-using-machine-learning-in-python/(date of access: 24.05.2022).

[34] – C. Yuan, S. Li, H. Cai, Vision-based excavator detection and tracking using hybrid kinematic shapes and key nodes, J. Comput. Civ. Eng. 31 (1) (2017), 04016038, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000602.

[35] – F. Vahdatikhaki, A. Hammad, H. Siddiqui, Optimization-based excavator pose estimation using real-time location systems, Autom. Constr. 56 (2015) 76–92, https://doi.org/10.1016/j.autcon.2015.03.006.

[36] – J. Seo, S. Han, S. Lee, H. Kim, Computer vision techniques for construction safety and health monitoring, Adv. Eng. Inform. 29 (2) (2015) 239–251, https://doi.org/10.1016/j.aei.2015.02.001

[37] – H. Guo, Y. Yu, M. Skitmore, Visualization technology-based construction safety management: a review, Autom. Constr. 73 (2017) 135–144, https://doi.org/10.1016/j.autcon.2016.10.004.

[38] – H. Guo, Y. Yu, M. Skitmore, Visualization technology-based construction safety management: a review, Autom. Constr. 73 (2017) 135–144, https://doi.org/10.1016/j.autcon.2016.10.004.

[39] – R.J. Sandzimier, H.H. Asada, A data-driven approach to prediction and optimal bucket-filling control for autonomous excavators, IEEE Robotics Automation Letters 5 (2) (2020) 2682–2689, https://doi.org/10.1109/LRA.2020.2969944.

[40] – A. Hammad, F. Vahdatikhaki, C. Zhang, A novel integrated approach to project-level automated machine control/guidance systems in construction projects, J. Inform. Technol. Construction 18 (9) (2013) 162–181. https://www.itcon.org/p aper/2013/9.

[41] – M.M. Soltani, Z. Zhu, A. Hammad, Skeleton estimation of excavator by detecting its parts, Autom. Constr. 82 (2017) 1–15, https://doi.org/10.1109/TPAMI.2013.248.

[42] – M.M. Soltani, Excavator Pose Estimation for Safety Monitoring by Fusing Computer Vision and RTLS Data, Building Engineering, Concordia University, PhD, 2017. https://spectrum.library.concordia.ca/983390/.

[43] – S. Li, A.B. Chan, 3D Human Pose Estimation from Monocular Images with Deep Convolutional Neural Network, Springer, Asian Conference on Computer Vision, 2014, pp. 332–347, https://doi.org/10.1007/978-3-319-16808-1_23

[44] – V. Mazzia , S. Angarano , F. Salvetti , F. Angelini , M. Chiaberge , Action trans-former: a self-attention model for short-time pose-based human action recog- nition, Pattern Recognit. 124 (2022) 108487 .

[45] – H. Zhao , G. Yang , D. Wang , H. Lu , Deep mutual learning for visual object track- ing, Pattern Recognit. 112 (2021) 107796 .

[46] – Consul Agent Configuration Reference | Consul by HashiCorp. *Consul by HashiCorp*. URL: https://www.consul.io(date of access: 24.05.2022).

[47] – Getting started with Redis. *Redis*. URL: https://redis.io

[48] – J. Wang , X. Long , Y. Gao , E. Ding , S. Wen , Graph-PCNN: two stage human pose estimation with graph pose refinement, in: 2020 European Conference on Computer Vision (ECCV), vol. 12356, 2020, pp. 492–508 .

[49] – H. Zhao , G. Yang , D. Wang , H. Lu , Deep mutual learning for visual object track- ing, Pattern Recognit. 112 (2021) 107796 .

[50] – J. Wang , K. Sun , T. Cheng , B. Jiang , C. Deng , Y. Zhao , D. Liu , Y. Mu , M. Tan , X. Wang , W. Liu , B. Xiao , Deep high-resolution representation learning for visual recognition, IEEE Trans. Pattern Anal. Mach. Intell. 43 (10) (2021) 3349–3364.

[51] – Z. Cao , G. Hidalgo , T. Simon , S.-E. Wei , Y. Sheikh , Openpose: realtime multi- -person 2D pose estimation using part affinity fields, IEEE Trans. Pattern Anal. Mach. Intell. 43 (1) (2021) 172–186 .

[52] – Pillow. Pillow – Pillow (PIL Fork) 9.1.1 documentation. URL: https://pillow.readthedocs.io.

[53] – GitHub - numpy/numpy: The fundamental package for scientific computing with Python. GitHub. URL: https://github.com/numpy.

[54] – OpenCV – Overview- GeeksforGeeks. GeeksforGeeks.URL: https://www.geeksforgeeks.org/opencv-overview/.

[55] – Introduction to PyTorch for Deep Learning - KDnuggets. KDnuggets. URL: http://kdnuggets.com.

ДОДАТКИ

Users > home > Downloads > defines.py > ...

```python
import os

###
#   COCO 1.0 defines
#
CATEGORIES_COCO_KEY = "categories"
ANNOTATIONS_COCO_KEY = "annotations"
ATTRIBUTES_COCO_KEY = "attributes"
ATTRIBUTES_ID_COCO_KEY = "Id"
ID_COCO_KEY = "id"
NAME_COCO_KEY = "name"
BBOX_COCO_KEY = "bbox"
CATEGORY_ID_COCO_KEY = "category_id"
IMAGE_ID_COCO_KEY = "image_id"
TYPE_COCO_KEY = "Type"

###
#   DEFAULT defines
#
DEFAULT_TYPE_NAME = "DEFAULT"
ILLEGAL_VAL = -1
VIDEO_FILE_EXT = '.mp4'

###
#   AWS defines
#


###
#   Dataset Creation
#
DATASET_MIN_TO_SIZE_RELATION = 2.0
DATASET_INTSANCE_SKIP = 6
DATASET_MAX_INSTANCES_MIN_LIMIT = 3
```

Users > home > Downloads > anomaly_detection.py > ...

```python
import logging
import argparse
import json
import os
import re
import tempfile
import shutil
import zipfile
# local imports
from cvat_utils import *
from video_utils import *
from utils import *



def main(in_txt_path, video_path, out_zip_path) :
    data = {}
    frame_num = 0
    # build pattern
    pattern_frame      = re.compile("Frame \#\:  (\d+)")
    pattern_detection = re.compile("Tracker ID: (\d+), Class: (\w+),  BBox Coords \(xmin, ymin, xmax, ymax\): \((\d+), (\d+), (\d+), (\d+)\)")
    # start
    with open(in_txt_path) as file :
        for line in file :
            # frame num line
            res = pattern_frame.match(line)
            if res is not None :
                frame_num = int(res.group(1))
                if frame_num not in data.keys() :
                    data[frame_num] = {}
                else :
                    logging.error(f"Detection results for frame {frame_num} exists twice...")
                continue
            # detection line
            res = pattern_detection.match(line)
            if res is not None :
                oid = int(res.group(1))
                ocategory = res.group(2)
                xmin = int(res.group(3))
                ymin = int(res.group(4))
                xmax = int(res.group(5))
                ymax = int(res.group(6))
                # compose data (t, l, w, h)
                data[frame_num][oid] = (ocategory, ymin, xmin, xmax - xmin - 1, ymax - ymin - 1)
                continue
    if len(data) == 0 :
        logging.error(f"Failed to collect any data from {in_txt_path}")
    else :
        (task_name, _) = get_filename_extention_from_file(out_zip_path)
        logging.info(f">>> finished data pasring from {in_txt_path}")
        # get video meta
        (w, h, _, max_frame) = get_video_meta(video_path)
        #
        (task_json, annotation_json) = CVAT_create_task(task_name, data, max_frame)
        # open tmp dir for save tmp files
        with tempfile.TemporaryDirectory() as temp_dir :
            logging.info(f"Going to save in tmp dir: {temp_dir}")
            # dump jsons
            with open(temp_dir + "/task.json", 'w') as f :
                json.dump(task_json, f, indent=4)
            with open(temp_dir + "/annotations.json", 'w') as f :
                json.dump(annotation_json, f, indent=4)
            # open dir for video
            data_path = temp_dir + "/data"
            os.mkdir(data_path)
            shutil.copy(video_path, data_path)
            # zip data
            with zipfile.ZipFile(out_zip_path, 'w') as zipo :
                for dirpath, dirnames, filenames in os.walk(temp_dir):
                    for filename in filenames :
                        file_path = os.path.join(dirpath, filename)
                        archive_file_path = os.path.relpath(file_path, temp_dir)
                        zipo.write(file_path, archive_file_path)
             # Report
            logging.info(f">>> successfully finished task build  {out_zip_path}")

```

```
if __name__ == "__main__":
    # logging base config
    logging.basicConfig(level=logging.DEBUG)
    # argparser
    parser = argparse.ArgumentParser(description='Enriches body detections (AWS format) to missing frames & converts to COCO 1.0')
    parser.add_argument('--i_txt', help='path to the input annotation txt file')
    parser.add_argument('--i_vid', help='path to the input video file')
    parser.add_argument('--o_zip', help='path to the output annotation task path')
    args = parser.parse_args()
    # lets do the work
    main(args.i_txt, args.i_vid, args.o_zip)
```

```
# method for getting annotations from detector
def Detector():
    pass


# convert bboxes annotation from [x, y, w, h] to [x1, y1, x2, y2]
def convert_bbox(bboxes):
    converted_bboxes = []
    for x in bboxes:
        y = x.copy()
        y[0] = x[0] - x[2] / 2  # top left x
        y[1] = x[1] - x[3] / 2  # top left y
        y[2] = x[0] + x[2] / 2  # bottom right x
        y[3] = x[1] + x[3] / 2  # bottom right y
        converted_bboxes.append(y)

    return converted_bboxes


def calculate_iou(boxA, boxB):
    # determine the (x, y)-coordinates of the intersection rectangle
    xA = max(boxA[0], boxB[0])
    yA = max(boxA[1], boxB[1])
    xB = min(boxA[2], boxB[2])
    yB = min(boxA[3], boxB[3])

    # compute the area of intersection rectangle
    interArea = abs(max((xB - xA, 0)) * max((yB - yA), 0))
    if interArea == 0:
        return 0
    # compute the area of both the prediction and ground-truth ectangles
    boxAArea = abs((boxA[2] - boxA[0]) * (boxA[3] - boxA[1]))
    boxBArea = abs((boxB[2] - boxB[0]) * (boxB[3] - boxB[1]))

    # compute the intersection over union by taking the intersection
    # area and dividing it by the sum of prediction + ground-truth areas - the interesection area
    iou = interArea / float(boxAArea + boxBArea - interArea)

    return iou
```

```
152  def upload_to_aws(img, bucket):
153
154      img_name = os.path.basename('camera_id1.jpeg')
155      #client = boto3.client('s3', aws_access_key_id= ACCESS_KEY, aws_secret_access_key= SECRET_KEY)
156      #transfer = S3Transfer(client)
157      s3 = session.resource('s3')
158
159      try:
160          object = s3.Object(bucket, img_name +'.jpeg')
161          object.put(Body=img)
162          logging.info("Upload Successful")
163          return True
164      except FileNotFoundError:
165          logging.info("The file was not found")
166          return False
167      except NoCredentialsError:
168          logging.info("Credentials not available")
169          return False
170
171
172
173  def notifier(bboxes):
174      if bool(bboxes):
175          logging.debug("Anomaly were found!")
176          logging.debug(bboxes)
177      else:
178          logging.debug("There is no anomaly")
```