

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

Автоматизації і інформаційних технологій
(факультет)

Інформаційних технологій проектування та прикладної математики
(кафедра)

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО КВАЛІФІКАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ МАГІСТРА**

на тему: «Інформаційна система аналізу постави на основі методів штучного
інтелекту»

Йовенко Дмитро Сергійович
(прізвище, ім'я та по батькові магістра повністю)

Київ 2025 р

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

Автоматизації і інформаційних технологій
(факультет)

Інформаційних технологій проектування та прикладної математики
(кафедра)

ЗАТВЕРДЖУЮ

Завідувач кафедри ІТППМ

Бородавка Є. В.

« _____ » _____ 2025 року

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО КВАЛІФІКАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ МАГІСТРА**

на тему: «Інформаційна система аналізу постави на основі методів штучного
інтелекту»

Виконав: студент 2-го курсу, групи ІСТм-24

Спеціальності: 126 «Інформаційні системи і
технології»

(шифр і назва напрямку підготовки, спеціальності)

Магістрант: Йовенко Д. С.
(прізвище та ініціали)

Керівник: Поплавський О. А.
(прізвище та ініціали)

Рецензент: _____
(прізвище та ініціали)

Київ 2025 р

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БУДІВНИЦТВА І АРХІТЕКТУРИ

Факультет: Автоматизації та інформаційних технологій

Кафедра: Інформаційних технологій проектування та ПМ

Освітній ступінь: «магістр за ОПП»

Спеціальність: 126 «Інформаційні системи і технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри ІТППМ

Бородавка Є. В

« » 2025 року

ЗАВДАННЯ ДО ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ МАГІСТРА

Йовенко Дмитро Сергійович

1. Тема роботи: «Інформаційна система аналізу постави на основі методів штучного інтелекту»

затверджена наказом ректора КНУБА № 1619/23/25 від « 29 » вересня 2025 року

2. Керівник роботи: Поплавський О. А.

3. Строк подання студентом роботи до захисту: 18.12.2025

4. Зміст пояснювальної записки розділами:

Р.1. Аналіз предметної області та постановка задачі

Р.2. Математичне забезпечення

Р.3. Інформаційне забезпечення

Р.4. Навчання згорткових нейронних мереж

Р.5. Проектування та реалізація системи

5. Інформаційні слайди:

С.1. _____

С.2. _____

С.3. _____

С.4. _____

С.5. _____

С.6.

С.7.

6. Календарний план виконання кваліфікаційної випускної роботи:

Види робіт та їх зміст	Дата виконання
Розділ 1	Вересень 2025
Розділ 2	Жовтень 2025
Розділ 3	Жовтень 2025
Розділ 4	Листопад 2025
Розділ 5	Листопад 2025
Остаточне оформлення роботи	Грудень 2025
Направлення роботи для перевірки на плагіат	Грудень 2025
Попередній захист роботи на випусковій кафедрі	Грудень 2025
Направлення роботи на рецензування	Грудень 2025

7. Консультанти розділів кваліфікаційної випускної роботи

Розділ	Прізвище, ініціали та посада консультанта	Перевірив	
		дата	підпис
Розділ 1	Поплавський О. А.	30.09.2025	
Розділ 2	Поплавський О. А.	20.10.2025	
Розділ 3	Поплавський О. А.	3.11.2025	
Розділ 4	Поплавський О. А.	14.11.2025	
Розділ 5	Поплавський О. А.	28.11.2025	

8. Дата видачі завдання: 29.09.2025

Керівник

(підпис)

Поплавський О.А

(прізвище на ініціали)

Магістрант

(підпис)

Йовенко Д. С.

(прізвище на ініціали)

РЕЗЮМЕ

Київський національний університет будівництва і архітектури

Йовенко Дмитро Сергійович

Факультет автоматизації і інформаційних технологій,

група ІСТм-24

Тема атестаційної випускної роботи:

«Інформаційна система аналізу постави на основі методів штучного інтелекту»

освітній рівень: магістр,

спеціальність: 126 «Інформаційні системи і технології»,

Науковий керівник: Поплавський О. А.

Обсяг роботи. Атестаційна випускова робота магістра складається з 5 розділів, 139 сторінок, 62 рисунків, завдання, анотації, вступу, висновків та списку використаних джерел.

Актуальність теми. Робота присвячена розробці інформаційної системи для аналізу постави людини з використанням методів штучного інтелекту. Актуальність теми зумовлена значною поширеністю порушень опорно-рухового апарату та необхідністю вдосконалення методів їхньої діагностики. Сучасні клінічні та інструментальні методи оцінки постави часто є складними або вимагають значних часових витрат. У відповідь на потребу пришвидшення процесу діагностики постави, розробка інформаційної системи для аналізу постави людини на базі методів штучного інтелекту є доречною. Система на основі штучного інтелекту пропонує швидке, точне та доступне рішення для пришвидшення процесу виявлення відхилень постави.

У вступі обґрунтовано актуальність теми, сформульовано мету та основні завдання роботи, визначено об'єкт і предмет дослідження, а також окреслено наукову новизну та практичну значущість отриманих результатів.

У першому розділі «Аналіз предметної області та постановка задачі» розглянуто опис опорно-рухового апарату, аналіз об'єкта дослідження — пос-

тави людини, класифікацію порушень постави та причини їх виникнення, а також визначено цілі роботи та сформульовано задачу.

У другому розділі «Математичне забезпечення» описано постановку задачі машинного навчання, розглянуто метод градієнтного спуску та його оптимізатори, а також представлено функції втрат і функції активації, необхідні для навчання нейронних мереж.

У третьому розділі «Інформаційне забезпечення» детально описано згорткові нейронні мережі, метод зворотнього поширення помилки, задачу знаходження ключових точок на зображеннях та проаналізовано відповідні методи, зокрема Yolo-pose та Keypoint R-CNN, а також метрики для оцінювання моделей.

У четвертому розділі «Навчання згорткових нейронних мереж» викладено етапи збору, анотації та аугментації навчальних даних, а також представлено процес та результати навчання моделей Yolo-pose та Keypoint R-CNN для визначення ключових точок постави.

У п'ятому розділі «Проектування та реалізація системи» запропоновано архітектуру та концепцію побудови клієнт-серверної інформаційної системи, детально описано проектування та реалізацію її модулів, зокрема модуля аналізу постави, та використані технології (FastAPI, PostgreSQL, Vue.js), а також наведено тестовий приклад.

Ключові слова: штучний інтелект, машинне навчання, згорткові нейронні мережі, Yolo-pose, Keypoint R-CNN, постава, опорно-руховий апарат, ключові точки, FastAPI, PostgreSQL, Vue.js.

Keywords: artificial intelligence, machine learning, convolutional neural networks, Yolo-pose, Keypoint R-CNN, posture, musculoskeletal system, key points, FastAPI, PostgreSQL, Vue.js.

Якість оформлення проекту. Атестаційна випускна робота магістра оформлена у відповідності до діючих нормативних документів та методичних вказівок до виконання дипломних робіт для студентів спеціальності 126 «Інформаційні системи і технології».

Загальний висновок стосовно роботи та присвоєння авторіві освітньо-кваліфікаційного рівня «магістр». Робота виконана на високому рівні, студент продемонстрував високий рівень теоретичної підготовки та сформованих практичних навичок в області сучасних інформаційних технологій. Заслуговує оцінки «відмінно».

Науковий керівник: _____ **Поплавський О. А.**
(підпис) (прізвище та ініціали)

Посада, місце роботи: _____

« _____ » _____ 2025 р.

АНОТАЦІЯ

Йовенко Д. С. «Інформаційна система аналізу постави на основі методів штучного інтелекту»

Кваліфікаційна випускна робота магістра за спеціальністю: 126 «Інформаційні системи і технології». – Київський національний університет будівництва та архітектури. – Київ, 2025.

Робота присвячена розробці інформаційної системи для автоматизованого аналізу постави людини з використанням методів штучного інтелекту. Мета полягає в автоматизації діагностики шляхом виявлення ключових точок на тілі людини за допомогою моделей на основі згорткових нейронних мереж, зокрема Yolo-pose та Keypoint R-CNN. Розроблений застосунок дозволяє підвищити ефективність та доступність діагностики.

Ключові слова: штучний інтелект , згорткові нейронні мережі, Yolo-pose, Keypoint R-CNN, постава, ключові точки, FastAPI, PostgreSQL, Vue.js.

SUMMARY

Yovenko D. S. “Information system for posture analysis based on artificial intelligence methods”

Master’s degree final thesis in the specialty: 126 “Information systems and technologies”. – Kyiv National University of Construction and Architecture. – Kyiv, 2025.

The work is devoted to the development of an information system for automated analysis of human posture using artificial intelligence methods. The goal is to automate diagnostics by detecting keypoints on the human body using models based on convolutional neural networks, in particular Yolo-pose and Keypoint R-CNN. The application was developed to increase the efficiency and accessibility of diagnostics.

Keywords: artificial intelligence, convolutional neural networks, Yolo-pose, Keypoint R-CNN, posture, keypoints, FastAPI, PostgreSQL, Vue.js.

Зміст

Вступ.....	11
Розділ 1. Аналіз предметної області та постановка задачі.....	13
1.1 Опис предметної області	13
1.2 Аналіз об'єкта дослідження	17
1.2.1 Постава людини.....	17
1.2.2 Класифікація порушень постави та причини їх виникнення.....	19
1.2.3 Способи діагностики постави	22
1.3 Опис предмету дослідження	25
1.3.1 Поняття штучного інтелекту.....	25
1.3.2 Штучні нейронні мережі	27
1.3.2 Навчання штучного інтелекту.....	29
1.4 Аналіз актуальності.....	30
1.5 Аналіз вже існуючих рішень.....	32
1.6 Визначення цілей роботи та постановка задачі	35
Розділ 2. Математичне забезпечення	38
2.1 Постановка задачі машинного навчання	38
2.2 Метод градієнтного спуску	40
2.3 Оптимізатори методу градієнтного спуску	44
2.4 Функції втрат	47
2.5 Функції активації.....	51
Розділ 3. Інформаційне забезпечення.....	55
3.1 Згорткові нейронні мережі	55
3.1.1 Згорткові шари.....	58
3.1.2 Агрегувальні шари	60
3.1.3 Повнозв'язні шари.....	62
3.2 Метод зворотнього поширення помилки.....	64
3.3 Задача знаходження ключових точок на зображеннях	66
3.4 Методи знаходження ключових точок на зображеннях.....	69
3.4.1 YOLO v11	71
3.4.2 Keypoint R-CNN.....	73
3.5 Метрики для оцінювання моделей	76

3.5.1 Метрики для класифікації	76
3.5.2 Метрики для детекції об'єктів	77
3.5.3 Метрики для моделей визначення ключових точок	78
3.6 Підготовка та обробка даних для навчання.....	80
3.6.1. Збір та формування набору даних	81
3.6.2. Анотація даних	81
3.6.3. Попередня обробка та аугментація даних	83
Розділ 4. Навчання згорткових нейронних мереж	85
4.1 Збір навчальних даних	85
4.2 Анотація даних	87
4.3 Підготовка та аугментація даних.....	90
4.4 Навчання моделей нейронних мереж.....	92
4.4.1 Навчання моделі Yolo-pose	93
4.4.2 Навчання Keypoint-RCNN	99
Розділ 5. Проектування та реалізація системи	104
5.1 Архітектура та концепція побудови системи.....	104
5.2 Проектування модулів системи	106
5.3 Опис модуля аналізу постави.....	110
5.4 Використані технології.....	117
5.5 Реалізація програмних модулів.....	119
5.5.1 Серверна частина.....	119
5.5.2 Клієнтська частина.....	123
5.6 Тестовий приклад	125
Висновок.....	135
Список використаних джерел	136

Вступ

Розвиток інформаційних технологій та систем штучного інтелекту (ШІ) відкриває нові, перспективні можливості у сфері медичної діагностики, зокрема захворювань опорно-рухового апарату. Актуальність теми роботи зумовлена значною поширеністю порушень постави, які охоплюють понад 150 різних захворювань і станів. Як зазначається у дослідженнях, ці стани характеризуються хронічним больовим синдромом та суттєвими обмеженнями рухливості, що критично впливає на якість життя пацієнтів та їхню працездатність.

Сучасні методи медичної діагностики опорно-рухового апарату, хоча й ефективні, часто є трудомісткими, вимагають часу та залучення висококваліфікованих спеціалістів і дорогого обладнання. Така ситуація вимагає вдосконалення підходів для забезпечення швидшого, точнішого та доступнішого аналізу, що є надзвичайно важливим для профілактики та забезпечення своєчасного лікування, особливо на ранніх стадіях. У цьому контексті застосування методів штучного інтелекту відкриває нові можливості для автоматизації та об'єктивізації процесу обстеження.

Мета роботи полягає у розробці інформаційної системи для аналізу постави людини з використанням методів штучного інтелекту для автоматизації та спрощення оцінки стану опорно-рухового апарату, а також виявлення відхилень у поставі.

Об'єктом дослідження є постава людини як інтегральний показник функціонування опорно-рухового апарату та процес діагностики захворювань, сфокусований на оцінці постави людини.

Предметом дослідження є методи та моделі штучного інтелекту, зокрема згорткові нейронні мережі, для автоматизованого визначення ключових точок на зображеннях з метою аналізу постави.

Методи дослідження. Для досягнення поставленої мети використовувалися методи математичного моделювання, алгоритмічний аналіз та машинне навчання, зокрема застосування згорткових нейронних мереж. Основні проектні рішення

роботи полягають у використанні цих методів для обробки зображень та інтеграція у інформаційну систему.

Практична значущість результатів роботи полягає в оптимізації процесу діагностики, спрощенні роботи медичних фахівців та підвищенні доступності раннього виявлення відхилень постави, що є важливим для профілактики та лікування численних захворювань.

Структура роботи. Пояснювальна записка складається зі вступу, п'яти розділів, висновків, списку використаних джерел та додатків. Розділ 1 містить аналіз предметної області (опорно-руховий апарат, постава, діагностика) та постановку задачі. Розділ 2 описує математичне забезпечення системи (машинне навчання, градієнтний спуск, функції втрат і активації). Розділ 3 деталізує інформаційне забезпечення (згорткові нейронні мережі, методи знаходження ключових точок та метрики оцінювання). Розділ 4 присвячений етапам навчання згорткових нейронних мереж (збір, анотація, розширення даних та результати навчання моделей Yolo-pose і Keypoint-RCNN). Розділ 5 охоплює проектування, архітектуру, реалізацію та тестовий приклад

Таким чином, дане магістерське дослідження має на меті створення високоефективної інформаційної системи на базі методів штучного інтелекту, зокрема згорткових нейронних мереж, для автоматизованої та точної діагностики порушень постави людини.

Розділ 1. Аналіз предметної області та постановка задачі

1.1 Опис предметної області

Ця освітньо-кваліфікаційна робота присвячена розробці інформаційної системи, призначеної для аналізу постави людини з використанням методів штучного інтелекту. Основна мета проекту полягає в автоматизації та спрощенні оцінки стану опорно-рухового апарату, зокрема виявлення відхилень у поставі, що є критично важливим для профілактики та ранньої діагностики численних захворювань. Впровадження такої системи має на меті значно оптимізувати процес обстежень, перетворюючи його з трудомісткої та довгої процедури на швидкий та ефективний аналіз, що займає секунди замість хвилин. Це рішення покликане спростити роботу медичних фахівців та покращити доступність діагностики.

Опорно-руховий апарат це складною системою, що формує каркас тіла та забезпечує його рухливість. Він складається з двох основних компонентів: пасивної частини – скелета, та активної – м'язів[5]. На рисунку 1.1 зображено опорно-рухову систему людини.

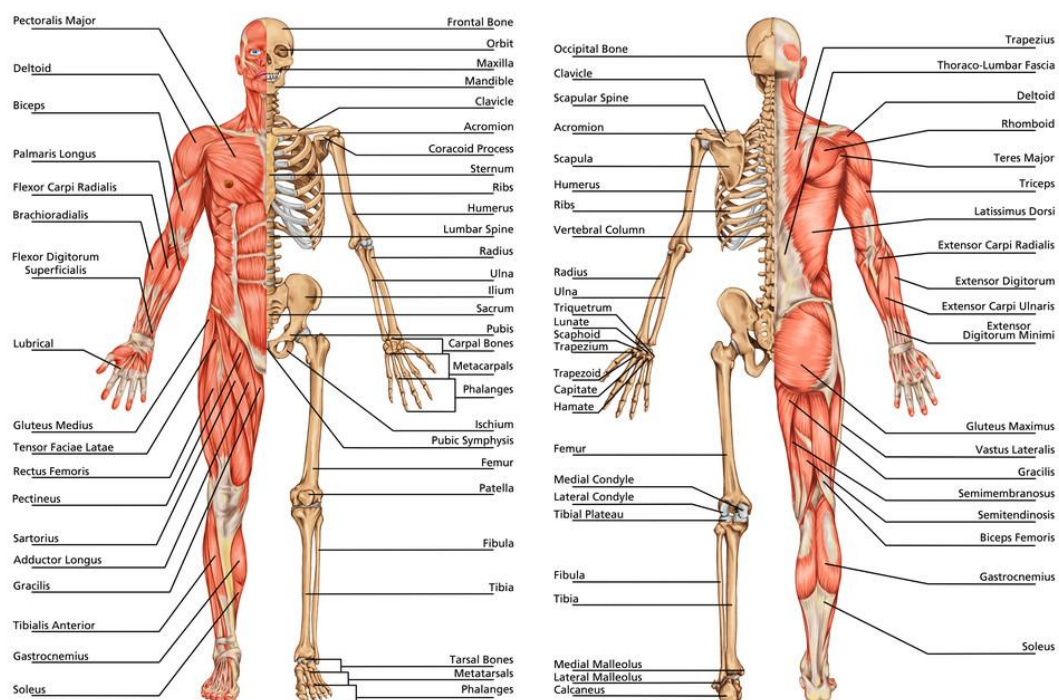


Рисунок 1.1 – Опорно рухова система

Кістки скелета виступають у ролі важелів, що приводяться в рух м'язами, забезпечуючи зміну положення частин тіла одна відносно одної, переміщення тіла у просторі та виконання опорної функції. Важливо зазначити, що опорно-руховий апарат також виконує анти гравітаційну функцію, оскільки людина постійно взаємодіє із силами земного тяжіння.

Обидві складові, скелет і м'язи, мають спільне ембріональне походження з мезодерми, що підкреслює їх тісний анатомічний та функціональний зв'язок.

Скелет є комплексом кісток, хрящів, зв'язок та інших щільних тканин, що формують механічну опору тіла людини. Як пасивна частина рухового апарату, кістки, з'єднуючись, утворюють важелі, що приводяться в рух м'язами. Людський скелет налічує стоні кісток, кожна з яких є живим організмом, що складається з різних типів тканин, кісткового мозку, кровоносних судин та нервів.

За формою кістки поділяються на довгі, короткі, плоскі та мішані. Довгі кістки (наприклад, кістки кінцівок) мають трубчасту будову з порожниною, заповненою жовтим кістковим мозком. Короткі кістки мають приблизно однакові розміри, плоскі – значну довжину та ширину при малій товщині, а мішані – поєднують елементи коротких та плоских кісток[13].

М'язи забезпечують рух кісток, скорочуючись та переміщуючи їх у суглобах. Скелетні м'язи, що кріпляться до кісток, розташовані антагоністичними групами навколо суглобів. Їхня робота контролюється нервовою системою, яка посиляє електричні сигнали, що спричиняють скорочення м'язів. В організмі людини є три види м'язової тканини (рис. 1.2), кожен з яких виконує свої унікальні функції:

- **Скелетна м'язова тканина** – відповідає за рух тіла, прикріплена до кісток і контролюється свідомо.
- **Гладка м'язова тканина** – утворює стінки внутрішніх органів і скорочується мимовільно.
- **Серцева м'язова тканина** – формує стінки серця і відповідає за його скорочення, що перекачує кров по всьому тілу. Вона також функціонує мимовільно[16].

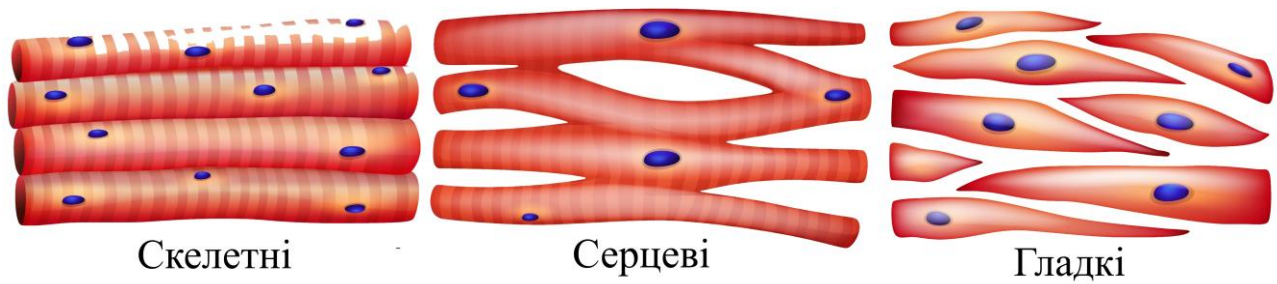


Рисунок 1.2 – Типи м'язової тканини

У контексті опорно-рухового апарату розглядаються скелетні та гладкі м'язи. Скелетна мускулатура забезпечує довільні рухи тіла, дозволяючи виконувати різноманітні дії – від повсякденної ходьби до комплексних фізичних навантажень. Натомість, гладка мускулатура, що є складовою внутрішніх органів і кровоносних судин, відповідає за їхню життєдіяльність, регулюючи процеси травлення та кровообігу. Важливо зазначити, що діяльність як серцевих, так і гладких м'язів відбувається незалежно від свідомого контролю людини. Серцевий м'яз, розташований у серці, виконує ключову функцію з ритмічного перекачування крові по всьому організму.

Під час скорочення м'язи стягуються або стискаються, впливаючи на прикріплені до них кістки. Ця дія призводить до виникнення руху в суглобах. Скелет, у свою чергу, виконує роль механічної опори, що дозволяє м'язам ефективно здійснювати тягові та стискаючі рухи, забезпечуючи необхідне переміщення.

З вище сказаного, очевидно, що опорно-руховий апарат, що є складною і взаємопов'язаною системою. Він виконує низку життєво важливих функцій, які забезпечують не тільки рух, а й загальну цілісність та життєдіяльність організму. Функціональне призначення опорно-рухового апарату є комплексним і розпочинається з опорної складової. Скелет формує міцний кістковий каркас, який не лише підтримує вагу тіла та визначає його форму, але й слугує основою для кріплення м'язів, що фіксують положення корпусу, забезпечуючи стабільність і рівновагу. На базі цієї статичної основи реалізується рухова функція: завдяки злагодженій взаємодії кісток, з'єднаних суглобами, та скороченню прикріплених

до них м'язів, людина отримує здатність переміщуватися у просторі та виконувати різноманітні дії.

Паралельно з цим система гарантує надійний захист життєво важливих органів, утворюючи довкола них кісткові порожнини: череп оберігає головний мозок, хребет — спинний, грудна клітка захищає серце та легені, а кістки таза разом із м'язами живота убезпечують органи черевної порожнини та розмноження. Окрім механічних завдань, кістки скелета активно залучені до метаболічних процесів, виступаючи депо для кальцію та фосфору, необхідних для фізіологічної діяльності організму. Водночас червоний кістковий мозок відіграє визначальну роль у кровотворенні, забезпечуючи постійне оновлення клітин крові, тоді як нервові закінчення у м'язах перетворюють їх на своєрідні органи чуття, що інформують нервову систему про положення тіла[25].

Скоординована діяльність м'язів та скелета важлива для підтримання правильної постави та загальної стабільності тіла. Зокрема, м'язи спини та черевного пресу, працюючи разом зі скелетом, забезпечують адекватну підтримку хребта, що допомагає уникнути деформацій та порушень. Ця синергія має вирішальне значення для профілактики травм та підтримки загального стану здоров'я.

Отже, взаємодія між м'язовою системою та скелетом визначає не тільки рухові можливості людини, але й її загальний стан здоров'я та функціонування організму в цілому. Важливість рухової активності у життєдіяльності людини неможливо переоцінити. Фізична активність не лише сприяє посиленню функцій опорно-рухового апарату, але й активізує обмін речовин та підвищує стійкість організму до різноманітних захворювань.

Таким чином, детальний аналіз предметної області виявив глибокий взаємозв'язок між станом опорно-рухового апарату, руховою активністю та загальним здоров'ям людини.

1.2 Аналіз об'єкта дослідження

1.2.1 Постава людини

Постава це звичне, невимушеним положенням тіла людини у стані стоячи та під час руху, яке формується в процесі фізичного розвитку та становлення статико-динамічних функцій. Вона є комплексним показником здоров'я опорно-рухового апарату та всього організму[3].

Порушення функціонування опорно-рухового апарату охоплюють понад 150 різних захворювань та станів, що характеризуються ураженнями цих структур і призводять до тимчасових або довічних обмежень у функціонуванні та участі у повсякденній діяльності. Такі стани, як правило, проявляються болем (часто хронічним) та обмеженнями рухливості й спритності, що суттєво знижує здатність людей працювати та брати участь у житті суспільства. Біль у структурах опорно-рухового апарату є найпоширенішою формою неонкологічного болю[17].

Проблеми з опорно-руховим апаратом актуальні протягом усього життя – від дитинства до похилого віку. Вони можуть виникати раптово і бути короткочасними наприклад, переломи, розтягнення, що супроводжуються болем та обмеженням функцій, або ж мати хронічний характер, як у випадку хронічного болю в попереку чи остеоартрозу.

Людський хребет функціонує як центральна опорна структура, що підтримує всю вагу тіла та забезпечує захист спинного мозку. Його анатомічна будова включає 33-34 хребці, які розділені міжхребцевими дисками, що складаються з волокнистої тканини. Ця складна та гнучка вісь має три природні вигини – шийний, грудний та поперековий, які сприяють оптимальному розподілу ваги та навантажень по всьому тілу. S-подібна форма хребта є ключовою для підтримання рівноваги та запобігання падінням[29].

Хребет традиційно поділяють на п'ять основних відділів(рис.1.3):

- шийний (7 хребців);
- грудний (12 хребців);
- поперековий (5 хребців);

- крижовий (5 хребців);
- куприковий (4-5 хребців).



Рисунок 1.3 – Структура хребта людини

Правильна постава характеризується оптимальним взаємним розташуванням частин тіла: пряма спина, розведені плечі, розвинена грудна клітина та піднята голова. При такому вирівнюванні внутрішні органи розташовані так, що створюються найкращі умови для їх ефективної роботи.

Неправильна постава часто стає причиною прогресуючих деформацій хребта, що з часом може спричинити значні ускладнення. Одним із таких наслідків є компресія нервових структур, що відходять від спинного мозку, розташованого у хребетному каналі. Це може проявлятися хронічним болем у ділянці спини, шиї та голови, а також сприяти розвитку дегенеративних захворювань.

1.2.2 Класифікація порушень постави та причини їх виникнення

Порушення постави являють собою стійкі відхилення від звичного, фізіологічно правильного положення тіла. У дітей такі порушення найчастіше асоціюються з проблемами розвитку опорно-рухового апарату та дисфункціями хребта, що відповідає за його опорну, захисну та амортизовану функції. У результаті цих порушень хребет втрачає свою здатність повноцінно виконувати зазначені функції, що, своєю чергою, призводить до розвитку системних захворювань організму у значної частини пацієнтів (від 69% до 82,5%).

Відхилення постави класифікують за трьома ступенями вираженості. До I ступеня відносять незначні зміни, які можуть бути усунені завдяки цілеспрямованій концентрації уваги дитини. II ступінь характеризується порушеннями, що коригуються після розвантаження хребта в горизонтальному положенні або при підвішуванні за пахвові западини. Натомість, III ступінь включає порушення, які залишаються незмінними навіть після розвантаження хребта.

Розрізняють декілька варіантів порушень постави, що відображають зміни у фізіологічних вигинах хребта[29]:

- "Сутулість": характеризується посиленням грудного кіфозу у верхніх відділах при згладженому поперековому лордозі.
- "Кругла спина": проявляється загальним збільшенням грудного кіфозу.
- "Увігнута спина": визначається збільшенням лордозу в поперековому відділі.
- "Кругло-увігнута спина": є поєднанням збільшеного грудного кіфозу та посиленого поперекового лордозу.
- "Плоска спина": характеризується згладжуванням всіх природних вигинів хребта.
- "Плоско-увігнута спина": спостерігається зменшення грудного кіфозу при нормальному або збільшеному поперековому лордозі.

Ці порушення постави безпосередньо корелюють з викривленнями хребта, основними типами яких є сколіоз, лордоз і кіфоз(рис. 1.4), що відрізняються своїми причинами, проявами та клінічним перебігом.

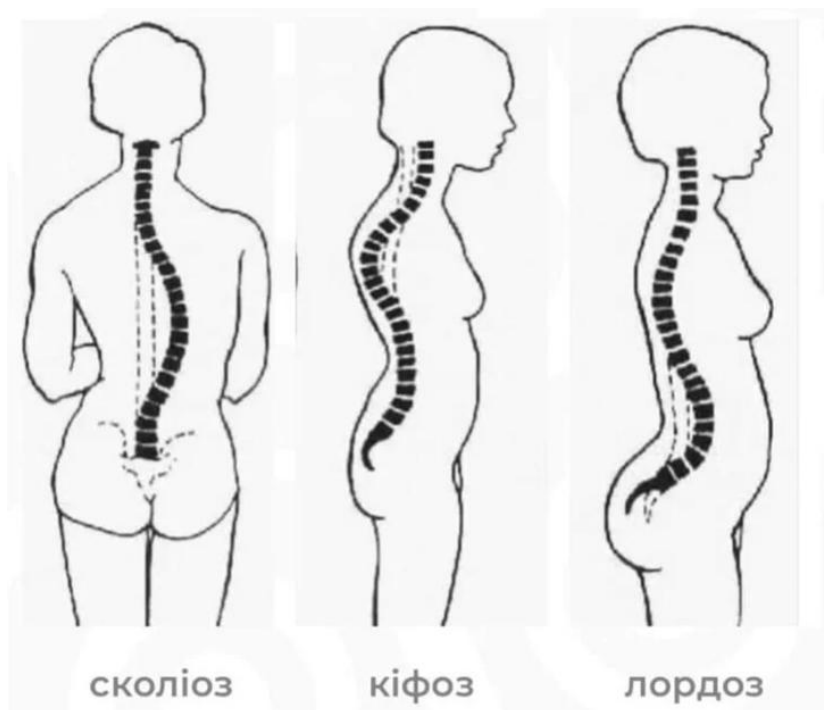


Рисунок 1.4 – Типи викривлень хребта

Сколіоз є комплексною, три площинною деформацією хребта, що охоплює фронтальну, горизонтальну та сагітальну площини. Це не просто косметичний дефект постави у дитини, а серйозне захворювання хребта, яке спричиняє зміни у статиці та динаміці опорно-рухового апарату, що, своєю чергою, призводить до порушення життєдіяльності всього організму і формування так званої сколіотичної хвороби.

Залежно від напрямку та кількості викривлень, сколіоз поділяють на С-подібний (з однією дугою), S-подібний (з двома дугами) та Z-подібний (з трьома дугами).

Кіфоз – це захворювання, що характеризується викривленням осі хребта (опуклістю назад) у сагітальній площині, найчастіше у грудному відділі.

Виникнення кіфозу зумовлене комплексом різноманітних причин. До основних передумов розвитку цієї патології належать порушення внутрішньоутробного розвитку та спадкова схильність. Окрім цього, значний вплив мають набуті чинники, зокрема травми хребта та слабкість м'язів спини.

При кіфозі пацієнти можуть скаржитися на загальну слабкість у кінцівках, а також на труднощі з диханням, травленням та функціонуванням серця, що пов'язано зі змінами об'єму грудної та черевної порожнини.

Лордоз це захворювання, що проявляється дугоподібною деформацією хребетного стовпа (вигин вперед) у сагітальній площині. Патологічна ж форма лордозу може розвинути у будь-якому віці.

Розвиток патологічного лордозу тісно пов'язаний із порушеннями функціонування кістково-м'язової системи, що нерідко посилюється на тлі ожиріння. Клінічна картина захворювання характеризується порушенням постави, хронічним больовим синдромом, зміною ходи та обмеженням рухливості хребта при згинанні, а зміщення центру тяжіння створює додаткові труднощі з утриманням голови у вертикальному положенні. На рисунку 1.5 зображено ілюстрацію людини з порушеною та правильною поставами. Зверху – правильна постава, знизу – порушена.

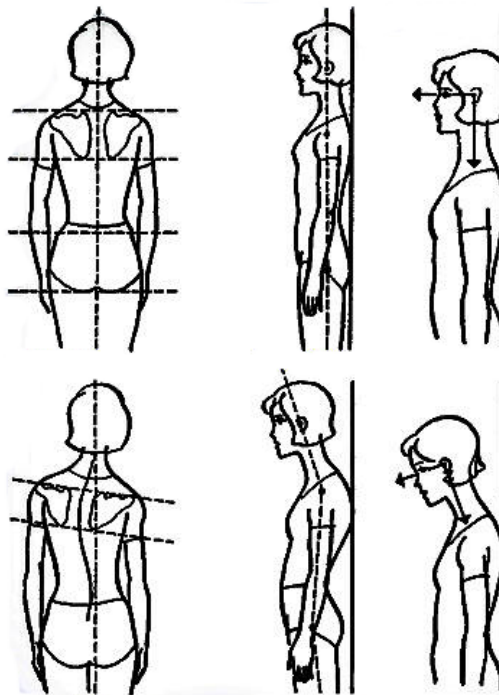


Рисунок 1.5 – Ілюстрація здорової та порушеної постави

Причини порушень постави є багатофакторними і можуть бути зумовлені поєднанням способу життя, генетичної схильності та впливу різноманітних захворювань. Розуміння цих факторів є ключовим для профілактики та ефективної корекції.

Однією з провідних причин розвитку порушень постави, як у дітей, так і у дорослих, є малорухомий спосіб життя (гіподинамія). Недостатня фізична активність призводить до ослаблення м'язового корсета – ключових м'язів спини та живота, які відповідають за стабільність хребетного стовпа та підтримання правильного положення тіла. Ослаблені м'язи не здатні адекватно утримувати хребет у фізіологічному положенні, що створює передумови для його викривлення. Сюди ж відноситься неправильна організація робочого або навчального місця, коли тривале перебування у неергономічній позі (наприклад, сидіння за невідповідною партою, робота за комп'ютером з неправильною висотою монітора та стільця) спричиняє асиметричне навантаження на хребет і м'язи. Крім того, часті та надмірні фізичні навантаження на хребет, особливо без дотримання правильної техніки виконання, також можуть сприяти розвитку деформацій.

Поряд із факторами способу життя, на формування патологічної постави впливають і чинники, пов'язані зі станом здоров'я. До них належить неправильне або незбалансоване харчування, яке може призвести до дефіциту важливих мікроелементів та вітамінів, необхідних для нормального розвитку кісткової та м'язової тканин. Хронічні захворювання шлунково-кишкового тракту можуть опосередковано впливати на поставу, порушуючи засвоєння поживних речовин. Важливу роль відіграють і вроджені захворювання опорно-рухового апарату, такі як ахондроплазія або спондилолітез, які безпосередньо зумовлюють структурні аномалії хребта.

1.2.3 Способи діагностики постави

Діагностування порушень постави та супутніх особливостей хребта вимагає комплексного підходу, що включає широкий арсенал сучасних методів. Цей процес охоплює як клінічні, так і різноманітні інструментальні дослідження, спрямовані на всебічну оцінку стану опорно-рухового апарату та виявлення можливих системних порушень[29].

1. Клінічне дослідження

Первинна діагностика починається з ретельного клінічного обстеження.

Воно включає:

- Лікарський огляд: Візуальна оцінка постави пацієнта в різних положеннях (стоячи, сидячи, при нахилах), виявлення асиметрії, вимірювання довжини кінцівок, огляд шкіри на предмет складок та рубців, що можуть вказувати на деформації.
- Збір анамнезу: Детальний збір інформації про скарги пацієнта (біль, дискомфорт, обмеження рухів), історію розвитку захворювання, перенесені травми та операції, сімейний анамнез, умови праці або навчання, рівень фізичної активності.
- Ортопедичне обстеження: Оцінка обсягу рухів у суглобах, визначення рухливості та торсії (скручування) хребта, виявлення зон локального болю, а також ознак остеохондрозу та інших вертебральних чи екстравертебральних синдромів.
- Мануальне м'язове тестування: Цей метод набуває все більшого значення в останні роки. Він дозволяє з мінімальними витратами оцінювати силу окремих груп м'язів за допомогою заданих рухів. Це допомагає виявити м'язовий дисбаланс, що є частою причиною порушень постави, сприяє диференційній діагностиці, визначенню локалізації ураження, а також надає цінну інформацію для прогнозування та моніторингу ефективності лікування.

2. Інструментальні методи діагностики

Для об'єктивізації та уточнення діагнозу застосовуються різноманітні інструментальні методи:

- Рентгенографія хребта (спондилографія, спондилометрія): Класичний метод, що дозволяє візуалізувати кісткові структури, оцінити ступінь викривлення хребта (наприклад, кут Кобба при сколіозі), виявити аномалії розвитку хребців та інші кісткові зміни.

- Спіральна комп'ютерна томографія (КТ) та магнітно-резонансна томографія (МРТ): Ці методи надають високо деталізовані, багато площинні зображення. КТ краще візуалізує кісткові структури та їхні деформації, тоді як МРТ є незамінною для оцінки стану м'яких тканин, міжхребцевих дисків, спинного мозку та нервових корінців, дозволяючи виявити грижі, запальні процеси чи компресії.
- Електроміографія (ЕМГ) паравертебральних м'язів: Дослідження електричної активності м'язів, що розташовані вздовж хребта. Допомагає виявити м'язові спазми, слабкість або асиметрію активності, що є важливим для розуміння біомеханічних порушень.
- Комп'ютерна топографічна фотометрія, стабілографія та стабілометрія: Це неінвазійні методи, що дозволяють об'єктивно оцінити форму тулуба, симетричність, розподіл навантаження на стопи, а також стабільність пози та рівновагу тіла.
- Остеосцинтиграфія: Радіонуклідне дослідження, що використовується для виявлення запальних процесів, переломів, пухлин або метаболічних порушень у кістковій тканині.
- Ультразвукове (УЗД) та доплерографічне дослідження: Застосовуються для візуалізації м'яких тканин (м'язів, зв'язок), суглобів, а також для оцінки кровотоку у судинах, що може бути порушений при значних деформаціях хребта.

3. Додаткові та лабораторні методи

Окрім вищезазначених, для повної картини стану здоров'я пацієнта можуть бути призначені:

- Функціональні дослідження:
 - Спірометрія та спірографія: Оцінка функції зовнішнього дихання, яка може бути порушена при виражених деформаціях грудної клітки, спричинених сколіозом чи кіфозом.

- Електрокардіографія (ЕКГ) та фонокардіографія (ФКГ): Дослідження роботи серця, оскільки значні викривлення хребта можуть впливати на його положення та функціонування.
- Лабораторні дослідження: Включають обстеження на рівень глікозаміногліканів та оксипроліну в крові та сечі (показники стану сполучної та кісткової тканини), оцінювання остеотропного гормонального профілю (для виявлення порушень кісткового метаболізму), а також генетичні дослідження, особливо при підозрі на вроджені чи синдромальні форми порушень постави.

Таким чином, детальний аналіз постави людини як об'єкта дослідження підкреслює її надзвичайну складність та багатоаспектність. Постава є не лише зовнішнім проявом фізичного стану, а й інтегральним показником функціонування всього опорно-рухового апарату та загального стану організму. Виявлення та класифікація порушень постави, таких як сколіоз, кіфоз та лордоз, є критично важливими, оскільки ці відхилення можуть призводити до значних деформацій хребта, хронічних больових синдромів, обмеження рухливості та, що найважливіше, до системних порушень роботи внутрішніх органів та систем.

Особливе значення це набуває у дитячому та підлітковому віці, коли відбувається інтенсивний ріст і формування опорно-рухового апарату, а фактори сучасного способу життя, такі як гіподинамія та неправильна організація навчального середовища, значно підвищують ризики розвитку патологій. З огляду на широкий спектр причин порушень постави — від генетичних аномалій та вроджених захворювань до впливу зовнішніх факторів та набутих хвороб — актуальність своєчасної та точної діагностики є беззаперечною.

1.3 Опис предмету дослідження

1.3.1 Поняття штучного інтелекту

Штучний інтелект (ШІ) – це вершина еволюції комп'ютерних наук, що активно формує майбутнє технологій. Ця галузь займається розробкою систем,

які, подібно до живих організмів, можуть сприймати інформацію з навколишнього середовища та діяти оптимальним чином для досягнення своїх цілей. ШІ втілюється у різноманітних програмних рішеннях, алгоритмах та апаратних платформах, наділяючи їх унікальними можливостями: від глибокого розуміння складних даних до автономного навчання, гнучкої адаптації та ефективного вирішення проблем, що традиційно вимагали людського інтелекту.

Фундаментальна праця "Artificial Intelligence: A Modern Approach" [21] розглядає чотири основні підходи до визначення штучного інтелекту, які відображають різні аспекти інтелектуальної поведінки:

- 1. Системи, що мислять по-людськи (Thinking Humanly):** Цей підхід зосереджений на розумінні внутрішніх механізмів людського мислення. Метою є не просто імітація поведінки, а розробка моделей, які відображають хід думок людини. Для цього використовуються когнітивна наука та інтроспекція, прагнучи створити програми, що мислять аналогічно людському розуму.
- 2. Системи, що діють по-людськи (Acting Humanly):** Цей підхід фокусується на здатності системи демонструвати поведінку, яка є невідрізною від людської. Класичним тестом для цього є Тест Тюрінга, запропонований Аланом Тюрінгом у 1950 році. Згідно з цим тестом, машина вважається "інтелектуальною", якщо сторонній спостерігач не може відрізнити її відповіді від відповідей людини під час письмового спілкування. Цей підхід охоплює такі області ШІ, як обробка природної мови, представлення знань, автоматичне міркування та машинне навчання.
- 3. Системи, що мислять раціонально (Thinking Rationally):** Цей підхід базується на логіці та прагне розробляти системи, які мислять "правильно" або "логічно". Метою є створення інтелектуальних систем, які здатні робити правильні висновки на основі доступних знань. Однак, незважаючи на привабливість логічного підходу, він стикається з обмеженнями у складних та невизначених реальних середовищах.

4. Системи, що діють раціонально (Acting Rationally): Сучасний підхід до ШІ, який є найпоширенішим у практичних застосуваннях. Він зосереджений на розробці агентів, які діють "раціонально", тобто роблять правильні речі для досягнення своїх цілей, максимізуючи очікувану вигоду. Раціональний агент прагне досягти найкращого результату або, у разі невизначеності, найкращого очікуваного результату. Цей підхід є більш гнучким, ніж логічний, оскільки він дозволяє діяти раціонально навіть за відсутності повних або точних знань, орієнтуючись на максимізацію успіху в даному середовищі.

Штучний інтелект є широкою і багатогранною дисципліною, що постійно розвивається, інтегруючи знання з інформатики, математики, філософії, психології та інших наук для створення систем, здатних вирішувати складні завдання та взаємодіяти зі світом інтелектуальним способом.

Для реалізації інтелектуальних можливостей, штучний інтелект послуговується великою різноманітністю різних методів та технологій. Серед них ключове місце посідають штучні нейронні мережі – моделі, що дозволяють системам навчатися на даних та виявляти складні закономірності, а також різноманітні підходи в рамках машинного навчання, які визначають способи набуття знань цими інтелектуальними системами.

1.3.2 Штучні нейронні мережі

Штучні нейронні мережі (ШНМ) являють собою математичні моделі, розроблені для імітації принципів функціонування біологічних нейронних мереж у людському мозку. Ці мережі складаються з великої кількості взаємопов'язаних штучних нейронів, які спільно працюють для виконання певних завдань.

Архітектура нейронної мережі характеризується ієрархічною та модульною будовою, що базується на тісній взаємодії кількох ключових компонентів. Основними обчислювальними елементами виступають нейрони (вузли), які, подібно до біологічних клітин, відповідають за приймання, обробку та генерацію

сигналів. Взаємодія між цими вузлами забезпечується через канали зв'язку, кожен з яких має асоційований числовий параметр — вагу. Вага відображає силу впливу сигналу на активацію приймаючого нейрона та є ключовим параметром, що підлягає налаштуванню під час навчання; саме адаптація ваг дозволяє мережі виявляти закономірності у вхідних даних[7].

Головним етапом обробки інформації всередині нейрона є застосування функції активації, яка трансформує зважену суму вхідних сигналів у вихідне значення. Основна роль цієї функції полягає у введенні нелінійності, без якої нейронна мережа, незалежно від її глибини, функціонувала б як проста лінійна модель, нездатна вирішувати складні завдання. Структурно нейрони організовані у послідовні шари, кожен з яких має своє функціональне призначення. Процес розпочинається з вхідного шару, що приймає первинні дані та відповідає їхній розмірності. Далі інформація передається через приховані шари, які не взаємодіють із зовнішнім середовищем безпосередньо, а виконують основні обчислення та виокремлюють абстрактні ознаки. Завершується процес вихідним шаром, який генерує фінальний результат, наприклад, клас об'єкта або прогнозоване значення, залежно від поставленої задачі.

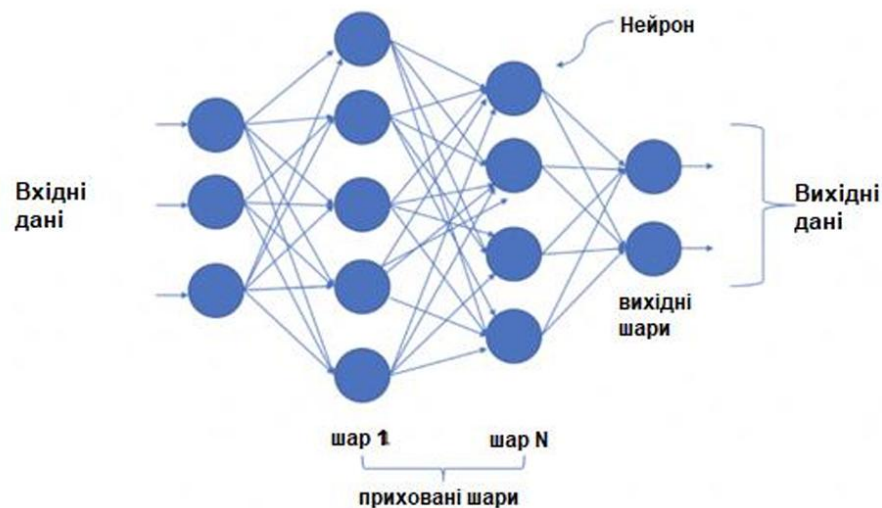


Рисунок 1.6 – Структурна схема штучної нейронної мережі

Організація нейронів у шари та їхні зв'язки визначають архітектуру нейронної мережі. На рисунку 1.6 зображено структурну схему повнозв'язної штучної нейронної мережі, де кожен нейрон одного шару з'єднаний з кожним

нейроном наступного шару, що забезпечує складну взаємодію та обробку інформації.

1.3.2 Навчання штучного інтелекту

Фундаментальною основою функціонування будь-якої нейронної мережі є процес навчання — динамічний механізм, завдяки якому модель здатна самостійно виявляти приховані закономірності у вхідних даних та адаптуватися до них для здійснення точних прогнозів. Ця процедура є центральним елементом ширшої галузі — машинного навчання, що як напрям штучного інтелекту зосереджений на розробці алгоритмів, здатних набувати знань емпіричним шляхом без необхідності явного програмування інструкцій.

Методи машинного навчання умовно поділяються на кілька основних парадигм, серед яких домінуючим є навчання за прецедентами (індуктивне навчання). На відміну від дедуктивного навчання, що історично пов'язане з експертними системами, індуктивне навчання фокусується на виведенні загальних правил із конкретних прикладів. Сьогодні саме навчання за прецедентами є найбільш популярним, оскільки експертні системи зіштовхуються з труднощами у взаємодії з сучасними базами даних та їх масштабуванням.

Навчання за прецедентами поділяється на три ключові типи[27]:

- **Контрольоване навчання (Supervised Learning):** Використовується, коли доступні великі обсяги **маркованих даних**, тобто даних, для яких заздалегідь відомі правильні відповіді або категорії. Мета такого навчання — створити алгоритм, який, ґрунтуючись на цих "учительських" мітках, зможе вірно розпізнавати або класифікувати нові, раніше небачені дані. Це дозволяє машині засвоювати ознаки та закономірності для подальшої категоризації.
- **Неконтрольоване навчання (Unsupervised Learning):** Застосовується в ситуаціях, коли доступні немарковані дані (наприклад, зображення без описів або аудіозаписи без коментарів). Основне завдання машини тут —

самостійно виявляти приховані зв'язки, закономірності, структурувати дані або групувати їх (кластеризувати) без попередньо заданих категорій. Прикладом може слугувати робота рекомендаційних систем, які аналізують попередні покупки клієнтів для пропозиції нових товарів.

- **Навчання з підкріпленням (Reinforcement Learning):** Це динамічний варіант навчання, де "вчителем" виступає саме навколишнє середовище. Машина, або "агент", взаємодіє зі середовищем, виконуючи певні дії. Середовище реагує на ці дії, надаючи агенту "нагороди" або "штрафи" (підкріплення), на основі яких агент поступово вчиться оптимізувати свою стратегію поведінки для досягнення максимальної винагороди

Порівняння цих типів навчання схематично представлено на рисунку 1.7.



Рисунок 1.7 – Порівняння контрольованого та неконтрольованого навчання

1.4 Аналіз актуальності

Актуальність цього дослідження зумовлена кількома взаємопов'язаними факторами: зростаючою глобальною проблемою порушень постави та захворювань опорно-рухового апарату, значним соціально-економічним навантаженням, яке вони створюють, а також потенціалом сучасних технологій штучного інтелекту для революціонізації методів їх діагностики та моніторингу.

Порушення постави це не лише естетичний недолік, а серйозне хронічне захворювання опорно-рухового апарату, яке, як було зазначено, має значні негативні наслідки для загального здоров'я людини, особливо критичні у

дитячому та підлітковому віці. Зміна способу життя, що характеризується гіподинамією та тривалим перебуванням у статичних нефізіологічних позах, сприяє широкому розповсюдженню таких патологій, як сколіоз, кіфоз та лордоз. Ці викривлення хребта, якщо їх не діагностувати та не коригувати вчасно, можуть призвести до серйозних ускладнень, включаючи компресію нервів, хронічні больові синдроми та порушення функціонування внутрішніх органів.

За даними Всесвітньої організації охорони здоров'я (ВООЗ), хронічні неінфекційні захворювання опорно-рухового апарату є одними з найпоширеніших у світі, вражаючи приблизно 1,71 мільярда людей. Вони є одними з провідних чинників обмеження фізичної працездатності населення, зумовлюючи значні витрати на медичні та реабілітаційні послуги. Більше половини осіб, які потребують реабілітації, страждають саме від цих захворювань, що робить проблему надзвичайно важливою як для дитячого, так і для дорослого населення. Країни з високим рівнем доходу є найбільш ураженими за кількістю людей – 441 мільйон, за ними йдуть країни регіону Західної частини Тихого океану (427 мільйонів) та регіону Південно-Східної Азії (369 мільйонів). Захворювання опорно-рухового апарату також є найбільшим чинником років життя з інвалідністю (YLDs) у всьому світі, становлячи приблизно 149 мільйонів YLDs, або 17% від усіх YLDs у світі[17].

Порушення постави не тільки негативно впливають на фізичний розвиток організму, але й призводять до формування структурних деформацій хребта та больових синдромів, що реалізуються через патологічні рухові та статичні стереотипи. Дослідники ще наприкінці ХХ ст. зазначали, що в подальшому це може призвести до розвитку остеохондрозу, стійкого больового синдрому та формування гриж міжхребцевих дисків, які трапляються у 60–80% дорослого працездатного населення і можуть спричиняти інвалідизацію[29].

З огляду на ці потенційно руйнівні довгострокові наслідки, особливого значення набуває своєчасне виявлення та запобігання прогресуванню відхилень у поставі. Саме тому фокус сучасної медицини зміщується в бік ранньої та доступної діагностики, що, втім, стикається з низкою викликів.

Ефективність будь-якого лікування значною мірою залежить від ранньої діагностики порушень постави та попередження їхнього прогресування. Регулярний моніторинг стану постави є ключовим для виявлення патологій на початкових стадіях, що значно підвищує шанси на успішне лікування та запобігання незворотним змінам. Однак існуючі традиційні методи діагностики постави, хоч і є ефективними, часто мають обмеження, такі як інвазивність (наприклад, рентгенографія хребта з радіаційним навантаженням), висока вартість обладнання, потреба у кваліфікованому персоналі та відносна тривалість процедур візуального та інструментального обстеження постави.

Саме в цьому контексті застосування штучного інтелекту, зокрема таких передових методів набуває особливої актуальності. Розробка інструментів ШІ для оцінки діагностики порушень постави дозволить вирішити низку соціально-економічних проблем, надаючи лікарям, медичним працівникам та широкому загалу доступ до інноваційних та доступних засобів моніторингу стану постави.

Таке рішення здатне значно скоротити час аналізу постави, підвищити об'єктивність результатів обстеження пацієнта, а також надати ефективний механізм для довготривалого контролю за ходом лікування та реабілітації.

1.5 Аналіз вже існуючих рішень

Аналіз постави за допомогою штучного інтелекту є важливим напрямом розвитку медичних і фітнес-технологій, що поєднує комп'ютерне зір, машинне навчання та глибинне навчання для оцінки положення тіла, виявлення відхилень і надання рекомендацій щодо корекції. Сучасні рішення в цій сфері спрямовані на профілактику, корекцію постави та діагностику патологій опорно-рухового апарату[23].

Серед популярних рішень виділяються платформи, які застосовують комп'ютерне зір для обробки зображень або відеозаписів із камер смартфонів. Наприклад, APECS(рис. 1.8) використовує фотограмметричні алгоритми для точного аналізу постави на основі фотографій або відео, отриманих зі смартфонів.

Система оцінює передню, задню, праву та ліву площини тіла, виявляючи відхилення, такі як сколіоз, кіфоз, синдром нахиленої голови, а також вальгусні чи варусні деформації ніг.

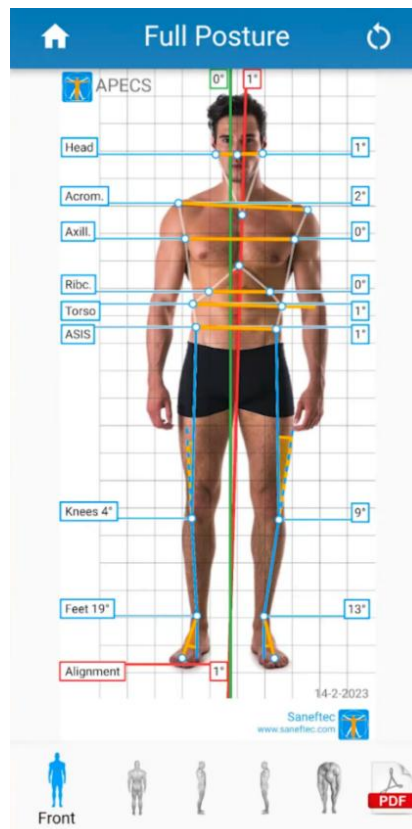


Рисунок 1.8 –Приклад фронтального аналізу постави APECS

APECS пропонує унікальні функції, такі як тест "Золотий Перетин" для оцінки пропорцій тіла, індекси симетрії тулуба (ATSI і POTSI), а також гоніометр для вимірювання кутів. Додаток підтримує динамічний аналіз через відео, включаючи режими ручного, автоматичного позиціонування та розпізнавання зелених маркерів, що робить його корисним для фізіотерапевтів, тренерів і дослідників. APECS генерує детальні звіти із можливістю кастомізації, а також надає щоденні поради та вправи для корекції постави. Однак точність залежить від якості зображень і умов освітлення, а інтеграція з медичними системами обмежена. Основні функції безкоштовні, але розширені можливості доступні за підпискою.

Для професійного використання призначена платформа **PostureScreen**, яка застосовує комп'ютерне зір для оцінки постави на основі фотографій або

відео(рис. 1.9). PostureScreen підтримує 2- та 4-видові оцінки постави, створюючи клінічні звіти, персоналізовані освітні звіти та звіти порівняння для відстеження прогресу. Додаток інтегрується з WebExercises для автоматичного підбору вправ на основі виявлених відхилень, таких як синдром нахиленої голови чи округлені плечі.

Розробники запевняють, що PostureScreen має високу надійність, підтверджену науковими дослідженнями. Система також підтримує віддалені оцінки через RemoteScreen та інтеграцію з деякими електронними медичними системами, що робить її цінним інструментом для фізіотерапевтів і хіропрактиків. На рисунку 1.9 зображено приклад звіту PostureScreen.



Рисунок 1.9 – Звіт аналізу постави виконаний PostureScreen

Отже, станом на сьогодні технології ШІ для аналізу постави активно розвиваються, що підтверджується зростанням інтересу до таких рішень. Таким чином, сучасні ШІ-системи для аналізу постави демонструють значний потенціал, але потребують вдосконалення для забезпечення високої точності, доступності та інтеграції з медичними платформами.

1.6 Визначення цілей роботи та постановка задачі

Головною метою даної роботи є розробка та реалізація програмної системи на основі технологій штучного інтелекту, призначеної для автоматизованого та неінвазійного аналізу постави людини за її фотозображеннями. Об'єктом дослідження є процес аналізу постави людини за візуальними даними. Предметом дослідження є методи та алгоритми штучного інтелекту, зокрема нейронні мережі та методи комп'ютерного зору, що застосовуються для автоматизованої оцінки постави.

Ця система має забезпечити новий рівень доступності та швидкості оцінки постави, дозволяючи фахівцям і самим пацієнтам отримувати точну інформацію без необхідності використання специфічного обладнання. Досягнення цієї мети сприятиме ранньому виявленню відхилень, своєчасному втручанню та моніторингу динаміки змін у поставі.

Для досягнення поставленої мети необхідно вирішити наступні основні завдання:

1. Аналіз вимог:

- Здійснити аналіз функціональних і нефункціональних вимог до програмної системи, включаючи вимоги до точності, швидкості, зручності інтерфейсу та безпеки даних.
- Визначити критерії успішності проекту, такі як мінімальний рівень точності розпізнавання ключових точок, час обробки одного зображення та інші параметри.

2. Формування та підготовка набору даних для навчання:

- **Збір та вибір даних:** Провести аналіз існуючих відкритих наборів даних для розпізнавання пози людини, таких як COCO, MPII або інших, і обрати найбільш відповідний для поставленого завдання.
- **Комбінований підхід:** У разі потреби, доповнити обраний відкритий набір даних невеликим власним набором стандартизованих фотозображень.

- **Анотація:** Використовувати вже існуючу розмітку в обраних наборах даних, а для власного набору – здійснити точну розмітку ключових анатомічних точок.

3. Розробка архітектури та навчання моделі ШІ:

- **Вибір архітектури:** Дослідити та обґрунтувати вибір найоптимальнішої архітектури нейронної мережі, що базується на методах комп'ютерного зору, для розпізнавання пози людини.
- **Навчання моделі:** Використати підготовлений і анотований набір даних для навчання обраної моделі або для тонкого налаштування попередньо навченої моделі.
- **Оптимізація та валідація:** Провести оптимізацію гіперпараметрів моделі для досягнення максимальної точності, а також валідацію її роботи на тестових даних.

4. Створення програмної системи та інтерфейсу користувача:

- **Імплементация моделі:** Інтегрувати навчену модель в програмне забезпечення, щоб забезпечити її функціонування.
- **Розробка інтерфейсу:** Створити зручний та інтуїтивно зрозумілий графічний інтерфейс, який дозволить користувачам легко завантажувати зображення, отримувати результати аналізу та візуалізувати знайдені ключові точки.
- **Аналітичний функціонал:** Реалізувати механізм автоматичного обчислення кутів та інших біомеханічних показників, що характеризують поставу, на основі даних, отриманих від моделі.

5. Комплексне тестування та оцінка ефективності:

- **Тестування:** Провести тестування системи з використанням незалежного набору даних, який не брав участі в навчанні моделі.
- **Підготовка рекомендацій:** Сформулювати рекомендації щодо подальшого вдосконалення системи та можливостей її використання в реальній медичній практиці.

Варто зазначити, що наведена послідовність завдань відображає повний життєвий цикл розробки інтелектуальної системи: від формування вимог та підготовки даних до програмної реалізації та валідації результатів. Для наочного представлення ієрархії завдань, їх декомпозиції та взаємозв'язків, структуру дослідження було візуалізовано. На рисунку 1.10 зображено дерево цілей даної роботи.

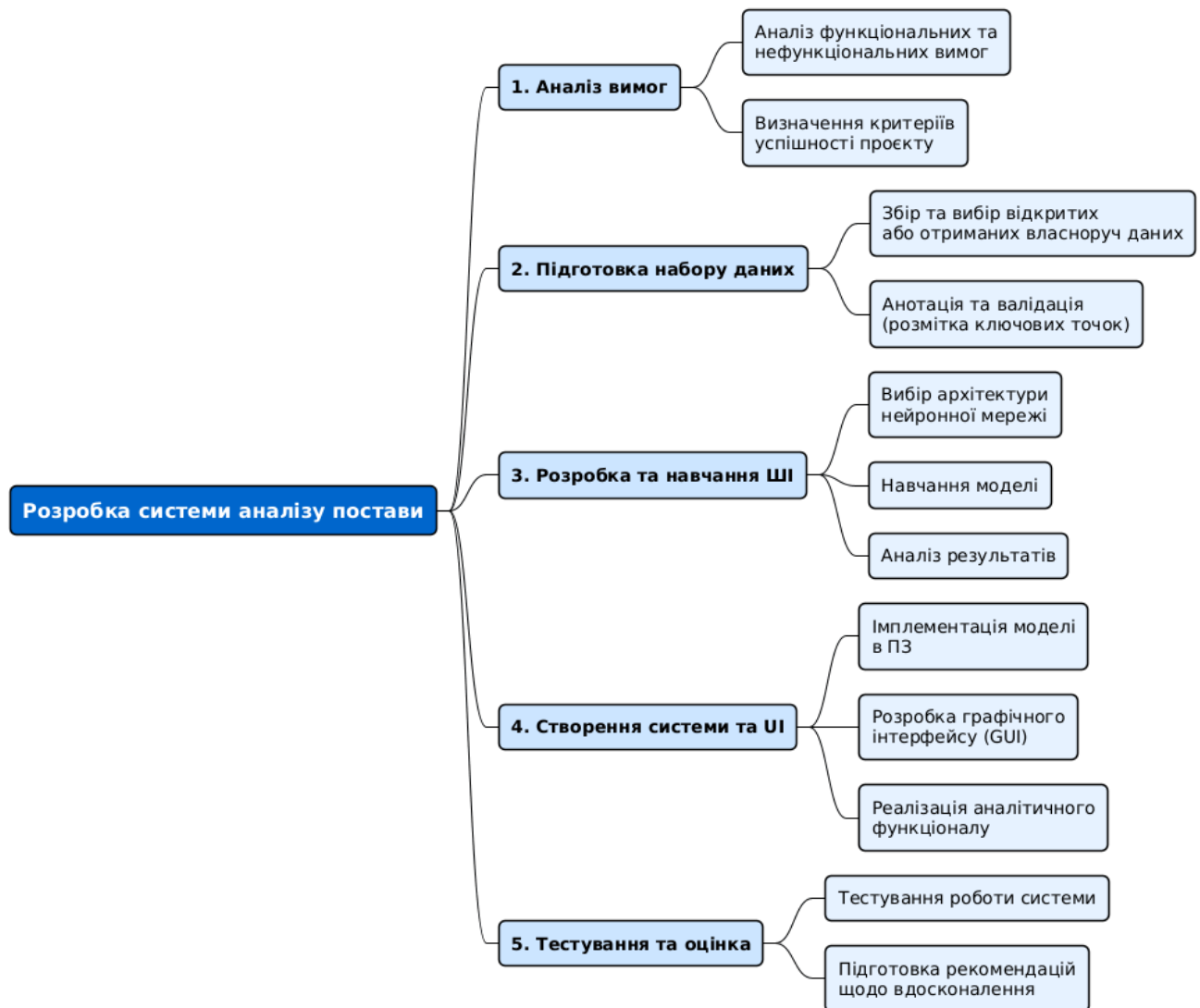


Рисунок 1.10 – Дерево цілей

Очікуваним результатом роботи є створення ефективного та доступного програмного інструменту, який може стати важливим допоміжним засобом для фахівців у галузі медицини та реабілітації, а також цінним інструментом для самоконтролю постави, що сприятиме загальному покращенню здоров'я населення.

Розділ 2. Математичне забезпечення

2.1 Постановка задачі машинного навчання

Машинне навчання є складовою галузі штучного інтелекту, що займається розробленням математичних моделей, алгоритмів та методів, здатних автоматично виявляти закономірності у даних та приймати рішення або робити передбачення без явного програмування цих правил.

Ідея машинного навчання полягає у тому, щоб побудувати функцію, яка узагальнює спостереження, отримані з минулого досвіду, і здатна коректно реагувати на нові, раніше невідомі вхідні дані[7].

У межах даної магістерської роботи задача аналізу постави людини розглядається як задача побудови математичної моделі, здатної узагальнювати закономірності у даних та робити передбачення положення ключових точок на тілі людини для нових прикладів.

Нехай задано множину прикладів спостережень:

$$D = (x_1, y_1), (x_2, y_2), \dots, (x_N, y_N), \quad (2.1)$$

де $x_i \in X$ – вхідне зображення людини у деякій проекції;

$y_i \in Y$ – відповідні мітки (набір координат ключових точок тіла).

У загальному випадку метою машинного навчання є побудова параметризованої моделі $f_\theta: X \rightarrow Y$, яка наближає невідому залежність між вхідними та вихідними змінними:

$$y \approx f_\theta(x), \quad (2.2)$$

де θ – вектор параметрів моделі (ваги, коефіцієнти, зміщення тощо).

У нашому випадку, метою навчання є побудова параметризованої моделі $f_\theta(x)$, яка для будь-якого нового зображення x забезпечує оцінку $\hat{y} = f_\theta(x)$, максимально наближену до істинного значення y . Параметри θ (ваги нейронної мережі) підбираються у процесі навчання шляхом мінімізації функції втрат $L(\theta)$.

Це формулюється як задача оптимізації:

$$\theta^* = \arg_{\theta} \min L(\theta) = \arg_{\theta} \min \frac{1}{N} \sum_{i=1}^N \ell(f_{\theta}(x_i), y_i), \quad (2.3)$$

де $L(\theta)$ – функціонал помилки (функція втрат);

$\ell(\cdot)$ – часткова функція втрат для одного прикладу;

N – кількість навчальних прикладів.

Результатом процесу машинного навчання є побудована модель із певним набором параметрів θ^* , яка описує залежність між вхідними даними x та очікуваними виходами y . У математичному сенсі, така модель визначає гіперплощину у просторі ознак, що найкраще апроксимує розподіл навчальних даних.

Під гіперплощиною розуміють багатовимірну узагальнену площину, яка розділяє простір ознак на області, що відповідають різним класам або значенням прогнозу. У випадку лінійних моделей, таких як лінійна регресія або лінійний класифікатор, гіперплощина описується рівнянням:

$$w_1 x_1 + w_2 x_2 + \dots + w_n x_n + b = 0, \quad (2.4)$$

де w_i – вагові коефіцієнти (параметри моделі);

x_i – компоненти вектора вхідних даних;

b – зсув (bias).

У цьому випадку навчання полягає у підборі таких параметрів $w = (w_1, w_2, \dots, w_n)$ та b , щоб гіперплощина найкраще узгоджувалася з навчальними прикладами, тобто мінімізувала середню помилку між реальними значеннями y_i та передбаченими $\hat{y}_i$.

Геометрично процес оптимізації можна інтерпретувати як поступове «обертання» та «зміщення» гіперплощини у просторі даних, доки вона не досягне положення, при якому втрати $L(\theta)$ набудуть мінімального значення.

Таким чином, задача машинного навчання в контексті аналізу постави – це оптимізаційна задача, у якій модель набуває таких параметрів, що мінімізують середню похибку передбачення розташування ключових точок на тілі людини.

2.2 Метод градієнтного спуску

Як зазначалося у попередньому підрозділі, задачу машинного навчання можна розглядати як задачу оптимізації деякої функції втрат (цільової функції), мінімізація якої дозволяє знайти оптимальні параметри моделі. Одним із найпоширеніших підходів до розв’язання таких задач є метод градієнтного спуску – чисельний алгоритм оптимізації, що базується на властивостях градієнта.

Градієнт – це вектор, який показує напрямок найшвидшого зростання певної скалярної функції. Його позначають як grad або символом ∇ . За своєю величиною (модулем) градієнт характеризує швидкість зміни функції у вибраному напрямку. Для кращого розуміння можна навести аналогію: якщо уявити собі поверхню землі, де значення функції відповідають висоті над рівнем моря, то градієнт у кожній точці вказує напрямок найкрутішого підйому, а його довжина – крутизну схилу [10].

На рисунку 2.1 схематично показано, як операція обчислення градієнта перетворює поверхню (ліворуч) у поле векторів (праворуч). Видно, що вектори спрямовані у напрямку зростання функції, а їхня довжина залежить від крутизни поверхні.

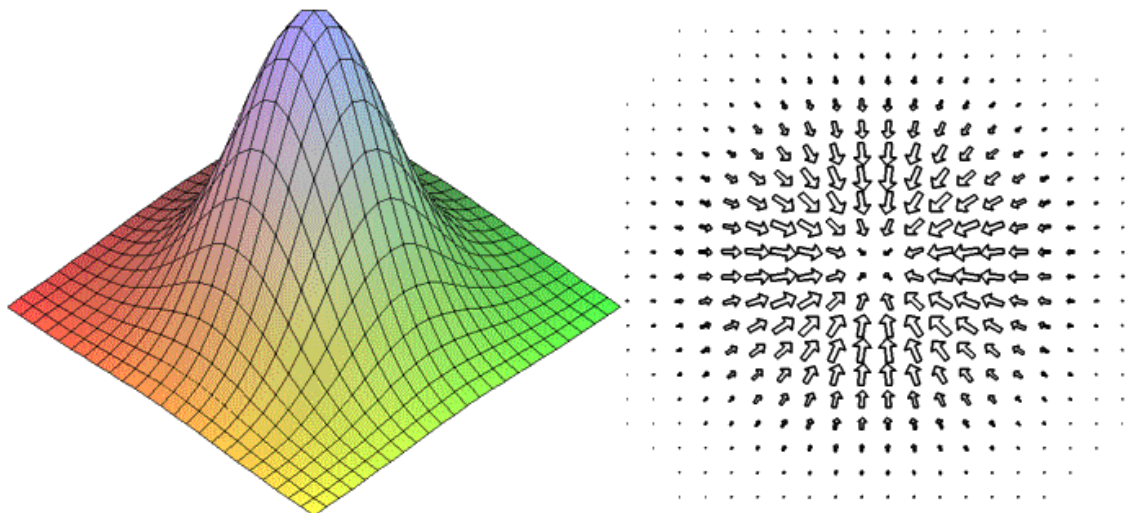


Рисунок 2.1 – Візуалізація градієнту

Для скалярного поля $U(x, y, z)$ градієнт визначається формулою:

$$\nabla U = \text{grad}U = \frac{\partial U}{\partial x}i + \frac{\partial U}{\partial y}j + \frac{\partial U}{\partial z}k, \quad (2.5)$$

Метод градієнтного спуску базується на ключовому спостереженні: якщо функція втрат $F(w)$ є неперервною та диференційовною за параметрами моделі w , то значення цієї функції найшвидше зменшується у напрямку, протилежному до градієнта $\nabla F(w)$ [28].

Інтуїтивно це можна пояснити тим, що градієнт вказує напрямок зростання функції, а отже, для мінімізації потрібно рухатися в протилежному напрямку – антиградієнту.

Таким чином, нове значення параметрів отримують шляхом віднімання градієнта:

$$w_{new} = w_{old} - \nabla F(w_{old}), \quad (2.6)$$

де, w – вектор параметрів моделі;

$\nabla F(w_{old})$ – градієнт функції втрат за цими параметрами.

У такому випадку, величина зміни параметрів може сильно коливатися, залежно від крутизни схилу функції, тому для нормалізації цього параметру застосовується додатковий коефіцієнт η .

Коефіцієнт навчання η визначає довжину кроку оновлення і виконує роль нормалізатора зміни параметрів. Тоді загальний процес градієнтного спуску можна подати у вигляді рекурентного оновлення:

$$w_{t+1} = w_t - \eta \nabla F(w_t), \quad (2.7)$$

де t – номер ітерації.

Іншими словами, член $\eta \nabla F(w_t)$ віднімається від w_{t+1} , оскільки ми хочемо рухатися проти градієнту, тобто вниз до мінімуму. Враховуючи це спостереження, починають з припущення x_0 про локальний мінімум F , і розглядають таку послідовність $w_0, w_1, w_n \dots$ що:

$$w_{n+1} = w_n - \eta_n \nabla F(w_n), n \geq 0, \quad (2.8)$$

Ми маємо:

$$F(w_0) \geq F(w_1) \geq F(w_2) \geq \dots, \quad (2.9)$$

і тому сподіваємося, що послідовність $F(w_n)$ збігається до бажаного локального мінімуму.

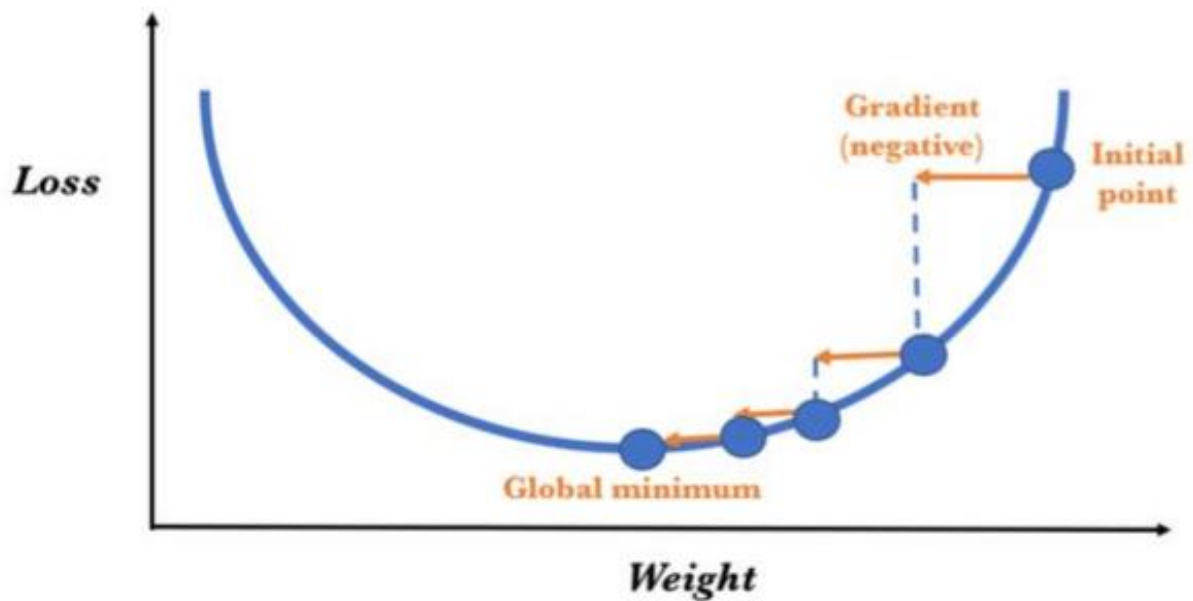


Рисунок 2.2 – Візуалізація роботи градієнтного спуску

На рисунку 2.2 візуалізовано роботу градієнтного спуску. На кожному кроці обчислюється градієнт функції втрат відносно поточних параметрів, і параметри оновлюються у напрямку, який зменшує значення функції. Цей процес повторюється доти, доки зміни у функції втрат не стають незначними або досягнуто заздалегідь визначеної кількості ітерацій.

Хоча алгоритм градієнтного спуску є одним із найефективніших і найпоширеніших методів оптимізації в машинному навчанні та штучному інтелекті, його використання супроводжується низкою обмежень, які необхідно враховувати при побудові моделей.

Одним із головних недоліків методу є те, що градієнтний спуск не гарантує знаходження глобального мінімуму функції втрат. У просторах із великою кількістю параметрів поверхня функції втрат може мати складний ландшафт із численними локальними мінімумами, плато або сідловими точками. У таких випадках алгоритм може збігатися лише до локального мінімуму, що не завжди забезпечує оптимальне рішення задачі. На рисунку 2.3 зображено візуалізовано проблему пошуку глобального мінімуму.

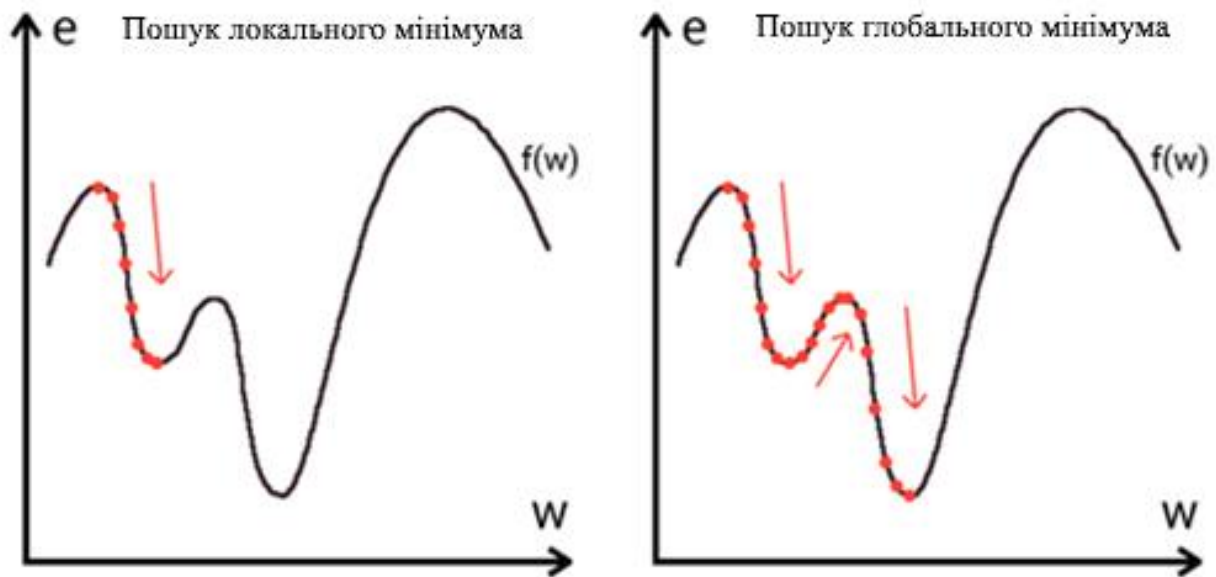


Рисунок 2.3 – Візуалізація проблеми локального мінімуму

Проблема ускладнюється тим, що вибір початкових значень параметрів істотно впливає на траєкторію збіжності. Вибір початкових значень параметрів w_0 суттєво впливає на траєкторію пошуку мінімуму. Невдалий вибір початкових параметрів може спричинити передчасну збіжність до локального мінімуму або до плато, на якому зміни функції втрат незначні, що сповільнює процес навчання

Також важливою особливістю є чутливість алгоритму до вибору швидкості навчання (коефіцієнта η), яка визначає довжину кроку оновлення параметрів. Якщо цей параметр обрано надто великим, градієнтний спуск може не збігатися, а навпаки, коливатися навколо мінімуму або навіть розбігатися. Якщо ж швидкість навчання занадто мала, процес навчання стає повільним і може потребувати великої кількості ітерацій для досягнення прийнятного результату. Тому належний підбір коефіцієнта η є важливою складовою стабільної та ефективної роботи алгоритму.

Суттєвим недоліком градієнтного спуску також є його обчислювальна складність. Для моделей із великим числом параметрів або при роботі з великими обсягами даних обчислення градієнта на кожній ітерації потребує значних ресурсів. Така ситуація характерна для класичного варіанту алгоритму, коли оновлення параметрів виконується лише після обробки всього навчального набору. Це робить метод енергоємним та уповільнює процес оптимізації.

Додатково, через використання похідних, метод потребує, щоб цільова функція була диференційованою. У випадках, коли функція має розриви, ділянки неперервності або різко змінює значення, ефективність алгоритму помітно знижується, або зовсім відсутня. У таких ситуаціях застосовують апроксимаційні підходи або спеціальні модифікації, що дозволяють частково обійти ці обмеження.

2.3 Оптимізатори методу градієнтного спуску

Як було розглянуто у попередньому підрозділі, метод градієнтного спуску є базовим інструментом для мінімізації функції втрат у процесі навчання моделей машинного навчання. Однак його класична форма має певні обмеження – зокрема, чутливість до вибору швидкості навчання, повільну збіжність і схильність до застрягання в локальних мінімумах. Для подолання цих недоліків було розроблено низку модифікацій алгоритму, відомих як оптимізатори градієнтного спуску.

Перед розглядом конкретних оптимізаторів варто окремо зазначити, що навіть базовий градієнтний спуск може мати кілька варіантів реалізації, які різняться за способом обчислення градієнта на навчальних даних. Основними з них є пакетний (batch), міні-пакетний (mini-batch) та стохастичний (stochastic) градієнтний спуск.

У пакетному градієнтному спуску (Batch Gradient Descent) для кожного кроку оновлення ваг обчислюється повний градієнт на всьому наборі навчальних даних. Це забезпечує стабільну збіжність і точне визначення напрямку мінімуму функції втрат, проте вимагає значних обчислювальних ресурсів, особливо при великих обсягах даних. Формально оновлення параметрів можна подати у вигляді:

$$w_{t+1} = w_t - \eta \nabla F(w_t), \quad (2.10)$$

де $\nabla F(w_t)$ – градієнт функції втрат, обчислений для всіх прикладів набору даних.

На противагу цьому, стохастичний градієнтний спуск (Stochastic Gradient Descent, SGD) використовує лише один випадковий приклад із набору даних для кожного оновлення. Формально виглядає так само(див. 2.10), проте $\nabla F(w_t)$ –

градієнт функції втрат, обчислений для одного екземпляру з навчальної вибірки. Це значно прискорює навчання та дозволяє уникати застрягання в локальних мінімумах за рахунок додавання стохастичності, однак може призводити до коливань навколо точки збіжності.

Компромісним варіантом є міні-пакетний градієнтний спуск (Mini-Batch Gradient Descent), який використовує невеликий під набір (батч або міні батч) даних розміром, наприклад, 32, 64 або 128 елементів. Такий підхід поєднує стабільність класичного методу та ефективність стохастичного, а також дозволяє ефективно використовувати векторизовані обчислення на GPU.

Оптимізатори спрямовані на підвищення ефективності та стабільності процесу навчання за рахунок адаптивної зміни параметрів оновлення ваг. Нижче наведено огляд найбільш поширених варіантів, які застосовуються у сучасних нейронних мережах[28].

Momentum

Однією з проблем класичного SGD є те, що під час навчання на «кривих» поверхнях функції втрат (з глибокими долинами або плато) рух градієнту може бути нестабільним. Для зменшення цих коливань використовується метод моменту (Momentum), який враховує не лише поточний градієнт, а й попередню «швидкість» зміни параметрів.

$$\begin{aligned} v_t &= \beta v_{t-1} + (1 - \beta) \nabla F(w_t) \\ w_{t+1} &= w_t - \eta v_t \end{aligned} \quad (2.11)$$

де v_t – «швидкість» або накопичений градієнт, а β – коефіцієнт згладжування (зазвичай $\beta \in [0.8, 0.99]$)

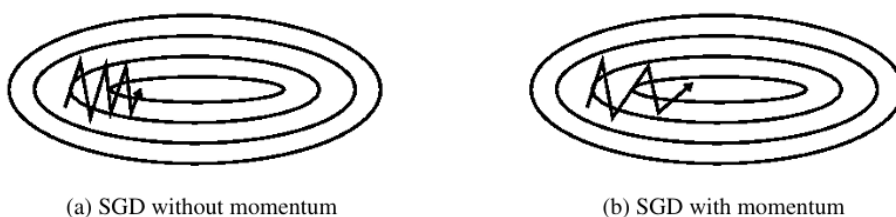


Рисунок 2.3 – Порівняння використання методу моменту

Такий підхід дозволяє алгоритму рухатись більш впевнено у напрямку загального зменшення функції втрат, ігноруючи незначні коливання локальних похідних.

Root Mean Square Propagation

Іншим удосконаленням є RMSProp (Root Mean Square Propagation), який адаптує швидкість навчання для кожного параметра окремо, залежно від середнього квадрата його попередніх градієнтів. Це особливо ефективно при роботі з нестационарними даними або в умовах, коли різні параметри мають різні масштаби впливу.

Формули оновлення виглядають так:

$$\begin{aligned} s_t &= \rho s_{t-1} + (1 - \rho)(\nabla F(w_t))^2 \\ w_{t+1} &= w_t - \eta \nabla F(w_t) / (\sqrt{s_t} + \varepsilon) \end{aligned} \quad (2.12)$$

де ρ – коефіцієнт згладжування, а ε – мале число для запобігання діленню на нуль.

RMSProp допомагає уникати надмірних коливань градієнтів і дозволяє швидше досягати стабільної збіжності навіть на складних функціях втрат.

Adaptive Moment Estimation

Одним із найпоширеніших сучасних оптимізаторів є Adam (Adaptive Moment Estimation), який поєднує ідеї Momentum та RMSProp. Adam зберігає як середнє значення градієнтів, так і середній квадрат градієнтів, що забезпечує баланс між стабільністю руху і адаптивністю швидкості навчання.

Алгоритм визначається наступними рівняннями:

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1 - \beta_1) \nabla F(w_t), \\ v_t &= \beta_2 v_{t-1} + (1 - \beta_2) (\nabla F(w_t))^2, \\ w_{t+1} &= w_t - \eta * (m_t / (\sqrt{v_t} + \varepsilon)), \end{aligned} \quad (2.13)$$

де β_1 та β_2 – коефіцієнти згладжування для першого і другого моментів відповідно.

Adam має високу стабільність і ефективність навіть у випадках, коли дані є шумними або неоднорідними. Саме тому цей оптимізатор найчастіше використовується для навчання глибоких нейронних мереж

Таким чином, розвиток оптимізаторів градієнтного спуску став одним із ключових факторів успіху сучасних алгоритмів глибокого навчання. Від простого стохастичного спуску до адаптивних методів із корекцією моментів – усі ці підходи спрямовані на підвищення стабільності, швидкості збіжності та здатності моделей ефективно працювати з великими, нерівномірними або шумними наборами даних. Застосування оптимізаторів є важливим етапом налаштування нейронних мереж, який безпосередньо впливає на точність і швидкість їх навчання.

2.4 Функції втрат

У попередніх підрозділах було розглянуто, що процес навчання моделей машинного навчання полягає у мінімізації функції втрат за допомогою методів оптимізації, зокрема градієнтного спуску та його варіацій. Функція втрат (англ. loss function або cost function) є ключовим елементом у математичному забезпеченні навчання, оскільки саме вона визначає, наскільки добре модель узгоджується з наявними даними.

З формальної точки зору, функція втрат – це скалярна функція, що відображає відстань між передбаченнями моделі $\hat{y}$ і реальними значеннями y . Її мінімізація є метою процесу оптимізації:

$$L(\hat{y}, y) = \text{Невідповідність істини та передбачення}, \quad (2.14)$$

У контексті постановки задачі машинного навчання, розглянутої раніше, функцію втрат можна розглядати як критерій якості апроксимації гіпотези $f(x; w)$, параметризованої вагами w . Вона відіграє роль зворотного зв'язку, що сигналізує моделі, наскільки добре вона наближає цільову функцію. У загальному випадку, для N прикладів навчальної вибірки, середнє значення функції втрат визначається як:

$$J(w) = \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i; w)), \quad (2.15)$$

де $J(w)$ – загальна функція вартості (cost function), яка мінімізується під час навчання,

$L(y_i, f(x_i; w))$ – функція втрат для одного прикладу.

Таким чином, навчання нейронної мережі полягає у знаходженні такого вектору параметрів w , який мінімізує середню похибку на всьому наборі даних:

$$w = \underset{w}{\operatorname{argmin}} J(w) \quad (2.16)$$

Функції втрат для регресійних задач

У регресійних моделях, де необхідно передбачити числове значення, найчастіше застосовуються такі функції втрат[18]:

– Середньоквадратична похибка (MSE, Mean Squared Error):

$$L(y, \hat{y}) = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2, \quad (2.17)$$

Ця функція надає велику вагу великим відхиленням, що робить її чутливою до викидів, але водночас ефективною для гладких задач оптимізації.

– Середня абсолютна похибка (MAE, Mean Absolute Error):

$$L(y, \hat{y}) = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|, \quad (2.18)$$

На відміну від MSE, MAE менш чутлива до великих відхилень і є більш стійкою до шуму.

Функції втрат для задач класифікації.

У випадку, коли завдання полягає у визначенні класу об'єкта, використовуються ймовірнісні підходи, що базуються на функціях крос-ентропії:

– Бінарна крос-ентропія:

$$L(y, \hat{y}) = \frac{-1}{N} \sum_{i=1}^N [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)], \quad (2.19)$$

Ця функція використовується для двокласових задач і стимулює модель до максимальної впевненості у правильних прогнозах.

– **Категоріальна крос-ентропія:**

$$L(y, \hat{y}) = \frac{-1}{N} \sum_{i=1}^N \sum_{c=1}^C y_{i,c} \log(\hat{y}_{i,c}), \quad (2.20)$$

де C – кількість класів, а $y_{i,c}$ – індикатор, що дорівнює 1, якщо зразок належить класу c .

– **Спеціалізовані функції втрат**

У задачах оцінювання поз людини до яких належить і аналіз постави, особливе значення має точність визначення координат ключових точок тіла – суглобів, орієнтирів та анатомічно значущих позицій. Для цього використовується різноманітні адаптації функцій втрат, яка вимірюють відхилення між передбаченими та істинними координатами точок[24].

На відміну від традиційних функцій втрат, що порівнюють скалярні значення або класи, Keypoint Loss враховує просторову природу даних, оскільки кожен об'єкт описується множиною координат у дво або тривимірному просторі. Формально для зображення з K ключовими точками функція втрат визначається як середнє квадратичне відхилення (або інша метрика відстані) між передбаченими $\hat{p}_k = (\hat{x}_k, \hat{y}_k)$ та еталонними $p_k = (x_k, y_k)$ координатами:

$$L_{keypoint} = \frac{1}{K} \sum_{k=1}^K |p_k - \hat{p}_k|_2^2, \quad (2.21)$$

де $|\cdot|_2$ – евклідова норма, що визначає відстань між точками у просторі.

У більш загальному випадку, якщо мережа генерує не координати, а теплові карти (heatmaps), що показують ймовірність знаходження ключової точки у кожному пікселі, то функція втрат визначається як відмінність між прогнозованими й цільовими тепловими картами. У цьому випадку застосовується середньоквадратична похибка між двовимірними матрицями:

$$L_{heatmap} = \frac{1}{K} \sum_{k=1}^K \sum_{i,j} \left(H_k(i,j) - \hat{H}_k(i,j) \right)^2, \quad (2.22)$$

де $H_k(i, j)$ – цільова теплокарта для точки k ;

$\widehat{H}_k(i, j)$ – прогнозована мережею.

Такий підхід дозволяє моделі враховувати не лише координати, а й просторову невизначеність у положенні точок. Це особливо важливо для даних, де точна локалізація може бути утруднена через шум, рух або часткове перекриття частин тіла.

Функція втрат у моделі Keypoint R-CNN є складною композицією декількох часткових функцій, кожна з яких відповідає за оптимізацію певного аспекту моделі – виявлення об’єкта, точність визначення його меж та локалізацію ключових точок[9].

Загальна функція втрат L для Keypoint R-CNN визначається як сума декількох компонентів:

$$L = L_{\text{cls}} + L_{\text{box}} + L_{\text{mask}} + L_{\text{keypoint}}, \quad (2.23)$$

де L_{cls} – функція втрат для класифікації об’єктів (використовується Cross-Entropy Loss);

L_{box} – функція втрат для регресії координат прямокутника, який обмежує об’єкт (Smooth L1 Loss);

L_{mask} – опціональна складова, що використовується в Mask R-CNN для сегментації об’єктів;

L_{keypoint} – функція втрат для передбачення координат ключових точок.

Остання складова, L_{keypoint} , є найбільш значущою для нашої задачі. Її метою є мінімізація різниці між передбаченими координатами ключових точок $(\hat{x}_i, \hat{y}_i)$ та їх істинними координатами (x_i, y_i) . У типовій реалізації Keypoint R-CNN для цього використовується Pixel-wise Sigmoid Cross-Entropy Loss між передбаченою та еталонною тепловою картою (heatmap).

Кожна ключова точка i представляється у вигляді 2D-теплової карти, де значення інтенсивності в кожному пікселі відповідає ймовірності наявності

ключової точки у даній позиції. Таким чином, функція втрат для ключових точок має вигляд:

$$L_{\text{keypoint}} = \frac{1}{N} \sum_{i=1}^N \text{BCE}(\widehat{H}_i, H_i), \quad (2.24)$$

де N – кількість ключових точок;

$\widehat{H}_i$ – передбачена тепловакарта для i -тої ключової точки;

H_i – істинна тепловакарта, що має гаусівське розподілення навколо істинної координати ключової точки;

BCE – бінарна крос-ентропія (Binary Cross-Entropy)(див. 2.19).

Іншими словами, модель не просто прогнозує координати ключових точок напряму, а навчається формувати теплові карти, які наближують просторовий розподіл ймовірностей для кожної ключової точки. Такий підхід має низку переваг:

- Підвищує стабільність навчання;
- Дозволяє враховувати невизначеність у локалізації;
- Забезпечує кращу здатність моделі до узагальнення.

У підсумку, функція втрат для Keypoint R-CNN поєднує класифікаційні та регресійні аспекти, забезпечуючи багатозадачне навчання, яке узгоджує виявлення об'єктів і точну локалізацію їхніх ключових точок у єдиній архітектурі.

Належно обрана функція втрат забезпечує не лише стабільність навчання, а й адекватну здатність моделі до узагальнення. Саме тому розробка і вибір функцій втрат є одним із ключових аспектів у побудові математичного забезпечення інформаційних систем на основі штучного інтелекту.

2.5 Функції активації.

У попередніх підрозділах було розглянуто постановку задачі машинного навчання як задачу оптимізації та описано методи, що дозволяють оновлювати параметри моделі. Наступним важливим аспектом математичного забезпечення

нейронних мереж є функції активації, які визначають, як нейрон реагує на вхідні сигнали, та безпосередньо впливають на здатність мережі моделювати складні нелінійні залежності.

Функція активації – це математичне перетворення, яке застосовується до зваженої суми вхідних сигналів нейрона. Вона визначає, наскільки сильно конкретний нейрон буде "активований", тобто наскільки його вихід впливатиме на наступні шари мережі. Формально, вихід нейрона можна записати у вигляді:

$$y = f(w^T x + b), \quad (2.25)$$

де x – вектор вхідних сигналів;

w – вектор вагових коефіцієнтів;

b – зсув (bias);

$f(\cdot)$ – функція активації.

Функції активації є невід’ємною складовою будь-якої нейронної мережі, адже саме вони надають моделі здатність відображати складні, нелінійні залежності між вхідними та вихідними даними.

Без функцій активації нейронна мережа, навіть з багатьма шарами, зводиться до лінійної моделі, що не може адекватно відтворювати реальні процеси, які зазвичай є нелінійними. Тому активаційні функції визначають обчислювальну виразність моделі, її стійкість до шумів, швидкість навчання та точність прогнозів.

У сучасних глибоких нейронних мережах застосовується низка різних функцій активації, вибір яких залежить від конкретної архітектури та задачі. Нижче наведено найпоширеніші з них[1].

Лінійна функція (Linear або Identity):

$$f(x) = x, \quad (2.26)$$

Це найпростіша форма активації, яка не змінює вхідних даних. Вона використовується переважно у вихідному шарі моделей, призначених для задач регресії, де потрібно зберегти безпосереднє числове співвідношення між вхідними та вихідними значеннями.

Сигмоїдна функція (Sigmoid):

$$f(x) = \frac{1}{1 + e^{-x}}, \quad (2.27)$$

Ця функція стискає вхідні значення у діапазон $[0,1]$, що дозволяє інтерпретувати вихід як ймовірність. Сигмоїда історично широко використовувалася в нейронних мережах, особливо у вихідних шарах для бінарної класифікації. Недоліком є схильність до насичення, що призводить до зникання градієнтів у глибших шарах.

Softmax:

$$f(x_i) = \frac{e^{x_i}}{\sum_{j=1}^K e^{x_j}}, \quad (2.28)$$

де K – кількість класів.

Функція Softmax перетворює вхідні значення на вектор ймовірностей, які сумуються до одиниці. Вона широко застосовується у вихідному шарі для задач багато класової класифікації, зокрема у нейронних мережах для розпізнавання об'єктів або ключових точок.

Прямолінійна випрямлена функція (ReLU – Rectified Linear Unit):

$$f(x) = \max(0, x), \quad (2.29)$$

Одна з найпопулярніших функцій активації у глибокому навчанні. ReLU пропускає лише додатні значення, встановлюючи всі від'ємні в нуль. Це забезпечує швидке навчання та розрідженість активацій, що підвищує ефективність обчислень. Проте існує проблема так званих мертвих нейронів, коли вага нейрона ніколи не оновлюється через нульовий градієнт.

Leaky ReLU:

$$f(x) = \begin{cases} x, & x > 0 \\ 0.01x, & x \leq 0 \end{cases} \quad (2.30)$$

Модифікація ReLU, що дозволяє пропускати невелике негативне значення для від'ємних вхідних сигналів. Це частково вирішує проблему "мертвих" нейронів і сприяє стабільнішому навчальному процесу.

ELU (Exponential Linear Unit):

$$f(x) = \begin{cases} x, & x > 0 \\ \alpha(e^x - 1), & x \leq 0 \end{cases} \quad (2.31)$$

Завдяки експоненційному нахилу у від'ємній частині ця функція забезпечує плавнішу зміну активацій та зменшує зміщення середніх значень, що покращує стабільність навчання.

SiLU (Sigmoid Linear Unit):

$$(x) = x \cdot \sigma(x) = \frac{x}{1 + e^{-x}} \quad (2.34)$$

Гладка, немонотонна функція активації, яка використовує механізм «самовентилляції» (self-gating), зважуючи вхідне значення його власною сигмоїдою. На відміну від ReLU, вона диференційована в усіх точках, що сприяє кращому проходженню градієнтів у глибоких мережах та підвищує ефективність навчання сучасних архітектури комп'ютерного зору (зокрема, у моделях сімейства YOLO).

Функції активації також регулюють передачу інформації між шарами нейронної мережі. Деякі функції, зокрема Sigmoid, стискають вихідні значення до певного діапазону, що дозволяє уникнути вибуху градієнтів під час навчання, проте водночас може спричиняти їх затухання. Інші функції, як-от ReLU та її похідні, зберігають більшу частину варіативності сигналу, забезпечуючи стабільну передачу інформації вглиб мережі без насичення.

Поведінка активаційної функції напряду впливає на ефективність навчання нейронної мережі. Якщо градієнти функції мають великі коливання або надто малі значення, алгоритм оптимізації може або не збігатися, або потребувати надмірної кількості епох. Функції типу ReLU виявилися надзвичайно успішними в глибоких моделях саме через їхню обчислювальну простоту та стабільність градієнтів. Вони дають ненульовий градієнт для додатних значень, забезпечуючи сталий потік інформації, що сприяє швидшому та ефективнішому навчанню.

Розділ 3. Інформаційне забезпечення

3.1 Згорткові нейронні мережі

Згорткові нейронні мережі (ЗНМ, англ. Convolutional Neural Network, CNN) – це спеціалізований клас глибоких штучних нейронних мереж прямого поширення, який став де-факто стандартом для задач аналізу візуальних даних. Запропонована Яном Лекуном та його колегами у праці “Gradient-based learning applied to document recognition”[8], архітектура LeNet-5 (рис. 3.1) була націлена на ефективне розпізнавання рукописних цифр шляхом імітації процесів у зоровій корі людини.

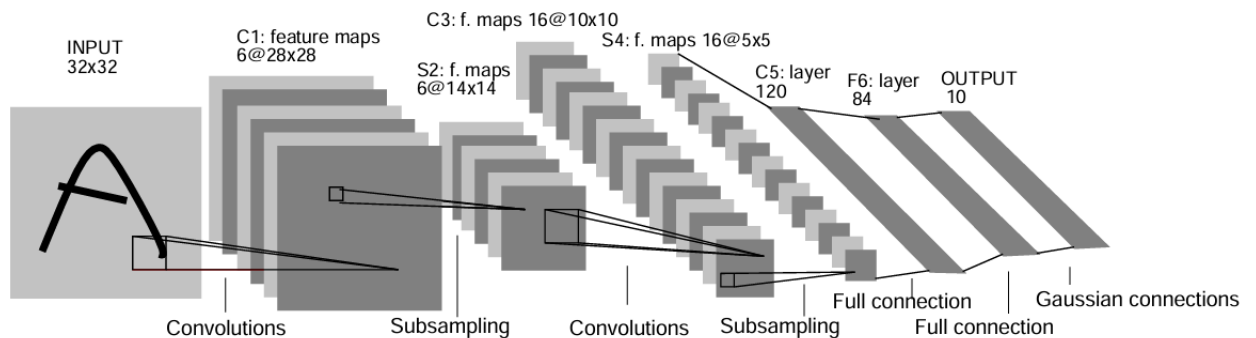


Рисунок 3.1 – Архітектура згорткової мережі LeNet-5

Подібно до того, як у зоровій корі головного мозку різні нейрони реагують на певні просторові ознаки в полі зору – такі як орієнтація ліній, контраст чи напрямок руху, – згорткові мережі використовують шари фільтрів, які «активуються» при наявності певних візуальних патернів. На ранніх рівнях мережі фільтри виявляють прості елементи (наприклад, грані або кути), а на наступних – складніші комбінації цих ознак, що відповідає ієрархічній структурі обробки зорової інформації у біологічних системах. Таким чином, CNN фактично моделюють принцип роботи нейронів у первинній зоровій корі, які спеціалізуються на розпізнаванні локальних характеристик зображення, поступово формуючи узагальнене уявлення про об’єкт.

На сьогоднішній день, ЗНМ входять до складу технологій глибокого навчання і продемонстрували видатні результати в завданнях класифікації, визначання об’єктів, сегментації зображень та багатьох інших. На відміну від

стандартних повнозв'язних мереж, які погано масштабуються для даних високої розмірності, архітектура ЗНМ базується на трьох фундаментальних принципах, що робить їх високоефективними[31].

- **Локальні рецептивні поля.** Кожен нейрон у згортковому шарі з'єднаний лише з невеликою локальною областю (рецептивним полем) нейронів попереднього шару. Це дозволяє мережі на ранніх етапах виявляти базові локальні ознаки (наприклад, грані, кути, кольорові плями).
- **Спільне використання ваг** полягає у тому, що один і той самий набір ваг, який називається ядром або фільтром (kernel), застосовується до *всіх* локальних рецептивних полів на вхідному зображенні. Цей процес (операція згортки) дозволяє фільтру виявляти одну й ту саму ознаку (наприклад, вертикальну лінію) незалежно від її положення на зображенні. Це забезпечує трансляційну інваріантність та кардинально зменшує загальну кількість параметрів, що навчаються, порівняно з повнозв'язними мережами.
- **Ієрархічне представлення ознак.** ЗНМ будуються шляхом послідовного застосування кількох шарів. Перші шари вчать розпізнавати прості ознаки (грані). Наступні шари комбінують ці прості ознаки для формування складніших (текстури, прості форми). Глибші шари комбінують їх далі, формуючи уявлення про об'єкти або їхні частини (очі, колеса автомобілів тощо).

Згорткові нейронні мережі сприймають та обробляють дані у формі тензорів[22]. На рисунку 3.2 наведено візуалізацію представлення вхідного зображення для згорткової нейронної мережі. Зображення обробляється як тривимірний тензор (або масив даних), що має три виміри: ширину (позначено як I_1), висоту (I_2) та глибину (I_3). Для стандартного кольорового зображення, як-от у прикладі, глибина I_3 дорівнює 3 та відповідає трьом окремим колірним каналам: червоному (Red), зеленому (Green) та синьому (Blue). Саме такий формат даних, де зберігається просторова та канална інформація, дозволяє

згортковим шарам мережі ефективно оперувати з візуальною високо розмірною інформацією.

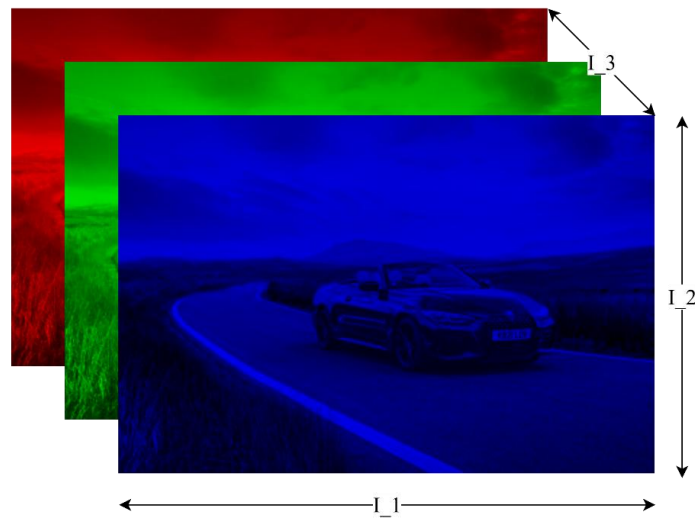


Рисунок 3.2 – Представлення вхідного зображення

Загальна структура ЗНМ (продемонстрована на прикладі VGG16 на рис. 3.3) має вхідний шар, серію прихованих шарів та вихідний шар. Приховані шари зазвичай формують "тіло" мережі (вилучення ознак) і складаються з чергування згорткових шарів, функцій активації та шарів агрегування. "Голова" мережі (класифікатор) зазвичай складається з повнозв'язних шарів, які виконують фінальну класифікацію. Далі буде розглянуто ці основні елементи ЗНМ.

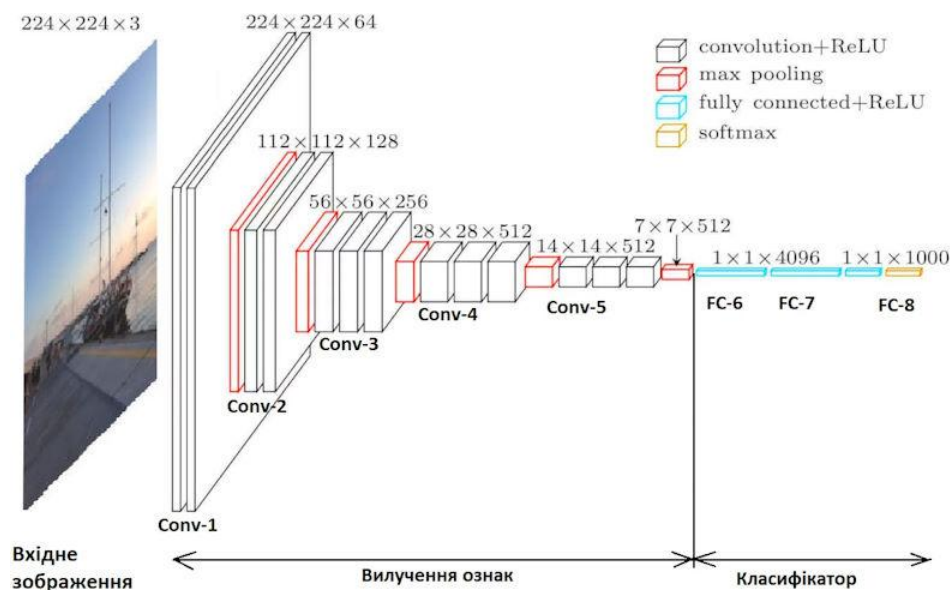


Рисунок 3.3 – Структура згорткової нейронної мережі

3.1.1 Згорткові шари

Згортковий шар є ядром всієї архітектури. Його головна задача – виявлення ознак у вхідних даних за допомогою операції згортки.

Згортка – це математична операція над двома матрицями: вхідною матрицею A (розміру $n_x \times n_y$) і ядром B (розміру $m_x \times m_y$), результатом якої є матриця $C = A \cdot B$. Фактично, ядро (фільтр) "ковзає" по вхідному зображенню, на кожній позиції виконуючи поелементне множення з подальшим сумуванням результатів[6]. Отримане скалярне значення стає елементом нової матриці, що називається картою ознак (feature map).

Ядро є набором параметрів, що навчаються. Кожен фільтр налаштовується під час тренування на виявлення специфічної ознаки (наприклад, певний кут, колірний перехід, текстура). Кількість фільтрів може бути гіперпараметром мережі, впливаючи на кількість потенційно виявлених ознак. Кількість фільтрів визначає кількість карт ознак, що визначає глибину вихідних даних та впливає на кількість параметрів на наступному рівні.

Гіперпараметрами згорткового шару є:

- **Розмір ядра (Kernel Size):** Визначає розмір рецептивного поля (наприклад, 3×3 , 5×5). Сучасні архітектури надають перевагу малим ядрам 3×3 , оскільки два послідовних 3×3 шари мають таке ж ефективне рецептивне поле, як один 5×5 , але з меншою кількістю параметрів і більшою нелінійністю.
- **Крок (Stride):** Визначає, на скільки пікселів ядро зсувається за один крок. Крок $S = 1$ означає повний обхід, $S > 1$ призводить до зменшення просторової розмірності вихідної карти ознак (downsampling).
- **Доповнення (Padding):** Контролює розмір вихідної карти. Без доповнення (VALID padding) карта ознак зменшується після кожної згортки. Щоб зберегти просторові розміри, часто використовують доповнення нулями (SAME padding) по краях вхідного зображення.

Загальний процес перетворень у згортковому шарі (для одного фільтра) описується рівнянням:

$$y = f(w * x + b), \quad (3.1)$$

де y – вихідна карта ознак;

w – ядро (фільтр);

x – вхідні дані;

b – параметр зсуву (bias);

f – нелінійна функція активації.

На рисунку 3.4 детально продемонстровано механізм роботи операції згортки, яка є центральною для згорткового шару. На схемі показано, як згорткове ядро (convolutional kernel) – у даному випадку, матриця розміром 3×3 з навченими вагами – застосовується до вхідної карти ознак (Source layer). Ядро "ковзає" по вхідному шару, і на кожній позиції виконується по елементне множення значень ядра на відповідні значення у локальному рецептивному полі вхідного шару. Як показано у розрахунку нижче, всі отримані добутки підсумовуються $(-1 \times 5) + (0 \times 2) + \dots + (0 \times 2) = 5$, формуючи єдине вихідне значення. Це значення стає елементом нової карти ознак (Destination layer). Цей процес повторюється для всієї вхідної карти, що дозволяє ядру систематично виявляти специфічну ознаку, на яку воно налаштоване, по всьому зображенню.

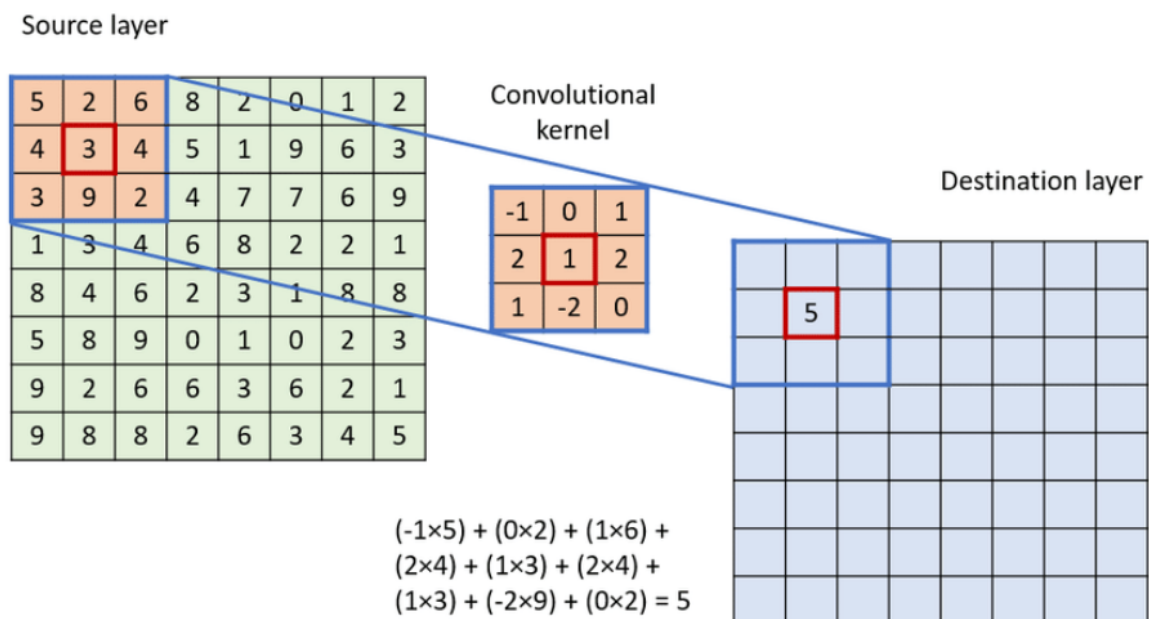


Рисунок 3.4 – Операція згортки

3.1.2 Агрегувальні шари

Агрегувальні шари, також відомі як шари субдискретизації або шари пулінгу (Pooling Layers), є невід'ємною складовою архітектури згорткових нейронних мереж поряд із згортковими шарами. Їх основне призначення полягає у зменшенні розмірності проміжних представлень даних при збереженні найважливіших ознак, що були виділені на попередніх етапах обробки. Такий підхід дає змогу зменшити кількість параметрів моделі, знизити обчислювальні витрати та запобігти перенавчанню, що є особливо важливим при роботі з великими обсягами візуальної інформації.

Пулінгові шари виконують операцію агрегування груп сусідніх нейронів із попереднього шару у єдиний представницький елемент нового шару. Це дозволяє мережі формувати більш узагальнене представлення просторових ознак, фокусуючись на ключових характеристиках об'єкта[22]. Важливо зазначити, що при цьому глибина карти ознак зберігається, тоді як її просторові розміри (висота та ширина) зменшуються. Зазвичай пулінг застосовується після однієї або кількох операцій згортки, що забезпечує баланс між точністю розпізнавання та ефективністю обчислень.

Найбільшого вжитку мають 2 типи пулінгу(рис. 3.5): максимальний (Max Pooling) та середній (Average Pooling).

У випадку Max Pooling з кожної області рецептивного поля вибирається максимальне значення. Такий підхід дозволяє мережі зосереджуватись на найбільш виразних ознаках, підвищуючи інваріантність до зсувів або деформацій об'єкта. У результаті формується карта ознак, яка містить найбільш значущі елементи попередньої карти.

Average Pooling, навпаки, обчислює середнє арифметичне значення елементів рецептивного поля. Цей метод створює більш згладжене представлення ознак, зменшуючи вплив випадкових шумів та локальних коливань інтенсивності.

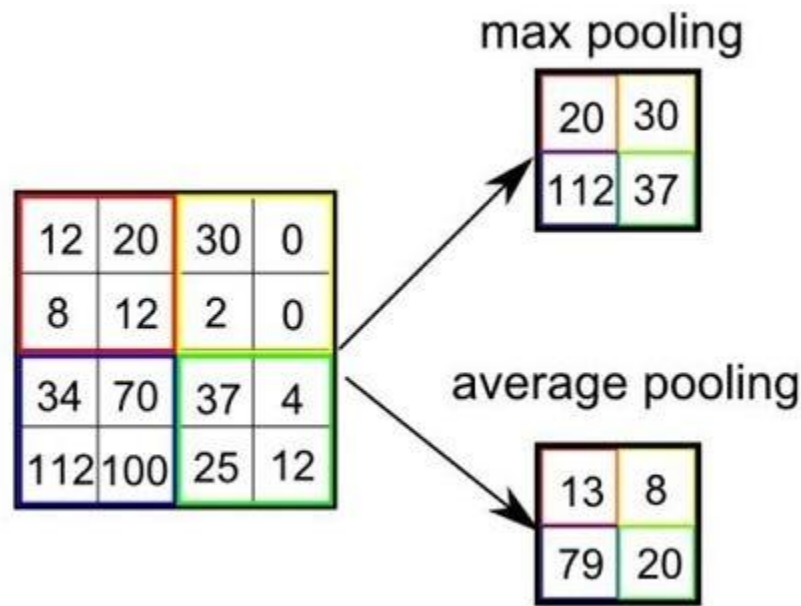


Рисунок 3.5 – Приклад роботи max та average pooling

Завдяки своїй здатності узагальнювати локальні просторові залежності, пулінгові шари підвищують стійкість моделі до збурень у вхідних даних – зокрема, до змін масштабу, кута огляду, положення чи освітлення об’єкта. Таким чином, навіть при незначних варіаціях вхідного зображення, виділені ключові ознаки залишаються стабільними, що сприяє кращій здатності мережі до генералізації.

З технічної точки зору, пулінгові шари мають ряд параметрів, аналогічних до параметрів згорткових операцій: розмір рецептивного поля, крок та можливість застосування доповнення нулями. Найчастіше використовується вікно розміром 2×2 із кроком 2, що забезпечує ефективне зменшення розмірності вдвічі при кожній операції пулінгу.

У сучасних архітектурах згорткових нейронних мереж іноді відмовляються від явних пулінгових шарів, замінюючи їх згортковими шарами з підвищеним кроком. Такий підхід дозволяє інтегрувати процеси згортки та зменшення розмірності в єдину операцію, що спрощує архітектуру мережі без втрати продуктивності. Проте класичні пулінгові шари залишаються важливим інструментом побудови глибоких моделей, особливо у випадках, коли необхідна висока інваріантність до локальних змін у зображеннях.

3.1.3 Повнозв'язні шари

Повнозв'язні (англ. Fully Connected) шари є завершальним етапом у класичній архітектурі згорткових нейронних мереж. Їх головна функція полягає у здійсненні високорівневої інтеграції ознак, виділених попередніми згортковими та агрегаційними шарами, з метою формування підсумкового рішення моделі – наприклад, класифікації, регресії чи прогнозування координат об'єктів.

На відміну від згорткових і пулінгових шарів, які працюють із просторовими структурами даних, повнозв'язні шари оперують одномірними векторами. Перед їх використанням тривимірні карти ознак, сформовані на попередніх етапах, «розгортаються» (flattening) у вектор ознак фіксованої довжини[6]. Кожен нейрон у повнозв'язному шарі має зв'язки з усіма нейронами попереднього шару, завдяки чому формується повна матриця ваг.

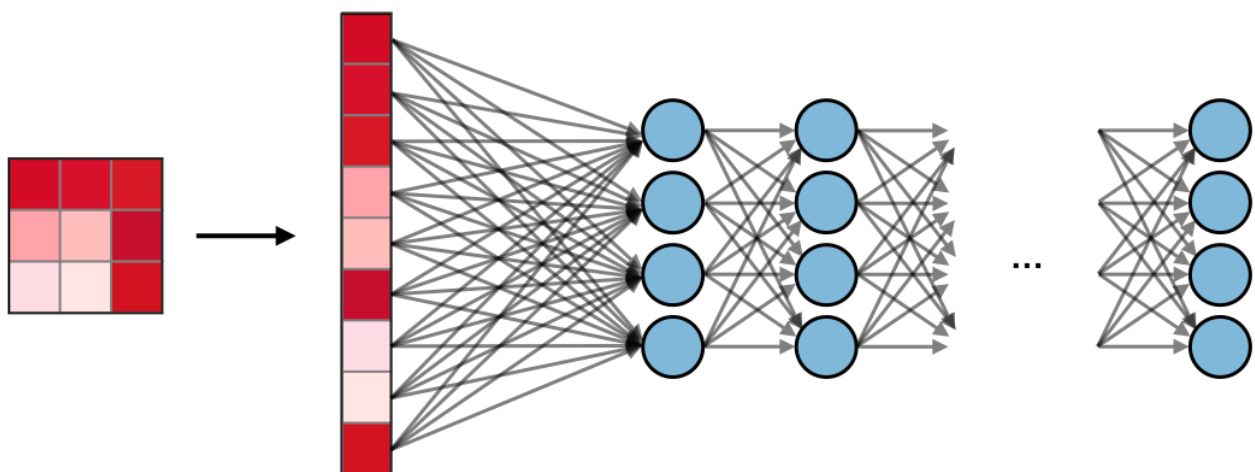


Рисунок 3.6 – Повнозв'язні шари згорткової мережі

На рисунку 3.6 схематично зображено цей процес. Вихідна карта ознак з останнього згорткового або агрегуючого шару (ліворуч, умовно 3×3) проходить операцію вирівнювання (Flattening). В результаті цієї операції, всі елементи матриці послідовно "розгортаються" в один довгий одномірний вектор (в центрі). Цей вектор ознак, який тепер не має просторової структури, але містить усю вилучену інформацію, подається на вхід першому повнозв'язному шару (праворуч), де кожен його елемент з'єднується з кожним нейроном наступного шару.

У процесі навчання повнозв'язні шари виконують роль нелінійного класифікатора, який навчається знаходити оптимальні комбінації ознак для досягнення поставленої мети. Зазвичай після кількох таких шарів додається вихідний шар, що використовує специфічну функцію активації відповідно до типу задачі: Softmax – для багатокласової класифікації, Sigmoid – для бінарної, або лінійну функцію – для регресійних моделей.

З математичної точки зору, вихід повнозв'язного шару можна описати як лінійну комбінацію вхідного вектора та матриці ваг із додаванням зсуву (bias):

$$z = W \cdot x + b, \quad (3.2)$$

де x – вхідний вектор ознак;

W – матриця ваг розмірності $n \times m$;

b – вектор зсувів.

Повнозв'язні шари мають велику кількість параметрів, що зростає пропорційно добутку кількості нейронів у суміжних шарах. Це робить їх потенційно схильними до перенавчання, особливо при роботі з невеликими наборами даних.

Роль повнозв'язних шарів у структурі згорткової нейронної мережі можна порівняти з етапом прийняття рішень у системі розпізнавання: якщо згорткові та пулінгові шари виступають у ролі механізму «зору», який виділяє важливі візуальні патерни, то повнозв'язні шари діють як «інтелектуальний модуль», що інтерпретує отриману інформацію та приймає фінальне рішення на її основі.

У сучасних архітектурах глибокого навчання роль повнозв'язних шарів поступово зменшується на користь адаптивних глобальних пулінгових операцій (Global Average Pooling, Global Max Pooling), які виконують подібну функцію з меншою кількістю параметрів[14]. Проте повнозв'язні шари залишаються невід'ємною частиною класичних моделей і широко застосовуються в задачах класифікації, оцінки ключових точок, визначення об'єктів та генеративних мережах.

3.2 Метод зворотнього поширення помилки

У попередніх підрозділах було розглянуто архітектуру згорткових нейронних мереж, зокрема їх основні структурні компоненти – згорткові, агрегаційні та повнозв'язні шари,. Проте ефективність роботи нейронної мережі визначається не лише її архітектурою, а й способом навчання, тобто процесом налаштування вагових коефіцієнтів таким чином, щоб мінімізувати похибку між фактичними та очікуваними результатами. Основним алгоритмом, який забезпечує цю оптимізацію, є метод зворотного поширення помилки (Backpropagation).

Метод зворотного поширення помилки є базовим механізмом навчання багатошарових нейронних мереж, і ґрунтується на використанні градієнтного спуску, який детально описаний у розділі 2.2, для мінімізації функції втрат, які описані у розділі 2.4. У межах цього методу ваги мережі оновлюються пропорційно до часткової похідної функції втрат за кожним параметром, що дозволяє поступово зменшувати помилку передбачення на навчальній вибірці.

Загальна мета навчання полягає у мінімізації функціонала помилки:

$$J(\theta) = \frac{1}{N} \sum_{i=1}^N \mathcal{L}(y_i, \hat{y}_i) \quad (3.3)$$

де $\hat{y}_i$ – вихід моделі для i -го прикладу;

y_i – істинна мітка;

L – часткова функція втрат для одного прикладу;

θ – вектор параметрів мережі (ваг та зміщень);

N – кількість прикладів у навчальній вибірці.

Для обчислення градієнтів функції втрат за вагами багатошарової мережі використовується принцип ланцюгового диференціювання, який забезпечує поширення сигналу помилки у зворотному напрямку – від вихідного шару до вхідного. На відміну від прямого проходу, під час якого сигнали передаються вперед через шари мережі для формування прогнозу, у фазі зворотного

поширення кожен шар отримує похибку від наступного шару і обчислює власний внесок у загальну помилку[7].

На математичному рівні процес корекції ваг здійснюється за правилом:

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} - \eta \frac{\partial L}{\partial w_{ij}}, \quad (3.4)$$

де w_{ij} – вага між нейронами i та j ;

η – коефіцієнт навчання (learning rate);

$\frac{\partial L}{\partial w_{ij}}$ – часткова похідна функції втрат за даним параметром.

На рисунку 3.7 зображено схему ітеративного процесу навчання штучної нейронної мережі за допомогою алгоритму зворотного розповсюдження помилки. Процес починається з прямого розповсюдження, під час якого вхідні дані проходять через шари мережі (вхідний, прихований та вихідний) з поточними вагами (w_k) для формування вихідного значення. Далі обчислюється помилка, за квадратичною функцією втрат. Якщо помилка є вищою за поріг, запускається етап зворотного розповсюдження помилки, який обчислює градієнт функції втрат по відношенню до кожної ваги ($\nabla F(w_k)$). На основі отриманих градієнтів виконується крок оновлення, що коригує ваги мережі ($w_k \Rightarrow w_{k+1}$) для мінімізації помилки на наступній ітерації.

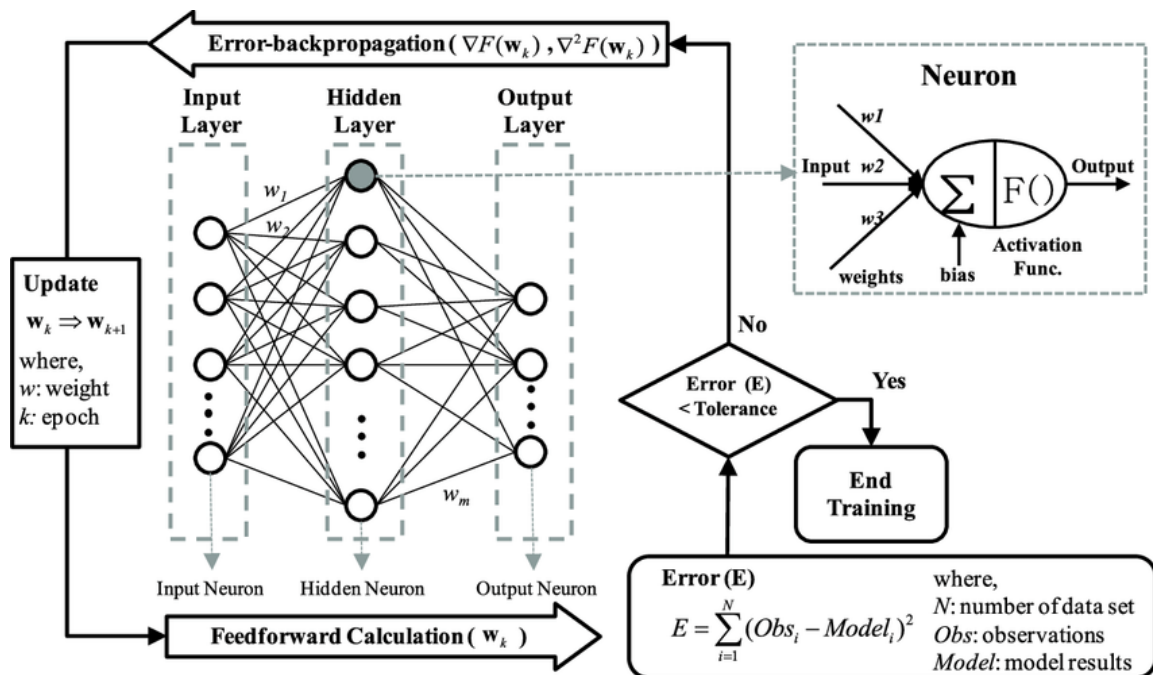


Рисунок 3.7 – Схематична діаграма алгоритму навчання зворотного поширення

3.3 Задача знаходження ключових точок на зображеннях

Однією з базових задач комп'ютерного зору, що безпосередньо пов'язана з використанням згорткових нейронних мереж є задача знаходження ключових точок на зображеннях.

Сутність цієї задачі полягає у визначенні координат характерних орієнтирів на об'єктах, зображених на фото або відео. Такі точки, що називаються ключовими, виступають структурними маркерами, які дозволяють машині ідентифікувати форму, положення або орієнтацію об'єкта в просторі [24].

У найпоширенішому випадку, коли об'єктом аналізу є людина, задача знаходження ключових точок збігається із задачею оцінки пози людини (pose estimation). Вона полягає у визначенні просторового розташування частин тіла – голови, тулуба, рук, ніг тощо (рис. 3.8). Аналогічні підходи застосовуються для розпізнавання обличчя (визначення координат очей, носа, рота).

Ключові точки – це визначені, анатомічні або структурні орієнтири, що мають семантичне значення. Наприклад, для людського тіла це можуть бути суглоби (лікоть, коліно, зап'ястя), для обличчя – кутики очей, кінчик носа, кутики губ, а для автомобіля – центри коліс, кути фар тощо. Ці точки у своїй сукупності описують позу, конфігурацію або стан деформації об'єкта.

Знаходження ключових точок це перший крок для вирішення багатьох прикладних задач високого рівня. До основних сфер застосування належать[12]:

- **Оцінка пози людини (Human Pose Estimation):** Аналіз рухів у спорті, моніторинг пацієнтів у медицині, системи віртуальної реальності (VR) та доповненої реальності (AR), безконтактне керування інтерфейсами.
- **Розпізнавання обличчя та емоцій:** Локалізація ключових точок обличчя є основою для систем ідентифікації особи, аналізу міміки та розпізнавання емоцій.
- **Робототехніка та автономні системи:** Роботизовані маніпулятори використовують ключові точки для захоплення та взаємодії з об'єктами.

Безпілотні автомобілі можуть використовувати їх для відстеження пішоходів та інших учасників руху.

- **Відстеження об'єктів (Object Tracking):** Визначення ключових точок дозволяє надійно відстежувати нежорсткі об'єкти, такі як люди або тварини, у часовій послідовності кадрів.
- **3D-реконструкція:** Використовуючи 2D ключові точки, отримані з декількох ракурсів, можна відновити тривимірну (3D) модель об'єкта.

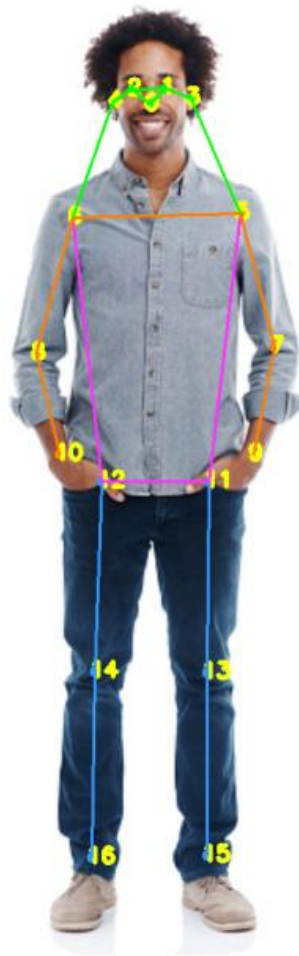


Рисунок 3.8 – Ключові точки на тілі людини

Ключові точки на зображенні визначаються на основі унікальних локальних ознак, які роблять їх легко відмінюваними від навколишніх пікселів. До основних властивостей ключових точок належать:

- **Унікальність:** здатність чітко виділятися серед інших ділянок зображення завдяки особливим характеристикам (яскравість, контраст, текстура);

- **Інваріантність:** стійкість до змін масштабу, обертання та умов освітлення;
- **Повторюваність:** можливість стабільного виявлення одних і тих самих точок при різних ракурсах або кадрах тієї ж сцени.

Ефективне розв’язання задачі визначення(детекції) ключових точок потребує від моделі високої стійкості до значних варіацій у вхідних даних. Одним із ключових викликів є оклюзія, коли частина орієнтирів може бути частково або повністю прихована іншими об’єктами або частинами самого тіла (так звана самооклюзія). У таких випадках система має або вірно прогнозувати положення невидимої точки, або позначати її як відсутню.

Іншою складністю виступає деформація та варіативність пози, адже об’єкти, що досліджуються – зокрема люди чи тварини – є нежорсткими і можуть набувати практично необмеженої кількості поз. Це створює складні просторові взаємозв’язки між ключовими точками, які модель має навчитися відтворювати незалежно від пози або конфігурації тіла.

Не менш суттєвим фактором є зміна ракурсу та масштабу. Об’єкт може бути зафіксований з різних напрямів – фронтально, збоку чи зверху – а також перебувати на різній відстані від камери[20]. Тому модель повинна зберігати інваріантність до таких геометричних перетворень, забезпечуючи стабільність результатів за будь-якого положення в просторі.

Додаткові труднощі створюють варіації освітлення. Різкі тіні, змінна інтенсивність світла, низький контраст або зйомка в умовах недостатнього освітлення суттєво ускладнюють виділення контурів та орієнтирів, що негативно впливає на точність визначення координат ключових точок.

Останнім, але не менш важливим аспектом є перевантажені сцени. У випадках, коли на зображенні присутня велика кількість людей, модель повинна не лише точно визначити всі ключові точки, а й правильно згрупувати їх, встановивши відповідність між кожною точкою та конкретною особою. Ця підзадача, відома як *multi-person pose estimation*, потребує поєднання детекції, асоціації та просторового аналізу, що робить її однією з найскладніших у сфері комп’ютерного зору[19].

3.4 Методи знаходження ключових точок на зображеннях

Історично розвиток методів визначення ключових точок на зображеннях пройшов шлях від класичних алгоритмів, заснованих на ручній інженерії ознак, до сучасних підходів глибокого навчання. У ранній період ця задача розв'язувалася переважно з використанням методів комп'ютерного зору, які спиралися на пошук локальних дескрипторів та аналіз градієнтних структур. Серед найвідоміших підходів можна відзначити SIFT (Scale-Invariant Feature Transform), SURF (Speeded-Up Robust Features) та HOG (Histogram of Oriented Gradients). Ці методи дозволяли виділяти стійкі точки на зображеннях – наприклад, кути або текстурні деталі – які зберігали свої характеристики при масштабуванні, поворотах або зміні освітлення. Однак такі точки не мали семантичного значення: алгоритм міг виявити «кут» або «ребро», але не визначав, чи відповідає ця точка певній анатомічній частині, наприклад, «лікоть» чи «коліно»[12].

Більш складні класичні підходи, як-от Deformable Part Models (DPM), намагалися моделювати просторові зв'язки між частинами об'єкта, проте їх точність та узагальнювальна здатність залишалися обмеженими. Вони були чутливими до варіацій у позі, деформацій та часткових перекриттів, що унеможливлювало їх ефективне застосування в реальних умовах.

Ситуація кардинально змінилася після появи згорткових нейронних мереж яка започаткувала нову епоху методів на основі глибокого навчання. Сучасні підходи до визначення ключових точок умовно поділяються на дві основні групи – регресійні моделі та моделі на основі теплових карт.

У регресійних методах нейронна мережа розглядається як функція, що безпосередньо перетворює зображення у набір координат ключових точок. На вхід подається зображення, після чого згорткова частина мережі (backbone), наприклад ResNet або MobileNet, видобуває високорівневі ознаки. Отриманий вектор ознак передається у повнозв'язні шари, які прогнозують координати

кожної точки. Такий підхід є надзвичайно швидким та ефективним з точки зору обчислень, що робить його придатним для використання на мобільних пристроях або в системах реального часу. Проте він має певні обмеження – через втрату просторової структури даних точність прогнозів зазвичай нижча, а навчання ускладнюється через необхідність апроксимації сильно нелінійних залежностей між піксельним простором та координатним простором ключових точок.

На відміну від цього, підходи на основі теплових карт (heatmap-based) стали домінуючими завдяки своїй високій точності. У таких моделях мережа не прогнозує координати напряду, а генерує набір двовимірних теплових карт – по одній для кожної ключової точки. Кожен піксель теплової карти відображає ймовірність того, що відповідна точка розташована в даній позиції. Під час навчання для кожної цільової точки створюється тепла карта, яка моделюється як двовимірна функція Гауса, центрована в істинній координаті. На етапі передбачення координати точки отримуються через пошук максимуму значення на тепловій карті.

Методи на основі теплових карт забезпечують високу точність завдяки збереженню просторової інформації, стійкості до оклюзій та здатності відображати невизначеність у прогнозах. Проте вони потребують значних обчислювальних ресурсів та пам'яті, особливо при генерації теплових карт високої роздільної здатності. Крім того, операція argmax не є диференційованою, що ускладнює процес зворотного поширення градієнта; для подолання цього обмеження іноді використовуються альтернативи, такі як soft-argmax або функції втрат, побудовані на піксельному рівні.

Таким чином, еволюція методів від ручного конструювання ознак до глибоких згорткових мереж суттєво підвищила ефективність і точність систем детекції ключових точок, відкривши шлях до сучасних архітектур, здатних працювати у складних реальних умовах із високим рівнем узагальнення.

Розглянемо дві сучасні архітектури, що широко застосовуються для задачі знаходження ключових точок: модель YOLO v11 (зокрема її режим «pose / keypoints») та Keypoint R-CNN.

3.4.1 YOLO v11

YOLO (You Only Look Once) – це клас архітектур для задачі об'єктного детектування, які побудовані за принципом одиничного проходу («single-stage») через зображення. У версії v11 автори продовжили покращення архітектури, що дозволило пришвидшити швидкодію моделі порівняно з минулими версіями[32].

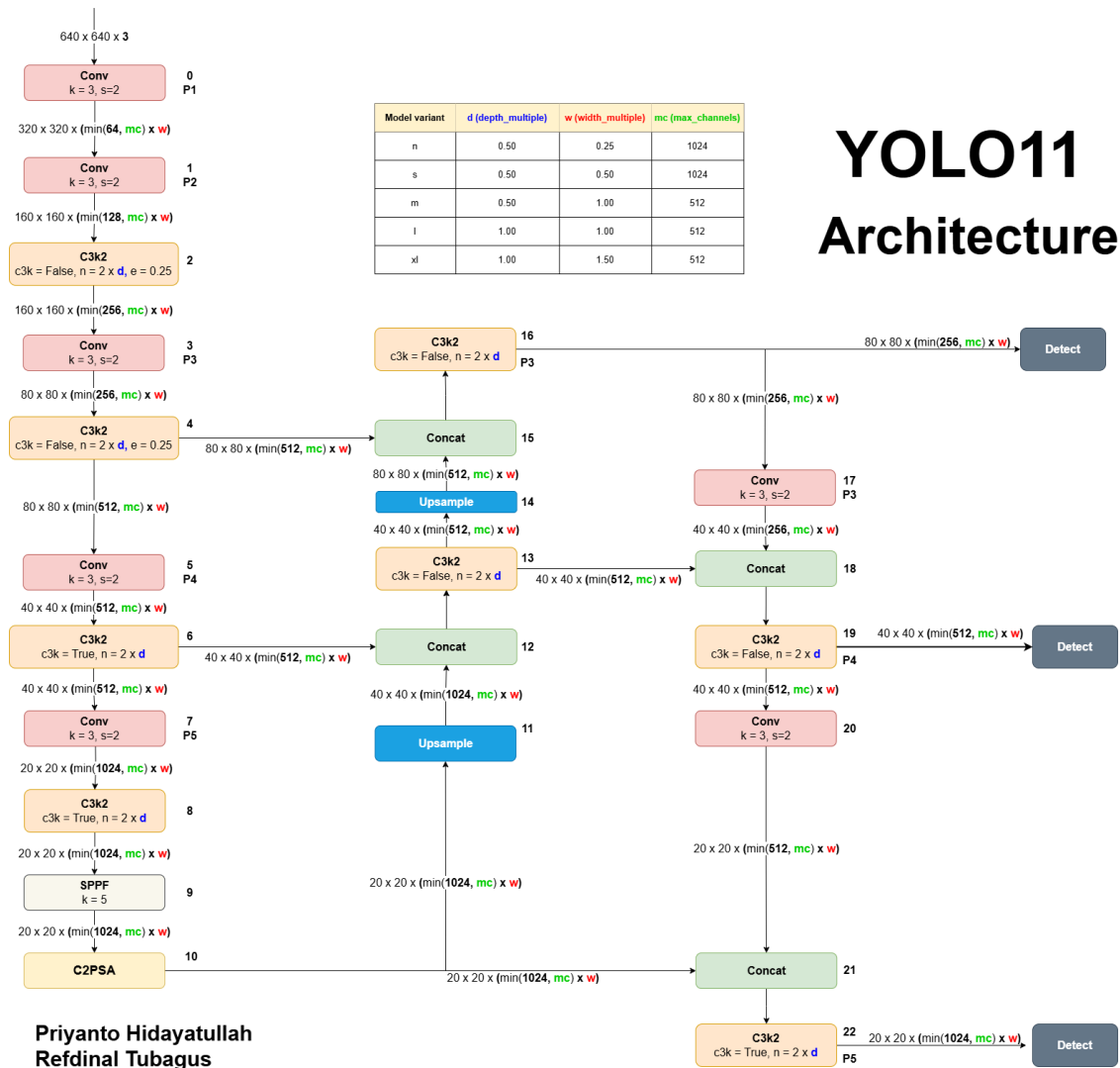


Рисунок 3.9 – Архітектура Yolo 11

Архітектурно YOLO v11 складається з трьох основних частин(рис. 3.9)[26]. "Хребет" (Backbone) приймає вхідне зображення 640x640 і послідовно зменшує його просторову роздільну здатність за допомогою Conv-шарів, одночасно поглиблюючи представлення ознак за допомогою блоків C3k2. "Хребет" завершується блоками SPPF (Spatial Pyramid Pooling Fast) та C2PSA для генерації багатомасштабних контекстуальних ознак на рівнях P3, P4 та P5.

Далі "Шия" (Neck) реалізує архітектуру типу PANet (Path Aggregation Network). Вона спочатку йде "зверху-вниз", поєднуючи за допомогою Upsample та Concat семантично багаті ознаки з P5 та P4 з просторово точними ознаками P3. Потім вона негайно будує шлях "знизу-вгору", знову агрегуючи ознаки, щоб забезпечити потужне представлення на всіх трьох вихідних рівнях.

Нарешті, "Голова" (Head) складається з трьох Detect блоків, кожен з яких підключений до одного з виходів "ший" (80x80, 40x40, 20x20). Це дозволяє моделі виявляти об'єкти різних розмірів на відповідних масштабах. Для адаптації цієї архітектури під задачу знаходження ключових точок (Pose Estimation), до неї додається паралельна "голова пози". Ця голова, замість використання теплових карт, виконує пряму регресію $K \times 3$ значень (x, y, v) для кожної з K ключових точок, що дозволяє зберегти високу швидкість роботи моделі[32].

Модель оптимізована для реального часу, архітектура v11 зменшує кількість параметрів у порівнянні з попередніми версіями, одночасно покращуючи точність (mAP) та швидкодію.

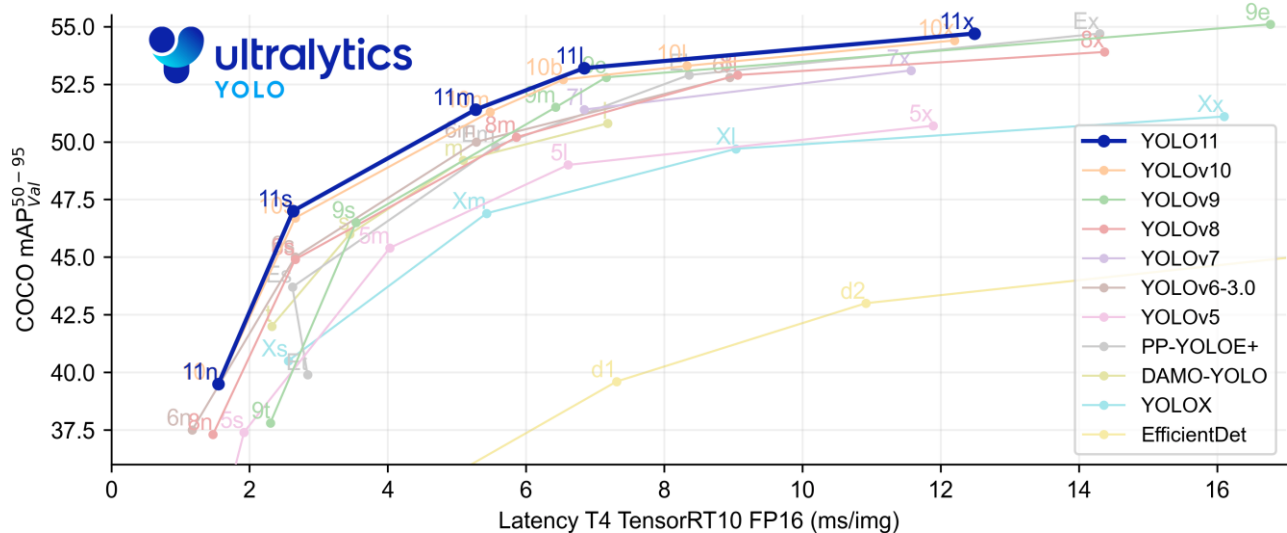


Рисунок 3.10 – Порівняння швидкодії моделей сімейства YOLO

На рисунку 3.10 представлено порівняльну діаграму, що ілюструє співвідношення між точністю детекції та швидкістю обробки для різних моделей, включно з серією YOLO11 та її попередниками і конкурентними архітектурами. Горизонтальна вісь відображає затримку (Latency) в мілісекундах на зображення на платформі T4 TensorRT10 FP16, що є метрикою швидкодії (менше – краще).

Вертикальна вісь показує точність, виміряну як COCO mAP з порогоми IoU від 0.50 до 0.95 (вище – краще). Модель YOLO11x досягає найвищої точності при затримці ~11.8 мс, перевершуючи за цим компромісом моделі YOLOv10-x та YOLOv9e. Аналогічно, легкі моделі, як-от YOLO11n, забезпечують конкурентну точність при значно нижчій затримці порівняно з іншими моделями.

Використання YOLO v11 у режимі оцінки поз дозволяє використовувати модель у мобільних пристроях, системах вбудованого відеоаналізу, спортивній аналітиці та медичних застосуваннях.

Разом із тим слід зазначити, що у задачі точного аналізу постави, коли необхідна висока просторово-семантична деталізація, можуть виникати обмеження: зокрема, одно стадійний підхід YOLO може поступатися багатоступеневим методам у деталізації позицій або у випадках, коли необхідне дуже високе розрізнення між ключовими точками.

3.4.2 Keypoint R-CNN

Одним із найбільш відомих і впливових підходів до задачі детекції ключових точок є архітектура Keypoint R-CNN, яка розвиває ідеї моделі Mask R-CNN, запропонованих у однойменній роботі[14]. Вона поєднує переваги рекурентних згорткових нейронних мереж (R-CNN) із додатковою гілкою, яка спеціалізується на локалізації ключових точок людського тіла. На відміну від повністю одно стадійних моделей, Keypoint R-CNN реалізує двоетапну архітектуру, що забезпечує високу точність та структуроване розуміння об'єкта на зображенні[11].

Архітектура Keypoint R-CNN (рис. 3.11 та рис. 3.12) базується на принципах ієрархічного поділу задачі: спочатку мережа виявляє області, які потенційно містять об'єкти (у випадку задачі оцінки пози – людей), а потім для кожного з детектованих регіонів виконує точну локалізацію набору ключових точок. Такий підхід дозволяє мережі ефективно працювати в умовах часткових перекриттів, складних поз та варіативності масштабу.

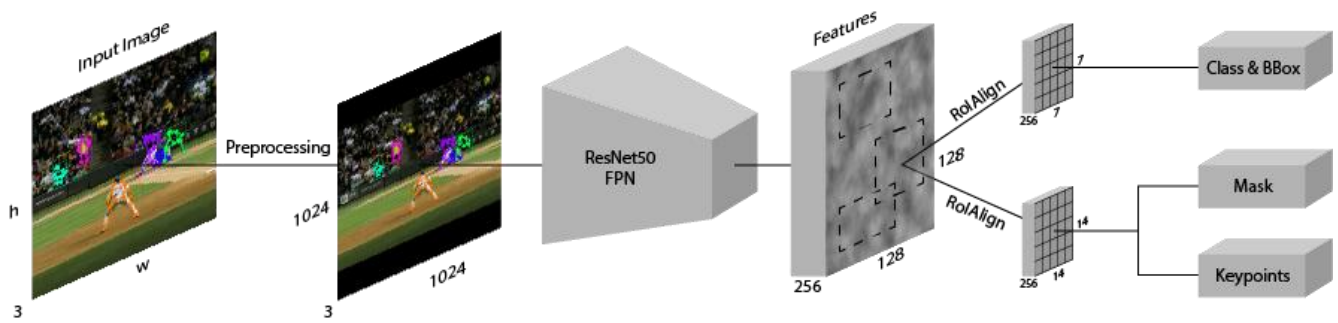


Рисунок 3.11 – Схематична архітектура keypoint rcnn

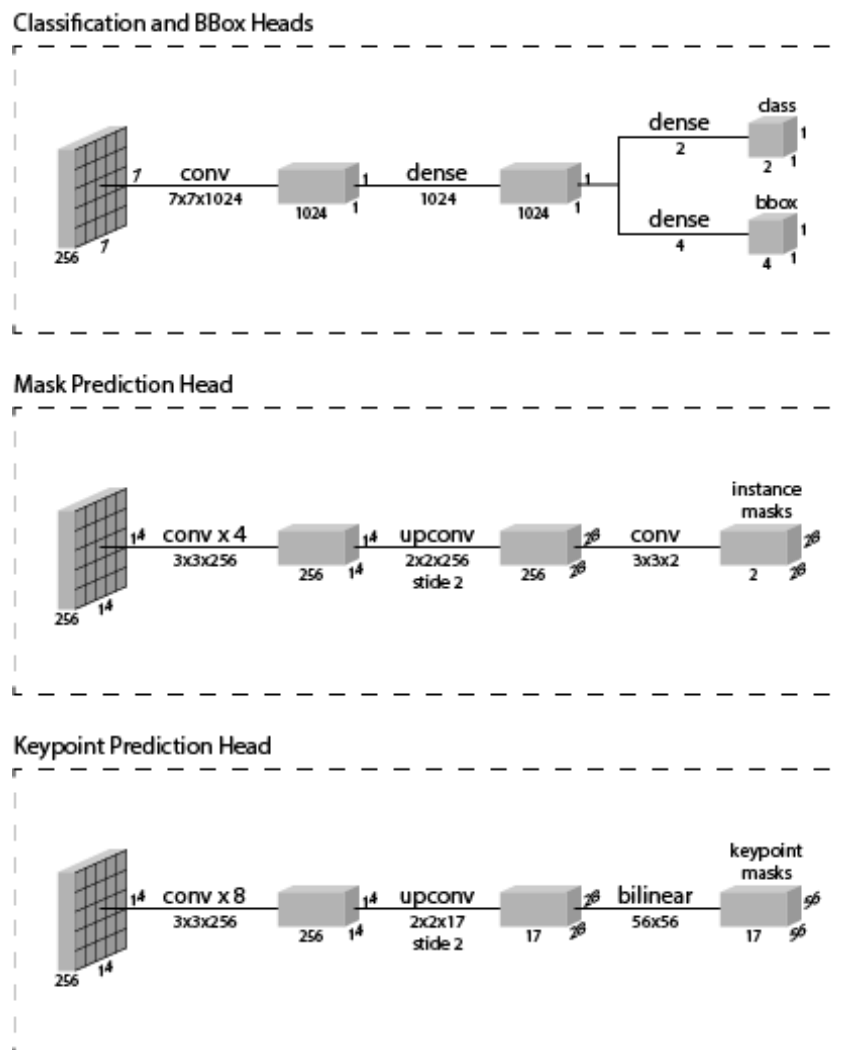


Рисунок 3.12 – Будова різних prediction head

У загальному вигляді Keypoint R-CNN складається з кількох основних компонентів:

- **Backbone** – базова згорткова мережа ResNet-50 у комбінації з Feature Pyramid Network (FPN). Цей компонент виконує роль екстрактора ознак, формуючи багаторівневе представлення зображення, яке містить як

низькорівневу інформацію (текстури, контури), так і високорівневу (частини тіла, орієнтація). FPN забезпечує можливість ефективно обробляти об'єкти різних розмірів завдяки побудові піраміди ознак із кількох масштабів.

- **RPN (Region Proposal Network)** – регіональна мережа, що генерує набір потенційних областей (bounding boxes), у яких можуть знаходитися об'єкти. RPN проходить по всіх рівнях піраміди ознак і пропонує обмежені області з найбільшою ймовірністю наявності об'єкта. Ці області згодом уточнюються на наступних етапах.
- **ROI Align** – модуль, який усуває проблему неточності дискретизації при витягненні ознак з областей, запропонованих RPN. На відміну від попереднього підходу ROI Pooling, який міг втрачати просторову точність через округлення координат, ROI Align використовує білінійну інтерполяцію для точнішої вибірки пікселів.
- **Keypoint Head** – додаткова гілка нейронної мережі, призначена спеціально для передбачення положення ключових точок. Вона складається з послідовності згорткових шарів, які опрацьовують ознаки з регіонів інтересу (ROI) і генерують heatmap для кожної ключової точки. На відміну від задачі класифікації, де мережа виводить одне значення, тут вона формує карту ймовірностей, де кожен піксель представляє ймовірність знаходження певної точки.

Keypoint R-CNN демонструє високу стійкість до змін пози, масштабу, часткових оклюзій і складних сцен. Завдяки глибокій архітектурі та використанню FPN вона здатна розпізнавати дрібні деталі навіть у зображеннях з низькою роздільною якістю.

Під час передбачення обробка зображення починається з нормалізації та подачі його до backbone, після чого RPN пропонує потенційні області[4]. Далі модуль ROI Align формує вибірки ознак, і гілка Keypoint Head генерує heatmaps для кожної області. Центр маси (або піксель з найвищим значенням) на heatmap визначається як координата ключової точки. При необхідності виконується

постобробка – масштабування координат назад до розміру оригінального зображення та фільтрація точок із низькою довірою.

3.5 Метрики для оцінювання моделей

Оцінювання якості роботи моделей комп'ютерного зору є критично важливим етапом для будь-якої системи, що виконує автоматизований аналіз візуальних даних. Незалежно від класу задачі – чи то класифікація, детекція об'єктів, семантична або інстанс-сегментація, чи локалізація ключових точок – метрики надають об'єктивний інструментарій для порівняння різних моделей, вимірювання прогресу в навчанні та визначення їхньої практичної застосовності. У цьому підрозділі розглянуто групи метрик, що використовуються у комп'ютерному зорі, з особливим акцентом на специфічних підходах до оцінювання моделей, які прогнозують набір ключових точок на зображенні.

3.5.1 Метрики для класифікації

У класичних задачах класифікації оцінювання часто ґрунтується на аналізі матриці невідповідностей, що включає показники True Positive (TP), False Positive (FP), False Negative (FN) та True Negative (TN). На основі цих значень розраховуються найпоширеніші метрики[2]:

- **Accuracy** – частка правильно класифікованих прикладів. Хоча є інтуїтивно зрозумілою, дана метрика може бути нерепрезентативною для незбалансованих даних.
- **Precision та Recall**. Precision показує, наскільки передбачення моделі є «влучними», тоді як Recall характеризує здатність моделі знайти всі відповідні об'єкти.
- **F1-score**, який є гармонічним середнім Precision та Recall, і дозволяє збалансовано оцінювати моделі, особливо коли важливе співвідношення між помилками I та II типу.

Ці метрики застосовуються переважно для задач, де моделі повертають дискретні класи, тому у задачах детекції та пози вони виступають лише допоміжними.

3.5.2 Метрики для детекції об'єктів

У задачах детекції об'єктів на зображенні широко застосовуються метрики, побудовані на основі величини Intersection over Union (рис.3.13). Показник IoU визначається як відношення площі перетину запропонованої моделлю рамки та істинної рамки до площі їх об'єднання.

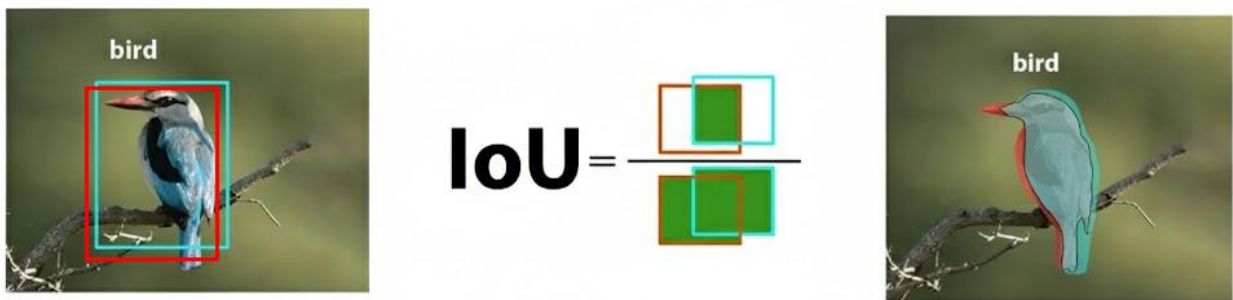


Рисунок 3.13 – Візуалізація IoU

Цей коефіцієнт визначається як відношення площі перетину (intersection) між рамкою, запропонованою моделлю, та істинною рамкою (ground truth) до загальної площі їх об'єднання (union). На основі IoU формуються ключові метрики для оцінювання детекторів. Зокрема, Average Precision (AP) розраховується як площа під кривою Precision-Recall, побудованою для певного порогу IoU. Наприклад, у популярному стандарті оцінювання COCO[15] використовуються різні пороги: AP@50 (вимагає $\text{IoU} > 0.5$), AP@75 та основна метрика AP@[.50:.95], яка усереднює AP за різними значеннями IoU від 0.5 до 0.95 з кроком 0.05. У свою чергу, mAP (mean Average Precision) є середнім значенням AP по всіх класах об'єктів і виступає стандартною метрикою для порівняння продуктивності різних архітектур детекторів. Хоча у задачах оцінки пози кожна людина також локалізується через обмежувальну рамку, mAP

використовується лише як допоміжна метрика, оскільки вона не відображає якості локалізації самих ключових точок.

У задачах оцінки пози людини, кожна людина також локалізується через bounding box, тому mAP іноді виступає допоміжною метрикою, однак не відображає якість саме ключових точок.

3.5.3 Метрики для моделей визначення ключових точок

Задача визначення ключових точок на тілі людини (pose estimation) потребує спеціалізованих метрик, оскільки результат моделі – це набір координат, які мають чітку просторову структуру. На відміну від класифікації, де помилка бінарна, або детекції, де помилка оцінюється відносно рамки, у ключових точках важливо враховувати як відстань між точками, так і їх видимість та масштаб об'єкта.

OXS (Object Keypoint Similarity)

Метрика OXS (Object Keypoint Similarity), представлена у датасеті COCO[15], є стандартом де-факто. Вона є аналогом IoU, але в просторі ключових точок. OXS порівнює передбачені та істинні координати ключових точок з урахуванням масштабу об'єкта та видимості кожної точки.

Означення OXS:

$$OXS = \frac{\sum_i \exp\left(-\frac{d_i^2}{2s^2k_i^2}\right) \cdot v_i}{\sum_i v_i}, \quad (3.5)$$

де d_i – евклідова відстань між передбаченою та істинною точкою;

$v_i \in [1,2,3]$ – коефіцієнт видимості точки;

s – масштаб об'єкта (площа його bounding box);

k_i – нормуючий коефіцієнт для кожного типу ключової точки.

Особливістю є те, що ця відстань нормалізується з урахуванням масштабу об'єкта та специфічного для кожного типу точки коефіцієнта, який відображає складність її точної локалізації. Також у розрахунку враховується коефіцієнт

видимості точки, ігноруючи невидимі точки. Така нормалізація дозволяє адекватно оцінювати помилки: невелике відхилення у пікселях є критичним для малого об'єкта, але припустимим для великого

На основі OKS, за аналогією з метриками детекції, формуються COCO-подібні показники, такі як $AP@OKS=0.50$, $AP@OKS=0.75$ та основна метрика $AP@[0.50:0.95]$, що усереднює точність за різними порогами OKS. Окрім цього, використовується AR (Average Recall), що вимірює середню повноту для різної кількості передбачень. Ці показники є галузевим стандартом для оцінювання сучасних моделей, зокрема Keypoint R-CNN, HRNet, RTMPose та YOLO-Pose.

PCK (Percentage of Correct Keypoints)

На відміну від OKS, яка обчислює "м'яку" міру подібності, PCK є "жорсткою" метрикою, що базується на бінарній умові: ключова точка вважається вірно ідентифікованою лише в тому випадку, якщо її прогнозована позиція знаходиться в межах певної порогової відстані від її істинної позиції[12].

Ключовою проблемою при застосуванні цієї метрики є визначення порогової відстані, оскільки фіксований поріг у пікселях є непридатним через значну варіативність масштабів об'єктів на зображеннях. Помилка у 10 пікселів може бути незначною для великого об'єкта на передньому плані, але абсолютно неприйнятною для малого, віддаленого об'єкта. Для вирішення цієї проблеми PCK використовує нормалізований поріг. Прогнозована точка i вважається правильною, якщо евклідова відстань d_i між нею та істинною точкою g_i не перевищує поріг, пропорційний масштабу об'єкта L :

$$d_i \leq \alpha \cdot L, \quad (3.12)$$

де α – це коефіцієнт;

L – нормалізуючий фактор;

Існують різні варіації цієї метрики, що відрізняються саме вибором фактора нормалізації L :

- **Стандартна PCK:** У цьому варіанті L зазвичай визначається як максимальний розмір обмежувальної рамки об'єкта, тобто $L = \max(w, h)$, де w та h – ширина та висота рамки, або іноді як довжина її діагоналі. Цей

підхід враховує загальний масштаб об'єкта, однак він є чутливим до сильних змін у позі. Наприклад, людина, що сидить, матиме значно меншу висоту рамки, ніж та, що стоїть, що вплине на поріг.

- **PCKh (Head-Normalized PCK):** Ця варіація, що стала стандартом де-факто у багатьох бенчмарках, використовує в якості L розмір голови об'єкта. Зазвичай L обчислюється як певна частка від довжини діагоналі обмежувальної рамки голови, або як відстань між конкретними точками на голові. PCKh вважається більш робастною метрикою, оскільки розмір голови є відносно стабільним незалежно від артикуляції тіла (пози),
- **PCKm (Torso-Normalized PCK):** Менш поширена варіація, де L нормалізується відносно розміру тулуба, наприклад, як відстань між лівим плечем та правим тазостегновим суглобом.

Кінцевий показник PCK розраховується як загальний відсоток ключових точок, що задовольнили цю умову. Часто результати подаються для конкретних значень коефіцієнта α , наприклад, PCK@0.5 або PCK@0.2. Незважаючи на домінування OKS у сучасних змаганнях, PCK та PCKh продовжують активно використовуватися завдяки своїй простоті та інтуїтивній інтерпретації, особливо в академічних дослідженнях та для оцінки на специфічних датасетах.

3.6 Підготовка та обробка даних для навчання

Ефективність та робастність моделей глибокого навчання у комп'ютерному зорі значною мірою залежать від коректності сформованого набору даних та методів його попередньої обробки. Підготовка даних передбачає цілісний комплекс методик, який охоплює їх збирання, формування еталонної розмітки, очищення, а також різноманітні техніки підвищення якості та варіативності вибірки. У кінцевому підсумку, якісну послідовність обробки даних стає фундаментом для забезпечення високої узагальнюючої здатності моделі та її стійкості до змін умов зйомки.

3.6.1. Збір та формування набору даних

Першим етапом традиційно є формування набору даних (датасет). Він повинен адекватно відображати варіативність умов реального світу, в яких модель буде застосовуватися. Збір даних може здійснюватися двома основними шляхами: використання існуючих публічних еталонних наборів даних та створення власного цільового датасету.

Публічні набори COCO (Common Objects in Context)[15] або MPII Human Pose надають тисячі високоякісно анотованих зображень і слугують стандартом для порівняння архітектур. Їх перевагою є великий обсяг, різноманітність сцен, умов освітлення, наявність часткового перекриття та верифікована розмітка. Використання таких датасетів є корисним для попереднього навчання моделей (pre-training), що дозволяє моделі вивчити загальні низькорівневі та середньорівневі ознаки.



Рисунок 3.14 – Приклад зображень з датасету COCO

Однак для вирішення вузькоспеціалізованих завдань, або визначення власних ключових точок, виникає необхідність у формуванні власного набору даних.

3.6.2. Анотація даних

Після збору "сирих" зображень наступним кроком є їх анотація – процес створення істинної розмітки (ground truth), яка слугуватиме еталоном для моделі

під час навчання. Тип анотації повністю залежить від поставленої задачі. Для задач детекції це обмежувальні рамки (bounding boxes), для сегментації – піксельні маски.

У задачах оцінки пози анотація є значно складнішим процесом, оскільки вимагає точної локалізації набору семантичних ключових точок. Кожна ключова точка i для кожного об'єкта на зображенні анотується у форматі (x, y, v) :

- x, y : координати точки у пікселях на зображенні.
- v : атрибут видимості, який є критично важливим для коректного навчання.

Зазвичай використовується тривірнева система за стандартом COCO:

- $v = 0$: Точка не анотована (знаходиться поза межами видимості кадру або не була частиною процесу анотації). Такі точки ігноруються під час обчислення функції втрат.
- $v = 1$: Точка анотована, але є перекритою іншим об'єктом або частиною самого тіла. Модель має бути здатною прогнозувати її ймовірне місцезнаходження, але функція втрат може застосовувати до неї меншу вагу.
- $v = 2$: Точка анотована і чітко видима на зображенні.

Процес анотації проводився з використанням спеціалізованого програмного забезпечення, такого як наприклад, CVAT (Computer Vision Annotation Tool), Roboflow, Label Studio(рис. 3.15).

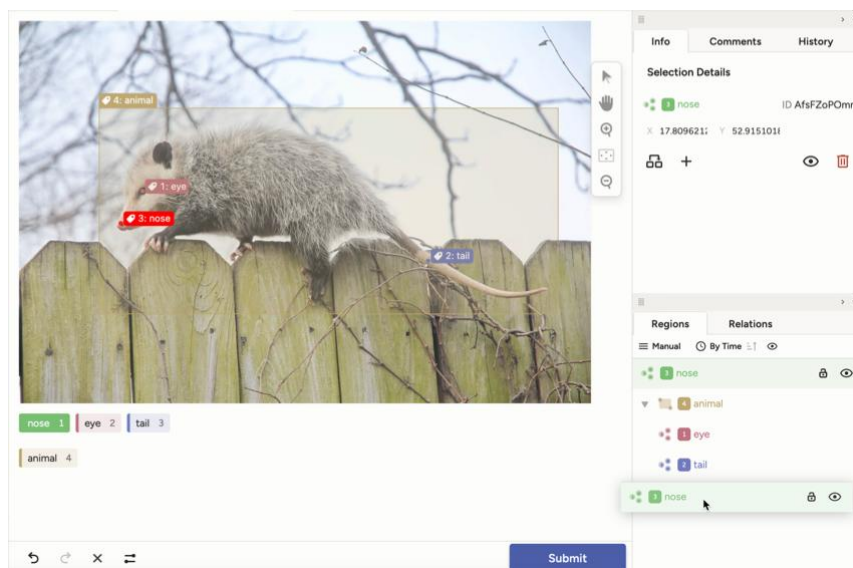


Рисунок 3.15 – Приклад анотації ключових точок у застосунку Label Studio

3.6.3. Попередня обробка та аугментація даних

Навіть великий та якісно анотований набір даних може бути недостатнім для тренування моделі, здатної протистояти перенавчанню та узагальнюватися на нові, небачені раніше дані. Для штучного розширення набору даних та покращення його варіативності застосовуються техніки аугментації даних.

Особливістю аугментації у задач визначення ключових точок є те, що будь-яке геометричне перетворення, застосоване до вхідного зображення, повинне бути симетрично застосоване до координат анотованих точок.

Усі перетворення, можна розділити на кілька груп[30]:

Попередня обробка:

Перед подачею в модель усі зображення та анотації проходять стандартизовану обробку:

- Масштабування: Зображення приводяться до єдиного вхідного розміру, який вимагає архітектура моделі. Відповідним чином перераховуються і координати ключових точок.
- Нормалізація: Значення пікселів зображення нормалізуються. Зазвичай це або приведення значень з діапазону $[0, 255]$ до $[0, 1]$

Геометричні перетворення:

Ці перетворення змінюють просторову орієнтацію та розмір об'єктів і вимагають синхронної модифікації координат (x, y) :

- Випадкове масштабування: Зображення та координати точок масштабуються на випадковий коефіцієнт. Це робить модель інваріантною до розміру об'єкта.
- Випадковий зсув: Зображення випадковим чином зсувається або обрізається. Координати точок, що вийшли за межі кадру, відповідним чином оновлюють свій атрибут видимості v .
- Горизонтальне віддзеркалення: Це одне з найефективніших перетворень для об'єктів з білатеральною симетрією, як-от тіло людини. При віддзеркаленні зображення з імовірністю 50% не лише інвертуються відбувається

семантична заміна парних точок: наприклад, "ліве плече" міняється місцями з "правим плечем", "ліве око" з "правим оком" тощо.

Фотометричні перетворення:

Ці перетворення змінюють візуальні характеристики зображення, але не впливають на координати ключових точок. Вони підвищують стійкість моделі до змін умов освітлення та якості зйомки:

- Зміна яскравості, контрасту та насиченості: Випадкове зміщення значень у каналах кольору.
- Додавання шуму: Імітація зйомки в умовах низької освітленості (наприклад, Гауссів шум).
- Розмиття: Застосування фільтрів розмиття для імітації нечітких кадрів.

Рисунок 3.16 ілюструє приклад застосування типових технік аугментації даних до вхідного зображення. Процес включає як геометричні перетворення так і фотометричні

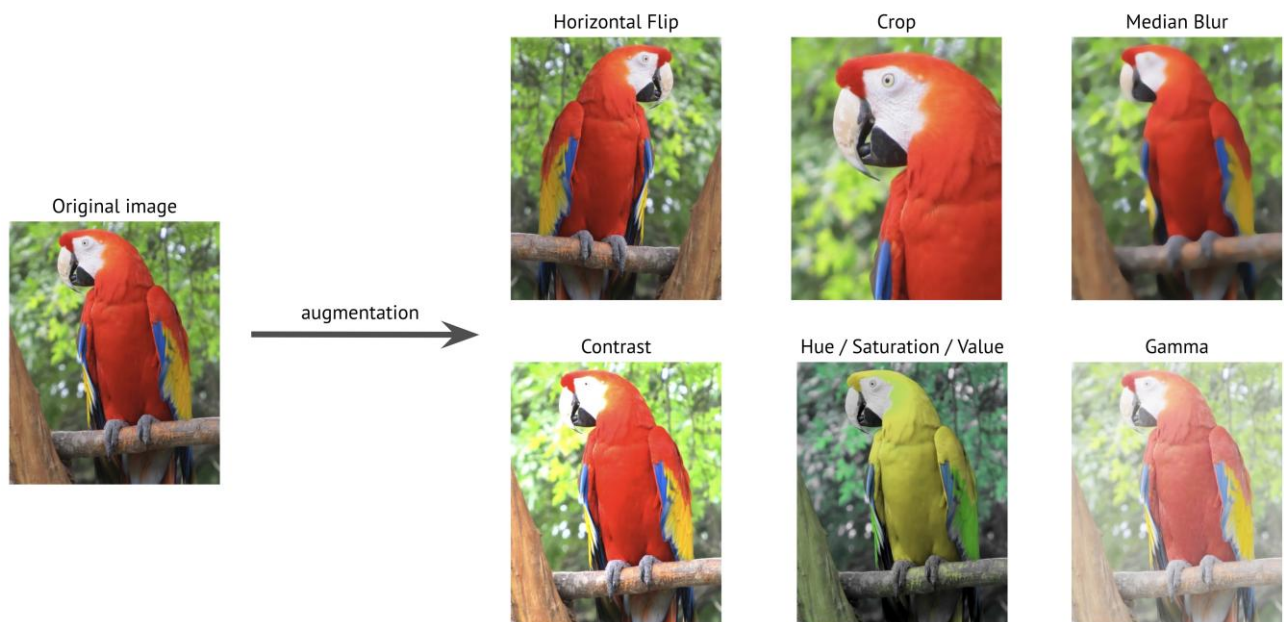


Рисунок 3.16 – Приклад аугментації зображення

Узагальнюючи, методи підготовки та обробки даних становлять системну частину ефективної побудови моделей комп'ютерного зору. Огляд цих методів демонструє, що саме якість даних, правильна організація анотацій, репрезентативність вибірки та продумана аугментація формують основу успішного навчання моделей для задач визначення ключових точок.

Розділ 4. Навчання згорткових нейронних мереж

4.1 Збір навчальних даних

Ефективність моделей глибокого навчання, орієнтованих на визначення ключових точок, безпосередньо залежить від якості та репрезентативності використаних зображень. Незважаючи на наявність великих публічних наборів даних, які розглянуті у розділі 3.5.2, вони здебільшого містять універсальні орієнтири та не враховують специфічні точки, що становлять інтерес у соматоскопічному аналізі. У зв'язку з цим було прийнято рішення сформувати власний датасет, який повністю відповідає клінічним вимогам, передбачає використання стандартизованих умов зйомки та містить усі необхідні анатомічні ключові точки.

Створення такого набору даних вимагало чітко визначеної процедури, що забезпечує однорідність зображень та зменшує вплив зовнішніх факторів[3]. Кожен знімок мав відображати пацієнта у повний зріст, у положенні стоячи, з оголеною верхньою частиною тулуба, що дозволяло точно ідентифікувати потрібні орієнтири. Важливим аспектом стало також забезпечення конфіденційності: оскільки зйомка проводилася у реальних клінічних умовах, задній фон на зображеннях був розмитий, щоб уникнути випадкового потрапляння в кадр сторонніх осіб.

Для забезпечення однаковості даних та зменшення варіацій у геометрії зображень було визначено та суворо дотримано єдиного протоколу збору.

1. Заповнення анкети пацієнта. Перед зйомкою проводилась реєстрація пацієнта з метою забезпечення контролю над отриманими даними та фіксації параметрів, важливих для подальшого аналізу.
2. Підготовка пацієнта. Пацієнта просили роздягнутися до нижньої білизни. Це забезпечувало відкритий доступ до анатомічних орієнтирів, які необхідно було позначити під час анотації.
3. Встановлення камери. Смартфон розміщувався на штативі на висоті приблизно 2/3 зросту пацієнта, на фіксованій відстані 1,6 м, а також

розташовувався строго паралельно до площини грудної клітки пацієнта. Це дозволяло уникати перспективних спотворень та нерівномірного масштабування ключових ділянок зображення.

4. Вирівнювання штатива. За допомогою вбудованого рівня контролювалась вертикальна стабільність конструкції. Точне вирівнювання знижує ризик нахилу камери та спотворень вертикальної лінії тіла.
5. Вирівнювання горизонту на камері. У смартфоні використовувався індикатор горизонту, і оператор забезпечував точність вирівнювання з допустимим відхиленням $\pm 0,2^\circ$, що є важливим для подальшого аналізу нахилів та симетрії.



Рисунок 4.1 – Процес збору знімків

На рисунку 4.1 представлено схематичне розташування камери, пацієнта та основних елементів процесу збору даних. Ця схема дозволяє візуально продемонструвати стандартизовані умови та повторюваність процедури.

У результаті застосування такого підходу вдалося сформувати компактний, але якісний датасет, повністю адаптований під завдання навчання моделей аналізу постави. Зображення характеризуються однорідними умовами зйомки, оптимальним відображенням ключових ділянок тіла та мінімальним рівнем шуму, що робить їх придатними для використання.

Таким чином, розроблений набір даних відповідає вимогам дослідження та забезпечує надійну основу для подальших етапів — анотації та навчання.

4.2 Анотація даних

Після завершення етапу збору та стандартизації зображень наступним ключовим кроком у підготовці набору даних стало проведення анотації, тобто визначення та точне маркування анатомічних ключових точок на кожному знімку. Якість анотації впливає на успішність навчання моделей, оскільки помилки або неоднорідність у розмітці безпосередньо впливають на здатність нейронної мережі правильно відтворювати просторову структуру тіла та прогнозувати положення орієнтирів на нових зображеннях. Враховуючи те, що попередні розділи вже охарактеризували популярні публічні набори даних, важливо відзначити, що їх стандартні схеми анотації не відповідали б специфіці клінічного завдання цього дослідження, тому визначено 14 ключових точок, релевантних саме соматоскопічному аналізу.

Набір ключових точок, використаний у даному дослідженні, охоплює базові анатомічні орієнтири, що дозволяють отримати достатньо повну картину положення голови, плечового поясу та тулуба на фронтальному зображенні. До цього набору входять: верхівка голови, обидва ока та вуха, ліве й праве плече, яремна точка, а також інші реперні точки верхньої частини тіла, необхідні для оцінки симетричності, нахилів та пропорцій. Візуально структура ключових точок у вигляді схематичної моделі людини, де кожна точка позначена певним маркером, зображена на рисунку 4.2.

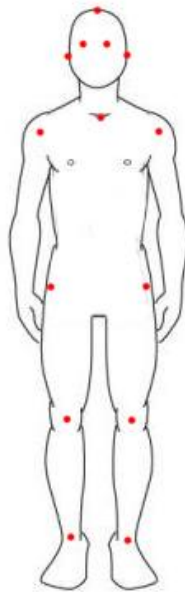


Рисунок 4.2 – Визначені ключові точки

Для виконання анотації було використано платформу **Roboflow**, яка забезпечує зручний інструментарій для роботи з кастомними наборами ключових точок, підтримує сучасні формати анотацій та має можливості інтеграції з численними фреймворками глибокого навчання. Перед початком розмітки було створено власний проект, у межах якого було визначено набір ключових точок, їхню послідовність, семантичні назви та тип маркування. Такий підхід дозволив стандартизувати логіку анотації та уникнути варіацій у її структурі.

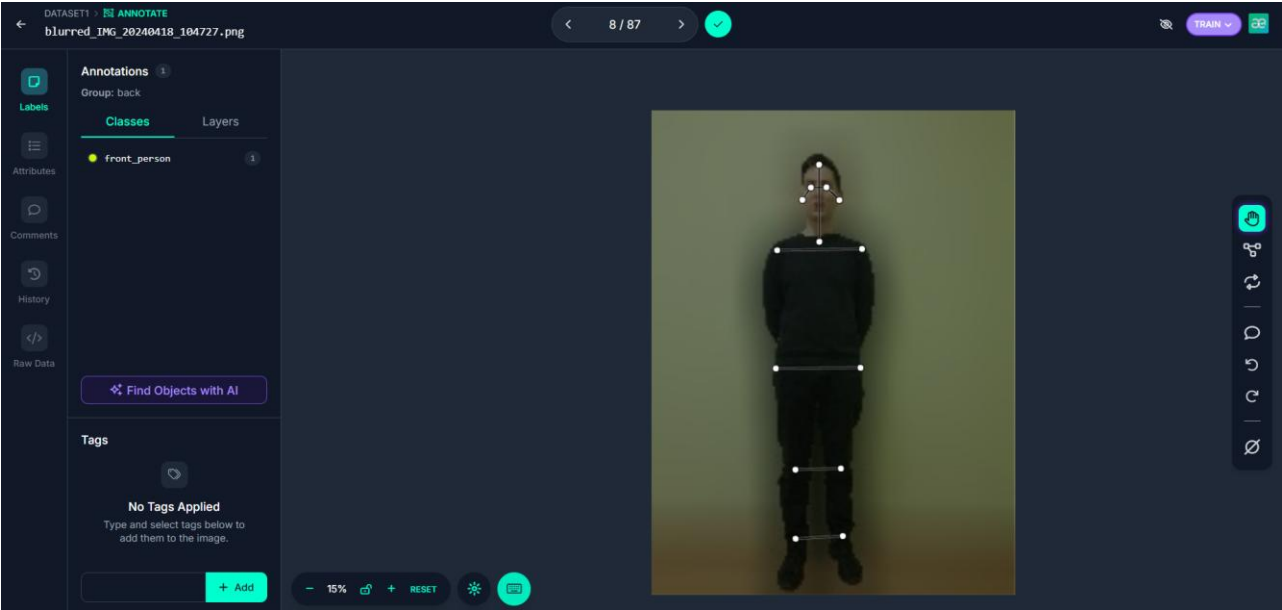


Рисунок 4.3 – Анотація зображення на платформі Roboflow

Процес розмітки виконувався вручну, із застосуванням інструментів Roboflow, зокрема використання розміщення ключових точок з попереднього зображення, для пришвидшеного позиціонування точок на тілі пацієнта. Платформа надає можливість масштабування зображень, підсвічування та корекції вже розмічених точок, що особливо важливо у випадках, коли орієнтири мають нечіткі межі або частково перекриті елементами тіла. Особлива увага приділялася симетричності відповідних парних точок, оскільки це суттєво впливає на коректність подальшого аналізу постави.

Завдяки використанню Roboflow вдалося забезпечити не лише зручність процесу анотації, але й автоматичну стандартизацію вихідних даних. Платформа дозволяє експортувати результати у різних форматах (зокрема COCO Keypoints, YOLO Pose), що значно спрощує інтеграцію з моделями навчання. Крім того, Roboflow автоматично перевіряє цілісність анотацій, що мінімізує ризик помилок, таких як відсутні точки або некоректні координати. У випадках, коли фон фотографії було розмито, система не фіксувала жодних критичних помилок у структурі об'єктів, оскільки ключові точки належать лише до цільового об'єкта — тіла пацієнта.

Важливою перевагою використання Roboflow стало також управління версіями датасету. Після завершення первинної розмітки відбувалося створення зафіксованої версії набору даних, що дозволило відтворювати результати експериментів та порівнювати різні варіанти навчання моделей на ідентичних вихідних умовах. Згодом, під час експериментів на Google Colab, були сформовані окремі версії датасету з різними модифікаціями, зокрема зі зменшеною роздільністю зображень або зі згенерованими аугментаціями, для тестування впливу цих факторів на точність моделі.

Завдяки поєднанню стандартизованої схеми збору та ретельної анотації, вдалося отримати структуровано оформлений та якісно розмічений набір даних, Саме цей датасет став основою для подальших експериментів з навчання моделей та аналізу результатів в наступних розділах.

4.3 Підготовка та аугментація даних

Процес підготовки та збагачення даних здійснювався за допомогою тієї ж платформи Roboflow, яка забезпечує зручні інструменти для автоматизації обробки, перетворення та аугментації датасетів. Робота з даними розпочиналася з попереднього впорядкування та нормалізації зображень, що включало узгодження їх орієнтації, приведення до єдиного простору розмірів, завдяки чому модель отримує однорідний набір вхідних прикладів. На цьому етапі застосовувалися операції Auto-Orient, та Resize, з параметром Fit within 640×480, що дозволяло адаптувати зображення до очікуваного формату вхідного шару нейронної мережі, не втрачаючи пропорції основних об'єктів.

Аугментація виконувалася з метою збільшення варіативності даних та покращення узагальнюючої здатності моделі. Аугментаційні перетворення були підібрані таким чином, щоб імітувати різні умови реального використання системи: зміни освітлення, контрастності, шумів сенсора чи незначних оптичних спотворень. Зокрема, для цього використовувалося регулювання насиченості в межах від -25% до +25%, зміна яскравості від -20% до +20%, застосування легкого розмиття до 1.2 px та додавання шуму на рівні до 1.4% пікселів. Варто зазначити, на платформі Roboflow є можливість автоматизованого формування навчальних вибірок, проте у нею було знехтувано, на противагу, формуванню вибірки «на льоту» під час навчання. Такий підхід дозволив полегшено експериментувати з розмірами навчальних вибірок.

Початковий набір містив 87 оригінальних фотографій, що є відносно невеликою вибіркою для задачі навчання моделі визначення ключових точок. Саме тому застосування аугментації дозволило збільшити обсяг даних і покращити різноманітність навчального набору. Платформа Roboflow, яка використовувалася для обробки та аугментації, у межах безкоштовного тарифного плану надає можливість збільшувати датасет максимум у три рази від початкового обсягу. Це означає, що на кожне вихідне зображення може бути згенеровано не більше двох додаткових аугментованих варіацій. У результаті

застосованих перетворень із наявних 87 знімків було сформовано 261 зображення, що відповідає максимально допустимому множнику 3. Таким чином, вдалося отримати розширений набір прикладів, який краще відображає можливі варіації зовнішніх умов та забезпечує суттєво вищу ефективність навчання моделі. Опис отриманого датасету на платформі Roboflow зображено на рисунку 4.4.

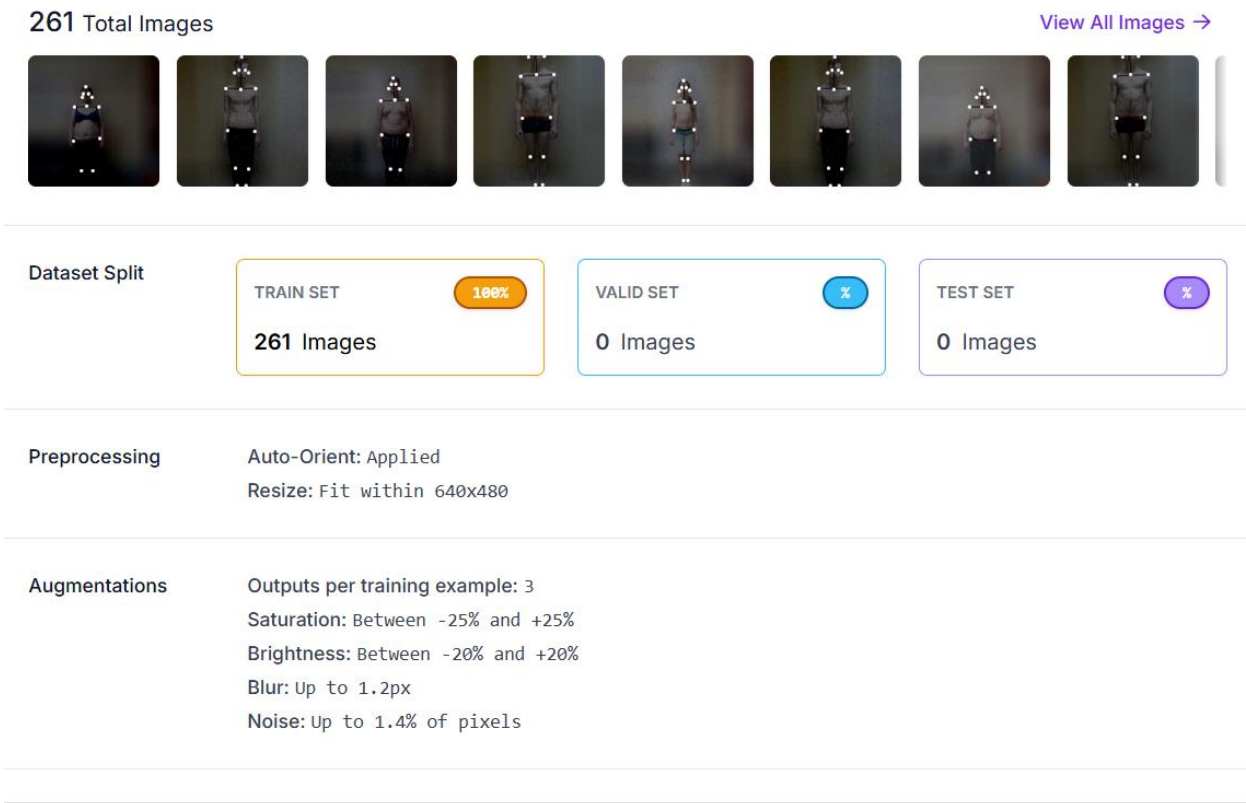


Рисунок 4.4 – Опис отриманого датасету на платформі Roboflow

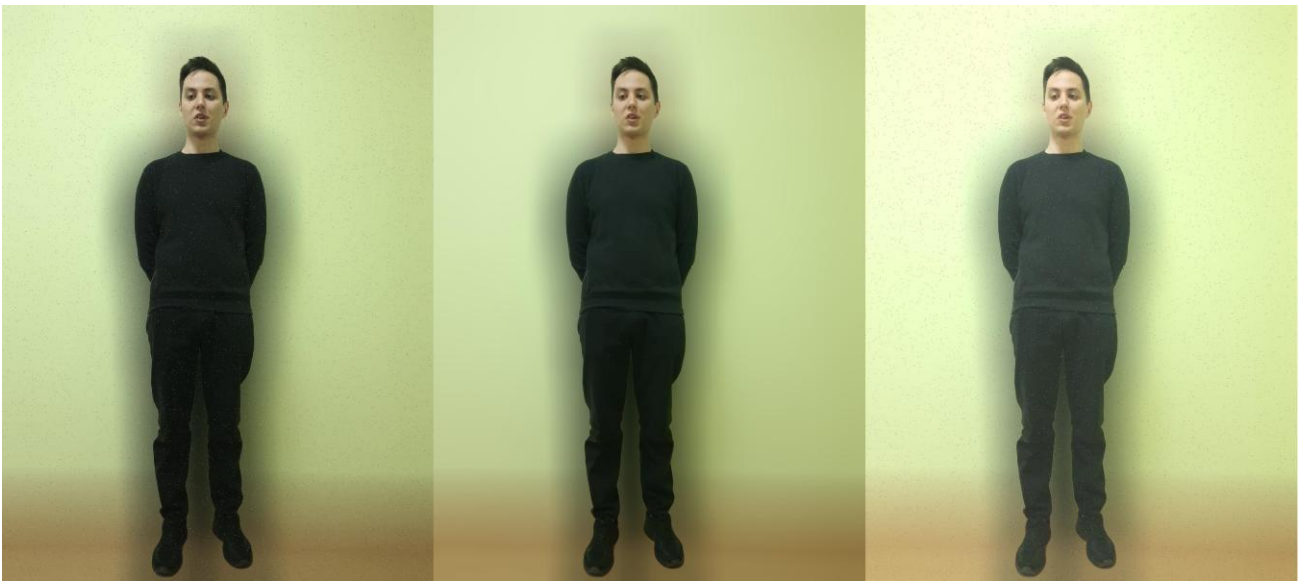


Рисунок 4.5 – Приклад аугментованих зображень

На рисунку 4.5 продемонстровано результати застосування методів аугментації до вихідних графічних даних.

У результаті виконаних етапів анотації, обробки та аугментації було сформовано два окремі варіанти датасету, кожен з яких оптимізовано під специфіку подальшого використання. Перший датасет експортували у COCO-форматі, який є стандартом де-факто для задач виявлення ключових точок і забезпечує сумісність із моделлю Keypoint R-CNN, реалізованою в PyTorch. Другий датасет було сформовано у YOLO-форматі, що дає змогу безпосередньо застосовувати його для навчання моделі YOLO11-Pose, яка вимагає власної структури анотацій та датасету. Наявність двох узгоджених наборів даних у різних форматах забезпечує гнучкість експериментів, можливість порівняння моделей між собою та підвищує валідність отриманих результатів.

Усі процедури, реалізовані через Roboflow, забезпечили автоматизований і відтворюваний робочий процес, що дозволив підготувати датасет належної структурної та змістової якості. Завдяки поєднанню первинної обробки та аугментації вдалося отримати збалансований набір даних, який адекватно відображає можливе різноманіття поз у реальних умовах та забезпечує необхідні передумови для ефективного навчання моделі визначення ключових точок.

4.4 Навчання моделей нейронних мереж

Оскільки обидві моделі мають відкрито доступні попередньо натреновані ваги, отримані на датасеті COCO Keypoints, процес навчання у межах даної роботи не розпочинався «з нуля». Натомість було застосовано підхід fine-tuning, який передбачає додаткове донавчання моделі на спеціалізованому, значно меншому за обсягом, користувацькому датасеті. Завдяки вже сформованим у моделях універсальним просторовим та семантичним ознакам, що охоплюють типові антропометричні структури людського тіла, fine-tuning дозволив швидко адаптувати їх до специфіки власних умов знімання зображень, власного набору з 14 ключових точок. Таким чином, перенесення навчання дало можливість

компенсувати обмеженість зібраного датасету та отримати конкурентоспроможні результати за значно меншу кількість епох і без ризику перенавчання, характерного для тренування моделей від нуля на малих вибірках.

Для проведення всіх обчислювальних експериментів використовувалася платформа Google Colab, яка надає доступ до апаратного прискорення у вигляді GPU. У рамках даної роботи було застосовано графічний процесор NVIDIA Tesla T4, який доступний для безкоштовних ресурсів Colab і забезпечує достатню продуктивність для навчання моделей середньої складності. T4 має доступних 15 GB відеопам'яті, підтримує змішану точність обчислень (FP32/FP16) та оптимізований для інференсу та тренування моделей, побудованих на фреймворках PyTorch та Ultralytics.

Використання Google Colab дозволило не лише забезпечити повторюваність експериментів, але й створити ізольоване середовище з фіксованими версіями бібліотек, відсутністю залежності від апаратної частини локального комп'ютера та можливістю швидкого масштабування конфігурацій. Саме в цьому середовищі було виконано підготовку та завантаження датасетів, налаштування гіперпараметрів, модифікацію архітектур за необхідності та проведено серію експериментів із варіативністю параметрів навчання.

Подальші підрозділи докладно описують етапи навчання обох моделей, починаючи від організації даних та вибору гіперпараметрів і закінчуючи аналізом отриманих результатів

4.4.1 Навчання моделі Yolo-pose

Процес навчання моделі YOLO11-Pose у межах даної роботи здійснювався із використанням фреймворку, представленого розробниками моделі, Ultralytics, який забезпечує високорівневий та стандартизований інтерфейс для тренування моделей сімейства YOLO. Використання цього фреймворку дозволило спростити підготовку навчального середовища, автоматизувати валідацію, логування та

контроль гіперпараметрів, а також забезпечити сумісність із попередньо натренованими вагами.

Навчання проводилося з використанням графічного прискорювача NVIDIA T4, що забезпечило достатньо швидке навчання, залежно від розміру моделі декілька секунд на епопу. Перед початком навчання модель ініціалізувалася вагами, натренованими на датасеті COCO-Keypoints.

У сімействі моделей YOLO11 передбачено кілька варіантів архітектур, що відрізняються масштабом, кількістю параметрів та обчислювальною складністю. Ці конфігурації позначаються літерами N, S, M, L, X. Моделі класу N (Nano) є найменшими та оптимізованими для роботи на малопотужних пристроях з обмеженою пам'яттю, тоді як S (Small) забезпечує кращий баланс між швидкістю та точністю. Архітектури M (Medium), L (Large) та X (Extra-Large) характеризуються збільшеною кількістю параметрів, ширшими шарами та глибшими блоками, що дозволяє досягати вищої точності за рахунок більших обчислювальних витрат.

У процесі навчання були налаштовані основні гіперпараметри, зокрема розмір батчу, початкова швидкість навчання, кількість епох, розмірність вхідного зображення та параметри оптимізатора.

Гіперпараметри задавалися безпосередньо у конфігураційному файлі Ultralytics або передавалися через параметри функції `model.train()`. Основними з них були:

- **epochs** – кількість епох навчання, під час експериментів варіювалася від 50 до 150;
- **batch** – розмір батчу, підбираний із урахуванням обмеження відеопам'яті (зазвичай 8–16);
- **lr0** – початкова швидкість навчання, яка впливала на швидкість збіжності;
- **lrf** – кінцеве значення коефіцієнта швидкості навчання (learning-rate final);
- **optimizer** – за замовчуванням використовувався AdamW, який добре підходить для fine-tuning;
- **imgsz** – розмір вхідного зображення, встановлений на рівні 640 пікселів

Навчання супроводжувалося автоматичним логуванням кривих втрат, метрик точності ($mAP@50$, $mAP@50-95$) та динаміки зміни ключових параметрів, що дозволило відстежувати прогрес і своєчасно коригувати конфігурацію моделі. Після кожної епохи проводилась оцінка на валідаційній частині датасету, що дозволяло контролювати стабільність навчання та уникати перетренування.

Для пошуку оптимальної конфігурації виконувались експерименти з різними пропорціями поділу датасету на навчальну, валідаційну та тестові вибірки. З урахуванням розміру датасету було створено 3 співвідношення виборок (train, valid, test) – 176/64/21(67,5/24,5/8); 192/64/5(73,5/24,5/2); 192/48/21(73,5/18/5/8). Проведено експериментальні навчання з цими трьома розмірностями вибірок на моделі розміру M, з однаковими гіперпараметрами – 50 епох, розмір батчу 8. На рисунку 4.6 зображено порівняльний графік трьох експериментальних навчань, де train – вибірка 176/64/21, train2- 192/64/5, train3 - 192/48/21. За оцінку ефективності навчання обрано втрати по ключовим точкам.

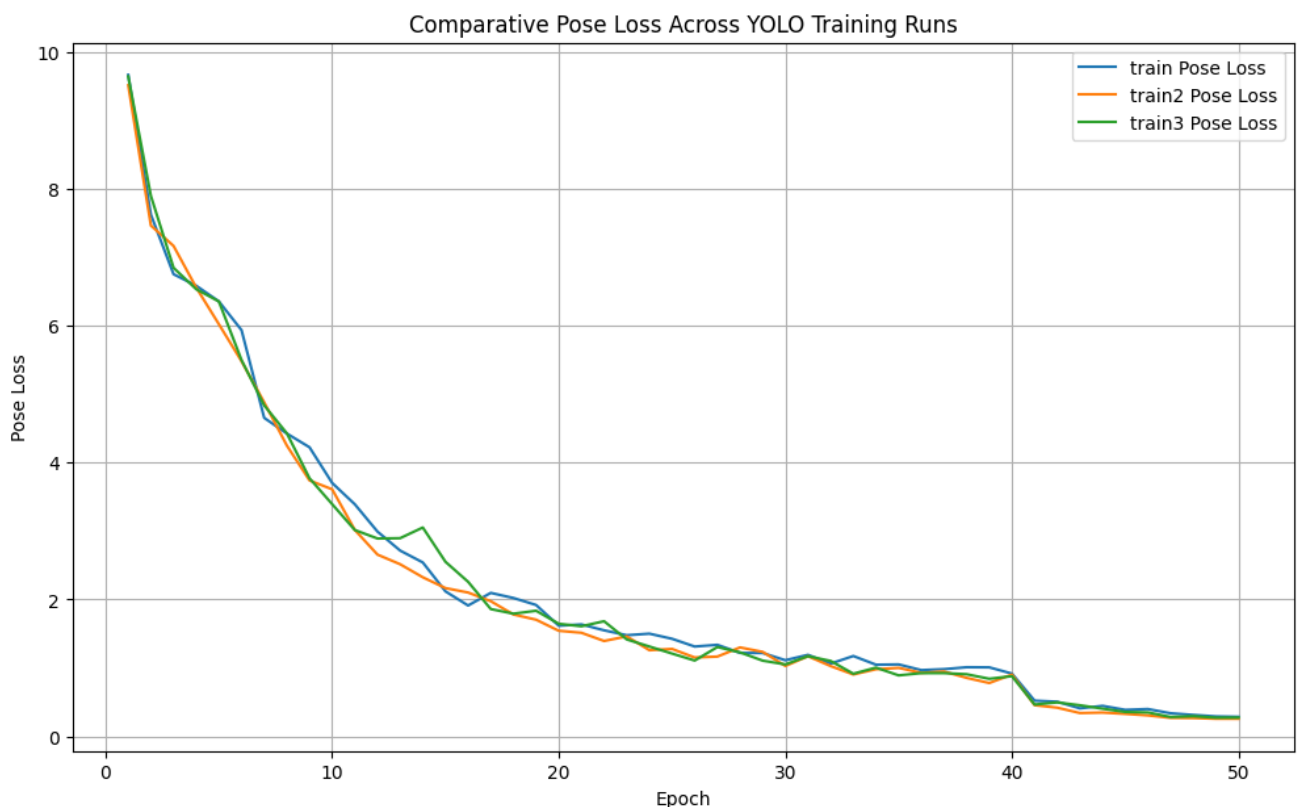


Рисунок 4.6 – Графік спадання втрат по ключовим точкам з різними розмірностями навчальних вибірок

Як видно з графіку, така різниця у розмірностях навчальних вибірок не впливає на результати навчання тому подальші експерименти виконувалися з середнім співвідношенням 73,5/18,5/8.

Далі проведено порівняння навчання моделей розміру М та Х. Проведено експериментальне навчання з кількістю епох 20, та отримано результати що зображені на рисунку 4.7. Як видно з графіку простіша модель швидше навчається.

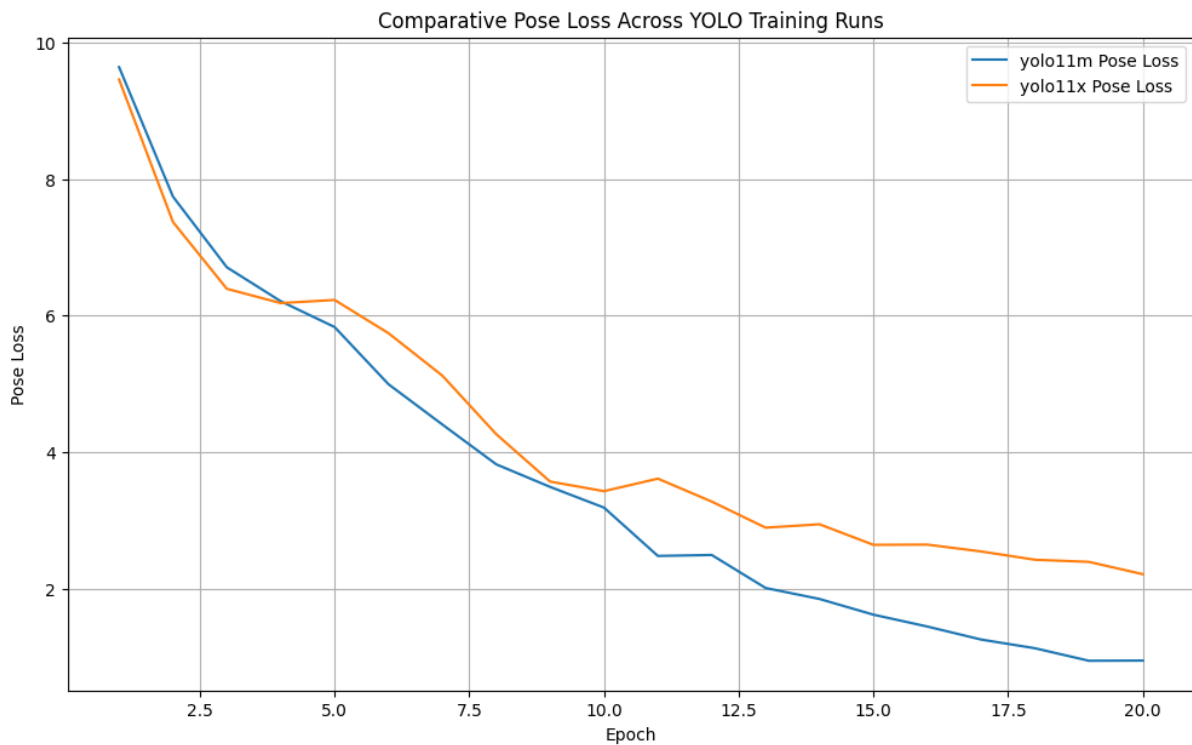


Рисунок 4.7 – Порівняння різних розмірностей моделей

Після інших низки експериментів, для фінального навчання було обрано модель yolo11l-pose для фінального навчання. Навчання здійснювалося на датасеті, 73,5/18,5/8, із використанням зображень розміром 640×480 пікселів, батч-розміром 8 та протягом 500 епох. Для підвищення стабільності моделі застосовано стратегія ранньої зупинки параметр $patience=100$, а також фіксований $seed=0$ для забезпечення відтворюваності.

У процесі навчання використовувались оптимізатор та схема зміни швидкості навчання, рекомендовані фреймворком Ultralytics, із початковим learning rate $lr_0=0.01$ та кінцевим значенням $lrf=0.01$, що забезпечило плавний та контрольований процес збіжності.

Детальний результат навчання, у вигляді показників втрат та метрик зображено на рисунку 4.8.

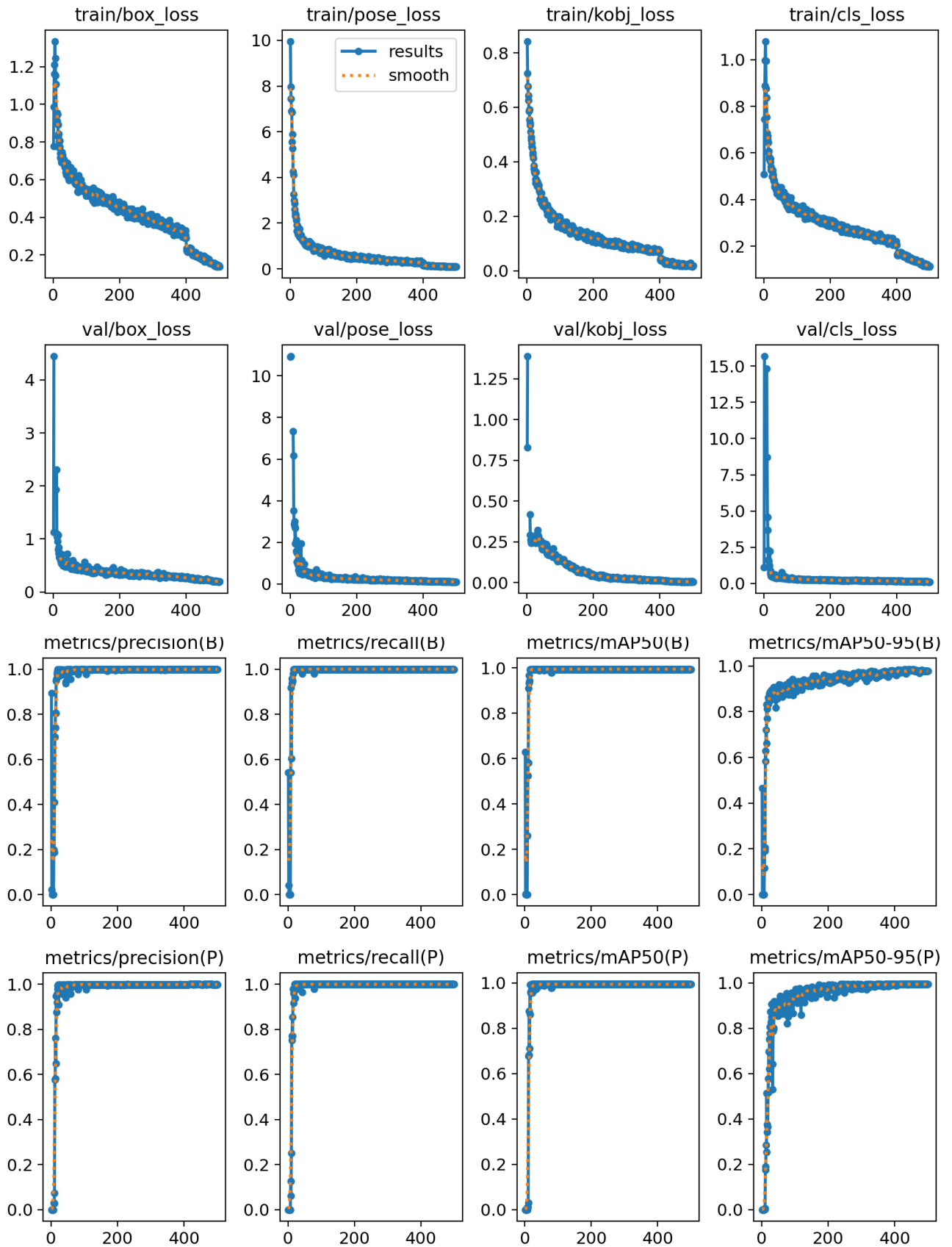


Рисунок 4.8 – Графіки метрики фінального навчання мережі

На рисунку 4.9 зображено приклад визначення ключових точок і обмежувальної рамки на зображенні яке не приймало участь у навчанні.

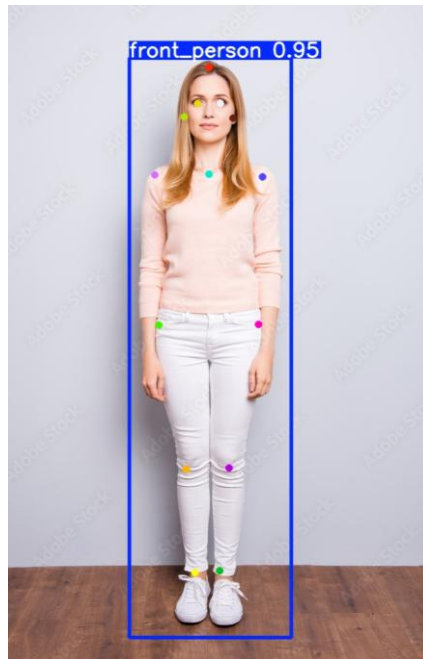


Рисунок 4.9 – Приклад визначення ключових точок моделлю

Варто зазначити, отримана модель, через однорідність датасету погано справляється з зображеннями, де люди стоять у відмінних позах від тієї що були у датасеті. Наприклад, на рисунку 4.10 зображено передбачення моделі для людини з ногами на ширині плеч, чого немає у датасеті.



Рисунок 4.10 – Приклад передбачення з позою якої не було у навчальній вибірці

Зниження точності моделі на зображеннях, що відрізняються від даних тренувального набору, є очікуваним наслідком надмірної однорідності датасету. Оскільки більшість зразків містили людей у схожих фронтальних позах із типовим положенням ніг і корпусу, модель сформувала глибоку залежність від цих повторюваних патернів. У результаті вона навчилася добре відтворювати ключові точки лише для тих положень тіла, які часто траплялися під час навчання, але виявилася менш здатною до генералізації на нові варіації постави.

Наприкінці навчання найкращі ваги автоматично зберігались фреймворком Ultralytics у форматі .pt, що забезпечує їхню подальшу інтеграцію в модулі аналізу постави.

4.4.2 Навчання Keypoint-RCNN

Навчання моделі Keypoint R-CNN проводилося у середовищі Google Colab із використанням фреймворку PyTorch та реалізації Keypoint R-CNN, доступної в модулі `torchvision.models.detection`. Попередньо, датасет було завантажено з платформи Roboflow у COCO форматі. По ходу реалізації перетворено у сумісний із PyTorch Dataset, де кожне зображення супроводжувалося масивом ключових точок, їх видимістю та обмежувальною рамкою.

У процесі навчання та подальшого аналізу ефективності моделі Keypoint R-CNN використовувався комплекс метрик, що дозволяв оцінити якість локалізації ключових точок та збіжність моделі. Базовими показниками були значення функції втрат на тренувальній та валідаційній вибірках, які відображали ступінь наближення моделі до цільових координат. Порівняння динаміки цих втрат дозволяло виявити як момент насичення під час навчання, так і потенційні ознаки перенавчання. Додатково застосовувалася середня відстань між передбаченими та реальними положеннями ключових точок (Average Keypoint Distance), що вимірювала абсолютну геометричну точність локалізації у пікселях або нормованому масштабі. На пізніших етапах аналізу використовувалася метрика точності виявлення ключових точок (Keypoint Detection Rate), яка оцінювала

відсоток ключових точок, передбачених у межах відхилення 8 пікселів від істинних координат, значення. Для більш формалізованої оцінки якості також розглядалася об'єктна схожість ключових точок (OKS) — метрика, аналогічна IoU у задачах детекції, яка враховує не лише відстань між точками, а й масштаб об'єкта та вагові коефіцієнти різних ключових точок. Разом ці показники забезпечували комплексний аналіз якості навчання та дозволяли об'єктивно порівнювати різні конфігурації моделі.

Перший етап передбачав експериментальне навчання моделі протягом 60 епох з розміром батчу 8, що дозволяло оцінити як динаміку збіжності, так і рівень узгодженості між тренувальною та валідаційною вибірками. Метрики цього навчання зображено на рисунку 4.11.

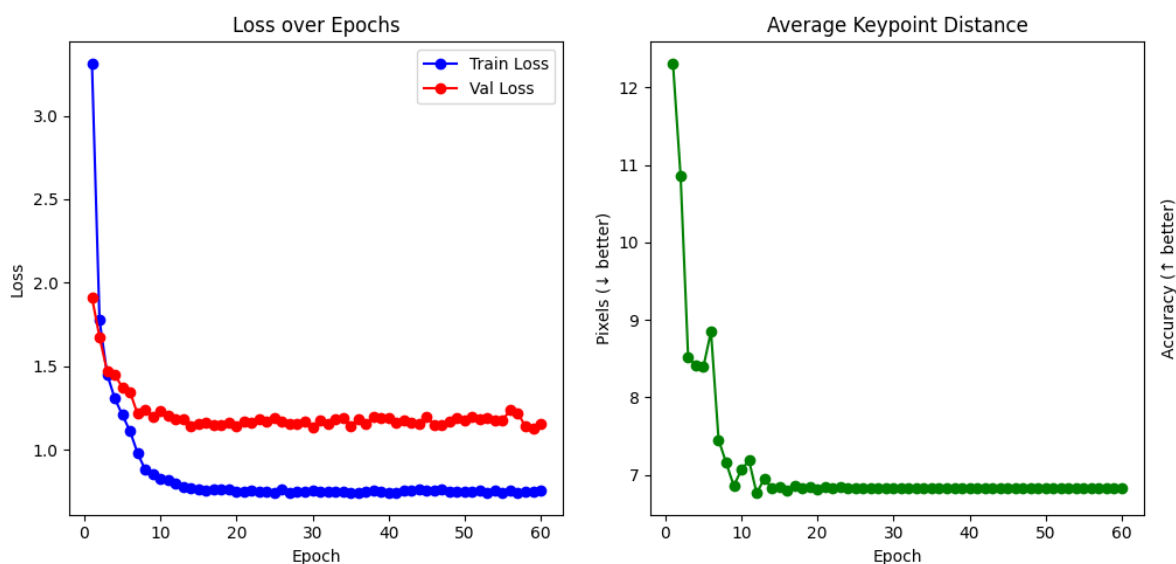


Рисунок 4.11 – Метрики пробного навчання

За підсумками цього експерименту було встановлено, що вже приблизно після 20 епох значення функції втрат на обох вибірках стабілізувалися, а подальше навчання не давало суттєвого покращення. Це свідчило про досягнення моделлю точки насичення, коли вона достатньо добре відтворює просторові залежності між ключовими точками, але не отримує нової інформації з подальших проходів по даних. Загалом, варто відзначити навчання відбувалося значно повільніше та з більшим використанням ресурсів, навіть, якщо порівнювати з yolo11x-rose, яка є співмірною по кількості параметрів з цією архітектурою.

З огляду на ці результати, подальші експерименти проводилися з обмеженням навчання до 20 епох, що дозволило оптимізувати час обчислень без втрати якості моделі. Такий підхід також зменшив ризик перенавчання, що особливо важливо для відносно невеликих та однорідних датасетів.

Далі експериментально було досліджено вплив розміру батчу. Проведено 2 навчання з однаковими гіперпараметрами та розмірами батчу 4 та 8. Графік навчання моделей зображено на рисунку 4.12, де а) – розмір батчу 8, б) – розмір батчу 4.

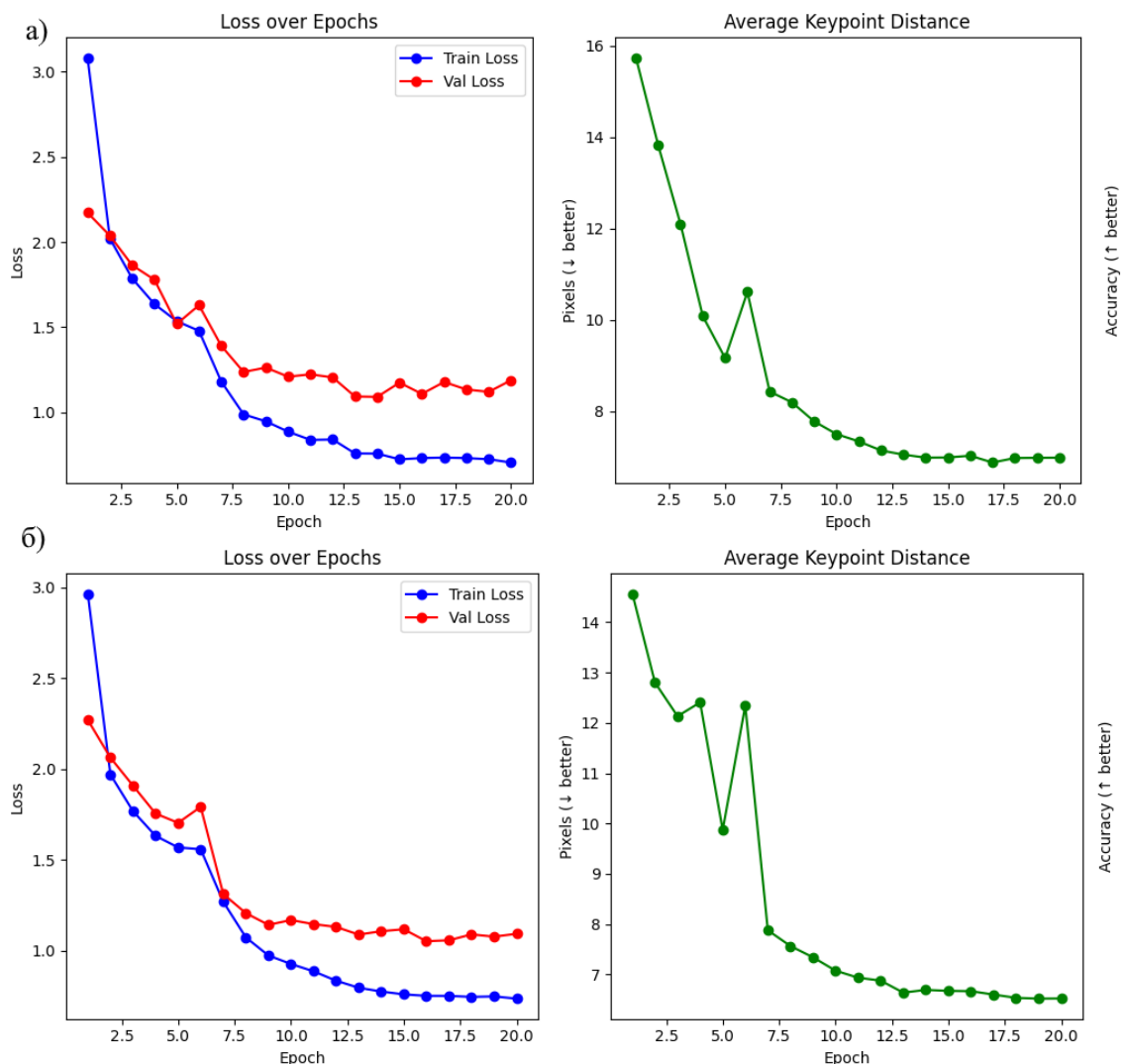


Рисунок 4.12 – Порівняння впливу розміру батча на навчання

У процесі навчання моделі було застосовано продуману тактику збереження результатів, що забезпечувала як безпечне відновлення прогресу, так і фіксацію найякіснішого стану моделі. Після завершення кожної епохи створювався чекпойнт, який містив поточний номер епохи, ваги моделі, стан оптимізатора та

значення тренувальних і валідаційних втрат. Такий підхід дозволяв у будь-який момент перервати навчання та пізніше відновити його з останнього стабільного стану без втрати результатів. Паралельно реалізовувався механізм відстеження найкращої моделі, коли після кожної епохи порівнювалися значення валідаційної втрати: якщо нове значення було нижчим за найкраще попереднє, ваги моделі зберігалися окремо як оптимальний варіант. Це гарантувало, що навіть якщо подальше навчання призводило до перенавчання або коливань значень втрат, користувач завжди мав доступ до найякіснішої версії моделі, отриманої протягом усього циклу тренування.

Після декількох експериментів, фінальна модель навчалася з використанням з типовими параметрами для Keypoint R-CNN: розміром батчу 4, початковою швидкістю навчання $lr=0.005$ та коефіцієнтом регуляризації $weight_decay=0.001$. Графік метрик навчання фінальної моделі зображено на рисунку 4.13.

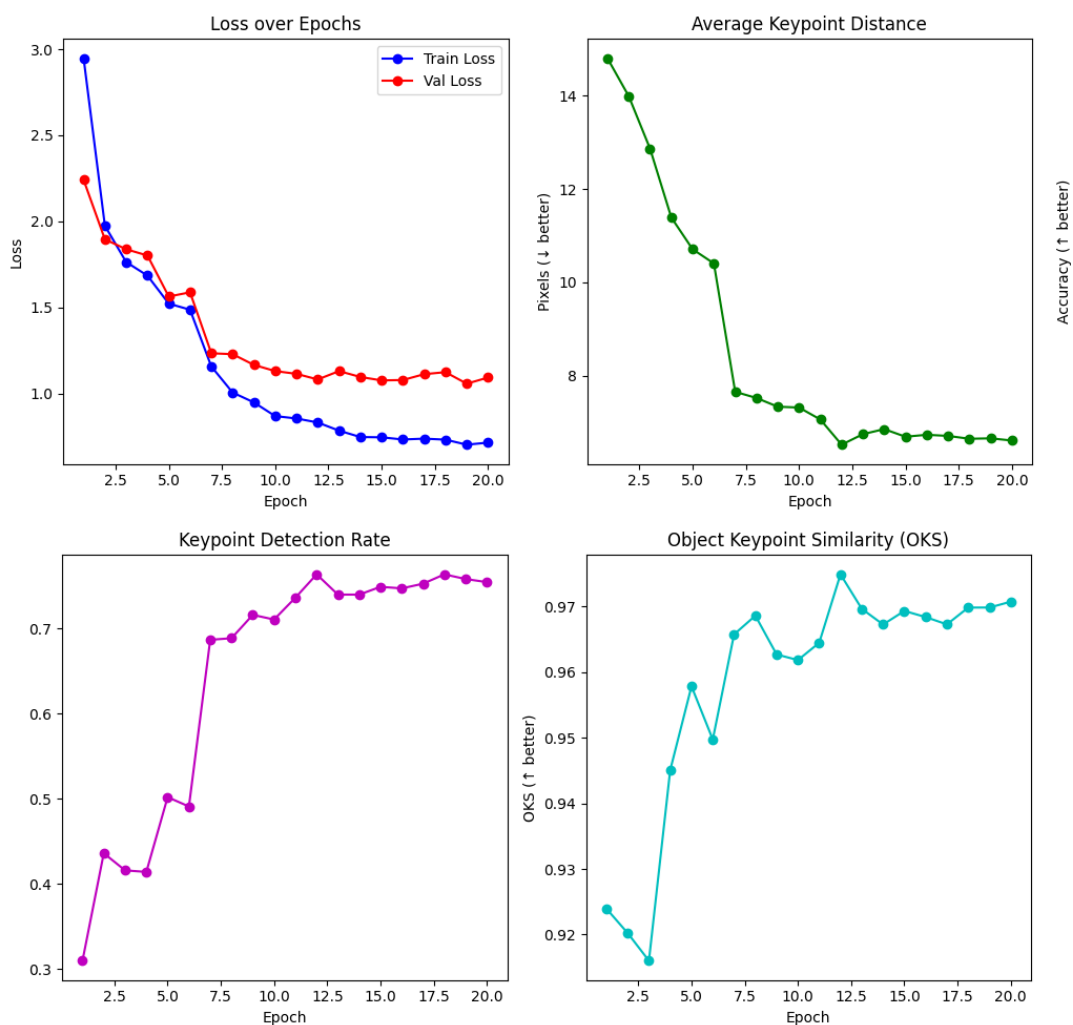


Рисунок 4.13 – Графіки фінального навчання Keypoint-RCNN

В результаті маємо модель з метрикою $OXS \approx 97\%$. Середнє відхилення від істинної точки 6 пікселів. Приклад передбачення ключових точок моделлю на зображенні, яке не приймало участі у навчання зображено на рисунку 4.16.

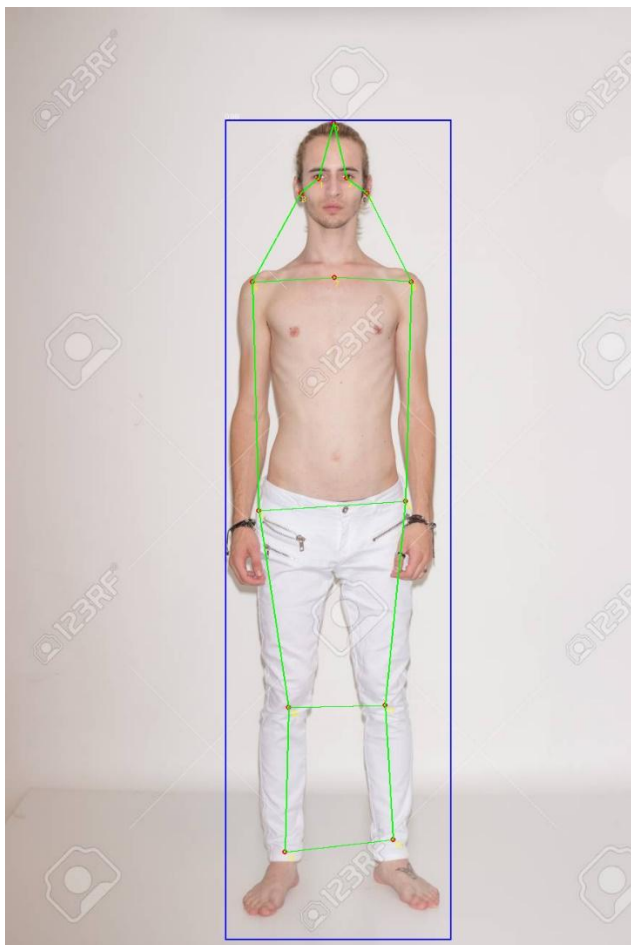


Рисунок 4.14 – Приклад передбачення моделлю Keypoint-RCNN

Як видно, з рисунку 4.14 модель, на відміну від Yolo, не має проблеми з нетиповою позою (ноги на ширині плеч), що робить її більш придатною до застосування. Це явище пояснюється прозорістю процесу навчання, навчання Keypoint-RCNN вимагає, хоч і шаблонної, ручної реалізації процесу навчання, що дозволяє точно налаштувати цей процес. В той час, фреймворк Yolo має власний алгоритм навчання з різними підходами регуляризації, через що модель хоч і починає навчання не з нуля, проте далі втрачає перед-треновані властивості.

Розділ 5. Проектування та реалізація системи

5.1 Архітектура та концепція побудови системи

Інформаційна система аналізу постави зпроектована як клієнт-серверний веб-застосунок, що поєднує засоби комп'ютерного зору, методи штучного інтелекту та інструменти візуалізації результатів. Архітектура системи структурована за принципом розмежування відповідальностей між клієнтською та серверною частинами. Клієнтська частина забезпечує взаємодію з користувачем, збір вхідних даних, відображення результатів та візуалізацію аналізу. Серверна частина виконує інтелектуальну обробку даних, включаючи визначення ключових точок на зображенні, обчислення метрик постави та формування підсумкового звіту.

Вибір клієнт-серверної архітектури для розроблюваної системи зумовлений необхідністю забезпечення високої продуктивності обчислень, гнучкості масштабування та зручності користувацької взаємодії. Обробка зображень і оцінка постави потребують значних обчислювальних ресурсів, тому виконання цих операцій на сервері дозволяє уникнути перевантаження клієнтських пристроїв, забезпечити стабільний час відповіді та використати оптимізовані середовища виконання. Водночас клієнтська частина може залишатися легкою, незалежною від апаратних характеристик користувача та доступною з будь-якого пристрою з веб-браузером. Така архітектурна модель спрощує оновлення програмних компонентів, централізоване керування моделями штучного інтелекту та подальше розширення функціональності системи.

Основою взаємодії між компонентами є REST-орієнтований підхід, який передбачає обмін даними у форматі JSON. Користувач надсилає зображення через веб-інтерфейс, після чого сервер приймає запит, здійснює обробку та повертає структурований результат аналізу, що включає координати ключових точок, розраховані кути та текстові інтерпретації отриманих значень. Такий підхід забезпечує чітке розмежування логіки представлення та логіки обробки, а також спрощує розширення функціональності.

Нижче на рисунку 5.1 наведено діаграму компонентів системи, що відображає її загальну структуру:

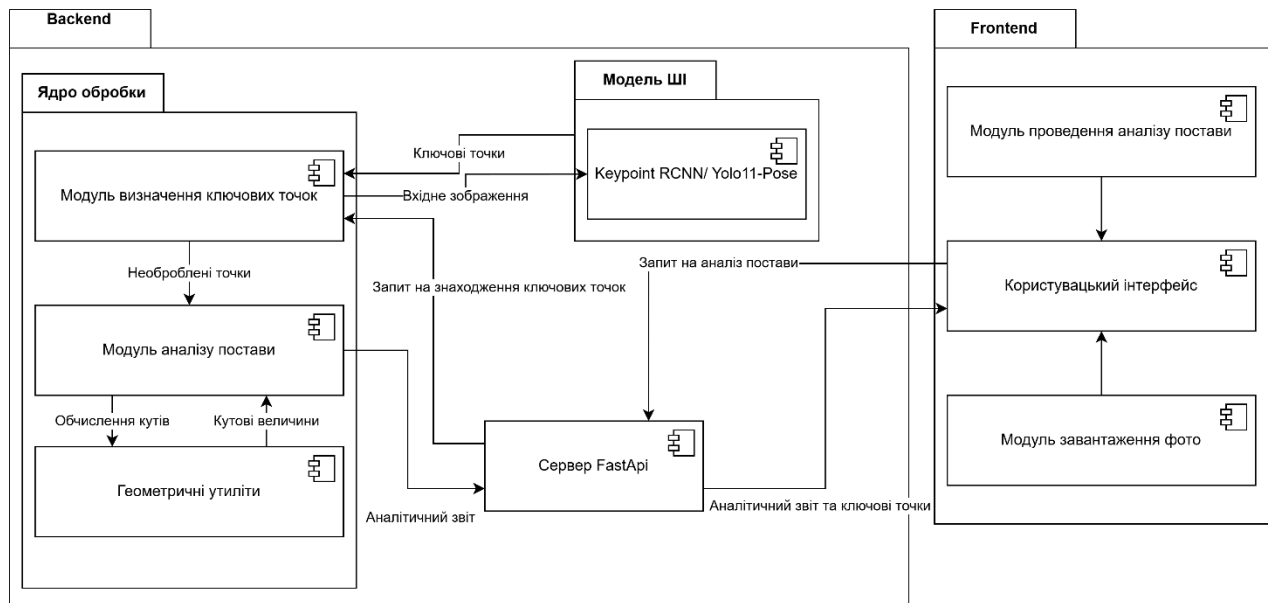


Рисунок 5.1 – Діаграма компонентів

Діаграма демонструє основні інформаційні потоки між компонентами системи та відображає послідовність виконання обчислювальних операцій. Центром взаємодії виступає сервер застосунку, який організовує роботу модулів визначення ключових точок, геометричних обчислень та підсумкового аналізу.

Процес взаємодії користувача з системою включає надсилання зображення та отримання результату в інтерактивному вигляді. Після того як користувач завантажує зображення, клієнтська частина ініціює запит до серверу, де зображення перетворюється у формат, придатний для подальшої обробки. На наступному етапі виконується визначення ключових точок за допомогою попередньо навченої моделі, інтегрованої у серверну частину як окремий модуль інференсу. Отримані координати передаються до модуля аналізу, де здійснюються геометричні розрахунки, визначаються кути та відхилення, після чого формується остаточний аналітичний звіт. Вже на клієнтській частині поверх зображення малюються знайдені ключові точки та скелет. На рисунку 5.2 зображена діаграма послідовності, яка подає типовий сценарій роботи системи.

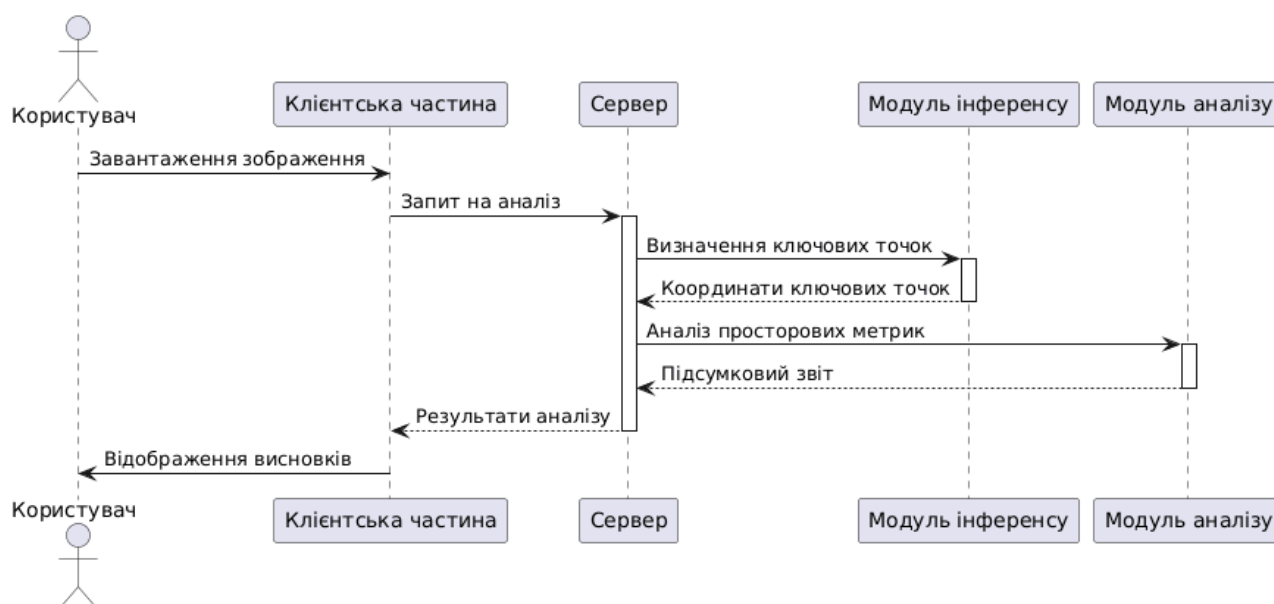


Рисунок 5.2 – Діаграма послідовностей

Ця діаграма відображає логіку обробки запиту від моменту, коли користувач надсилає зображення, до отримання фінального результату. Архітектурне рішення передбачає синхронну взаємодію між компонентами, що дозволяє формувати результат у реальному часі без додаткових затримок.

Інтеграція моделі штучного інтелекту реалізована у вигляді окремого компонента, що використовується виключно для виконання інференсу. Завантаження моделі здійснюється під час ініціалізації серверного застосунку, що мінімізує затримки під час роботи. Вхідні дані, такі як зображення, автоматично нормалізуються відповідно до вимог моделі.

Загалом, архітектура побудована таким чином, щоб забезпечити модульність, масштабованість, прозорість даних та можливість подальшого розвитку системи, включаючи заміну або вдосконалення окремих її компонентів без зміни загальної структури.

5.2 Проектування модулів системи

Розроблення системи аналізу постави ґрунтується на принципах клієнт–серверної архітектури, що забезпечує чітке розмежування функцій між обчислювальним ядром і користувацьким інтерфейсом. Такий підхід дає

можливість централізувати аналітичні обчислення, гарантувати узгодженість даних, масштабувати серверні ресурси та забезпечувати незалежний розвиток клієнтської частини, яка взаємодіє із системою через стандартизований API. Саме це дозволяє підтримувати розширюваність, оновлюваність і контроль без необхідності втручання у програмне забезпечення робочих станцій чи мобільних пристроїв користувачів.

У межах проектування система структурована як багатошарова, із концептуальним поділом на презентаційний рівень, сервісно-доменне ядро та допоміжні підсистеми, що дозволяє зменшити зв'язність та забезпечити структурну керованість програмного комплексу. Логічна декомпозиція модулів охоплює керування ідентичностями та доступом, керування сутностями користувача й пацієнта, роботу зі зображеннями та аналітичними записами, підсистему обробки біомеханічних даних, аудит і логування, модуль файлової абстракції та інтеграцію з підсистемою комп'ютерного зору.

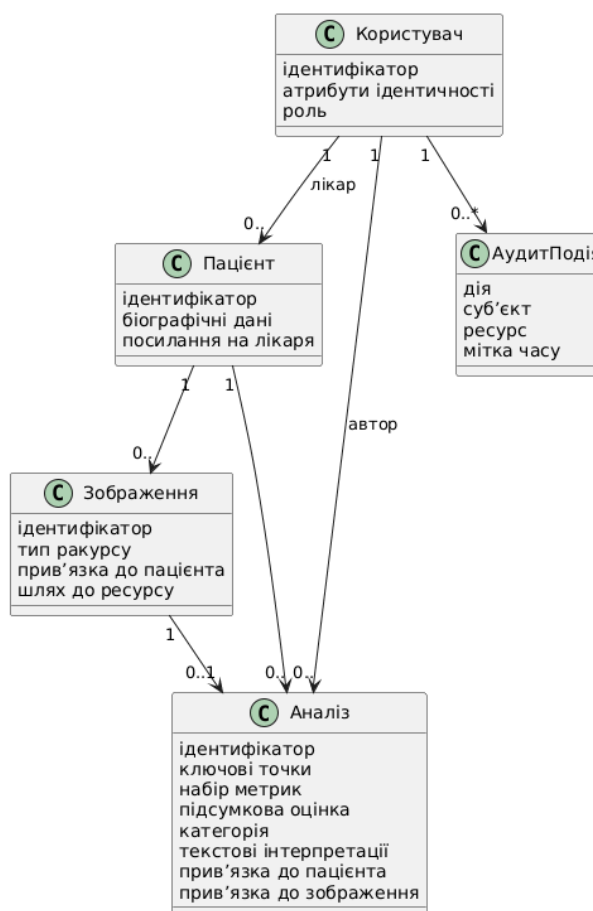


Рисунок 5.3 – Діаграма концептуальних класів системи

Ядро застосунку координує ініціалізацію, маршрутизацію та міжмодульну взаємодію, тоді як конфігураційний модуль забезпечує централізоване керування параметрами виконання. Сутності моделюють основні об'єкти – користувача, пацієнта, зображення, аналітичний запис і аудит – як незалежні стійкі структури, адаптовані до подальшого розширення та аналізу. Діаграму концептуальних класів зображено на рисунку 5.3.

Підсистема ідентифікації реалізує автентифікацію й авторизацію, формує токени доступу та примусове застосування правил безпеки. На рисунку 5.4 зображено діаграму послідовностей яка відображає взаємодію з підсистемаю ідентифікації.

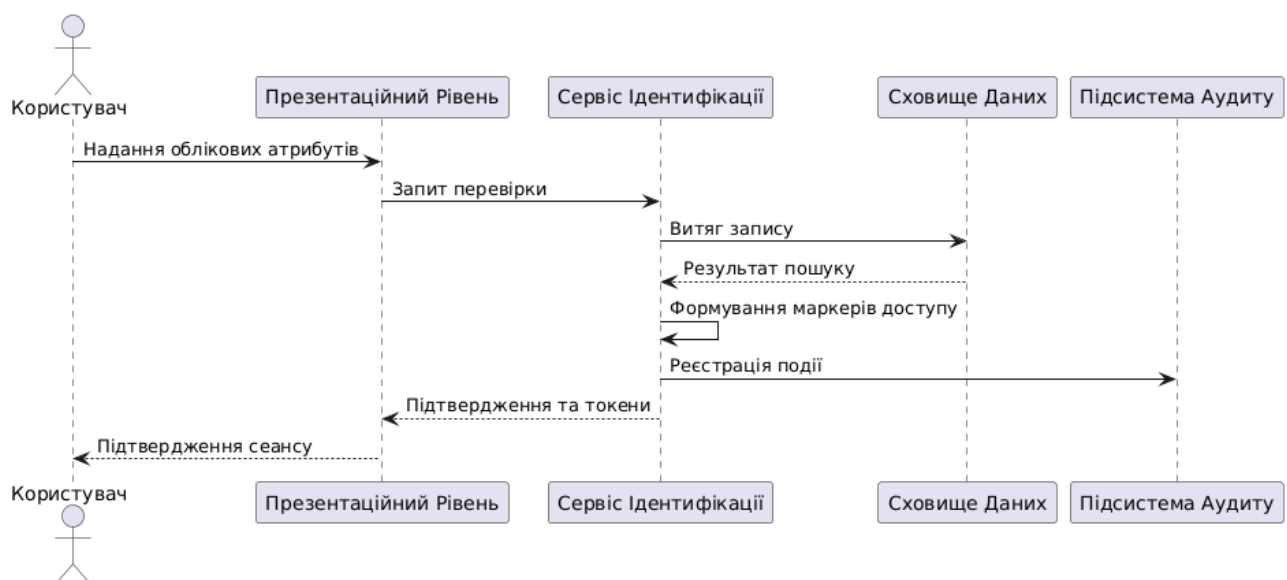


Рисунок 5.4 –Діаграму послідовностей процесу авторизації

Сервіс взаємодії із зображеннями організовує приймання та зберігання медіа даних, контролюючи їх цілісність і прив'язку до контексту користувача. Підсистема детекції постави виконує виділення ключових точок і їх нормалізацію, після чого аналітичний модуль метрик формує геометричні, симетричні та структурні показники, інтегральну оцінку та підсумкову категорію. Файлова абстракція відокремлює бізнес-логіку від фізичного середовища зберігання, дозволяючи вільно змінювати інфраструктурні рішення.

Концептуальна модель даних побудована навколо сутностей «Користувач», «Пацієнт», «Зображення» та «Аналіз постави», а також «АудитПодія» як ортогонального механізму контролю (див. рис. 5.3). Ключові точки представлені універсальними семантичними елементами з координатами й оцінкою достовірності, що дає змогу розширювати перелік анатомічних маркерів без зміни схеми даних.

Процес взаємодії між модулями при обробці запиту передбачає проходження через автентифікацію, отримання контексту даних, застосування бізнес-правил та взаємодію з аналітичними або файловими компонентами. Детекція та обчислення метрик пов'язані стандартизованим форматом ключових точок. Збереження результатів аналізу виконується атомарно з денормалізованими метриками для зменшення часу подальшого доступу. На рисунку 5.5 зображено взаємодію раніше описаних модулів у випадку процесу формування аналітичного звіту.

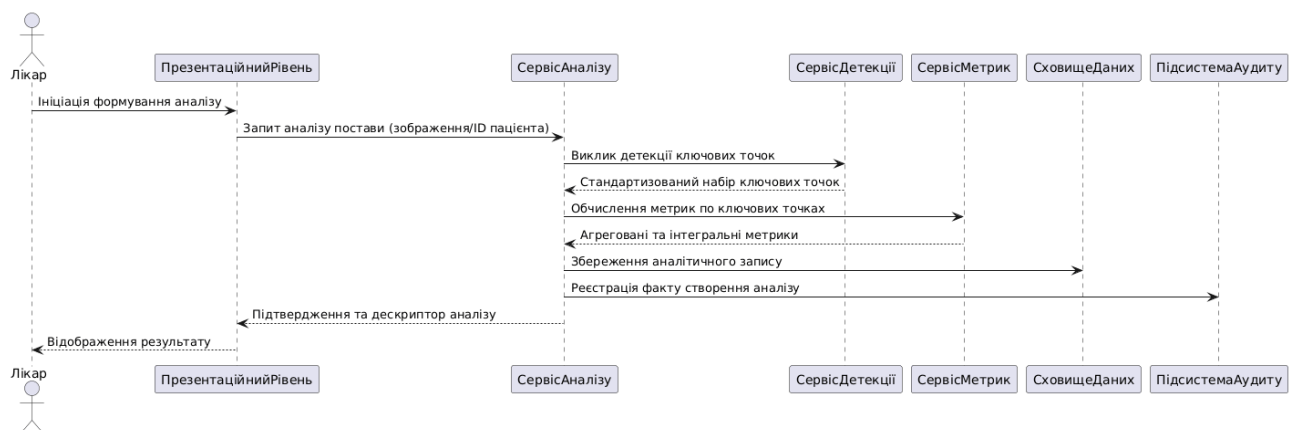


Рисунок 5.5 – Діаграма послідовності для процесу формування аналітичного звіту

Запропоновані архітектурні рішення забезпечують гнучкість і розширюваність програмного комплексу. Виділення аналітичних метрик у самостійний модуль дозволяє застосовувати різні методи обчислення без зміни решти системи. Атомарне створення аналізу забезпечує баланс між точністю та продуктивністю, а аудит підвищує рівень спостережуваності. Стандартизований формат ключових точок гарантує нейтральність щодо моделей детекції.

Архітектура системи передбачає перспективи природного розширення: додавання нових метрик, зміну або деталізацію політик доступу, розширення

формату даних, підключення нових моделей детекції або аналітичних підсистем, а також створення каналів експорту чи зовнішньої інтеграції. Така побудова забезпечує довгострокову життєздатність, масштабованість та відповідність вимогам медичного програмного забезпечення.

5.3 Опис модуля аналізу постави

Модуль аналізу постави базується на поєднанні моделей детекції анатомічних точок та математичного апарату, який дозволяє отримати комплексну оцінку постурального стану людини. Основою системи є метрики, що описують фронтальну симетрію, локальні кутові відхилення, лінійні співвідношення сегментів та агреговані інтегральні показники. Перед розрахунком метрик формується уніфікований набір ключових точок, що забезпечує незалежність аналітичного ядра від конкретної моделі детекції. Архітектура системи передбачає збирання вихідних предикцій, їх фільтрацію, стандартизацію та подальше застосування алгоритмів геометричної обробки.

Першим етапом роботи алгоритмічного конвеєра є детекція ключових точок. У системі реалізовано підтримку двох категорій моделей: однокрокової архітектури YOLO Pose та двокрокової Keypoint R-CNN. Модель YOLO Pose виконує одночасну локалізацію та регресію координат, приймаючи на вхід растрове зображення (BGR масив) і повертаючи тензори координат K_{xy} розмірності $(P, N, 2)$ та матрицю конфіденційності. Натомість архітектура Keypoint R-CNN працює послідовно, формуючи регіони інтересу та уточнюючи позиції точок у межах регіону. Незалежно від обраної моделі, реалізовано механізм уніфікації вихідних даних: результати інференсу перетворюються у стандартизований список словників виду $\{name, x, y, confidence\}$. Це забезпечує поліморфізм аналітичних алгоритмів та можливість розширення системи новими детекторами без зміни логіки обчислень.

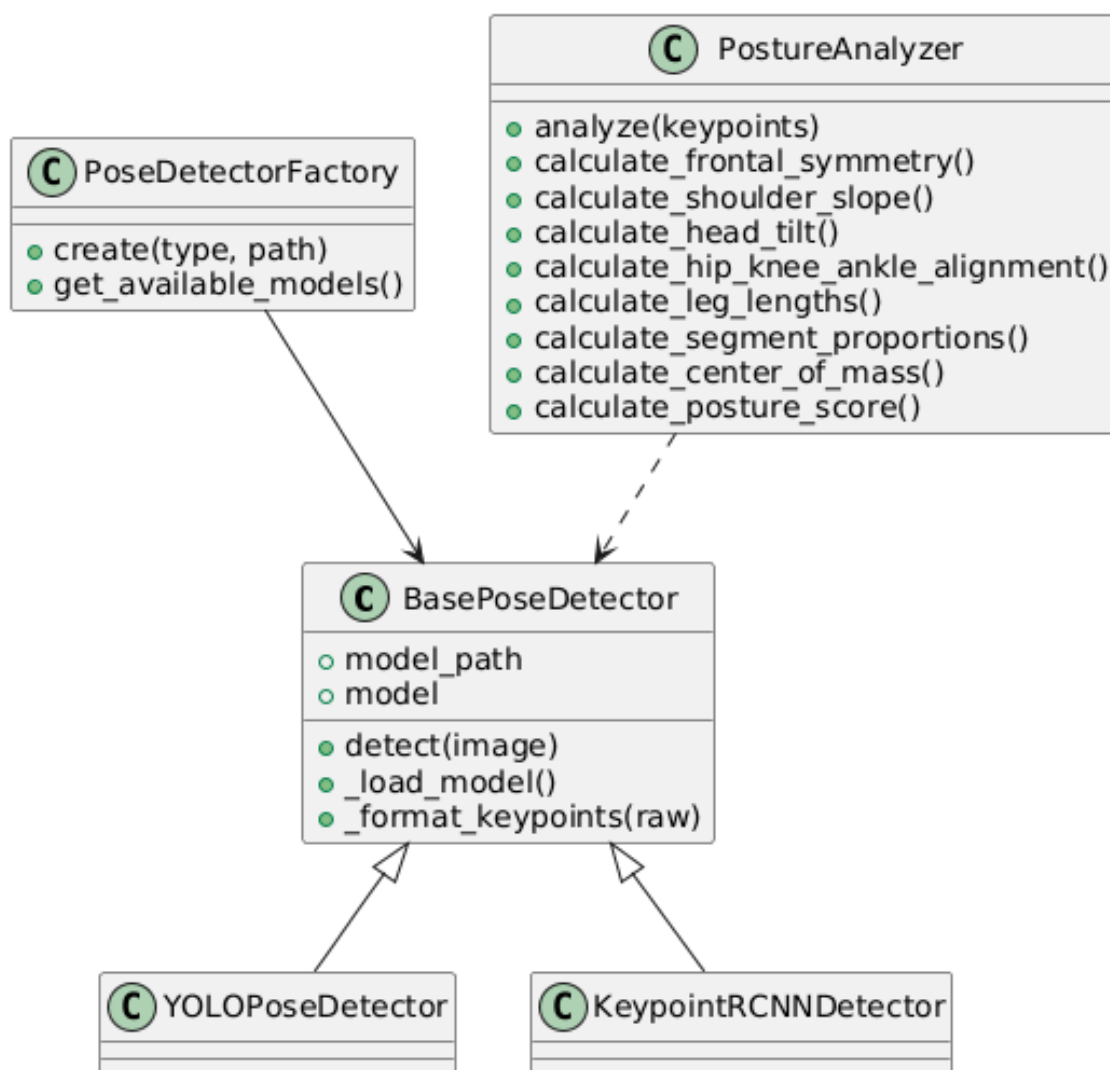


Рисунок 5.6 – Діаграма класів модуля аналізу постави

Діаграма на рисунку 5.6 демонструє застосування фабричного патерну для ізоляції конкретних реалізацій детекторів та окремий моноліт алгоритмів обчислення метрик, що оперує виключно уніфікованими координатами.

Застосовано набір метрик, що відображає фундаментальні властивості постави людини у фронтальній площині. Вибір цих метрик мотивовано клінічною інтерпретованістю (кореляція з порушеннями балансу, сколіотичними відхиленнями, асиметрією навантаження), геометричною стабільністю (опора на відносні показники та кути), а також простотою обчислень, які базуються на елементарних тригонометричних та метричних співвідношеннях. Додатково набір ключових точок – вершина голови, очі, вуха, плечі, яремна точка, стегна, коліна та щиколотки – створює достатній каркас для формування стійких вимірів без надмірної деталізації.

Оцінювання постави на основі ключових точок тіла передбачає обчислення низки геометричних та симетричних метрик, що відображають якість вертикальної орієнтації, симетрію тіла та пропорційність положення анатомічних орієнтирів. Кожна метрика ґрунтується на евклідовій геометрії та аналізі векторів, отриманих із набору детектованих ключових точок. Нижче подано метрики та математичні формули, на яких вони базуються.

Фронтальна симетрія

Фронтальна симетрія оцінює, наскільки парні анатомічні маркери (очі, вуха, плечі, тазові точки, коліна, щиколотки) знаходяться на однаковій висоті у фронтальній площині.

Для кожної пари (L, R) обчислюється вертикальний розрив:

$$\Delta y = |y_L - y_R| \quad (5.1)$$

та нормувальна ширина (горизонтальна відстань):

$$W = |x_R - x_L| \quad (5.2)$$

На їх основі визначається часткова симетрія:

$$S_{pair} = \begin{cases} 100 \cdot (1 - \min(\frac{\Delta y}{W} + 1)) & W > 0, \\ 0, & W = 0 \end{cases} \quad (5.3)$$

Загальна симетрія – середнє по всіх валідних парах:

$$S_{overall} = \frac{1}{M} \sum_{k=1}^M S_{pair}(k) \quad (5.4)$$

Значення поблизу 100 означають рівномірне положення парних сегментів та відсутність перекосів.

Нахил плечей

Нахил плечового поясу визначається як кут між лінією, що з'єднує ліве та праве плече, і горизонтальною віссю:

$$\theta_{shoulders} = |\arctan 2(y_R - y_L, x_R - x_L)| \cdot \frac{180}{\pi} \quad (5.5)$$

Чим більше значення, тим більший переко́с плечей. У нормі кут близький до 0°.

Нахил голови

Вектор від середини лінії між вухами до верхівки голови має бути максимально вертикальним.

Спочатку обчислюється середня точка між вухами:

$$E_{\text{mid}} = \left(\frac{x_{\text{earL}} + x_{\text{earR}}}{2}, \frac{y_{\text{earL}} + y_{\text{earR}}}{2} \right) \quad (5.6)$$

Створюємо вектор:

$$\vec{v} = P_{\text{vertex}} - E_{\text{mid}} \quad (5.7)$$

і порівнюємо його з еталонним вертикальним вектором:

$$\vec{u} = (0, -1) \quad (5.8)$$

Кут нахилу:

$$\theta_{\text{head}} = \arccos \left(\frac{\vec{v} \cdot \vec{u}}{|\vec{v}| |\vec{u}|} \right) \cdot \frac{180}{\pi} \quad (5.9)$$

0° – повна вертикальність; збільшення кута вказує на латеральний нахил голови.

Нахил тазу

Нахил тазу у фронтальній площині визначається як кут між лінією, що з'єднує ліву та праву клубові точки, і горизонтальною віссю. У нормі таз має залишатися максимально рівним, без підйому однієї зі сторін.

$$\theta_{\text{pelvis}} = \arctan 2 (y_R - y_L, x_R - x_L) \cdot \frac{180}{\pi} \quad (5.10)$$

Значення $\theta_{\text{pelvis}} \approx 0^\circ$ відповідає симетричному положенню таза. Збільшення кута свідчить про фронтальний нахил тазу — підняття однієї з клубових остей, що може бути ознакою м'язового дисбалансу, нерівномірного навантаження на нижні кінцівки або компенсаторної реакції на сколіотичні зміни.

Довжини ніг та їх симетрія

Довжина кожної ноги:

$$L_{\text{left}} = d(P_{\text{hipL}}, P_{\text{ankleL}}), L_{\text{right}} = d(P_{\text{hipR}}, P_{\text{ankleR}}), \quad (5.11)$$

де $d(\cdot)$ – евклідова відстань.

Різниця довжин:

$$\Delta L = |L_{left} - L_{right}| \quad (5.12)$$

Нормована симетрія:

$$S_{legs} = \begin{cases} 100 \cdot (1 - \frac{\Delta L}{\max(L_{left}, L_{right})}) & \max(L_{left}, L_{right}) > 0, \\ 0, & else \end{cases} \quad (5.13)$$

Значення близьке до 100 свідчить про рівні довжини кінцівок.

Пропорції сегментів нижньої кінцівки

Співвідношення довжин стегнового та гомілкового сегментів:

$$R_{leg} = \frac{d(P_{knee}, P_{ankle})}{d(P_{hip}, P_{knee})} \quad (5.14)$$

Великі між сторонні розбіжності або атипові значення можуть сигналізувати про неправильне визначення точок або особливості розвитку кінцівок.

Умовний центр мас

Для набору точок із вагами w_i барицентр визначається як:

$$x_{COM} = \frac{\sum w_i x_i}{\sum w_i}, \quad y_{COM} = \frac{\sum w_i y_i}{\sum w_i} \quad (5.15)$$

Зміщення центру мас убік корелює з латеральною асиметрією або функціональним перекосом.

Інтегральний бал постави

Для узагальненої оцінки всі метрики приводяться до шкали 0, 100.

Нормовані бали:

$$B_{shoulders} = 100 \cdot \max\left(0, 1 - \frac{A_s}{20}\right), \quad (5.16)$$

$$B_{head} = 100 \cdot \max\left(0, 1 - \frac{A_h}{20}\right), \quad (5.17)$$

$$B_{hka} = 100 \cdot \max\left(0, 1 - \frac{D_{hka}}{30}\right), \quad (5.18)$$

де $A_s = \theta_{shoulders}$ – кут нахилу плечей

$A_h = \theta_{head}$ – кут нахилу голови

D_{hka} – середнє відхилення стегно–колiно–щиколотка.

Узагальнений iндекс:

$$Score = 0.25S_f + 0.15B_{shoulders} + 0.20B_{head} + 0.20B_{hka} + 0.20S_{legs} \quad (5.19)$$

Категорії:

- Excellent: $Score \geq 85$
- Good: $Score \geq 70$
- Moderate: $Score \geq 50$
- Poor: $Score < 50$

Індекс дозволяє узагальнити рiзнорiднi геометричнi показники в єдину оцiнку, що зручна для монiторингу та порiвняння мiж пацiєнтами.

Логiка обробки помилок включає запобiгання дiленням на нуль, перевiрку наявностi парних точок перед розрахунками та повернення нульових значень, коли обчислення не можуть бути виконанi коректно. Якщо певна метрика не може бути визначена, вона не враховується у фiнальному зваженому iндекси, що пiдтримує стiйкiсть системи навіть на даних низької якостi.

Крім того, iнтеграція всiх обчислень у послiдовний програмний цикл забезпечує вiдтворюванiсть результатiв та можливiсть точного порiвняння динамiки стану пацiєнта в часi. Кожна метрика проходить однаковий етап нормалiзацiї, перевiрки цiлiсностi та масштабування, що унеможливорює вплив випадкових шумiв або локальних артефактiв зображення на остаточний результат. Такий пiдхiд дозволяє пiдтримувати стабiльнiсть iндексу навіть за умов варiативного освiтлення, рiзної пози камери або часткового перекриття анатомiчних маркерiв, роблячи метод придатним як для клiнiчного застосування, так i для використання у побутових або телемедицих сценарiях.

Описанi метрики працюють узгоджено в межах єдиного аналітичного конвеєра, де кожна метрика робить внесок у побудову узагальненого iндексу пози, що дозволяє як виконувати автоматизовану дiагностику, так i забезпечувати подальше масштабування системи. Завдяки модульностi пiдходу та прозоростi математичних трансформацiй алгоритми можуть бути адаптованi для рiзних типiв

камер, конфігурацій середовища та вимог клінічних протоколів, забезпечуючи надійну основу для подальшого розвитку інформаційної системи аналізу постави.

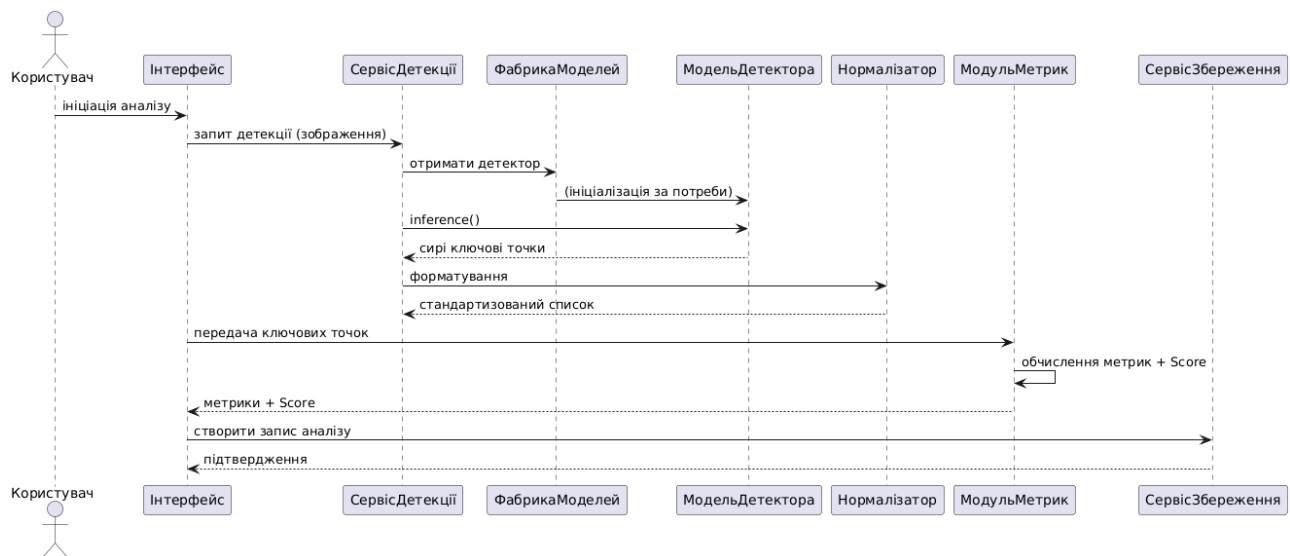


Рисунок 5.7 – Діаграма послідовностей процесу аналізу постави

Представлена на рисунку 5.7 діаграма послідовності відображає цілісний алгоритмічний конвеєр аналізу постави – від моменту ініціації запиту клінічним користувачем до формування та збереження структурованого результату.

Процес розпочинається на рівні інтерфейсу користувача, який передає зображення для детекції ключових точок. Сервіс детекції взаємодіє з фабрикою моделей для отримання відповідного детектора та викликає нейронну модель, яка виконує інференс і повертає сирі координати.

Далі ці дані проходять етап нормалізації, де координати узгоджуються до стандартизованого формату, що є необхідним для коректного розрахунку біомеханічних метрик. Нормалізований набір точок передається модулю метрик, який виконує обчислення геометричних характеристик, симетричних відхилень та інтегрального показника якості постави.

Після формування результату інтерфейс ініціює його збереження у відповідному сервісі, що забезпечує створення аналітичного запису і повертає підтвердження користувачеві.

5.4 Використані технології

Для реалізації програмної системи було застосовано низку сучасних технологій, інструментів та програмних засобів, які забезпечують коректне функціонування усіх компонентів – від збору та обробки даних до зберігання результатів аналізу та взаємодії з користувачем через веб-інтерфейс. Вибір технологій обумовлений необхідністю створення продуктивної, масштабованої та зручно підтримуваної системи.

Основною мовою програмування, використаною під час розробки, є **Python**, що завдяки своїй гнучкості, великій кількості бібліотек та підтримці машинного навчання дозволив ефективно реалізувати всі алгоритми аналізу ключових точок, обробки зображень і розрахунку метрик постави. Для реалізації серверної частини системи використано сучасний асинхронний фреймворк **FastAPI**, який забезпечує високу продуктивність, зручну типізацію та автоматичне генерування документації API у форматі OpenAPI/Swagger. FastAPI було обрано завдяки його можливості обробляти велику кількість запитів із мінімальною затримкою, простій інтеграції з моделями машинного навчання та підтримці асинхронного виконання, що є особливо корисним при роботі з ресурсномісткими операціями, такими як обробка зображень або виклик моделей глибинного навчання. Структура FastAPI надає чітке розділення маршрутів, схем даних та логіки обробки, що полегшує підтримку та розширення системи. Автоматично згенерована документація API суттєво спрощує тестування, налагодження та інтеграцію фронтенду, мобільних клієнтів або сторонніх сервісів.

Для реалізації клієнтської частини системи застосовано фреймворк **Vue.js**, що забезпечує реактивний інтерфейс та дозволяє створити динамічні елементи, такі як перегляд результатів аналізу, візуалізація ключових точок та інтерактивні графічні компоненти. Архітектура Vue.js забезпечує розділення логіки на компоненти та зручну організацію стану застосунку, що покращує підтримку і масштабованість фронтенду.

Для зберігання даних обрана реляційна система керування базами даних **PostgreSQL**, яка відзначається високою продуктивністю, підтримкою складних запитів та надійністю. Використання PostgreSQL дозволяє організувати зберігання інформації про користувачів, результати аналізу постави, журнал оброблених зображень та інші метадані системи. Взаємодія між Python-частиною та базою даних реалізована через ORM-бібліотеку **SQLAlchemy**, що забезпечує зручне моделювання таблиць у вигляді Python-класів та спрощує виконання CRUD-операцій.

Для виконання аналізу постави та обробки зображень застосовуються бібліотеки **Torch** і **Torchvision**, які забезпечують завантаження, виконання та інференс нейронних мереж, включаючи моделі детекції ключових точок. Додатково для моделей Yolo-Pose застосовується бібліотека Ultralytics, яка у свою чергу використовує Torch. Завдяки цим інструментам система може використовувати раніше навчені моделі, що у свою чергу дозволяє визначати координати ключових точок тіла необхідні для комплексної оцінки постави.

Для роботи із зображеннями додатково застосовуються бібліотеки **OpenCV** та **Pillow**, які дозволяють коректно виконувати попередню обробку даних, масштабування, конвертацію та фільтрацію зображень перед подачею їх у мережу. Це забезпечує стабільність результатів та уніфікацію вхідних даних.

Комунікація між клієнтською та серверною частиною реалізована за допомогою REST API, що забезпечує універсальність та сумісність із будь-якими клієнтами, здатними формувати HTTP-запити. Такий підхід спрощує інтеграцію системи у більші інформаційні комплекси, а також дозволяє розробляти мобільні застосунки або інші зовнішні модулі.

Таким чином, використані технології формують цілісну архітектуру, яка поєднує сучасні підходи до веб-розробки, глибинного навчання та управління даними. Їхня комбінація забезпечує стабільну роботу системи, високу точність аналізу, зручний інтерфейс та можливість подальшого розширення функціональності.

5.5 Реалізація програмних модулів.

5.5.1 Серверна частина

Реалізація серверної частини ґрунтується на структурованому REST-API, побудованому на базі фреймворку FastAPI із чіткою диференціацією відповідальностей між маршрутизаторами, доменними моделями, схемами даних і утилітарними сервісами. API організовано за логічними доменними префіксами: автентифікація (/auth), керування пацієнтами (/patients), робота зі зображеннями (/images), детекція ключових точок (/detection) та аналітична обробка (/analyses). Кожний ендпоінт формує чіткий контракт «вхід–вихід»: наприклад, POST /auth/login приймає облікові дані (username, password) та повертає JSON-структуру із маркерами доступу (access_token, refresh_token) та серіалізованим представленням користувача; POST /patients/ створює сутність пацієнта через адаптований DTO без експонування внутрішніх технічних полів; GET /patients/{id} повертає розширену композицію зі списком пов'язаних зображень і аналізів.

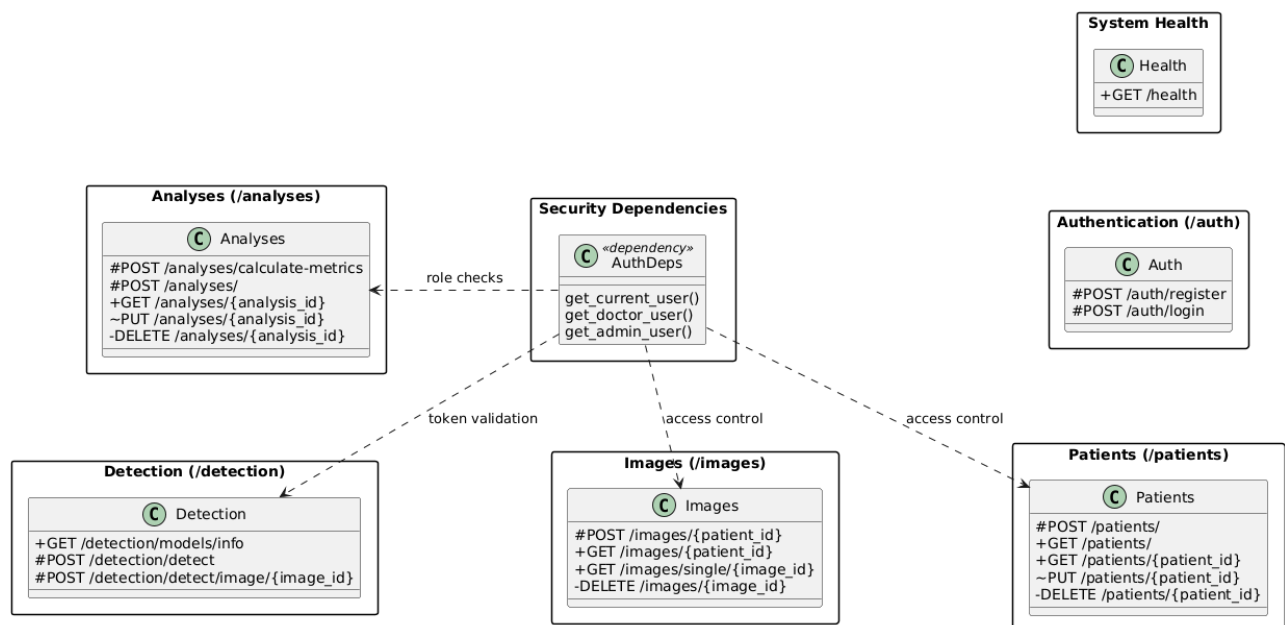


Рисунок 5.8 – Діаграма API endpoints

На рисунку 5.8 зображено діаграм, яка групує ендпоінти за логічними ресурсами з позначенням HTTP методів. Кожен клас відображає контракт REST

інтерфейсу без внутрішніх реалізацій. Блок залежностей показує централізоване застосування механізмів автентифікації й авторизації до доменних ресурсів.

Рівень безпеки забезпечується перевіркою JWT у dependency-функціях, що вбудовуються як декларативні параметри контролерів, з наступною перевіркою ролей і приналежності ресурсу.

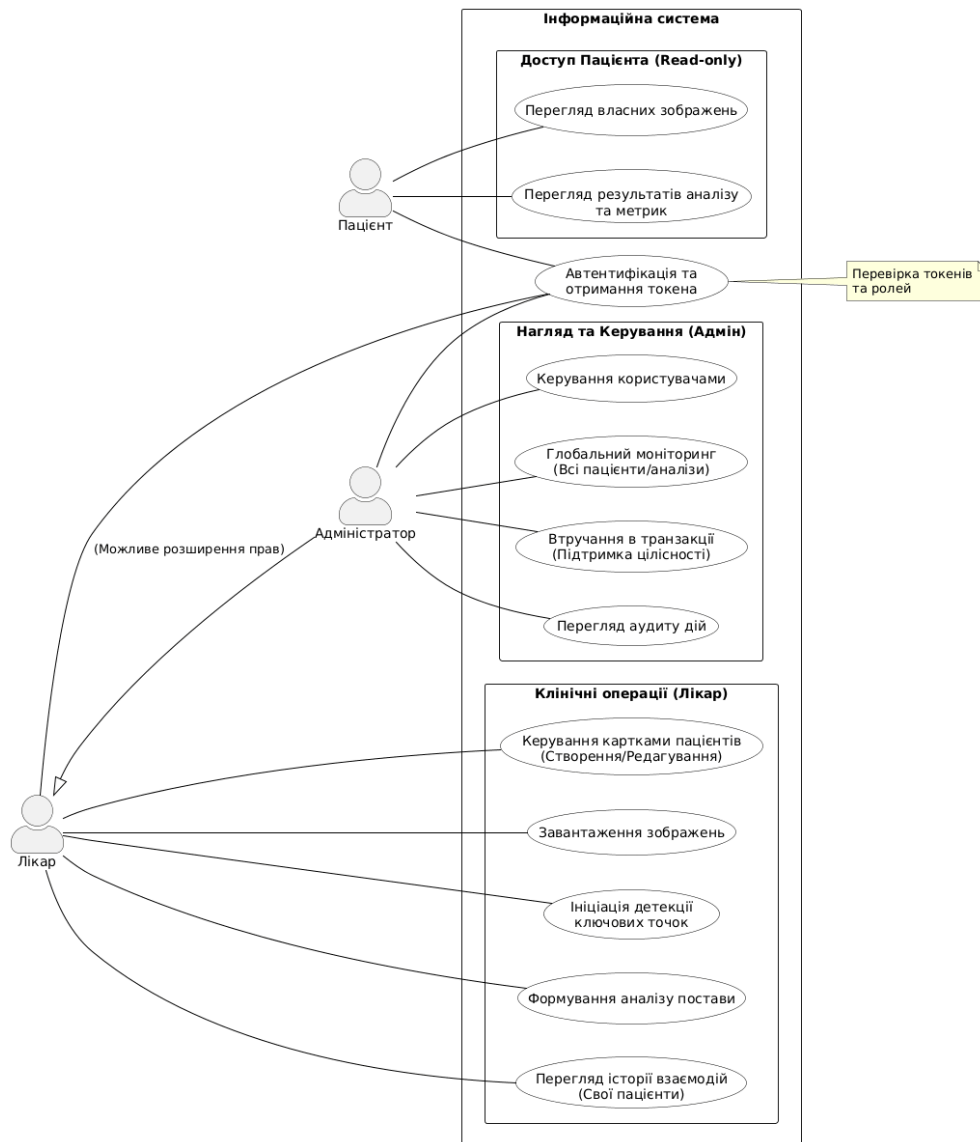


Рисунок 5.9 – Рольова модель доступу системи аналізу постави

Рольова модель системи, яка зображена на рисунку 5.9, передбачає три страти доступу: адміністратор, лікар та пацієнт. Адміністратор виконує над системні функції нагляду і керування – має глобальний огляд усіх пацієнтів, аналізів, користувачів та може втручатися у будь-яку транзакцію для підтримки цілісності й безпеки. Лікар виступає операційним суб'єктом процесів: створює і редагує картки пацієнтів, завантажує зображення, ініціює детекцію ключових

точок, формує аналізи постави й переглядає історію взаємодій лише тих пацієнтів, за яких відповідає. Пацієнт має мінімально необхідний обсяг читального доступу для перегляду власних зображень, аналізів та показників, не впливаючи на діагностичні записи та не модифікуючи клінічні дані.

Схема бази даних побудована на основі реляційної моделі та складається з п'яти нормалізованих сутностей, що забезпечують цілісність клінічних даних та безпеку доступу. Ядром системи автентифікації виступає сутність `users`, яка зберігає облікові дані, розмежовує ролі (адміністратор, лікар, пацієнт). Блок медичних даних організовано ієрархічно: сутність `patients` містить персональні дані та посилання на лікаря-куратора (`doctor_id`), а також має зв'язок "один-до-багатьох" із завантаженими знімками. Результати роботи інтелектуальних алгоритмів зберігаються в таблиці `analyses`, яка агрегує як розраховані біомеханічні метрики, так і текстові висновки, маючи суворий зв'язок із конкретним зображенням та лікарем, що ініціював аналіз. На рисунку 5.10 зображено ER діаграму бази даних

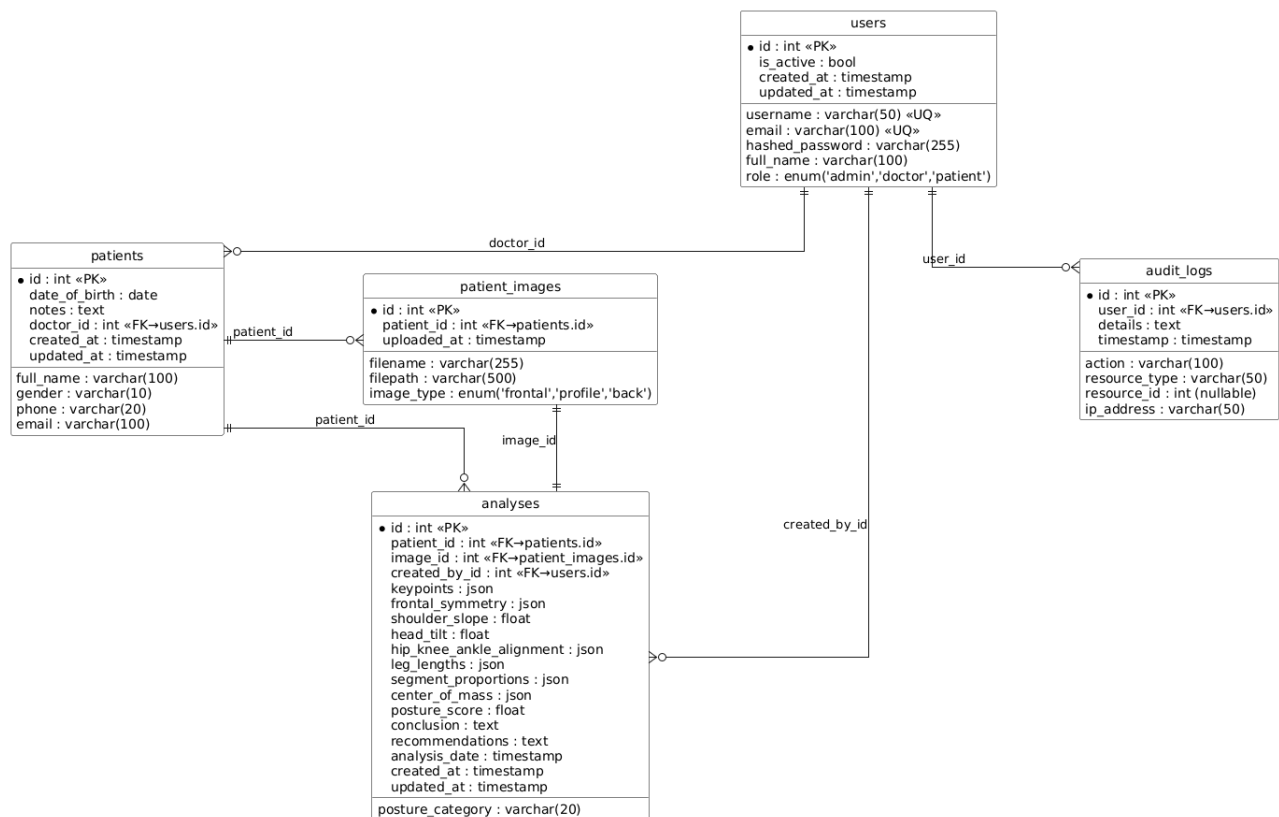


Рисунок 5.10 – ER діаграма бази даних

Механіка формування JSON-відповідей реалізує сувору ізоляцію шарів: доменні моделі, отримані з ORM, перетворюються у Pydantic-схеми з увімкненим режимом `from_attributes`. Це дає контроль над полями, що експонуються, забезпечує універсальність валідації типів і автоматичну генерацію OpenAPI документації. Композитні відповіді (наприклад, деталізований пацієнт із масивами вкладених аналізів та зображень) формуються через опції ORM-запиту з `joinedload` завантаженням залежних колекцій з метою зменшення латентності серіалізації. Метрики аналізу зберігаються в JSON полях моделі «Analysis» без повторного обчислення при читанні, забезпечуючи детермінований доступ і можливість історичного порівняння. Стандартизований патерн обробки передбачає отримання сесії бази даних через залежність `get_db()`, атомарне виконання транзакції та формування відповіді через Pydantic-схеми, що запобігає несанкціонованій серіалізації приватних атрибутів.

Алгоритм обробки завантажених зображень реалізовано через модульний медіа-потік: при виклику `POST /images/{patient_id}` запит у форматі `multipart` містить бінарний буфер файлу та тип зображення як формове поле. Спеціалізована утиліта перевіряє розширення, генерує унікальне ім'я файлу та розміщує його у дереві директорій, прив'язаному до ідентифікатора пацієнта. Після успішного збереження формується ORM-об'єкт з вказанням типу ракурсу і шляху до файлу, з наступним підтвердженням транзакції та аудит-записом дії «`upload_image`». Видалення зображення симетрично поєднує логічну перевірку доступу, фізичне вилучення файлу та видалення запису.

Процес виклику моделі детекції реалізовано у двох сценаріях:

- безпосередній детект ключових точок на завантаженому зображенні (`POST /detection/detect` із файлом);
- детект на вже збереженому ресурсі (`POST /detection/detect/image/{image_id}`);

У будь-якому сценарії, визначається тип моделі, ініціалізується інстанс детектора через фабрику, зображення перетворюється у необхідне представлення після чого виконується `inference` та формується список детекцій. Усі сирі вихідні

тензори приводяться до уніфікованого структурного формату списку словників з іменами ключових точок, координатами та показниками достовірності. Якщо модель не повернула валідні дані, застосовується набір синтетичних координат для забезпечення тестованості алгоритмів. Цей самий формат передається на аналітичний ендпоінт або інтегрується у створення аналізу.

5.5.2 Клієнтська частина

Клієнтська частина, реалізована на базі Vue.js та управління станом через Pinia, формує узгоджений клієнтський шар, що опосередковує всі взаємодії із API. Компоненти PatientsList, PatientDetail, UploadImage, Analysis, Report організовані за принципом функціональної спеціалізації: перший компонент відповідає за фільтрацію та відображення колекції пацієнтів, другий – за деталізований перегляд із вкладеними списками зображень та аналізів, третій – за ініціацію завантаження файлу, четвертий – за редагування або візуалізацію ключових точок, п'ятий – за генерацію структурованого звіту.

Діаграма на рисунку 5.11 відображає компонентну структуру клієнтського шару: сторінкові (View) компоненти оркеструються маршрутизатором і взаємодіють зі станом, що інкапсулює асинхронні виклики до бекенду через узгоджений сервіс api.js. Така топологія мінімізує зв'язність: сторінки не звертаються безпосередньо до бекенду, а лише через абстракцію сервісу і стан, що спрощує тестування, розширення та можливу заміну транспортного шару.

Ключовою особливістю архітектури є те, що Views не звертаються до бекенду напрямку. Натомість вони взаємодіють із шаром управління станом (State Management), який виступає "єдиним джерелом істини" та інкапсулює бізнес-логіку, включно з асинхронними операціями. У свою чергу, сховище стану делегує виконання мережевих запитів узгодженому сервісу api.js. Така багат шарова топологія забезпечує слабку зв'язність компонентів: зміни в структурі API або транспортних протоколах не впливають на візуальну частину, а

ізоляція логіки в окремих сервісах значно спрощує модульне тестування, масштабування системи та підтримку кодової бази.

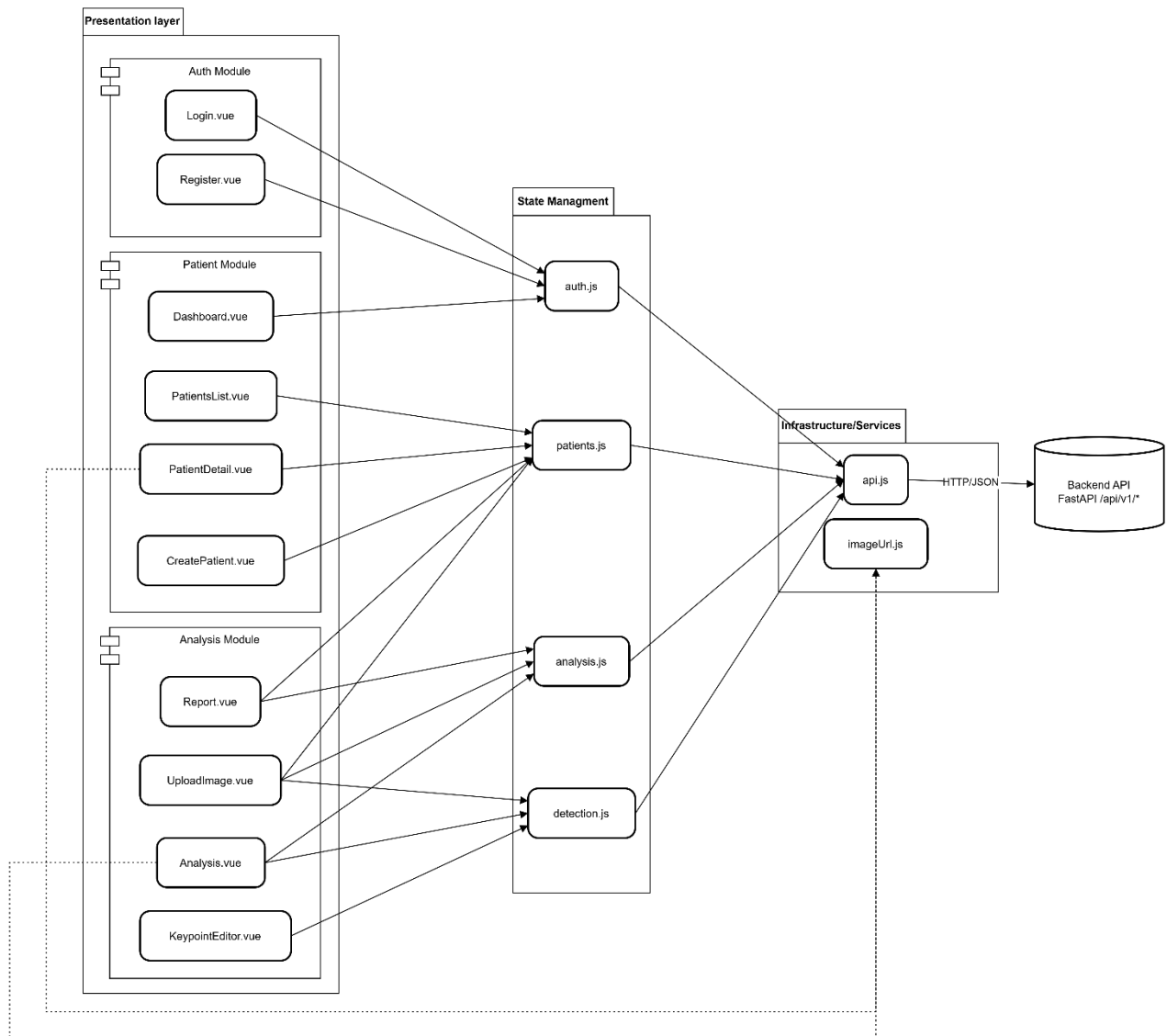


Рисунок 5.11 – Діаграма компонент фронтенду

Візуалізація ключових точок та метрик на інтерфейсі базується на двох узагальнених підходах:

- накладання координат точок поверх зображення через позиціонування HTML/Canvas-елементів із нормалізацією координат до розмірів контейнера,
- табличне або карточне представлення підсумкових метрик (бал, категорія, часткові симетрії).

Формат ключових точок (ім'я–координати–достовірність) дозволяє як числову, так і графічну інтерпретацію без додаткових перетворень.

Обробка відповідей сервера на фронтенді включає послідовні стадії: перевірка формату (очікування масиву чи об'єкта), оновлення реактивного стану, сповіщення користувача у випадку відмови та ініціювання вторинних запитів наприклад, після успішного створення пацієнта – навігація до сторінки його деталей. Структурна простота відповіді мінімізує ризик неконсистентних клієнтських перетворень і знижує когнітивні витрати при інтеграції нових компонентів.

Завдяки такій реалізації досягається низка інженерних переваг:

- прозорість інтерфейсів і відтворюваність поведінки,
- знижена зв'язність між аналітичним кодом та клієнтською візуалізацією,
- можливість незалежного масштабування компонентів детекції,
- трасованість дій через аудит без забруднення бізнес-логіки.

Архітектурна ортогональність модулів створює сприятливий фундамент для подальшого розширення (додавання нових метрик, альтернативних візуалізацій, експортних форм) без порушення існуючих зв'язків.

5.6 Тестовий приклад

Для демонстрації працездатності програмного комплексу та коректності виконання функцій аналізу постави було проведено тестовий прогін системи за повним сценарієм: від реєстрації лікаря до отримання кількісних показників аналізу зображення пацієнта.

Перед початком роботи з клієнтською частиною важливо зазначити, що архітектура системи базується на чіткому розмежуванні серверної та клієнтської логіки. Взаємодія між компонентами відбувається через REST API. Для зручності інтеграції та тестування ендпоінтів FastApi автоматично генерується інтерактивна документація, за допомогою Swagger/OpenAPI, яка описує структуру запитів та відповідей для сутностей користувачів, пацієнтів та результатів аналізу. На рисунках 5.12 та 5.13 зображені сторінки згенерованої документації за допомогою Swagger та OpenApi відповідно.

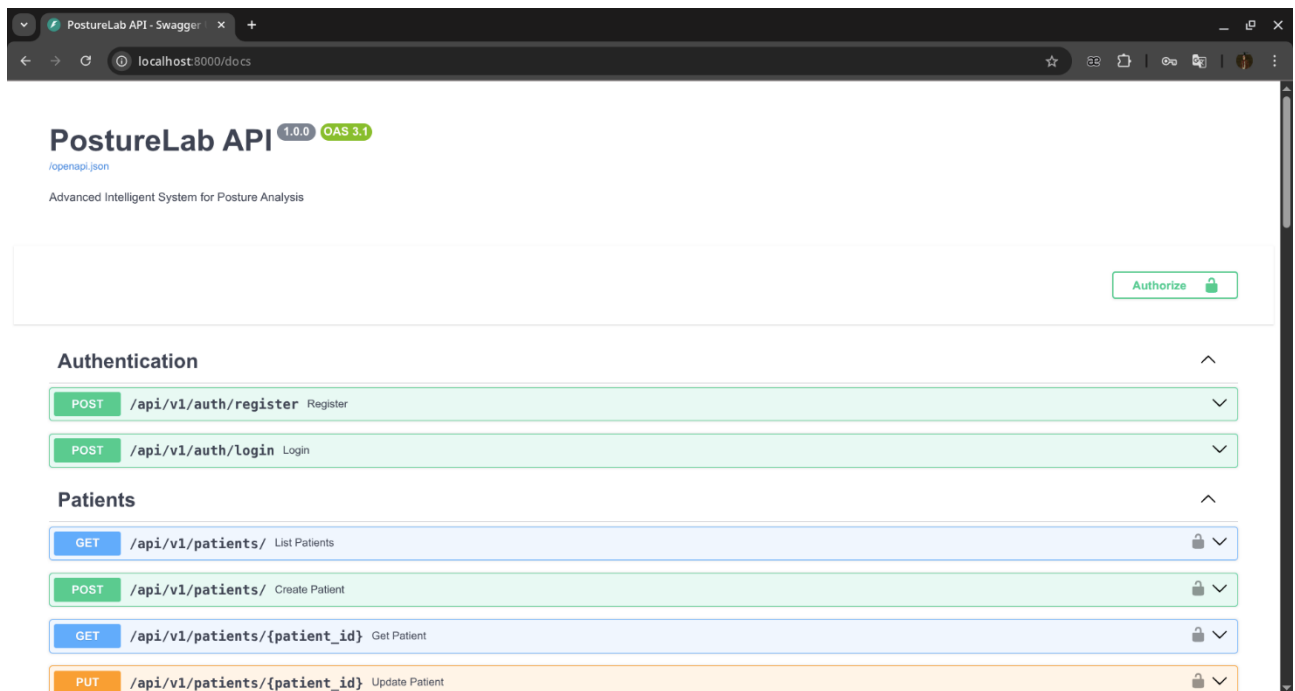


Рисунок 5.12 – Документація Swagger

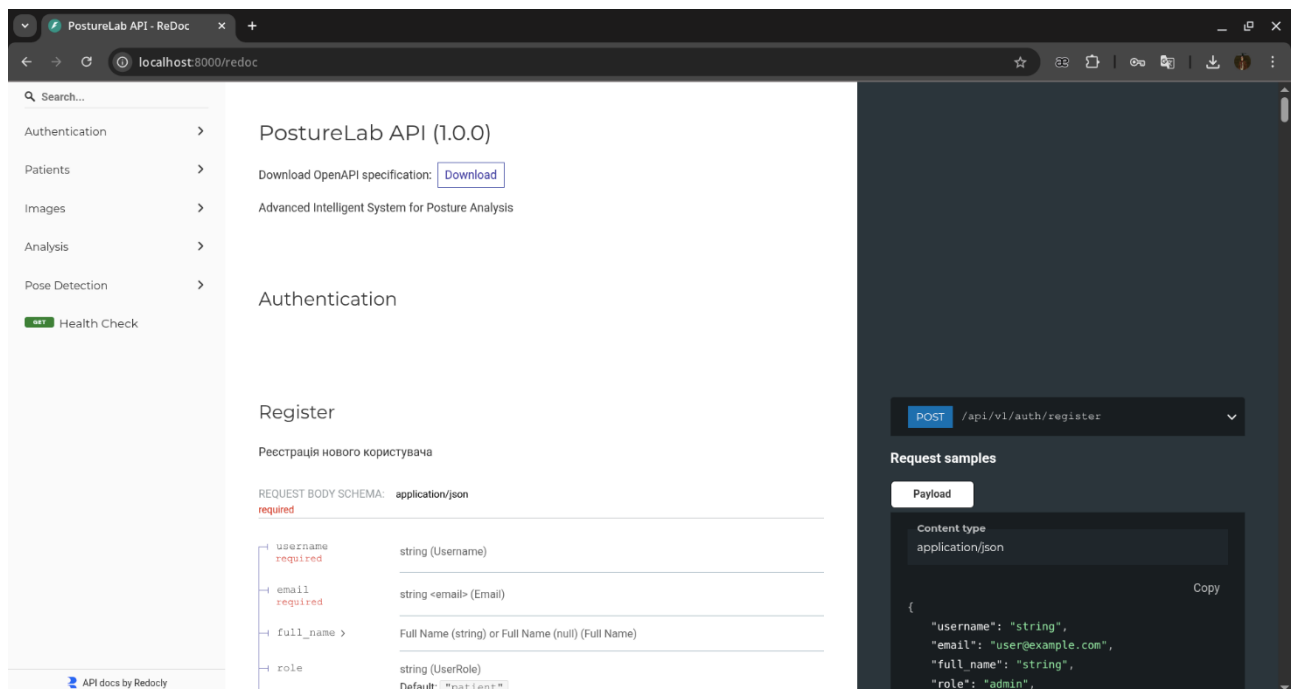
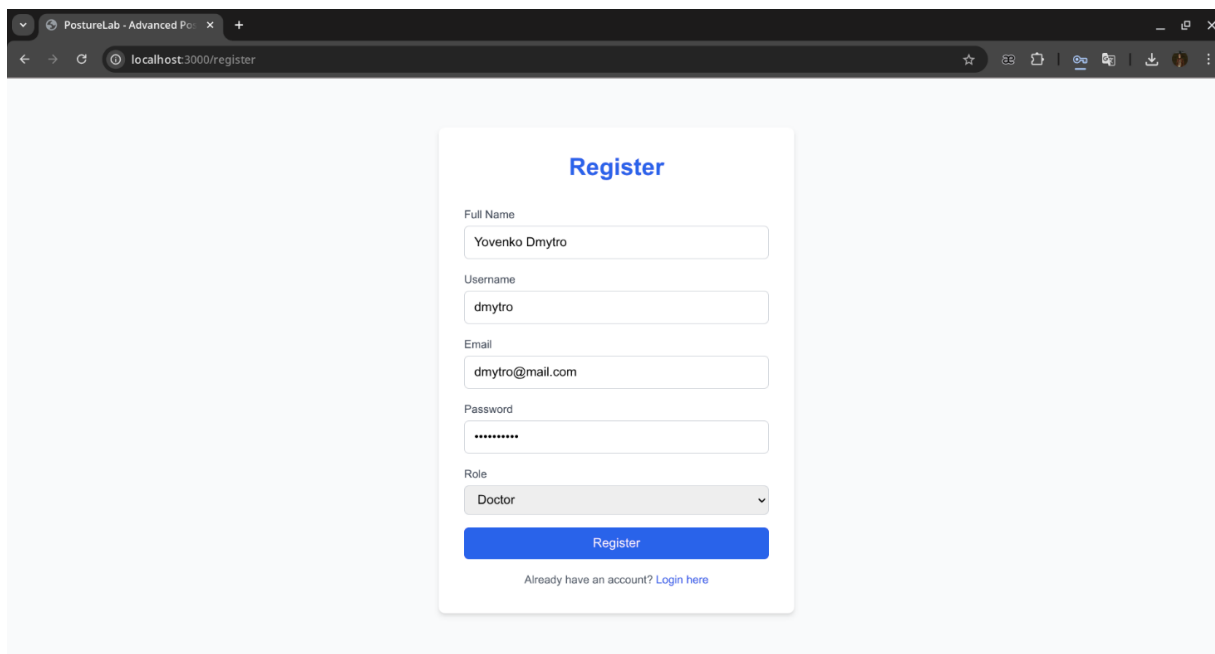


Рисунок 5.13 – Документація OpenApi

Використання Swagger значно спростило та пришвидшило процес розробки та тестування API. Завдяки інтерактивному інтерфейсу тестування API не потребує сторонніх застосунків.

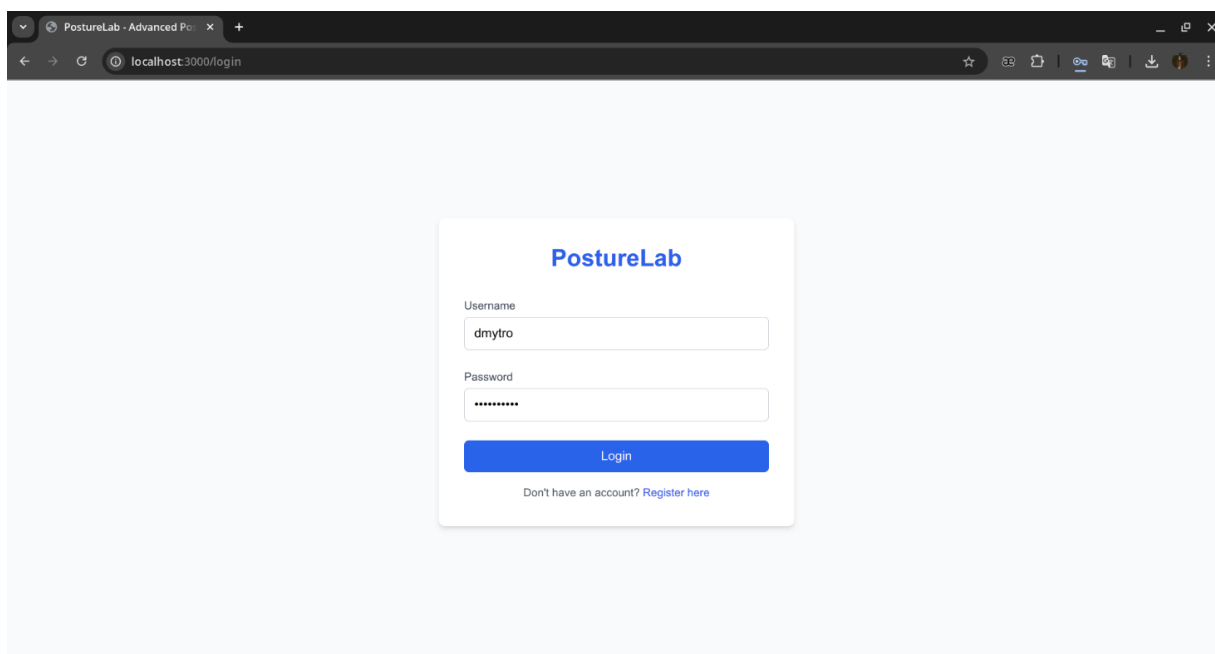
Робота користувача з веб-інтерфейсом розпочинається з процедури автентифікації. Якщо користувач не має облікового запису, він переходить до

форми реєстрації, що зображена на рисунку 5.14, де вказує повне ім'я, ім'я користувача, електронну пошту, пароль та обирає роль. Після успішної реєстрації або введення облікових даних у форму логіну(рис. 5.15), користувач отримує доступ до захищеної частини системи.



The screenshot shows a web browser window with the address bar displaying 'localhost:3000/register'. The page features a central registration form titled 'Register' in blue. The form contains the following fields: 'Full Name' with the value 'Yovenko Dmytro', 'Username' with 'dmytro', 'Email' with 'dmytro@mail.com', 'Password' with masked characters, and a 'Role' dropdown menu set to 'Doctor'. A prominent blue 'Register' button is located below the form. At the bottom of the form, there is a link that reads 'Already have an account? Login here'.

Рисунок 5.14 – Сторінка реєстрації



The screenshot shows a web browser window with the address bar displaying 'localhost:3000/login'. The page features a central login form titled 'PostureLab' in blue. The form contains the following fields: 'Username' with the value 'dmytro' and 'Password' with masked characters. A prominent blue 'Login' button is located below the form. At the bottom of the form, there is a link that reads 'Don't have an account? Register here'.

Риснуок 5.15 – Сторінка авторизації

Після входу в систему користувач потрапляє на головну панель (Dashboard). Ця сторінка слугує центром керування, відображаючи загальну статистику

(кількість пацієнтів, проведених аналізів) та надаючи швидкий доступ до основних дій – перегляду списку пацієнтів, створення нового пацієнта.

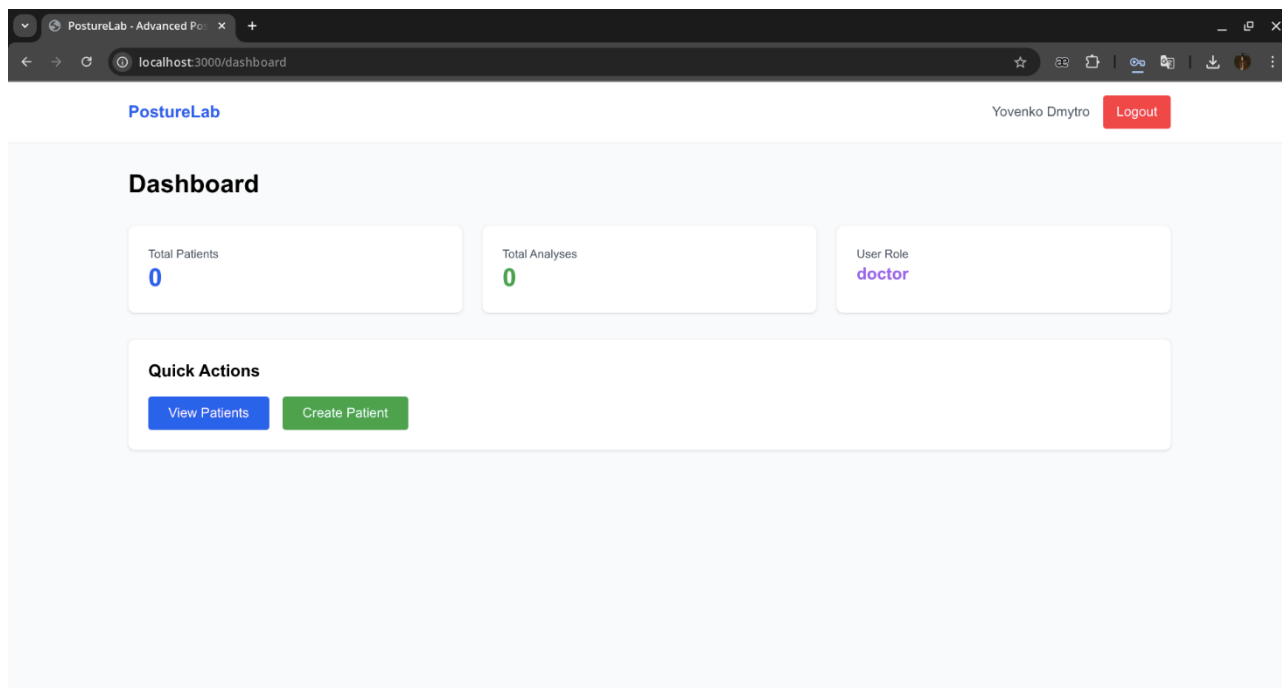


Рисунок 5.16 – Головна панель користувача

The screenshot shows a web browser window with the URL 'localhost:3000/patients/new'. The form is titled 'Create New Patient'. It contains the following fields: 'Full Name *' with the value 'Василь', 'Date of Birth *' with the value '01.01.2000', 'Gender' with a dropdown menu set to 'Male', 'Phone' with the value '0987654321', 'Email' with the value 'email@mail.com', and 'Notes' with the value 'Пацієнт1'. At the bottom of the form are two buttons: 'Create Patient' (blue) and 'Cancel' (grey).

Рисунок 5.17 – Форма створення нового пацієнта

Натискання кнопки створення пацієнта, яка розташована на головній панелі(див. рис. 5.16), відкриває форму, яка зображена на рисунку 5.17, де лікар вносить персональні дані: ПІБ, дату народження, стать та контактну інформацію.

Після збереження даних створюється профіль пацієнта(рис. 5.18). Інтерфейс профілю містить інформацію про пацієнта та історію завантажених зображень, збережених аналізів.

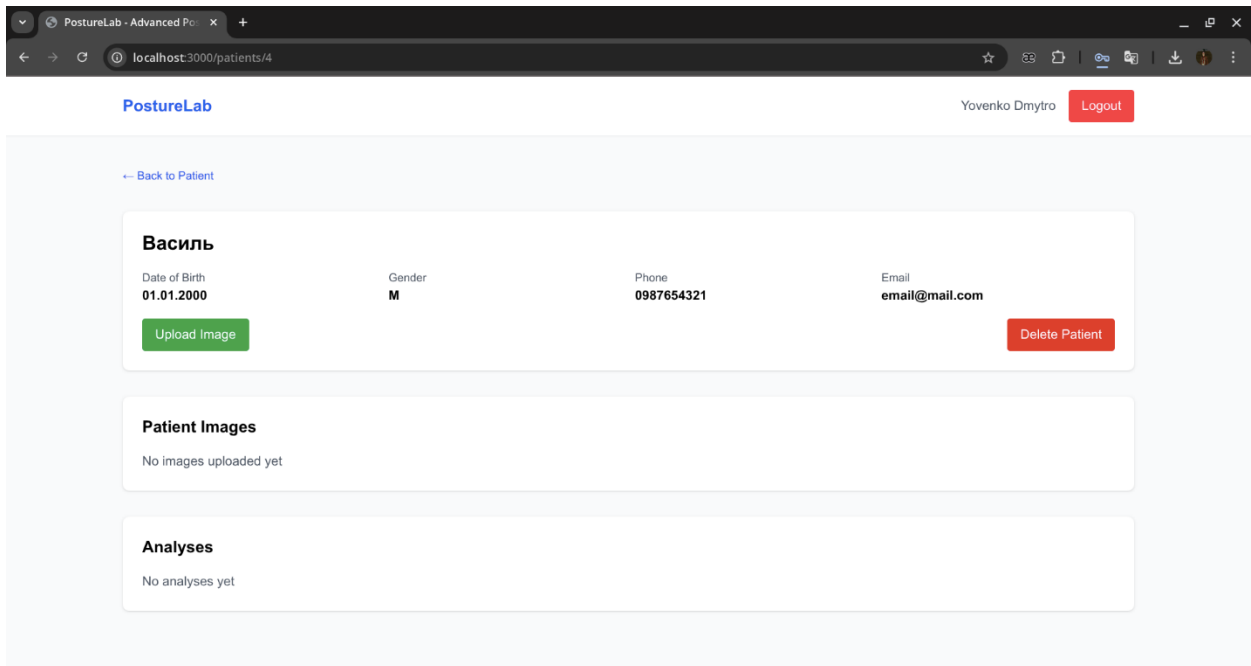


Рисунок 5.18 – Профіль пацієнта

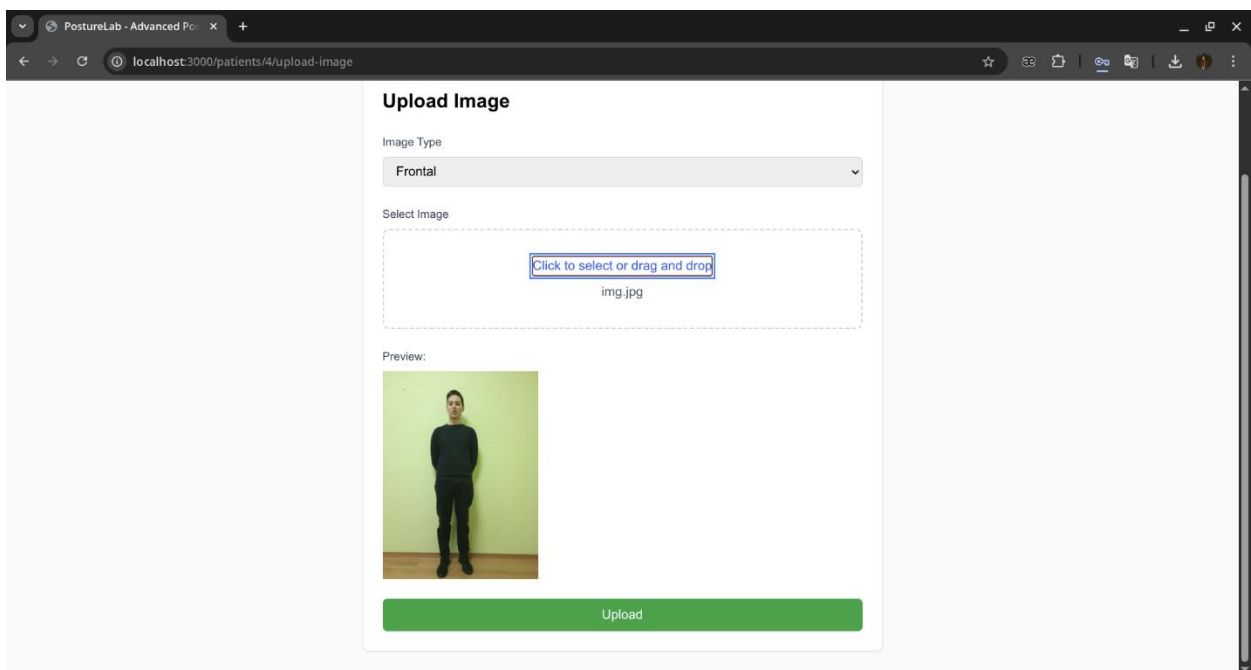


Рисунок 5.19 – Форма завантаження зображення

Процес завантаження реалізовано через спеціалізовану форму(зображена на рисунку 5.19), яка підтримує вибір типу зображення та механізм drag-and-drop.

Ключовим етапом роботи є безпосередній аналіз постави з використанням методів штучного інтелекту. За замовчуванням система використовує модель Yolo Pose для визначення ключових точок по завантаженні сторінки побудови аналізу(рис. 5.20)

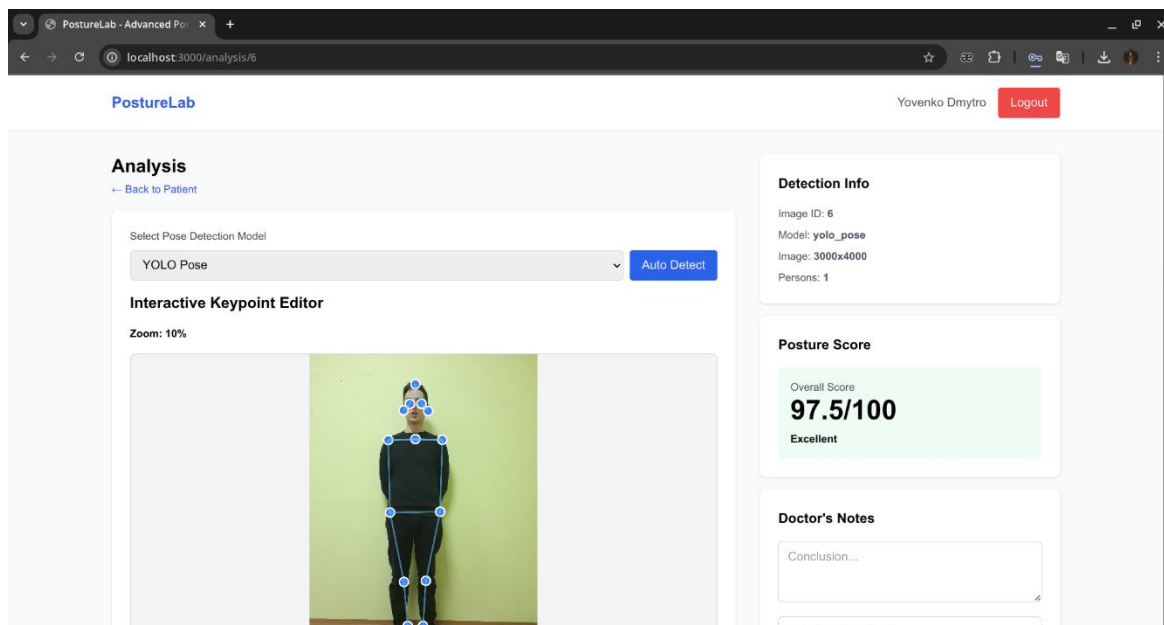


Рисунок 5.20 – Сторінка побудови аналізу з автоматично детекцією ключових

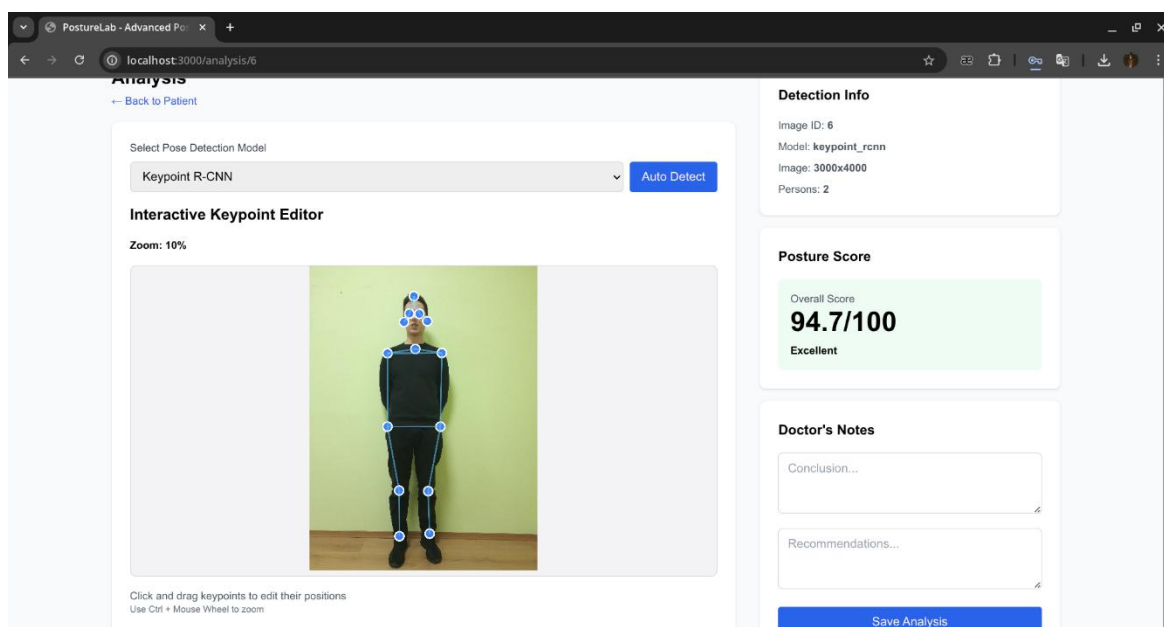


Рисунок 5.21 – Детекція точок за допомогою Keypoint-RCNN

Після обробки зображення нейромережею на екрані відображається візуалізація скелетної моделі, накладеної на фотографію пацієнта. Важливою особливістю інтерфейсу є інтерактивний редактор ключових точок ("Interactive

Keypoint Editor"), який дозволяє оператору вручну скоригувати положення маркерів у випадку, якщо автоматична детекція виявилася неточною через складні умови освітлення або одяг пацієнта.

На основі координат детектованих точок система автоматично розраховує біомеханічні метрики. На сторінці аналізу метрики виводяться у вигляді плиток, що включає оцінку фронтальної симетрії (по положенню очей, вух, плечей, стегон, колін та кісточок) та кутові метрики (нахил голови, плечей, тазу) та загальну оцінку постави (Posture Score). Також присутні поля для текстового висновку лікаря і рекомендацій.

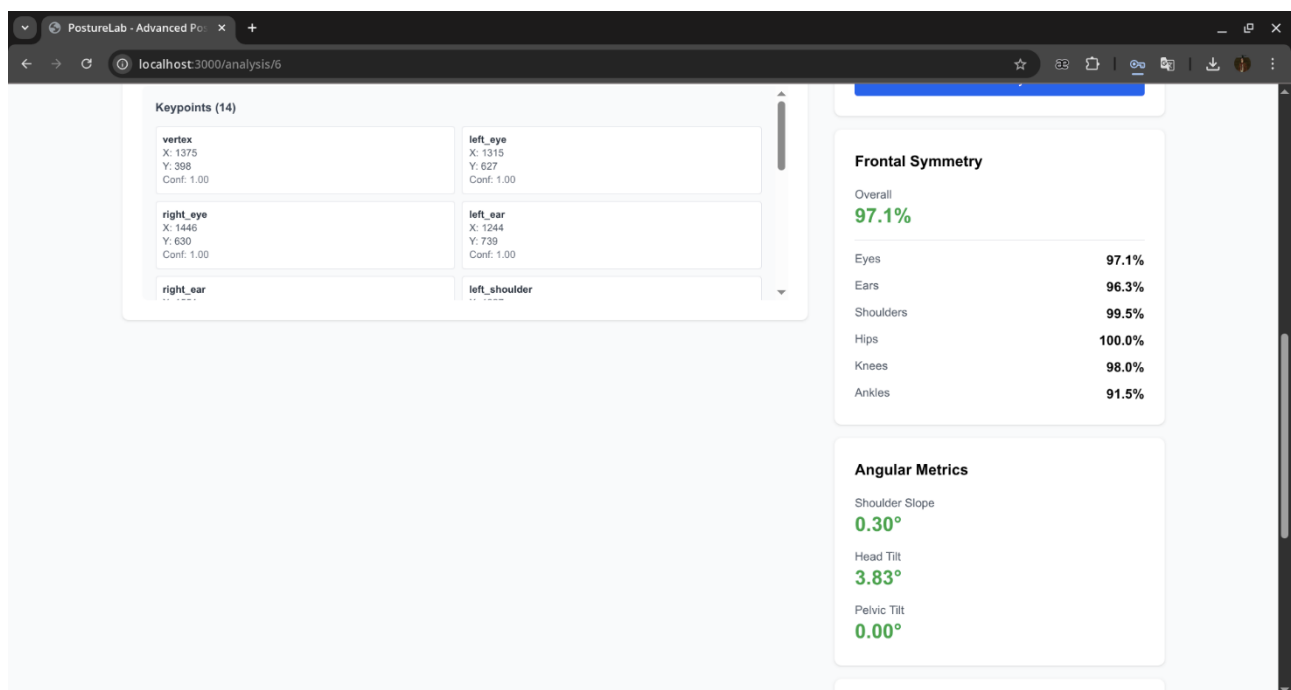


Рисунок 5.22 – Відображення метрик на сторінці побудови аналізу

Після перевірки місцезнаходження точок, складання висновку та рекомендацій, лікар може зберегти результати аналізу у профілі користувача. Сторінка відображення збереженого звіту зображена на рисунку 5.23 Для забезпечення можливості документування та передачі результатів пацієнту або інтеграції у зовнішні системи медичного документообігу, присутня можливість експорту звіту у форматі Portable Document File(рис.5.24).

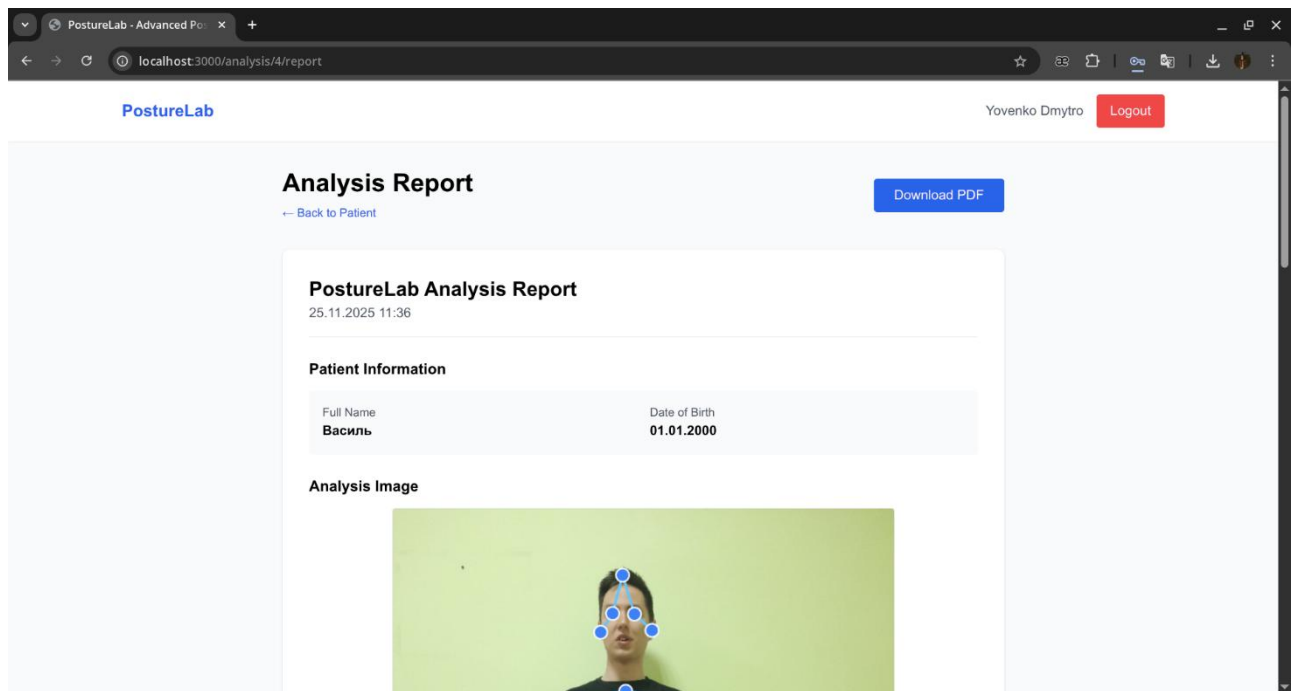


Рисунок 5.23 – Збережені результати аналізу

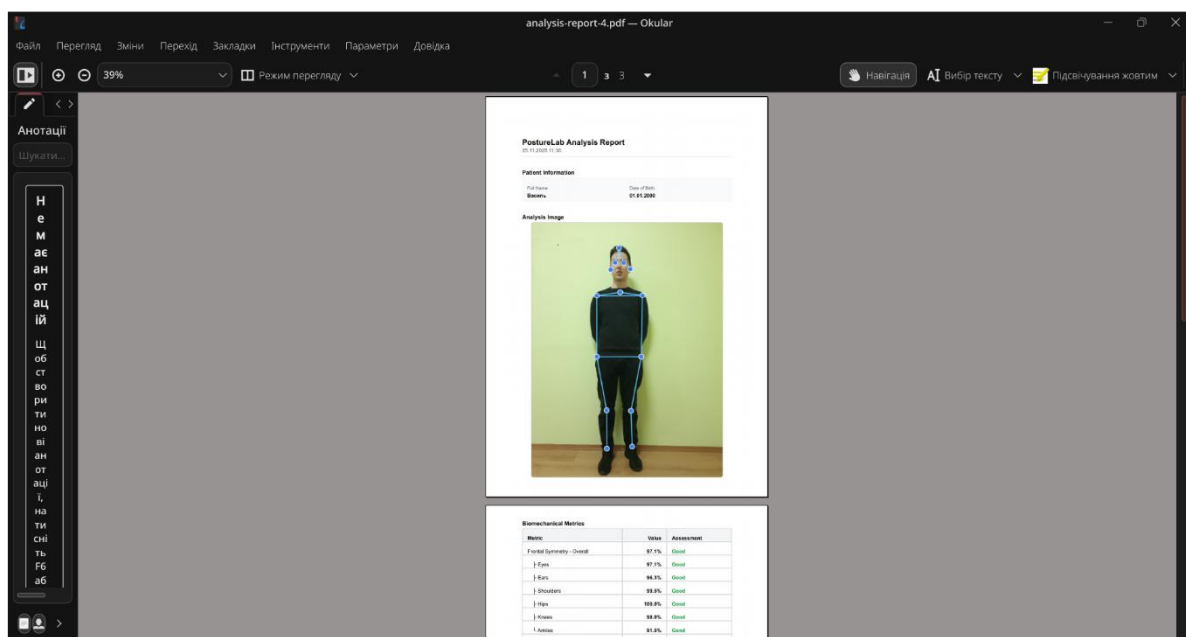


Рисунок 5.24 – Експортований pdf звіт

Згенерований PDF-файл повністю зберігає структуру веб-звіту, включаючи анотацію зображення, персональні дані пацієнта та таблицю показників.

Важливою функцією інформаційної системи є зручне управління реєстром пацієнтів. Для перевірки масштабованості інтерфейсу до бази даних було додано декілька тестових записів. Сторінка "Patients" відображає список усіх карток з

ключовою інформацією. Для швидкої навігації реалізовано механізм пошуку та фільтрації "на льоту". Введення пошукового запиту (наприклад, літери "П") миттєво відфільтровує список, залишаючи лише релевантні записи, що значно пришвидшує роботу лікаря при великій кількості пацієнтів.

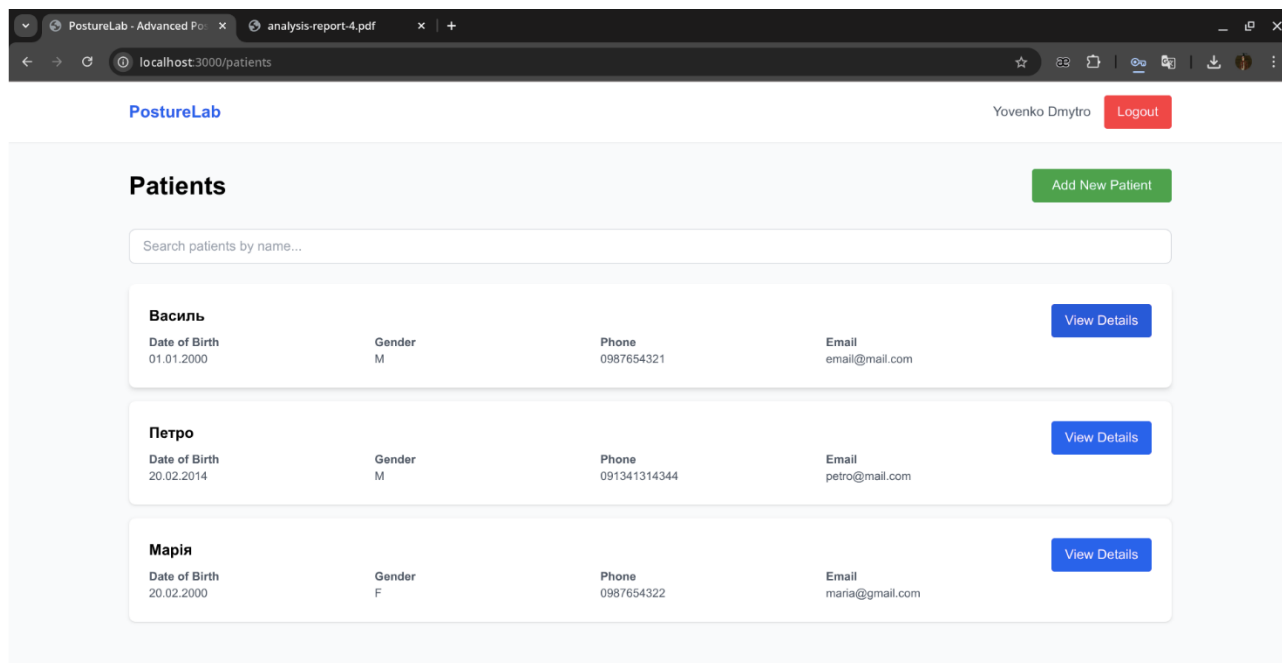


Рисунок 5.25 – Загальний реєстр пацієнтів

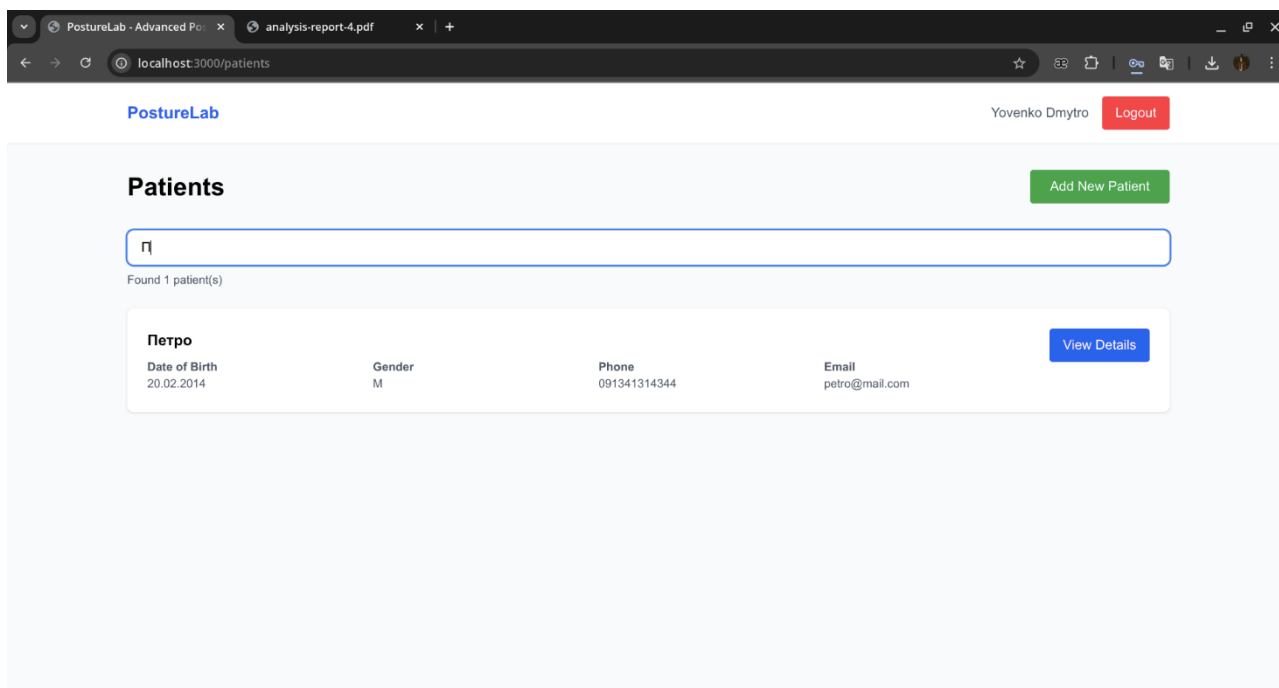


Рисунок 5.26 – Демонстрація роботи функції пошуку

Коректність роботи серверної частини та бази даних додатково підтверджується оновленням глобальної статистики. Після реєстрації нових користувачів та проведення досліджень, лічильники на головній панелі автоматично актуалізуються, відображаючи поточний стан системи.

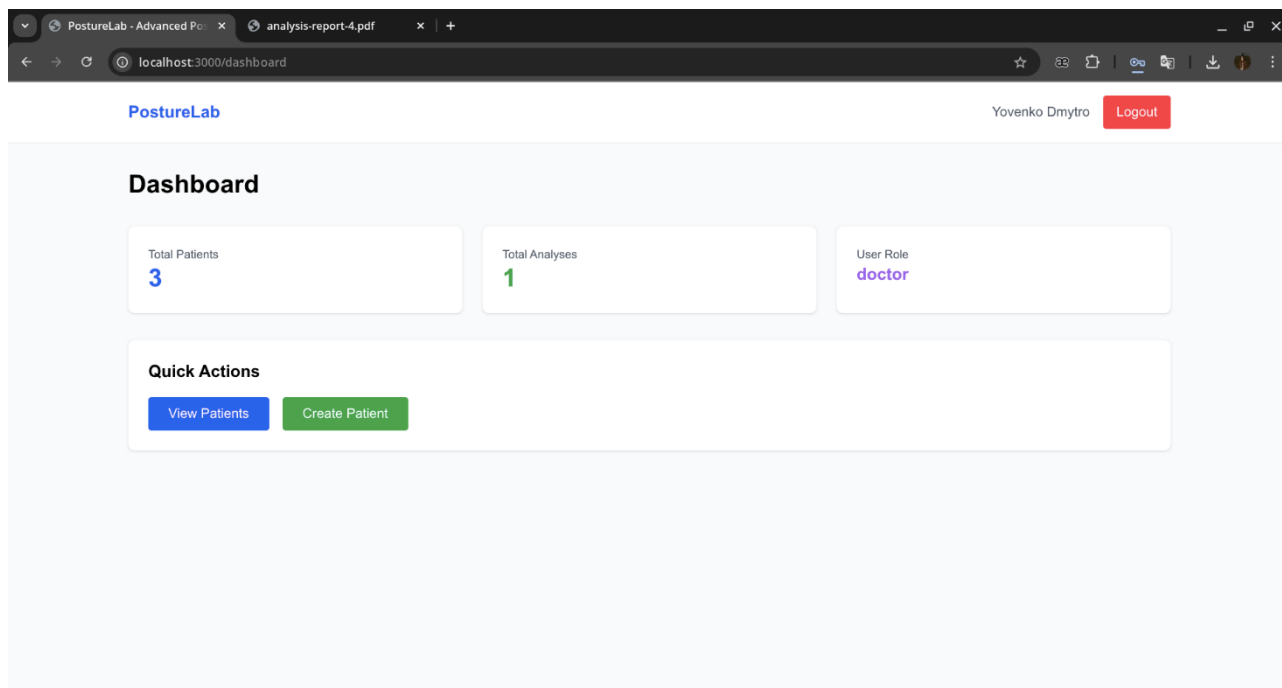


Рисунок 5.27 – Актуалізація статистики на головній панелі після додавання даних

Таким чином, проведений тестовий приклад продемонстрував повну працездатність системи, коректність виконання ключових алгоритмів аналізу постави, а також надійність допоміжних функцій управління даними.

Висновок

У ході виконання кваліфікаційної випускної роботи було розроблено та досліджено інформаційну систему аналізу постави, що поєднує сучасні методи комп'ютерного зору, машинного навчання та програмної інженерії. Метою роботи було створення інструменту, здатного автоматично визначати ключові точки людського тіла, розраховувати біомеханічні показники та формувати узагальнену оцінку постави на основі цифрових зображень.

У процесі дослідження спочатку було детально проаналізовано сучасні методи визначення ключових точок людського тіла, зокрема моделі YOLO Pose та Keypoint R-CNN, оцінено їх переваги та обмеження для задач соматоскопічного аналізу. Обидві моделі були додатково до навчені на сформованому власному датасеті, що забезпечило підвищення точності детекції та адаптацію мереж до специфічних анатомічних орієнтирів. Після етапу моделювання було спроектовано та реалізовано клієнт-серверну інформаційну систему, що включає серверну частину на основі FastAPI, клієнтську частину на Vue.js, базу даних PostgreSQL та модуль інтелектуального аналізу. Реалізовано алгоритми попередньої обробки зображень, визначення ключових точок, розрахунку кутових і лінійних метрик, а також формування комплексного індексу постави.

Автоматизація аналізу дозволяє зменшити суб'єктивність оцінювання, підвищити точність діагностики та спростити моніторинг динаміки змін у пацієнтів.

Результати роботи показують, що використання нейронних мереж у сфері соматоскопії є перспективним напрямом, який дозволяє переходити від ручного вимірювання до цифрових, відтворюваних та масштабованих методів. Запропонована система може слугувати основою для подальшого розвитку: інтеграції стереоскопічного аналізу, тривимірної реконструкції, автоматичної класифікації типів порушень постави та розробки рекомендацій щодо корекції.

Таким чином, поставлена мета була досягнута повністю. Робота має як наукову, так і практичну цінність та демонструє ефективність поєднання сучасних інформаційних технологій для вирішення задачі аналізу постави людини.

Список використаних джерел

- [1] Activation functions: comparison of trends in practice and research for deep learning / C. Nwankpa та ін. URL: <https://arxiv.org/abs/1811.03378>.
- [2] Archana R., Jeevaraj P. S. E. Deep learning models for digital image processing: a review. *Artificial intelligence review*. 2024. Т. 57, № 1. С. 11. URL: <https://doi.org/10.1007/s10462-023-10631-z>.
- [3] A visual scan analysis protocol for postural assessment at school in young students / M. E. Alves та ін. *International journal of environmental research and public health*. 2020. Т. 17, № 8. С. 2915. URL: <https://doi.org/10.3390/ijerph17082915> (дата звернення: 27.11.2025).
- [4] Cascaded pyramid network for multi-person pose estimation / Y. Chen та ін. 2017. URL: <https://doi.org/10.48550/arXiv.1711.07319>.
- [5] Contributors to Wikimedia projects. Human musculoskeletal system - Wikipedia. *Wikipedia, the free encyclopedia*. URL: https://en.wikipedia.org/wiki/Human_musculoskeletal_system (дата звернення: 27.11.2025).
- [6] CS 230 - convolutional neural networks cheatsheet. *Stanford University*. URL: <https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks> (дата звернення: 27.11.2025).
- [7] Deep learning. MIT Press, 2017. 800 с.
- [8] Gradient-based learning applied to document recognition / Y. Lecun та ін. *Proceedings of the IEEE*. 1998. Т. 86, № 11. С. 2278–2324. URL: <https://doi.org/10.1109/5.726791> (дата звернення: 27.11.2025).
- [9] Human Pose Estimation using Keypoint RCNN in PyTorch. *LearnOpenCV – Learn OpenCV, PyTorch, Keras, Tensorflow with code, & tutorials*. URL: <https://learnopencv.com/human-pose-estimation-using-keypoint-rcnn-in-pytorch/> (дата звернення: 27.11.2025).
- [10] IBM. What is gradient descent? | IBM. *IBM*. URL: <https://www.ibm.com/think/topics/gradient-descent> (дата звернення: 27.11.2025).

- [11] Keypointrcnn_resnet50_fpn. *PyTorch Documentation*.
URL: https://docs.pytorch.org/vision/stable/models/generated/torchvision.models.detection.keypointrcnn_resnet50_fpn.html#torchvision.models.detection.keypointrcnn_resnet50_fpn (дата звернення: 27.11.2025).
- [12] Кнап Р. Human modelling and pose estimation overview.
URL: <https://arxiv.org/abs/2406.19290>.
- [13] LibreTexts. 6.1A: огляд опорно-рухового апарату. *LibreTexts - Ukrayinska*.
URL: [https://ukrayinska.libretexts.org/Медицина/Анатомія_і_фізіологія/Книга:_Анатомія_і_фізіологія_\(Boundless\)/6:_Скелетна_система/6.1:_Огляд_скелетної_системи/6.1A:_Огляд_опорно-рухового_апарату](https://ukrayinska.libretexts.org/Медицина/Анатомія_і_фізіологія/Книга:_Анатомія_і_фізіологія_(Boundless)/6:_Скелетна_система/6.1:_Огляд_скелетної_системи/6.1A:_Огляд_опорно-рухового_апарату) (дата звернення: 27.11.2025).
- [14] Mask R-CNN / К. Хе та ін. *IEEE transactions on pattern analysis and machine intelligence*. 2020. Т. 42, № 2. С. 386–397.
URL: <https://doi.org/10.1109/tpami.2018.2844175> (дата звернення: 27.11.2025).
- [15] Microsoft COCO: common objects in context / Т.-Y. Lin та ін.
URL: <https://arxiv.org/abs/1405.0312>.
- [16] Muscle types | SEER training.
URL: <https://training.seer.cancer.gov/anatomy/muscular/types.html> (дата звернення: 27.11.2025).
- [17] Musculoskeletal health. *World Health Organization (WHO)*.
URL: <https://www.who.int/news-room/fact-sheets/detail/musculoskeletal-conditions> (дата звернення: 27.11.2025).
- [18] PyTorch Contributors. Loss functions. *PyTorch Documentation*.
URL: <https://docs.pytorch.org/docs/stable/nn.html#loss-functions> (дата звернення: 27.11.2025).
- [19] Realtime multi-person 2D pose estimation using part affinity fields / Z. Cao та ін. *2017 IEEE conference on computer vision and pattern recognition (CVPR)*. 2017. С. 1302–1310. URL: <https://doi.org/10.1109/CVPR.2017.143>.
- [20] Reference values for sagittal clinical posture assessment in people aged 10 to 69 years / O. Ludwig та ін. *International journal of environmental research and public*

- health*. 2023. Т. 20, № 5. С. 4131. URL: <https://doi.org/10.3390/ijerph20054131> (дата звернення: 27.11.2025).
- [21] Russell S. J., Norvig P. Artificial intelligence: a modern approach. Pearson Education, Limited, 2021.
- [22] Saxena A. An introduction to convolutional neural networks. *International journal for research in applied science and engineering technology*. 2022. Т. 10, № 12. С. 943–947. URL: <https://doi.org/10.22214/ijraset.2022.47789> (дата звернення: 27.11.2025).
- [23] Scoliosis management through apps and software tools / L. Bottino та ін. *International journal of environmental research and public health*. 2023. Т. 20, № 8. С. 5520. URL: <https://doi.org/10.3390/ijerph20085520> (дата звернення: 27.11.2025).
- [24] Seong B., Lee H. J., Lee R. Keypoint-wise adaptive loss for whole-body human pose estimation. 2023.
- [25] Skeletal muscle atrophy: from mechanisms to treatments / L. Yin та ін. *Pharmacological research*. 2021. Т. 172. С. 105807. URL: <https://doi.org/10.1016/j.phrs.2021.105807> (дата звернення: 27.11.2025).
- [26] Stunning-vision. Summary of YOLO11 Architecture · Issue #17102 · ultralytics/ultralytics. *GitHub*. URL: <https://github.com/ultralytics/ultralytics/issues/17102> (дата звернення: 27.11.2025).
- [27] Supervised vs unsupervised vs reinforcement learning - geeksforgeeks. *GeeksforGeeks*. URL: <https://www.geeksforgeeks.org/machine-learning/supervised-vs-reinforcement-vs-unsupervised/> (дата звернення: 27.11.2025).
- [28] Tapkir A. A comprehensive overview of gradient descent and its optimization algorithms. *Iarjset*. 2023. Т. 10, № 11. URL: <https://doi.org/10.17148/iarjset.2023.101106> (дата звернення: 27.11.2025).
- [29] Tkach O. F., Konoplitskyi V. S., Korobko Y. Y. Postural disorders in children. Modern views on classification, diagnosis, treatment and rehabilitation. *Paediatric surgery. ukraine*. 2024. № 4(85). С. 111–122. URL: [https://doi.org/10.15574/ps.2024.4\(85\).111122](https://doi.org/10.15574/ps.2024.4(85).111122) (дата звернення: 27.11.2025).

- [30] Ultralytics. Data Augmentation using Ultralytics YOLO. *Ultralytics YOLO Docs*. URL: <https://docs.ultralytics.com/guides/yolo-data-augmentation/> (дата звернення: 27.11.2025).
- [31] Vialatte J.-C., Gripon V., Coppin G. Learning local receptive fields and their weight sharing scheme on graphs. *2017 IEEE global conference on signal and information processing (globalsip)*. 2017. С. 623–627. URL: <https://doi.org/10.1109/GlobalSIP.2017.8309034>.
- [32] YOLO advances to its genesis: a decadal and comprehensive review of the You Only Look Once (YOLO) series / R. Sapkota та ін. *Artificial intelligence review*. 2025. Т. 58, № 9. URL: <https://doi.org/10.1007/s10462-025-11253-3> (дата звернення: 27.11.2025).