

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

Факультет автоматизації та інформаційних технологій

Кафедра інформаційних технологій

**ПОЯСНОВАЛЬНА ЗАПИСКА
ДО КВАЛІФІКАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВР**

на тему: “Система ШІ для діагностики захворювань опорно-рухового апарату”

Йовенко Дмитро Сергійович

Київ 2024 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

Факультет автоматизації та інформаційних технологій

Кафедра інформаційних технологій

ЗАТВЕРДЖУЮ
Завідувач кафедри ІТ

„____” _____ 2024 року

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

на тему: “Система ШІ для діагностики захворювань опорно-рухового апарату”

Виконав: Йовенко Дмитро Сергійович
Спеціальність: 122 “Комп’ютерні науки”
Інформаційні управляючі системи і технології
Групи: КН 20-2
Керівник: Поплавський О. А.
к.т.н., доцент кафедри інформаційних технологій.
Ідентичність підтверджую

Київ 2024 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

Факультет: Автоматизації та інформаційних технологій

Випускова кафедра: Інформаційних технологій

Освітній ступінь: Бакалавр

Спеціальність: 122 “Комп’ютерній науки”

Освітня програма: Інформаційні управляючі системи і технології

ЗАТВЕРДЖУЮ
Завідувач кафедри ІТ

_____” _____ 2024 року

З А В Д А Н Н Я

**ДО ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ НА
ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВР**

Йовенко Дмитро Сергійович

1. Тема роботи: “Система ШІ для діагностики захворювань опорно-рухового апарату”

затверджена наказом ректора КНУБА № __ від «__» _____ 202__ року

2. Керівник роботи: Поплавський Олександр Анатолійович к.т.н., доцент кафедри інформаційних технологій

3. Строк подання Здобувачем роботи до захисту _____

4. Зміст пояснювальної записки за розділами:

P.1 Аналіз предметної області та постановка задачі

P.2 Математичне, алгоритмічне та інформаційне забезпечення

P.3 Проектні рішення. Розробка програмного забезпечення

P.4 Ергономіка інформаційних технологій

5. Інформаційні слайди:

C.1 Титульний слайд

C.2 Предметна область

C.3 Об’єкт дослідження

C.4 Об’єкт дослідження ч.2

C.5 Предмет дослідження

C.6 Предмет дослідження ч.2

С.7 Збір даних

С.8 Тестовий приклад

С.9 Тестовий знімок

С.10 Ергономіка в ІТ

С.11 Висновки

6. Календарний план виконання роботи:

Види робіт та їх зміст	Дата виконання
Розділ 1	Лютий 2024
Розділ 2	Березень 2024
Розділ 3	Квітень 2024
Розділ 4	Травень 2024
Остаточне оформлення роботи	Травень 2024
Направлення роботи для перевірки на плагіат	Червень 2024
Попередній захист роботи на випусковій кафедрі	Червень 2024
Направлення роботи на рецензування	Червень 2024

7. Консультанти розділів атестаційної випускної роботи

Розділ	Прізвище, ініціали та посада консультанта	Перевірив	
		дата	підпис
Розділ 1	Поплавський О.А., доцент	02.02.2024	
Розділ 2	Поплавський О.А., доцент	13.03.2024	
Розділ 3	Поплавський О.А., доцент	24.04.2024	
Розділ 4	Поплавський О.А., доцент	25.05.2024	

8. Дата видачі завдання 21.12.2023

Зав. кафедри _____
 (підпис) (прізвище та ініціали)

Керівник _____
 (підпис) (прізвище та ініціали)

Здобувач _____
 (підпис) (прізвище та ініціали)

АНОТАЦІЯ

Йовенко Д. С. Розробка системи штучного інтелекту для діагностування захворювань опорно-рухового апарату.

Атестаційна випускна робота бакалавра за спеціальністю: 122 «Комп'ютерні науки» – Київський національний університет будівництва та архітектури. – Київ, 2024.

Робота присвячена розробці системи штучного інтелекту для автоматизації діагностики захворювань опорно-рухового апарату. Основним інструментом реалізації системи є модель YOLOv8, яка використовується для виявлення ключових точок на тілі людини. Розроблена система має потенціал для впровадження у медичні заклади, підвищуючи ефективність та доступність медичної допомоги.

Ключові слова: штучний інтелект, машинне навчання, YOLOv8, медичні зображення, опорно-руховий апарат, ключові точки, нейронні мережі.

SUMMARY

Yovenko D. S. Development of an Artificial Intelligence System for Diagnosing Musculoskeletal Disorders.

Bachelor's Thesis in Specialty: 122 "Computer Science" – Kyiv National University of Construction and Architecture. – Kyiv, 2024.

This work is dedicated to the development of an artificial intelligence system for automating the diagnosis of musculoskeletal disorders. The primary tool for implementing the system is the YOLOv8 model, which is used to detect key points on the human body. The developed system has the potential to be integrated into medical institutions, enhancing the efficiency and accessibility of medical care.

Keywords: artificial intelligence, machine learning, YOLOv8, medical images, musculoskeletal disorders, key points, neural networks.

ЗМІСТ

ВСТУП	8
1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ	10
1.1 Опис предметної області	10
1.2 Аналіз об'єкта дослідження	15
1.3 Опис предмету дослідження	22
1.4 Аналіз актуальності	26
1.5 Стан вже існуючих рішень	28
1.6 Визначання цілей роботи та постановка задачі	29
2. МАТЕМАТИЧНЕ, АЛГОРИТМІЧНЕ ТА ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ	31
2.1 Математичне забезпечення системи	31
2.1.1 Метод градієнтного спуску	31
2.1.2 Функції активації нейронів	35
2.2 Алгоритмічне забезпечення системи	38
2.2.1 Згортова нейронна мережа	38
2.2.2 Метод зворотнього поширення помилки	44
2.3 Інформаційне забезпечення	48
2.3.1 Задача виявлення ключових точок	48
2.3.2 Методи знаходження ключових точок	52
2.4 Проектування системи	57
3. ПРОЕКТНІ РІШЕННЯ. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	61
3.1 Проектні рішення з програмного забезпечення	61
3.2 Збір даних для навчання	66
3.3 Опис класів	71
3.4 Тестовий приклад	73
4. ЕРГОНОМІКА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ	78
4.1 Визначення ергономіки в контексті ІТ	78
4.2 Основні принципи ергономіки у програмному забезпеченні	81
4.3 Підходи до проектування програмного забезпечення з урахуванням потреб користувачів медичних систем	82
4.4 Вимоги до інтерфейсу та проектування його компонентів	84
4.4.1 Аналіз вимог до проектування	84

	7
4.4.2 Аналіз інтерфейсу існуючої медичної системи FreeMED	87
4.4.3 Аналіз вимог від користувачів	90
4.5 Проектування інтерфейсу програмного забезпечення	92
ВИСНОВКИ	96
Список використаних джерел	97

ВСТУП

Розвиток інформаційних технологій та систем штучного інтелекту (ШІ) відкриває нові можливості для діагностики захворювань, зокрема захворювань опорно-рухового апарату. Сучасні методи медичної діагностики потребують удосконалення для забезпечення швидшого та точнішого виявлення патологій, що є надзвичайно важливим для покращення якості життя пацієнтів та ефективності лікування.

Мета даної роботи полягає у розробці системи штучного інтелекту для діагностування захворювань опорно-рухового апарату на основі оцінки постави людини. Необхідність розробки такої системи обґрунтовується потребою у швидкій, точній та доступній діагностиці, що зможе суттєво покращити виявлення захворювань на ранніх стадіях та забезпечити своєчасне лікування. Існуючі методи діагностики захворювань опорно-рухового апарату часто є трудомісткими, вимагають значних витрат часу та залучення висококваліфікованих спеціалістів та обладнання. Використання систем ШІ для автоматизації та підвищення точності діагностики є перспективним напрямком, що дозволяє вирішити ці проблеми. ШІ-системи здатні аналізувати великі обсяги медичних даних, виявляти приховані закономірності та надавати точні прогнози, що є важливим для медичної галузі.

Основні проектні рішення цієї роботи полягають у застосуванні сучасних методів машинного навчання, зокрема згорткових нейронних мереж, для обробки медичних зображень та даних пацієнтів. Використання таких методів дозволить досягти високої точності діагностики, що підтверджується численними дослідженнями та успішними впровадженнями в інших медичних системах. Результати роботи можуть бути застосовані у різних медичних закладах, включаючи лікарні, клініки та діагностичні центри. Система ШІ для діагностування захворювань опорно-рухового апарату може бути інтегрована у існуючі медичні інформаційні системи для поліпшення точності та швидкості діагностики, що сприятиме загальному підвищенню ефективності медичного обслуговування.

Актуальність роботи зумовлена необхідністю вдосконалення методів діагностики захворювань опорно-рухового апарату, особливо в умовах України, де доступ до якісної медичної допомоги є обмеженим у деяких регіонах. Впровадження систем ШІ у медицину є важливим кроком для підвищення рівня медичних послуг та забезпечення їх доступності.

Об'єктом дослідження є процес діагностики захворювань опорно-рухового апарату, а предметом дослідження – методи та алгоритми машинного навчання, що використовуються для аналізу медичних даних та зображень. Основними методами дослідження є математичне моделювання, алгоритмічний аналіз та машинне навчання. Зокрема, використання згорткових нейронних мереж.

Основні розділи роботи включають аналіз предметної області та постановку задачі, математичне, алгоритмічне та інформаційне забезпечення, проектні рішення та розробку програмного забезпечення, а також ергономіку інформаційних технологій.

Таким чином, дане дослідження має на меті створення системи ШІ, яка значно пришвидшить процес діагностики захворювань опорно-рухового апарату, забезпечивши швидкість, точність та доступність медичної допомоги.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Опис предметної області

Ця кваліфікаційна робота фокусується на розробці автоматизованої системи для діагностики захворювань опорно-рухового апарату. Метою цього проекту є спрощення та прискорення процесу виявлення патологій, забезпечуючи фахівців у галузі засобами для ефективного ведення обстежень. Нова система розрахована на значне зменшення часу, витраченого на проведення обстежень, перетворюючи його з хвилин на секунди. Це інноваційне рішення має на меті полегшити роботу медичних спеціалістів та зробити процес діагностики більш швидким та ефективним.

Опорно-руховий апарат складається зі скелета та м'язів. Скелет це пасивна частина, а м'язи – активна. Кістки скелета є важелями, які приводяться в рух м'язами. Внаслідок цього частини тіла змінюють своє положення стосовно одна одної, переміщують тіло в просторі й виконують опорну функцію. На рисунку 1.1 зображено опорно-рухову систему людини[1].

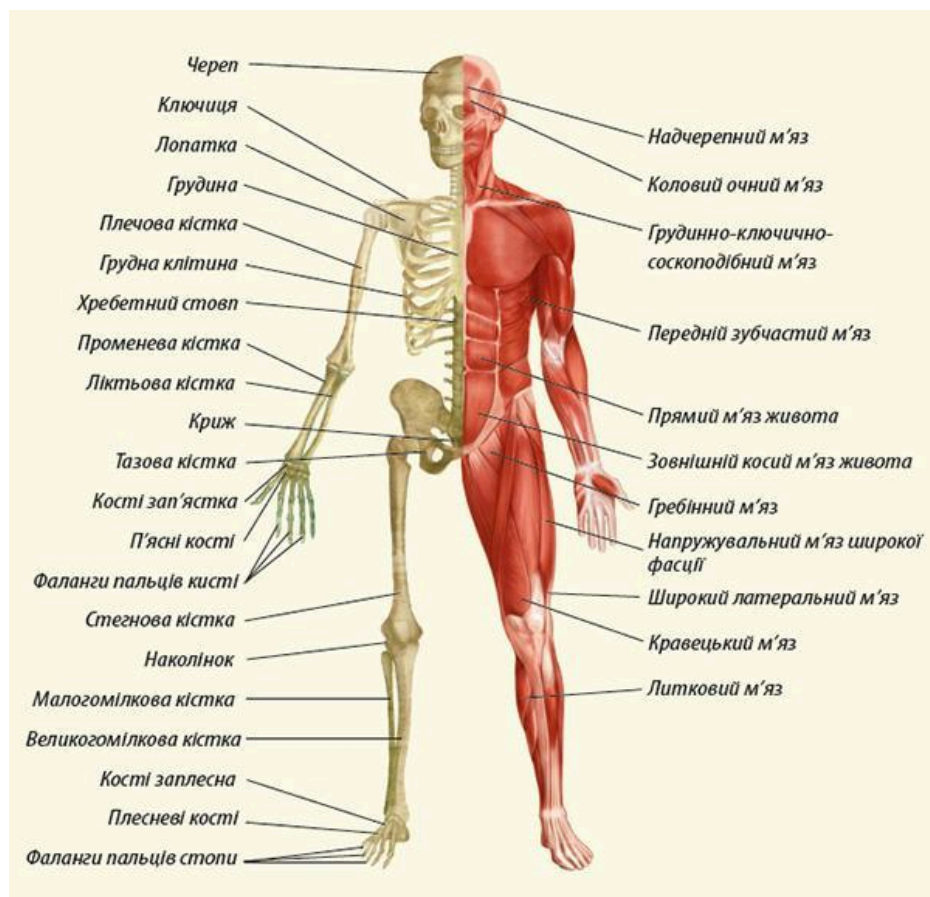


Рисунок 1.1 – Опорно рухова система

Людина народжується, розвивається й існує в умовах земного тяжіння – гравітації, і кожний рух пов'язаний з подоланням сил тяжіння, тому опорно-руховий апарат виконує ще й антигравітаційну функцію.

Обидві частини опорно-рухового апарату і розвиваються з одного й того самого зародкового листка – мезодерми, і тісно пов'язані між собою анатомічно й функціонально.

Опорно-руховий апарат має наступні функції:

- **Опорна.** Скелет формує кістковий каркас тіла, слугує місцем прикріплення м'язів. Скелетні м'язи фіксують тіло в певному положенні.
- **Рухова.** Здійснюється за допомогою з'єднаних суглобами кісток і скорочення прикріплених до них м'язів.
- **Захисна.** Знаходить свій вияв в утворенні кістками порожнин, у яких розташовані життєво важливі органи. Так, грудна клітка захищає від механічних впливів серце та легені; нижній відділ хребта, кістки таза й м'язи живота захищають внутрішні органи черевної порожнини – шлунок, кишечник, нирки; головний мозок захищений кістками черепа, а спинний мозок-хребта.
- **Метаболічна функція.** Бере участь в обміні мінеральних речовин.
- **Кровотворна.** Кожний рух здійснюється за допомогою кількох м'язів різнонаправленої дії. Крім того, скелетні м'язи є своєрідними органами чуття, тому що в м'язових волокнах і сухожиллях наявні нервові закінчення – рецептори, які посилають імпульси в центральну нервову систему.

Скелет – це сукупність кісток, хрящів, зв'язок та інших щільних утворів, які разом утворюють механічну опору для тіла людини. Крім того, деякі кістки виконують захисну функцію, утворюючи порожнини, в яких розміщуються життєво

важливі органи. Наприклад, череп містить головний мозок, грудна клітка – легені та серце, а порожнина тазу – органи розмноження та частину органів виділення.

Скелет також є пасивною частиною рухового апарату, оскільки кістки, з'єднуючись між собою, утворюють важелі, які рухаються за допомогою м'язів, що до них прикріплені. У людини налічується понад 200 кісток, кожна з яких є живим організмом, складаючись з кількох типів тканин (кісткової, хрящової), а також кісткового мозку, кровоносних судин та нервів.

Кістки поділяються за формою на довгі, короткі, плоскі та мішані. Довгі кістки, які включають більшість кісток кінцівок, мають трубчасту будову з порожниною всередині, заповненою жовтим кістковим мозком. Короткі кістки мають приблизно однакові розміри у всіх напрямках. Плоскі кістки мають значну довжину і ширину при малій товщині. Мішані кістки містять елементи коротких та плоских кісток.

М'язи здійснюють скорочення, щоб переміщувати кістки, які з'єднані в суглобі. Скелетні м'язи кріпляться до кісток і розташовані протилежними групами навколо суглобів. Ці м'язи іннервуються, тобто нерви проводять електричні сигнали від центральної нервової системи, що змушують м'язи скорочуватися.

В організмі існує три типи м'язової тканини(рис1.2):

- Скелетна;
- Гладка;
- Серцева;

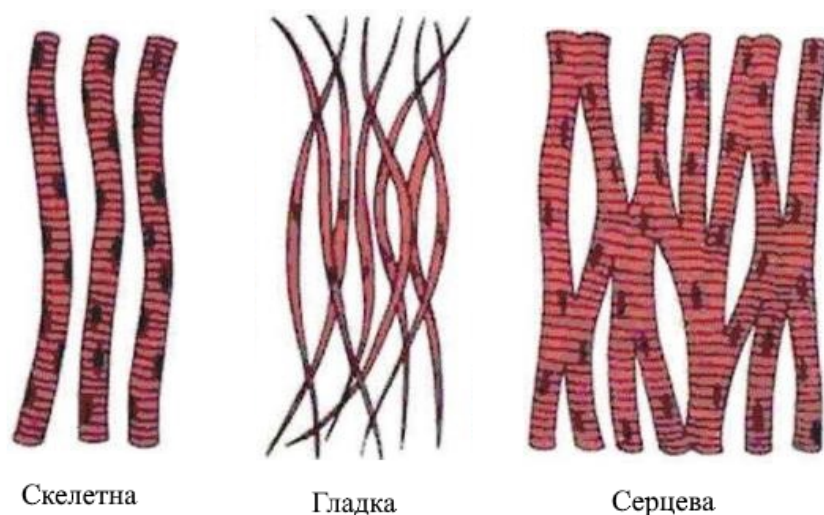


Рисунок 1.2 – Типи м'язової тканини

Скелетна і гладка мускулатура вважаються частиною опорно-рухового апарату. Скелетні м'язи відповідають за рух тіла, вони дозволяють виконувати різноманітні рухи, від ходьби до складних спортивних вправ[2]. М'язову систему людини зображено на рисунку 1.3.

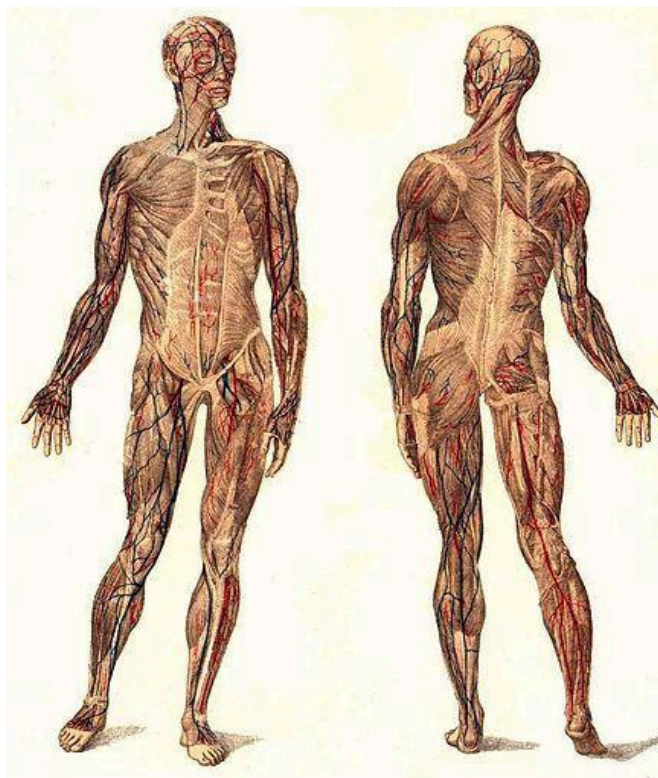


Рисунок 1.3 – М'язова система

Приклади гладкої мускулатури включають ті, що містяться в стінках кишечника та судин, і вони забезпечують роботу органів травлення та кровообігу.

Серцеві і гладкі м'язи характеризуються мимовільним рухом, тобто вони функціонують без свідомого контролю. Серцеві м'язи знаходяться в серці та забезпечують його ритмічну роботу, щоб перекачувати кров по всьому організму.

Зв'язок між м'язовою системою і скелетом глибокий і важливий для функціонування організму. Скелет не лише надає опору і захист внутрішнім органам, а й є базою для м'язів, які забезпечують рух і дозволяють виконувати різноманітні фізичні дії.

Коли м'язи скорочуються, вони тягнуться або стискаються, тягнучи за прикріплені до них кістки. Цей процес створює рух у суглобах. Скелет діє як механічна підтримка, що дозволяє м'язам ефективно тягнутися і стискатися, створюючи потрібний рух.

Крім того, скелет допомагає у збереженні ендокринної системи. Кістки також служать резервуаром для кальцію, необхідного для багатьох функцій організму, включаючи м'язову роботу і передачу нервових сигналів.

Узгоджена робота м'язів і скелета також важлива для підтримки правильної постави і стабільності тіла. Наприклад, м'язи спини і черевного пресу спільно зі скелетом забезпечують правильну підтримку хребта і уникнення спинного стовбура. Це важливо для запобігання травм і підтримки загального здоров'я.

Таким чином, взаємодія між м'язовою системою і скелетом визначає не лише рухові можливості людини, а й її загальний стан здоров'я і функціонування організму в цілому.

Рухова активність призводить до посилення функцій опорно-рухового апарату, активізації обміну речовин, підвищення стійкості організму до хвороб. Значення рухової активності в життєдіяльності людей має важливі роль. Наприклад, розвиток дрібної моторики рук у дитини сприяє функції зональному розвитку структур мозку.

Лише фізично активна людина є здоровою. Малорухливий спосіб життя (гіподинамія) може сприяти розвитку низки захворювань, порушений роботи опорно-рухового апарату, серцево-судинної системи, кровообігу, дихання, травлення.

1.2 Аналіз об'єкта дослідження

Один з найрозповсюдженіших типів хронічних неінфекційних захворювань – це ураження опорно-рухового апарату. За даними ВООЗ, близько 1,71 мільярда людей по всьому світу стикаються з цими розладами. Хвороби опорно-рухового апарату займають провідне місце серед причин інвалідності у світі. Понад 150 видів захворювань призводять до ураження опорно-рухового апарату, що зазвичай виражається болем, обмеженням рухливості та погіршенням функціональних можливостей, ускладнюючи трудову активність.

Пряма спина, відправлені плечі, розвинута грудна клітина та піднята голова – ці ознаки показують, що постава правильна. При правильній поставі внутрішні органи розташовані так, що створюються найкращі умови для їх ефективної роботи.

Хребет людини – це своєрідна колонка, що не лише утримує вагу всього тіла, але й захищає спинний мозок. Він складається з 33-34 хребців, розділених дисками, що виготовлені з тканини з волокнистого субсиду та спинного мозку[3].

Хребет складається із 5 відділів:

- Шийний відділ, який в свою чергу складається із 7 хребців;
- Грудний відділ, який складається із 12 хребців;
- Поперековий відділ, який складається із 5 хребців;
- Крижовий відділ, який складається із 5 хребців;
- Куприковий відділ, який складається із 4-5 хребців;

Ця складна і гнучка система(рис 1.4) має три природні вигини: шийний, грудний і поперековий. Вони сприяють органічному розподілу ваги та навантаження. Наприклад, S-подібна анатомічна форма хребта запобігає падінню людини вперед або назад.



Рисунко 1.4 – Хребет людини

Невірна постава може призвести до викривлення хребта, що відбувається рано або пізно і може спричинити більш серйозні пошкодження. Це може призвести до стиснення нервів, які виходять з спинного мозку, що розташований всередині каналу хребта. Людина може почати відчувати біль у спині, шиї та голові, а також розвивати хворобу остеохондроз.

Порушення постави може призвести до серйозних захворювань опорно-рухового апарату, від яких страждає велика кількість людей по всьому світу. Неправильна постава може стати причиною різноманітних хвороб і ускладнень, які впливають на якість життя та загальний стан здоров'я.

Однією з найпоширеніших проблем, пов'язаних з порушенням постави, є біль та дискомфорт у спині. Невірне розташування хребта може призвести до розвитку сколіозу, кифозу, лордозу (рис 1.5) та інших дегенеративних захворювань хребта, які перелічені далі[4].

Постійний напружений стан м'язів, який виникає при неправильній поставі, може призвести до їхньої слабкості та спазмів.

Крім того, порушення постави може призвести до деформації суглобів та зменшення їхньої мобільності. Наприклад, неправильна постава плечей може призвести до розвитку синдрому "плеча замикача", коли рухи в руці стають обмеженими та болючими.

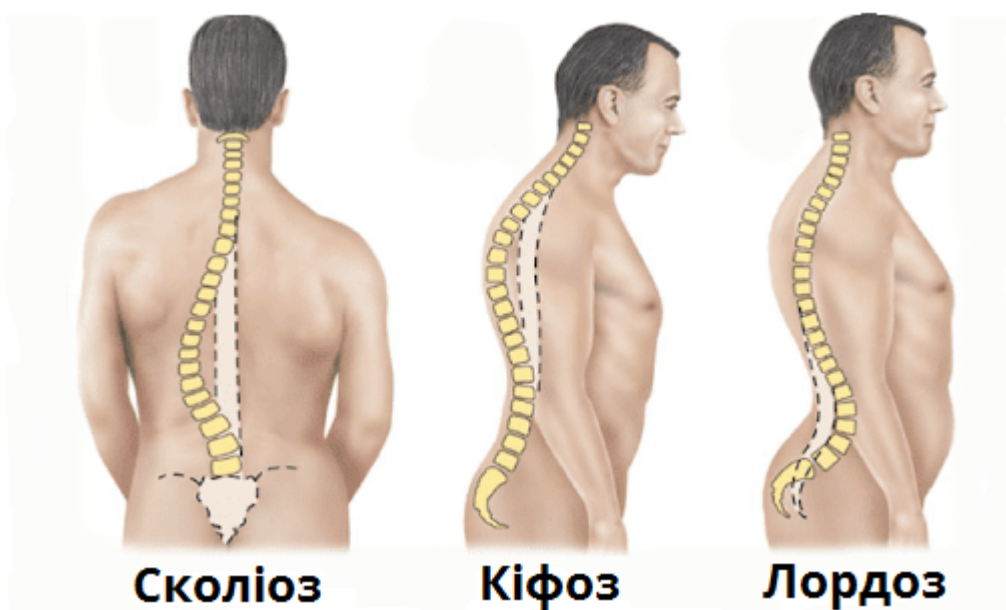


Рисунок 1.5 – Типи викривлень хребта

Ускладнення порушення постави також може впливати на роботу внутрішніх органів, зокрема легенів і серця. Наприклад, стиснення грудної клітини при поганий поставі може призвести до утрудненого дихання та погіршення кровообігу.

На рисунку 1.6 зображено ілюстрацію людини з порушеною та правильною поставами. Зверху – правильна постава, знизу – порушена.

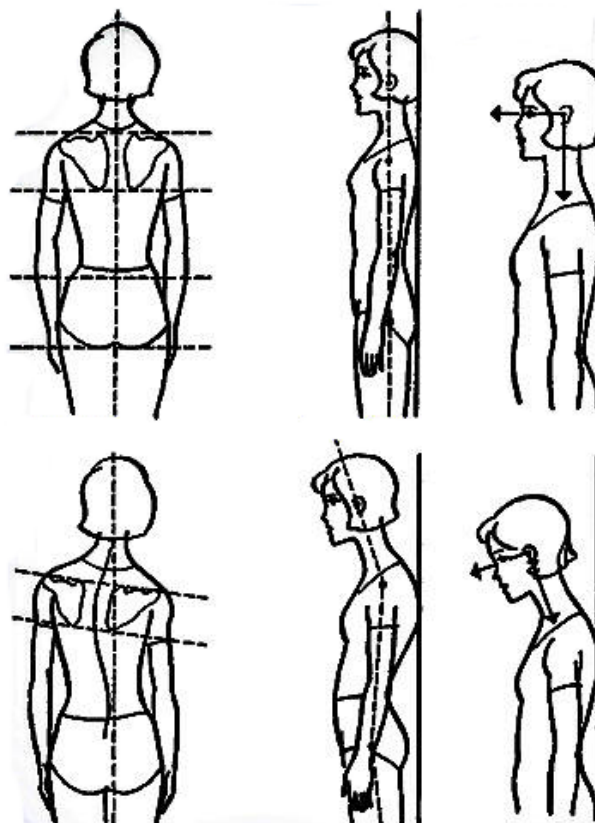


Рисунок 1.6 – Ілюстрація здорової та порушеної постави

Причини захворювання опорно-рухового апарату різноманітні, але всі вони мають однаковий результат – призводять до порушення в роботі органів, які відносяться до опорно-рухової системи людини.

У список причини включені:

- відсутність фізичних навантажень, малорухливий спосіб життя;
- отримані травми;
- спадкові захворювання ОРА;
- зловживання алкогольними напоями;
- зайва вага;
- надмірні фізичні навантаження (характерно для спортсменів);

- недостатня кількість вітамінних комплексів в організмі;
- погана екологія.

Опорно-рухові захворювання суглобів ускладнюють виконання певних функцій. Виходячи з особливостей їх прояву, всі захворювання опорно-рухового апарату діляться на дві основні групи: первинні і вторинні. Перша категорія включає в себе всі порушення, які можуть вважатися самостійними, а до другої-порушення в будові опорно-рухового апарату, вони відбуваються в результаті перенесених хвороб[5].

Найпоширеніші хвороби опорно-рухового апарату та їх короткий опис представлені у таблиці 1.1.

Таблиця 1.1

Остеохондроз	Хворі з дегенеративно-дистрофічними захворюваннями хребта. Проблеми з хребтом, в процесі яких відбуваються дистрофічні зміни тканин – кісткової і хрящової. Ці процеси відбуваються в міжхребцевої області, де знаходяться диски.
Артрит	Запалення суглобів.
Артрози	Вторинні порушення опорно-рухової системи, які досить часто стають наслідками артриту. В такому випадку у пацієнта відзначають запалення суглобової сумки, зрощення суглобів або обмеження їх рухливості.

Ревматоїдний артрит, спондилоартрит, недиференційованої захворювання сполучної тканини	Аутоімунні захворювання суглобової і сполучної тканини
Подагра	Слідство порушення метаболічного обміну (метаболічний синдром), результатом якого є захворювання ОДА

Діагностика опорно-рухового апарату є ключовим етапом у виявленні проблем і вчасному лікуванні різних захворювань. Існує велика методів, які використовуються для оцінки стану опорно-рухового апарату та виявлення можливих патологій.

Далі наведено основні методи діагностики:

- **Клінічний огляд:** Лікар проводить зовнішній огляд пацієнта, оцінює поставу, рухливість суглобів та м'язовий тонус, шукає ознаки травм або деформацій.
- **Рентгенографія:** Цей метод дозволяє отримати зображення кісток та суглобів, що допомагає виявити остеоартроз, переломи, деформації та інші патології.
- **Магнітно-резонансна томографія (МРТ):** МРТ використовується для отримання детальних зображень м'язів, суглобів та м'язевих тканин, що дозволяє виявити травми, запалення та інші патології.
- **Комп'ютерна томографія (КТ):** Цей метод дозволяє отримати тривимірні зображення кісток та суглобів для детального вивчення їхньої структури та стану.

- **Ультразвукове дослідження:** Ультразвукова діагностика може використовуватися для вивчення м'язів, сухожилків та суглобів, що дозволяє виявити запалення, травми та інші патології.
- **Лабораторні тести:** Деякі патології опорно-рухового апарату можуть супроводжуватися змінами у крові, такими як підвищення рівня запальних маркерів або показників біохімічного аналізу, тому лабораторні тести також можуть бути корисними для діагностики.

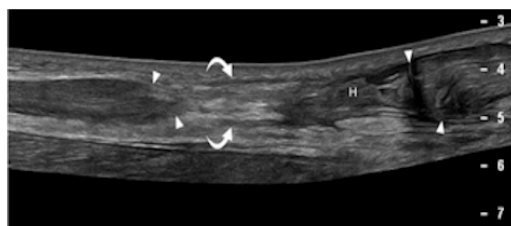
На рисунку 1.7 зображено приклади результатів діагностики опорно рухового апарату вище переразованими методами. а) – КТ знімок колінного суглобу, б) – Рентгенівський знімок стопи, в) – УЗД ахілесового сухожилля, г) – МРТ знімок хребта людини



а)



б)



в)



г)

Рисунок 1.7 – Результати досліджень різними методами

Посилаючись на отримані результати, лікар визначає стан організму пацієнта і ставить діагноз, після чого призначає певний курс лікування. Загальна рекомендація для всіх пацієнтів із захворюваннями опорно-рухового апарату берегти зв'язки, суглоби та інші пошкоджені області в спокої, не виконувати складні фізичні навантаження.

Якщо звернутися до статистичних даних то після 40 років життя рентгенологічні ознаки запальних і дистрофічних відхилень у хребті і суглобах присутні практично у 100% обстежених людей.

Така статистика настановлює на логічний висновок, що заходи, пов'язані з профілактикою захворювань опорно-рухової системи, слід починати на більш ранньому етапі – з дитинства. Дегенеративні та запальні патології впливають не тільки на суглоби, але так само і на стан організму в цілому.

1.3 Опис предмету дослідження

Штучний інтелект – це галузь комп'ютерних наук, що займається створенням систем, програм або алгоритмів, які наділені здатністю до розуміння, самонавчання, адаптації та вирішення завдань, які зазвичай вимагають інтелектуального спроможності людини. Це включає в себе розробку методів та технологій, таких як машинне навчання, обробка природної мови, комп'ютерне зору, обчислення на основі знань та інші.

Штучні нейронні мережі - це математичні моделі, які намагаються імітувати роботу нейронних мереж у людському мозку. Нейромережі складаються з великої кількості взаємопов'язаних штучних нейронів, які працюють разом для виконання певних завдань. Нейромережі можуть самонавчатися, тобто вони можуть адаптуватися до нових даних та удосконалювати свої здібності з часом.

Нейронні мережі в наш час мають величезну актуальність і широке застосування у багатьох сферах. Ось кілька причин, чому вони так важливі:

- **Обробка великих обсягів даних.** У сучасному світі генерується величезна кількість даних щодня. Нейронні мережі дозволяють ефективно аналізувати, класифікувати та використовувати ці дані для різних цілей, таких як прогнозування, рекомендації, виявлення аномалій тощо.
- **Зображення та відео.** Нейронні мережі показали вражаючі результати в обробці зображень та відео. Вони можуть використовуватися для розпізнавання об'єктів, визначення емоцій на обличчі, виявлення аномалій тощо. Це має велике значення для медицини, безпеки, аналізу зображень та інших галузей.
- **Мовне розпізнавання.** Нейронні мережі здатні до ефективного розпізнавання та розуміння мови. Вони використовуються в системах автоматичного перекладу, розпізнаванні мовлення, голосових помічників тощо.

- **Автономні системи.** В області автономних систем, таких як автомобілі самостійного водіння, дрони та роботи, нейронні мережі використовуються для розпізнавання оточуючого середовища, прийняття рішень та керування.
- **Медицина.** У медицині нейронні мережі використовуються для діагностики захворювань, аналізу медичних зображень, прогнозування вірогідності захворювань та реакції на лікування.

Нейронні мережі складаються з різних компонентів, які допомагають їм виконувати різноманітні завдання. Основні складові структури нейронної мережі включають:

- **Нейрони** (або вузли): Нейрони є основними обчислювальними одиницями нейронної мережі. Кожен нейрон приймає вхідні сигнали, обробляє їх і генерує вихідний сигнал. У зв'язку з цим, нейрони часто називають "вузлами" або "узлами обчислення".
- **Зв'язки** (або ваги): Зв'язки встановлюють зв'язок між нейронами. Кожен зв'язок має асоційований параметр, вагу, яка вказує на важливість вхідного сигналу для відповідного нейрона. Ваги налаштовуються під час процесу навчання нейронної мережі.
- **Функція активації:** Функція активації приймає зважену суму вхідних сигналів і визначає, чи буде активований нейрон. Ця функція може бути лінійною або нелінійною і дозволяє нейронній мережі моделювати складні залежності у даних.
- **Шари:** Нейронні мережі зазвичай організовані у вигляді шарів. Вхідний шар отримує вхідні дані, вихідний шар генерує вихідні результати, а проміжні шари, відомі як приховані шари, виконують обчислення між вхідним і вихідним шарами.

На рисунку 1.8 зображено структурну схему повнозв'язної штучної нейронної мережі.

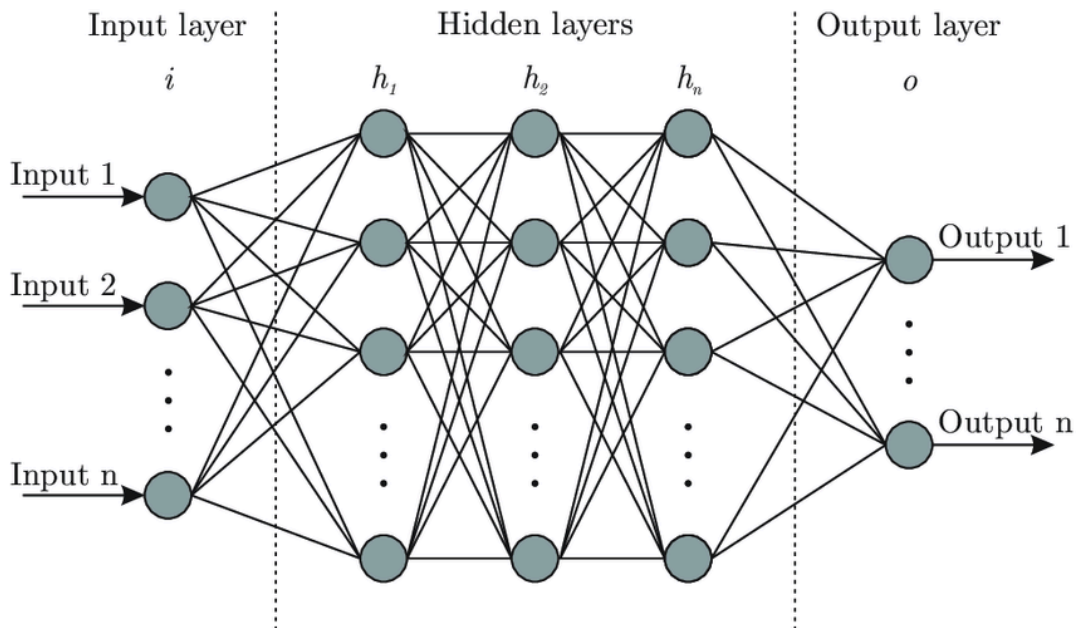


Рисунок 1.8 – Структурна схема штучної нейронної мережі

Невід'ємною частиною нейронної мережі є процес її навчання. Навчання нейронної мережі – це процес, під час якого модель здатна самостійно вчитися розпізнавати шаблони та здійснювати передбачення на основі вхідних даних.

Машинне навчання – важлива галузь штучного інтелекту, що досліджує розробку алгоритмів навчання різних типів штучного інтелекту.

Методи машинного навчання поділяються на два основних типи: навчання за прецедентами, відоме також як індуктивне навчання, і дедуктивне навчання. Оскільки останнє включається до сфери експертних систем, терміни "машинне навчання" і "навчання за прецедентами" можуть розглядатися як взаємозамінні. Метод навчання за прецедентами нині дуже популярний, тоді як експертні системи переживають занепад. Суперечності між базами знань, які використовуються в них, і реляційною моделлю даних ускладнюють ефективне використання промислових СУБД для заповнення баз знань експертних систем[6].

Навчання за прецедентами поділяється на три основні типи: контрольоване навчання, неконтрольоване навчання і навчання з підкріпленням.

Контрольоване навчання використовується у випадках, коли доступні великі обсяги маркованих даних, наприклад, тисячі фотографій домашніх тварин з

відповідними мітками. Мета полягає у створенні алгоритму, який здатний розпізнати на нових фотографіях, котрі раніше не були побачені, зображення кішок або собак. У цьому випадку людина, яка виставляє мітки, виступає в ролі "учителя". Машина вчиться розрізняти категорії на основі задалегідь обраних ознак. Знайдений алгоритм може бути легко адаптований для інших завдань, наприклад, розпізнавання курей і качок або обробки результатів комп'ютерної томографії.

Неконтрольоване навчання використовується в ситуаціях, коли доступні дані без міток, наприклад, зображення без підписів або аудіозаписи без коментарів. У такому випадку завдання машини полягає у виявленні зв'язків між даними, виявленні закономірностей, класифікації даних або структуруванні їх. Неконтрольоване навчання використовується, наприклад, у рекомендаційних системах, де на основі аналізу попередніх покупок клієнтам пропонуються товари, які їх можуть зацікавити.

Навчання з підкріпленням є варіантом контрольованого навчання, де "вчителем" є навколишнє середовище. Машина, або "агент", не має попередніх знань про середовище, але може взаємодіяти з ним. Середовище реагує на дії агента, надаючи йому дані, на основі яких він може вчитися. Навчання з підкріпленням використовується для розв'язання складних завдань, таких як навігація роботів, логістика та графічні ігри.

На рисунку 1.9 схематично зображено порівняння контрольованого (Supervised) і не контрольованого (Unsupervised) навчання.

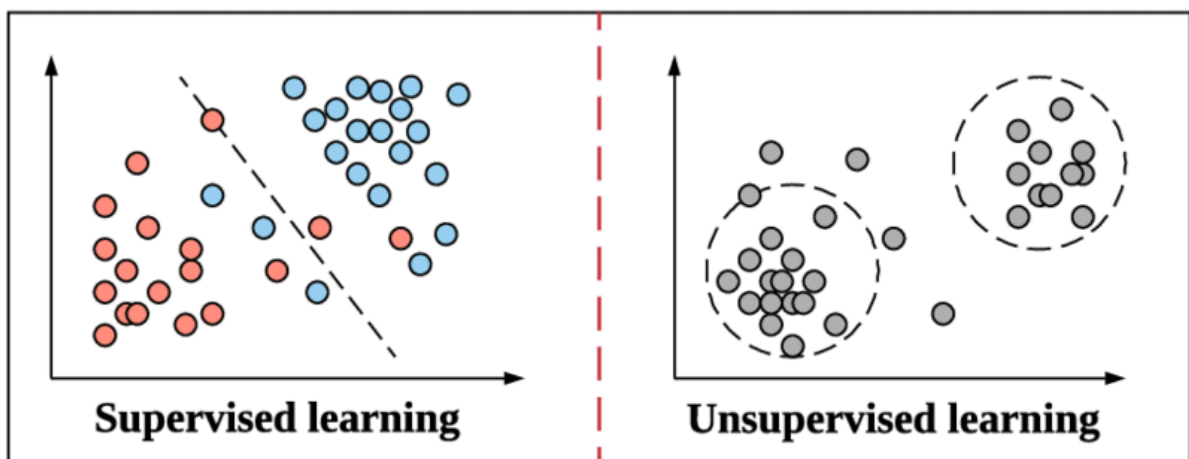


Рисунок 1.9 – Порівняння контрольованого та неконтрольованого навчання

1.4 Аналіз актуальності

У сучасному світі стикаємося з глибокою проблемою доступності якісної та доступної медичної допомоги для значної частини нашого населення. Це стає причиною серйозних соціально-економічних викликів, оскільки кожен рік мільйони людей по всьому світу змушені витратити значну частину свого доходу на медичні послуги, що ставить їх у небезпечне становище фінансової незахищеності. Надзвичайно тривожною є інформація, що серед цих людей є не лише ті, хто проживає у країнах з низьким рівнем розвитку, а й серед мешканців багатих країн, зокрема у Європі.

Хронічні неінфекційні захворювання, пов'язані з ураженням опорно-рухового апарату, є одними з найпоширеніших у світі. За даними Всесвітньої організації охорони здоров'я (ВООЗ), приблизно 1,71 мільярда людей по всьому світу страждають від цих захворювань. Вони є одними з найважливіших факторів, що призводять до обмеження фізичної працездатності населення.

Наразі відомо більше 150 різновидів захворювань опорно-рухового апарату, які часто супроводжуються болем, обмеженням рухливості та погіршенням функціональних здібностей загалом. Це обмежує здатність осіб до виконання різних видів діяльності, зокрема трудової.

Більше того, захворювання опорно-рухового апарату є основним стимулом для зростання попиту на реабілітаційні послуги. Серед тих, хто потребує таких послуг, більше половини складають люди з хворобами опорно-рухового апарату, що робить ці послуги важливими для дітей та дорослого населення. Ефективність будь-якого лікування залежить від ранньої діагностики та попередження прогресування захворювання. Щоб діагностувати захворювання опорно-рухового апарату на ранній стадії, необхідно регулярно контролювати стан здоров'я як дітей, так і дорослих. У переважній більшості випадків ці заходи підвищують шанси на успішне лікування виявлених патологій.

Цей проект спрямований на вирішення основних соціально-економічних проблем населення та дозволить лікарям, медичним працівникам та громадянам отримати доступ до верифікованих інструментів штучного інтелекту для оцінки

ризиків захворювань опорно-рухового апарату, що є частиною пріоритетів України та Європейського Союзу щодо розширення можливостей цифрових технологій та застосування інструментів ШІ в медицині.

Перевагами розробленої системи є: неінвазивність, безконтактність, безболісність методу попереднього моніторингу, діагностики та прогресування хронічних неінфекційних захворювань хребта, а також точність і надійність AI. Запропонована система значно скорочує час діагностики та підвищує точність результатів обстеження пацієнта, а також дає можливість контролювати хід лікування.

1.5 Стан вже існуючих рішень

Існують різні комп'ютерні автоматизовані системи, які використовуються для діагностики опорно-рухового апарату. Деякі з них використовують технології штучного інтелекту та машинного навчання для аналізу медичних зображень та надають сприяння лікарям у виявленні патологій.

Наприклад, системи комп'ютерної томографії (СТ) та магнітно-резонансної томографії (MRI) використовують програмне забезпечення для автоматичного визначення аномалій та патологій у зображеннях опорно-рухового апарату. Також існують системи, які аналізують рухові дані та допомагають визначити порушення рухової активності пацієнтів.

Зокрема, деякі стартапи та медичні компанії розробляють системи, які використовують алгоритми машинного навчання для розпізнавання аномалій у зображеннях рентгенівських, КТ- або МРТ-знімків, а також для аналізу біомеханіки та рухових паттернів пацієнтів.

Варто відзначити проєкт “Brain Scan” – система працює на базі штучного інтелекту і пришвидшує процес діагностики захворювань або ушкоджень головного мозку. Нейромережа аналізує зображення комп'ютерної томографії головного мозку в автоматичному режимі та виявляє патологічні зміни. І вже протягом 5 хвилин після обробки знімка програмою лікар може ухвалити рішення про подальше лікування.

Використання штучного інтелекту для аналізу КТ-знімків головного мозку в рамках проєкту “BrainScan” показало хороші результати [7].

1.6 Визначання цілей роботи та постановка задачі

Мета даної роботи полягає у розробці та реалізації системи штучного інтелекту для діагностики захворювань опорно-рухового апарату. Основним завданням є створення програмного забезпечення, яке здатне аналізувати знімки пацієнтів та визначати на них ключові точки, тим самим прискорюючи роботу лікарів. Мета включає дослідження та вибір алгоритмів та моделей штучного інтелекту, їх початок та валідацію через тестування на вхідних даних.

В результаті цієї роботи очікується створення ефективного інструменту для швидкої та точної діагностики стану опорно-рухового апарату, що може значно полегшити роботу медичних фахівців та покращити якість надання медичної допомоги пацієнтам.

На рисунку 1.10 зображено дерево цілей даної роботи

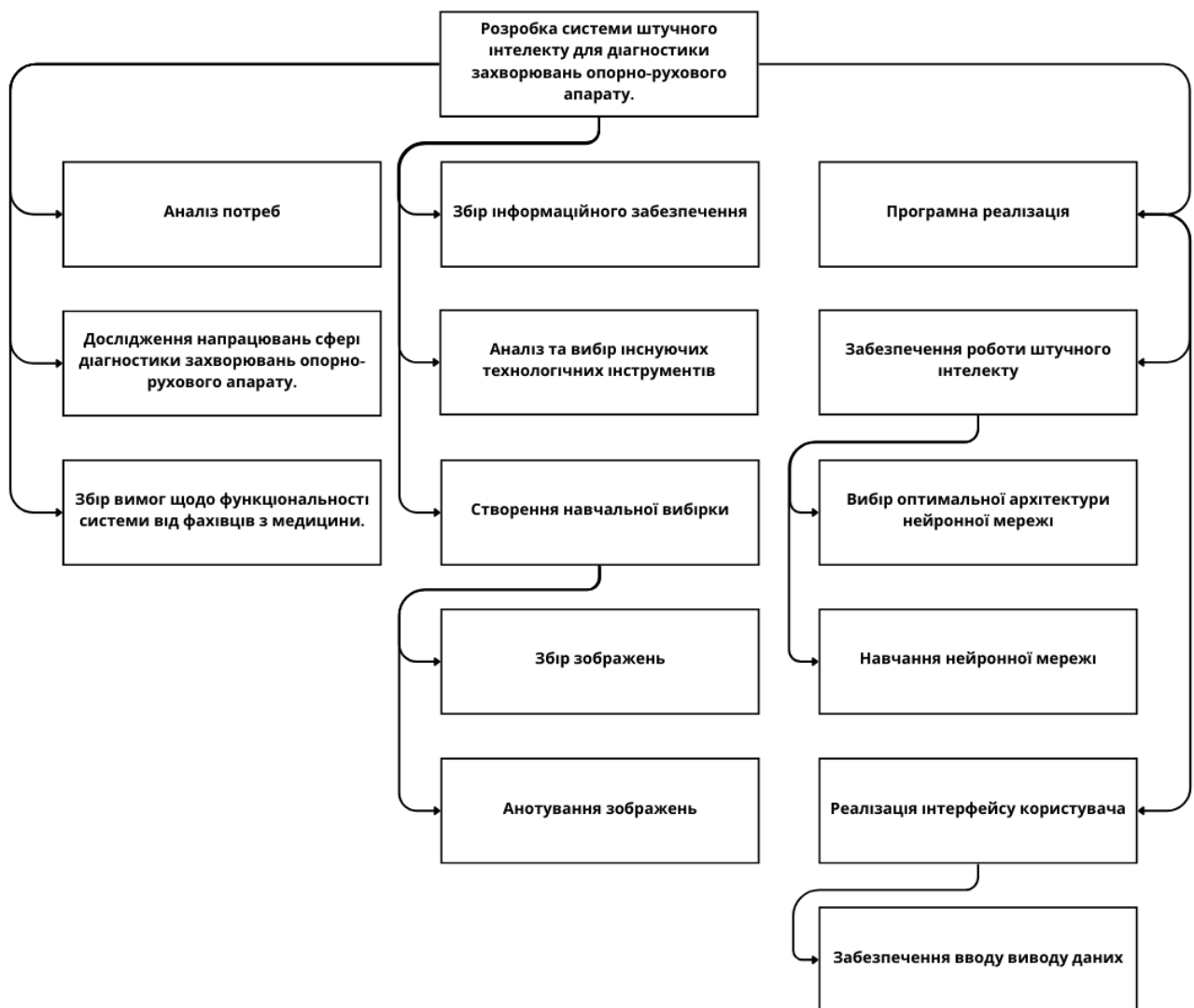


Рисунок 1.10 – Дерево цілей

Досягнення мети даної роботи ставить наступні завдання:

- 1) Збір даних для навчання нейронної мережі:
 - Організація збору стандартизованих знімків тіла людини, які будуть використовуватися для навчання системи.
 - Забезпечення відповідної якості та репрезентативності даних для ефективного навчання моделі.
- 2) Анотація даних:
 - Позначення ключових точок на тілі людини на зібраних зображеннях.
 - Розробка системи маркування та анотування даних для підготовки навчального набору.
- 3) Вибір архітектури нейронної мережі:
 - Проведення досліджень для визначення найоптимальнішої архітектури нейронної мережі для вирішення поставленої задачі.
 - Аналіз і порівняння різних моделей та їх ефективності на вхідних даних.
- 4) Навчання нейронної мережі:
 - Проведення процесу навчання моделі на підготовленому навчальному наборі даних.
 - Оптимізація параметрів мережі для досягнення найкращої продуктивності.
- 5) Розробка інтерфейсу для взаємодії з навченою моделлю:
 - Створення зручного та інтуїтивно зрозумілого інтерфейсу для взаємодії з системою діагностики.
 - Імплементация можливостей взаємодії з результатами діагностики та аналізу знайдених ключових точок.

Об'єктом дослідження даної роботи є опорно-руховий апарат людини та його захворювання.

Предметом дослідження є штучний інтелект, а саме нейронні мережі як один із його типів.

2. МАТЕМАТИЧНЕ, АЛГОРИТМІЧНЕ ТА ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Математичне забезпечення системи

2.1.1 Метод градієнтного спуску

Градiєнт – вектор, своїм напрямком вказуючий на напрямок найшвидшого зростання якоїсь скалярної величини, позначається grad .

По величині (модулю) градієнт швидкості рівня зростання величини у напрямку вектора. Наприклад, якщо взяти в якості висоти поверхні землі над рівнем моря, то її градієнт у кожній точці поверхні буде показувати «напрямок самого крутого під'єму», а його величина характеризувати крутизну схилу [8].

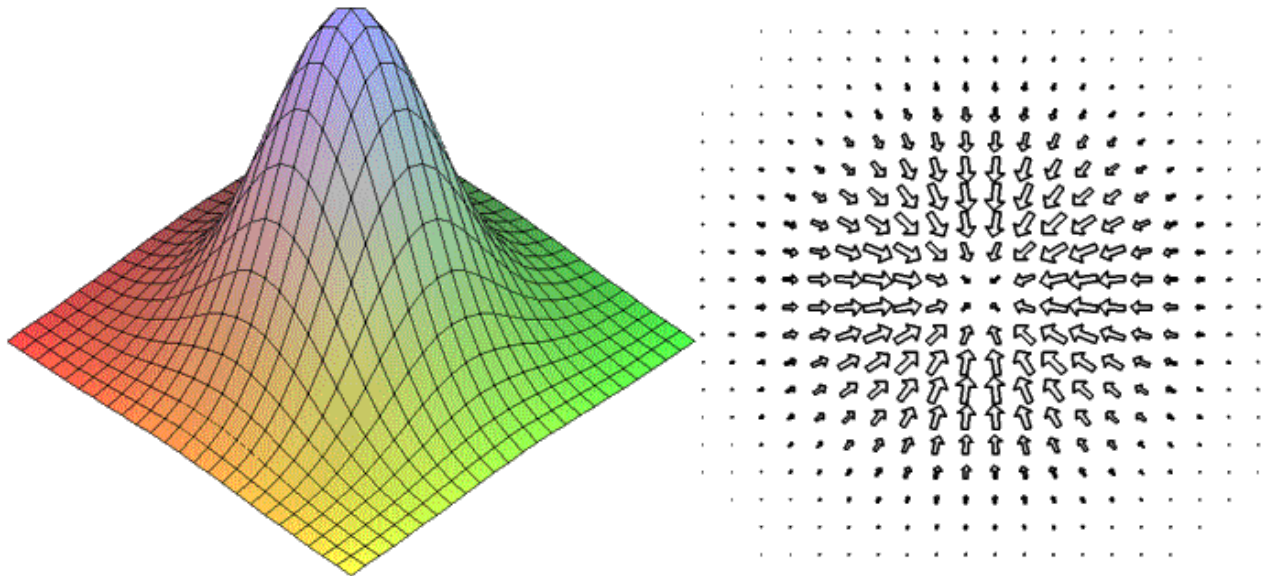


Рисунок 2.1 – Ілюстрація градієнту

На рисунку 2.1 операція градієнта перетворює пагорб (ліворуч), якщо дивитися на нього зверху, на поле векторів (праворуч). Видно, що вектори спрямовані «вгору», і чим крутіший нахил, тим вони довші.

Для скалярного поля $U(x,y,z)$ градієнт визначається формулою (2.1):

$$\nabla U = \text{grad}U = \frac{\partial U}{\partial x}i + \frac{\partial U}{\partial y}j + \frac{\partial U}{\partial z}k \quad (2.1)$$

Градiєнт лежить у основі методу градієнтного спуску.

Гرادієнтний спуск – це ітераційний алгоритм оптимізації першого порядку, в якому для знаходження локального мінімуму функції здійснюються кроки, пропорційні протилежному значенню градієнту (або наближеного градієнту) функції в поточній точці. Якщо натомість здійснюються кроки пропорційно самому значенню градієнту, то відбувається наближення до локального максимуму цієї функції; і ця процедура тоді відома як градієнтний підйом.

Градiєнтний спуск ґрунтується на тому спостереженні, що якщо функція кількох $\mathbf{F}(\mathbf{x})$ є визначеною та диференційовною в околі точки \mathbf{a} , то $\mathbf{F}(\mathbf{x})$ зменшується найшвидше, якщо йти від \mathbf{a} в напрямку, протилежному градієнтові $\nabla \mathbf{F}$ в \mathbf{a} , тобто в напрямку $-\nabla \mathbf{F}(\mathbf{a})$. З цього випливає, що якщо:

$$\mathbf{b} = \mathbf{a} - \gamma \nabla F(\mathbf{a})$$

Для достатньо малого γ , то $\mathbf{F}(\mathbf{a}) \geq \mathbf{F}(\mathbf{b})$. Іншими словами, член $\gamma \nabla \mathbf{F}(\mathbf{a})$ віднімається від \mathbf{a} , 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 $x_0, x_1, x_2 \dots$ що:

$$x_{n+1} = x_n - \gamma_n \nabla F(x_n), \quad n \geq 0$$

Ми маємо:

$$F(x_0) \geq F(x_1) \geq F(x_2) \geq \dots$$

і тому сподіваємося, що послідовність (x_n) збігається до бажаного локального мінімуму.

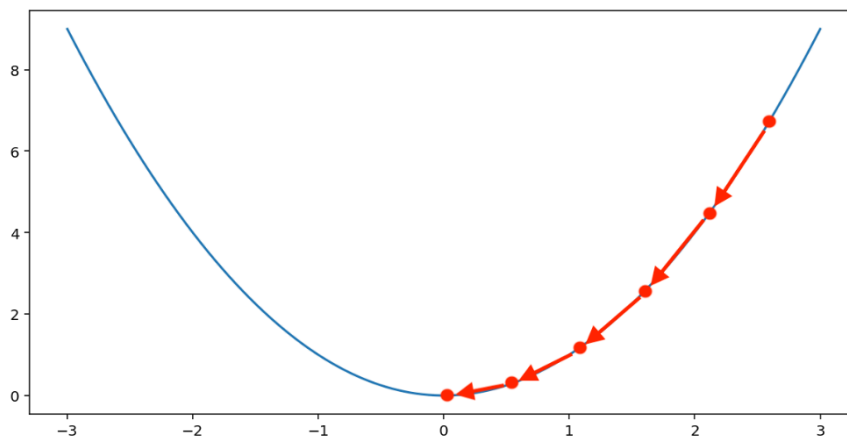


Рисунок 2.2 – Ілюстрація роботи градієнтного спуску

На рисунку 2.2 проілюстровано роботу градієнтного спуску. Спуск починається в якійсь точці. Далі проводиться серія кроків з обчисленням похідної та оновленням значення змінної доти, доки не буде досягнуто мінімуму.

Хоча алгоритм градієнтного спуску є потужним і широко використовується для оптимізації функцій, він також має деякі недоліки:

- **Пошук локальних мінімумів:** Градієнтний спуск не гарантує знаходження глобального мінімуму, а лише збігається до локального мінімуму. Це може призвести до підходів, що застрягають у локальних мінімумах і не можуть дійти до глобальної оптимальності.
- **Чутливість до вибору кроку:** Вибір оптимальної швидкості навчання (кроку) γ . Крок є важливим аспектом градієнтного спуску. Якщо швидкість навчання занадто велика, алгоритм може зациклюватися навколо мінімуму або навіть розбігатися. З іншого боку, якщо швидкість навчання занадто мала, алгоритм може зайняти багато часу для збігу.
- **Час виконання:** Градієнтний спуск може бути обчислювально витратним, особливо для великих обсягів даних або складних функцій втрат.
- **Чутливість до початкових значень:** Початкове значення x_0 може значно вплинути на шлях, який пройде градієнтний спуск. Недостатньо обдумані або випадкові початкові значення можуть призвести до швидкого застрягання в локальному мінімумі.
- **Необхідність диференційовності функції:** Градієнтний спуск потребує, щоб функція була диференційовною. Це може бути проблемою в разі виникнення неперервних функцій або випадків, коли функція має розриви або інші особливості.

На рисунку 2.3 зображено проілюстровано проблему пошуку глобального мінімуму.

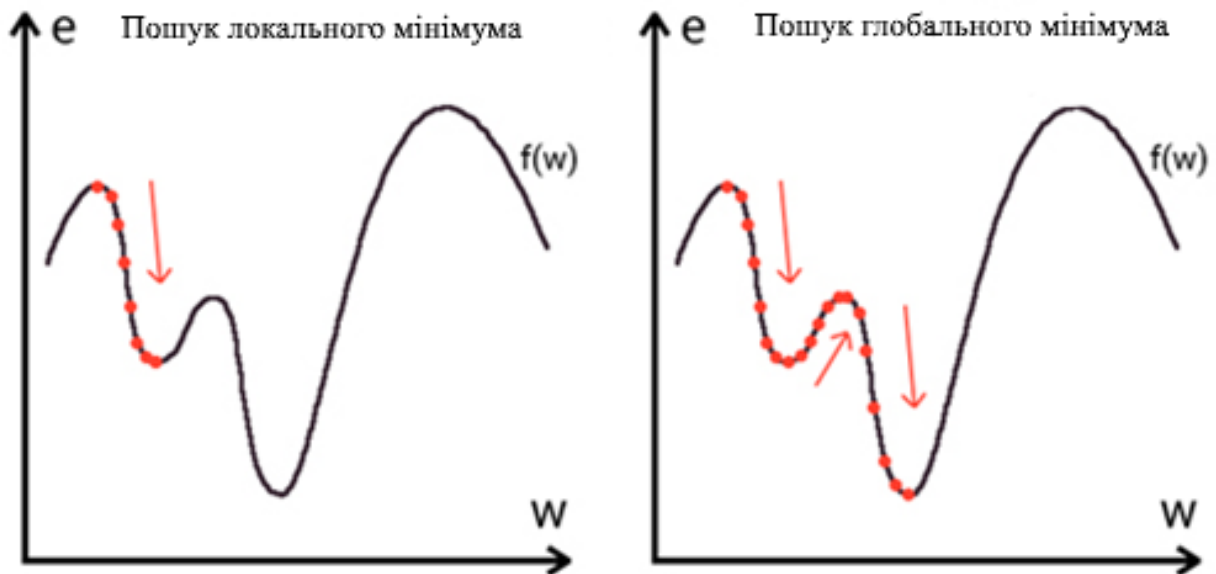


Рисунок 2.3 – Ілюстрація

Незважаючи на ці недоліки, градієнтний спуск залишається корисним і ефективним алгоритмом для оптимізації в багатьох випадках, особливо коли враховувати його різноманітні модифікації (оптимізатори) та підходи до вдосконалення.

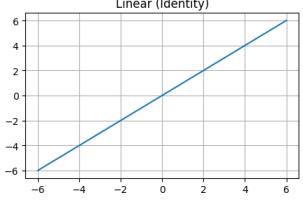
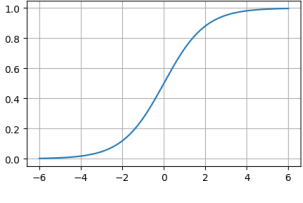
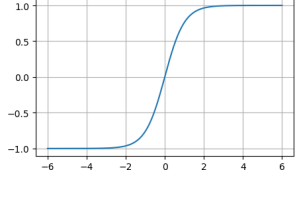
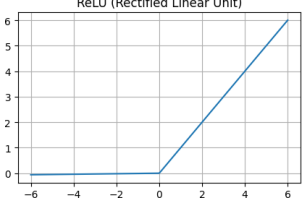
Метод градієнтного спуску є важливою складовою алгоритму навчання Back Propagation (зворотнє розповсюдження помилки), який детально оглянутий у розділі 2.2.2.

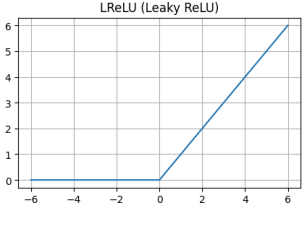
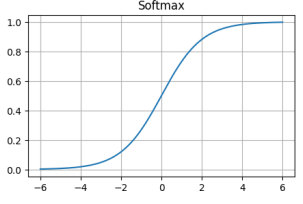
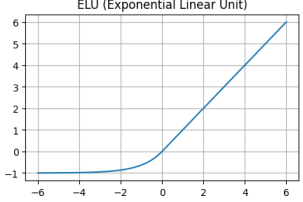
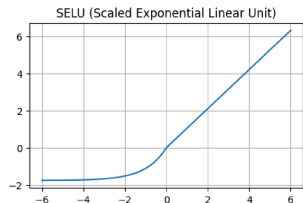
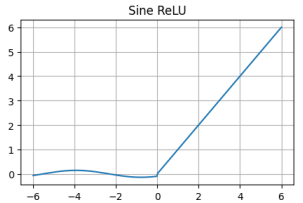
2.1.2 Функції активації нейронів

Функція активації нейрона – залежність вихідного сигналу штучного нейрона від вхідного.

Функції активації відіграють критичну роль у нейронних мережах, додаючи нелінійність до моделі, що дозволяє їй виявляти складніші патерни в даних. Вони визначають, як нейрон буде активований, залежно від суми вхідних сигналів. Популярні функції активації включають ReLU, Sigmoid, Tanh, Leaky ReLU, та інші, кожна з яких має свої переваги та сценарії застосування. На таблиці 2.1 [9] зображено популярні функції активації та їх характеристики.

Таблиця 2.1

№	Функція активації	Опис	Використання	Графік
1	Linear (або Identity)	Повертає вхід без змін. $ff(xx) = xx$	Застосовується в випадках, коли потрібно зберегти розподіл вхідних даних, наприклад, у вихідному шарі для задач регресії.	
2	Sigmoid	Приводить вихідні значення до діапазону між 0 та 1.	Часто використовується в вихідному шарі для бінарної класифікації.	
3	Tanh (гіперболічний тангенс)	Схожа на сигмоїд, але виводить значення між -1 та 1. $ff(xx) = \tanh(xx)$	Добре підходить для прихованих шарів завдяки своїй центрованості близько нуля.	
4	ReLU (Rectified Linear Unit)	Виводить вхід, якщо він позитивний; інакше повертає нуль. $ff(xx) = \max(0, xx)$	Найпопулярніша функція для прихованих шарів у глибоких нейронних мережах завдяки	

			простоті та ефективності	
5	LReLU (Leaky ReLU)	Схожа на ReLU, але пропускає маленьке значення для негативних вхідних даних. $f(x) = \max(0.01x, x)$	Допомагає уникнути проблеми «мертвих нейронів», яка може виникати при використанні ReLU.	
6	Softmax	Перетворює вектор чисел на вектор ймовірностей, що сумуються в 1.	Застосовується у вихідному шарі для багатокласової класифікації.	
7	ELU (Exponential Linear Unit)	Подібна до ReLU, але з експоненціальним нахилом для негативних вхідних даних.	Допомагає зменшити зсув активацій, може покращити швидкість навчання.	
8	SELU (Scaled Exponential Linear Unit)	Самонормалізуюча версія ELU.	Забезпечує самонормалізацію вхідних даних, ефективна для глибоких мереж.	
9	Sine ReLU	Використовує синусоїду для активації негативних значень замість лінійного або експоненційного нахилу.	Експериментальна функція для введення періодичності в активації, може бути корисною в специфічних задачах	

Вибір відповідної функції активації нейрона - це ключовий аспект у процесі проектування нейронних мереж.

Вибір та використання функцій активації суттєво впливають на функціонування мережі, оскільки без них мережа не може навчатися розпізнавати нелінійні закономірності та розподіли. Якщо рівні мережі не мають функції активації, між ними утворюється лінійна залежність, що робить кількість рівнів непливовою, а отже, робить ці рівні еквівалентними одному.

Ця функція визначає, як нейрон реагує на вхідні сигнали і передає їх на виході, і може вплинути на їхню ефективність та здатність вирішувати різноманітні завдання.

2.2 Алгоритмічне забезпечення системи

2.2.1 Згорткова нейронна мережа

Згорткові нейронні мережі (англ. convolutional neural network) – це клас глибоких штучних нейронних мереж прямого поширення, який застосовується для задач аналізу зображень. Запропонована Яном Лекуном у 1988 році та націлена на ефективне розпізнавання образів, входить до складу технологій глибокого навчання і успішно зарекомендувала себе в завданнях розпізнавання та класифікації зображень. Назва типу мережі походить від операції згортки, де кожен фрагмент зображення перемножується на матрицю (ядро) згортки, а отриманий результат додається до вихідного зображення [10].

Згорткові нейронні мережі сприймають та обробляють дані у формі тензорів, що дозволяє їм працювати з зображеннями у природному форматі. Кожне вхідне зображення має наступні параметри: ширина, висота, та глибина, яка визначається кодуванням зображення. Найбільш поширеним форматом є RGB-кодування, де кожен піксель на зображенні кодується трьома значеннями для червоного, зеленого та синього кольорів. Ці значення, які описують зображення, називаються каналами. Отже, згорткові нейронні мережі розглядають кожне зображення як чотиривимірний масив даних.

Загальна структура згорткової нейронної мережі має вхідний шар, кілька прихованих шарів і вихідний шар. Приховані шари зазвичай включають в себе згорткові шари, шари агрегування (пулінгу), шари нормалізації та повнозв'язні шари. Ці шари взаємодіють між собою за допомогою визначених функцій активації. На рисунку 2.4 зображено структуру згорткової нейронної мережі VGG16 як демонстрацію загальної структури мереж даного типу.

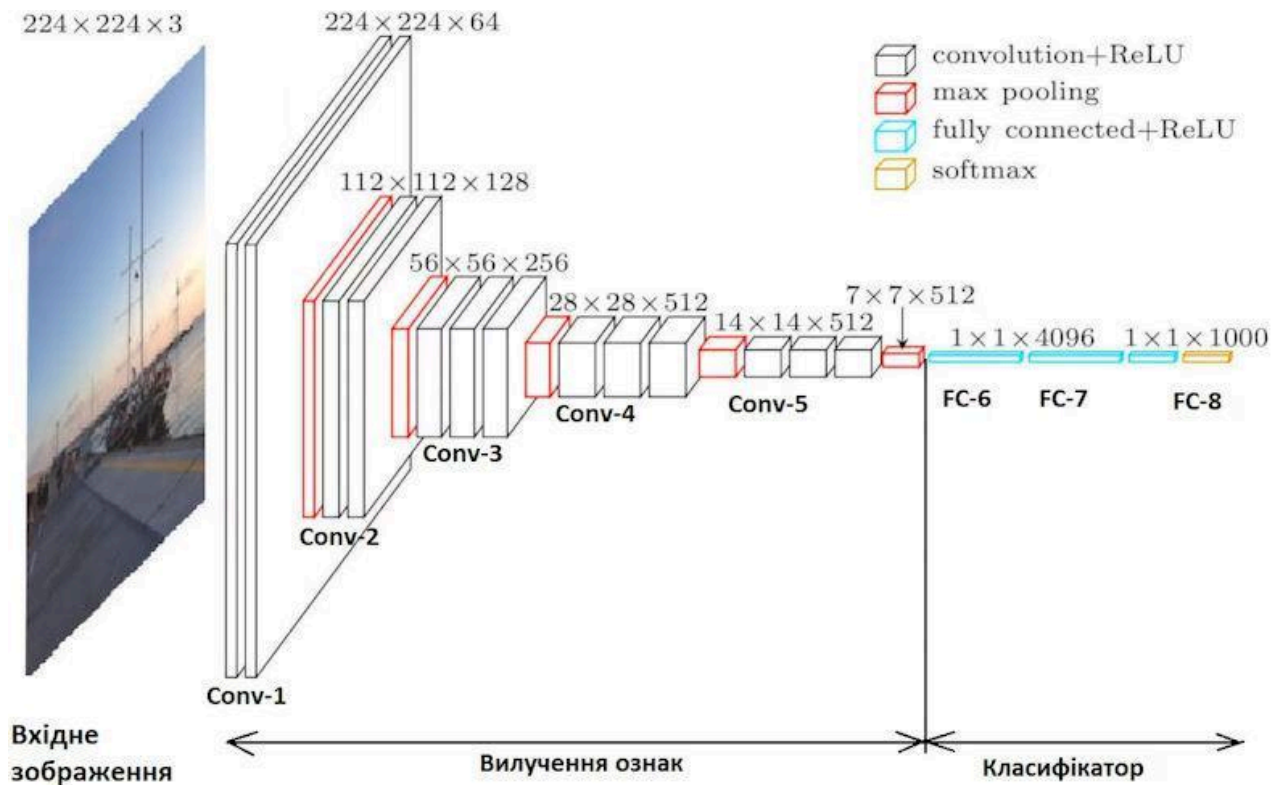


Рисунок 2.4 – Структура згорткової нейронної мережі

Згорткові шари

Згортка – операція над парою матриць A (розміру $n_x \times n_y$) і B (розміру $m_x \times m_y$), результатом якої є матриця $C = A * B$ розміру $(n_x - m_x + 1) \times (n_y - m_y + 1)$.

Операції згортки, як виконуються у згорткових шарах нейронної мережі, є ключовим елементом згорткового шару. Згортка, фактично, представляє собою математичну процедуру, яка застосовує ядро (або детектор ознак) до вхідного зображення у вигляді тензора, отримуючи на виході карту ознак. Кожен детектор ознак містить певну кількість фільтрів, кожен з яких є набором параметрів, що сканують частини зображення в ширину та висоту, зберігаючи розміри в глибину. Ці фільтри є частиною параметричної структури мережі, що піддається навчанню. Крім виявлення ознак та переходів, фільтри також можуть використовуватися для різкості або розмиття зображення. 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. Під час навчання фільтри вчаться визначати ознаки, які групуються в карту ознак. Кожен фільтр відповідає за створення однієї карти

ознак, переміщуючись зображенням і аналізуючи різні регіони. Крок переміщення є параметром моделі, впливаючи на зсув фільтра та розмір карт ознак. Хоча абсолютне положення ознак на карті не зберігається, відносне положення залишається сталим, зберігаючи властивості зображення[11].

На рисунку 2.5 зображено приклад операції згортки.

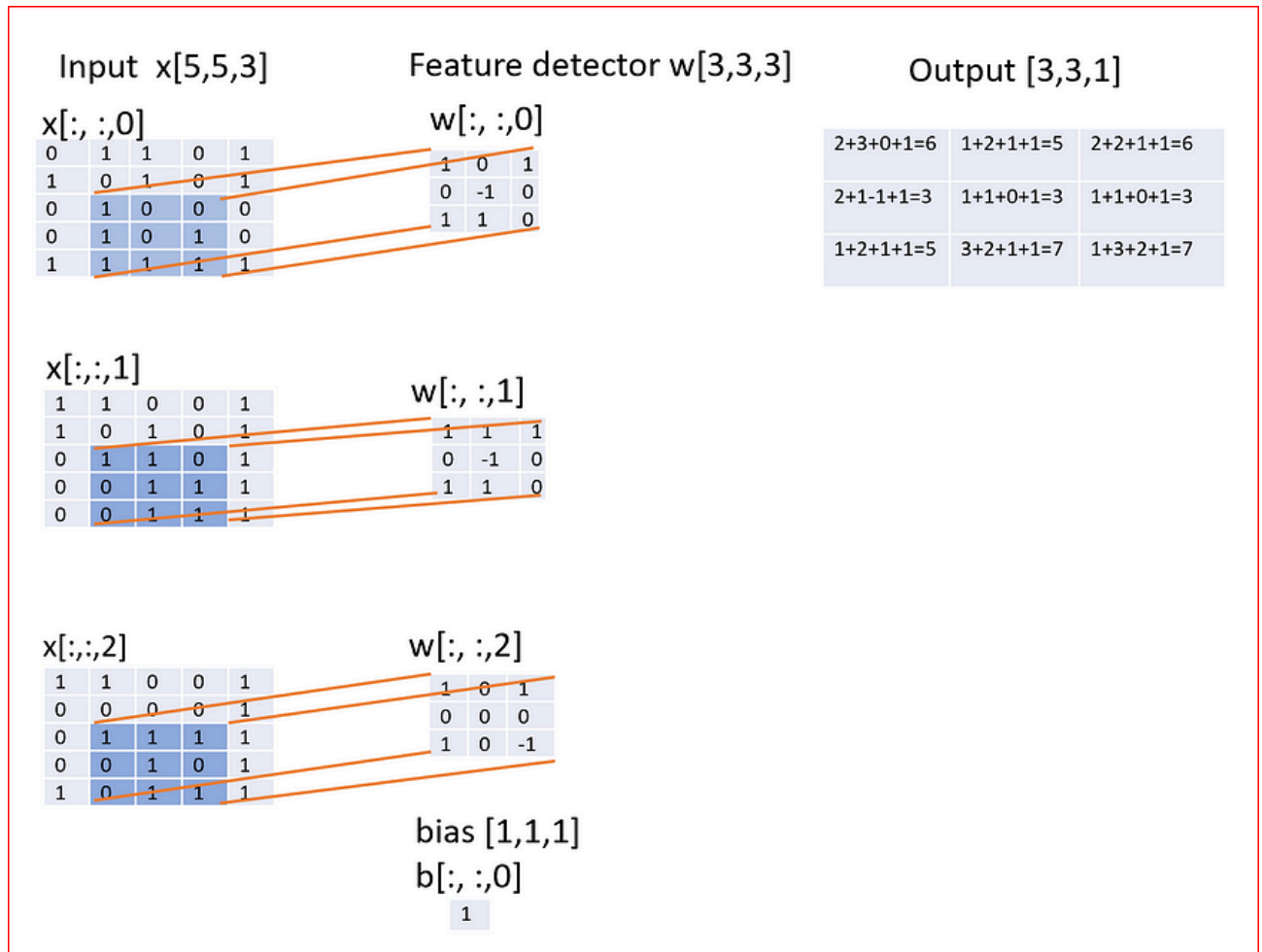


Рисунок 2.5 – Операція згортки

Загальний процес перетворень у згортковому шарі може бути описаний рівнянням:

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

де, y – вихідний результат обчислень на згортковому шарі,

w – детектор ознак,

x – вихідні дані,

b – зсув.

Агрегувальні шари

Агрегувальні шари, також відомі як шари субдискретизації, або шари пулінгу, є ключовими компонентами згорткових нейронних мереж, нарівні зі згортковими шарами. Основна мета агрегаційних шарів полягає у зменшенні розмірності даних, зберігаючи при цьому ключові характеристики шляхом встановлення залежностей між групами елементів (нейронами) з попереднього шару та єдиним елементом даного шару. Тому вони зазвичай використовуються з певною періодичністю між згортковими шарами при конструюванні мережі[12].

Важливо зауважити, що агрегаційні шари зберігають глибину вхідних даних, уникаючи при цьому перенавчання. Ці шари мають два основних типи: середнє (**Average Pooling**) та максимальне (**Max Pooling**). Середнє значення кожного елементу визначається як середнє значення відповідних елементів групи нейронів з попереднього шару, у той час як максимальне значення обирається найбільшим з них.

Max Pooling – це операція об'єднання, яка вибирає максимальний елемент із області карти об'єктів, охопленої фільтром. Таким чином, результатом після шару максимального об'єднання буде карта ознак, яка містить найвидатніші функції попередньої карти ознак. На рисунку 2.6 зображено приклад використання Max Pooling.

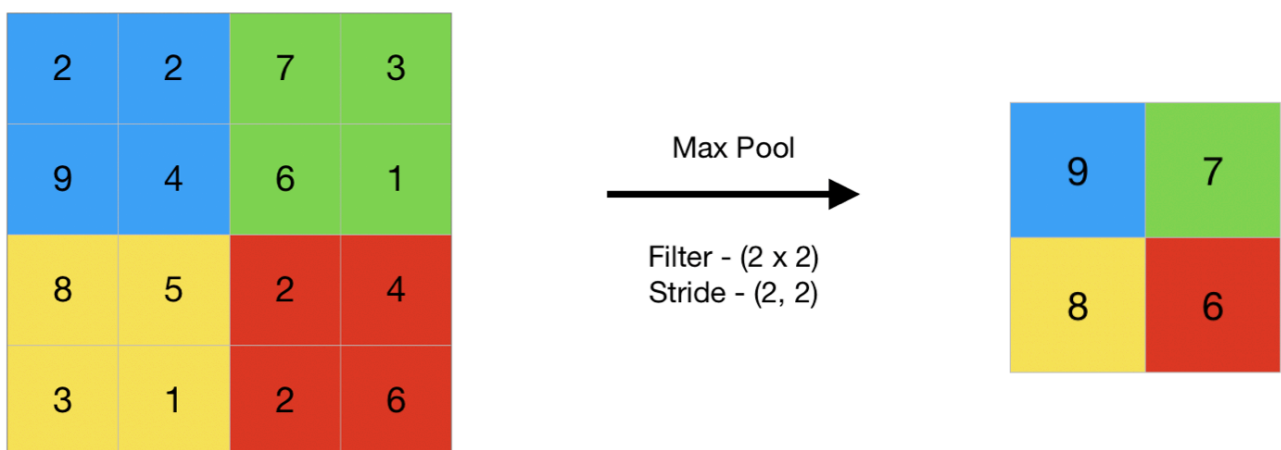


Рисунок 2.6 – приклад Max Pooling

Average Pooling – обчислює середнє значення елементів, присутніх в області карти об'єктів, охопленої фільтром. Таким чином, у той час як максимальне об'єднання дає найвидатнішу ознаку в карти ознак, Average Pooling дає середнє значення, серед присутніх на карті. На рисунку 2.7 зображено приклад виконання Average Pooling

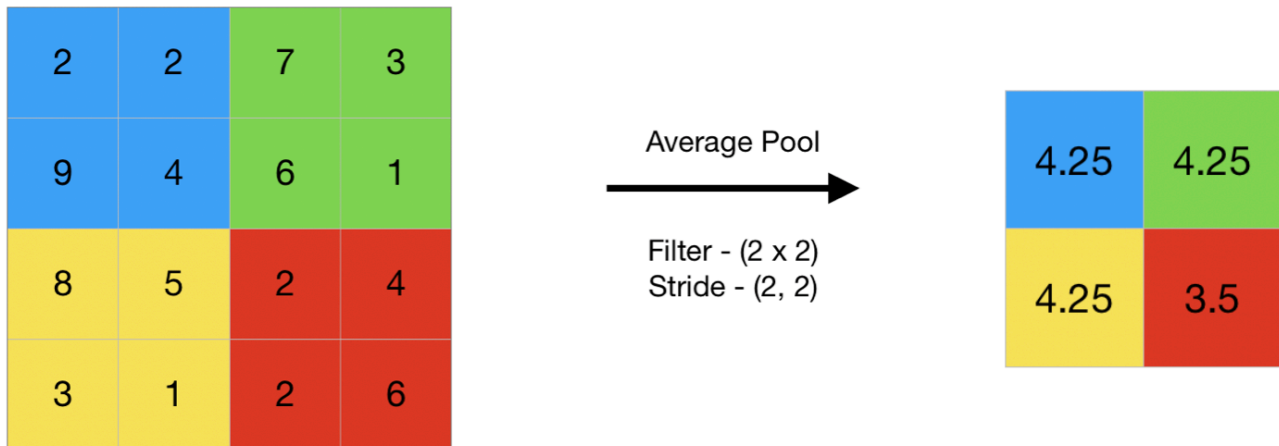


Рисунок 2.7 – Приклад Average Pooling

Агрегаційні шари виконують важливу функцію в забезпеченні стійкості мережі до збурень в даних, таких як зміна кута огляду об'єкта або його положення. Це досягається шляхом вибору значень з множини за певним критерієм, таким як середнє або максимальне, які зменшують вплив незначних змін у вхідних даних на вихідні результати агрегаційного рівня. Таким чином, схожі ознаки, наприклад, переходи або кути, будуть мати схожі вихідні значення незалежно від деяких змін у положенні або освітленні. Для таких завдань частіше за все використовується максимізаційне агрегування.

Агрегаційні шари мають декілька спільних гіперпараметрів з операцією згортки, таких як крок обходу зображення, доповнення нулями та розмірність рецептивного поля, проте зазвичай використовується вікно розміром 2x2. Вони дозволяють значно зменшити розмірність вхідних даних та мінімізувати вплив окремих елементів з невеликими значеннями на результат, що робить використання малих рецептивних полів доцільним. Існують архітектури згорткових мереж, які взагалі не використовують агрегаційні шари, замінюючи їх згортковими шарами зі збільшенням кроку.

Повнозв'язні шари

Повнозв'язні шари в згорткових нейронних мережах це ті шари де всі нейрони з наступного шару з'єднані з нейронами попереднього шару, подібно до більшості шарів у звичайних штучних нейронних мережах(рис 2.8).

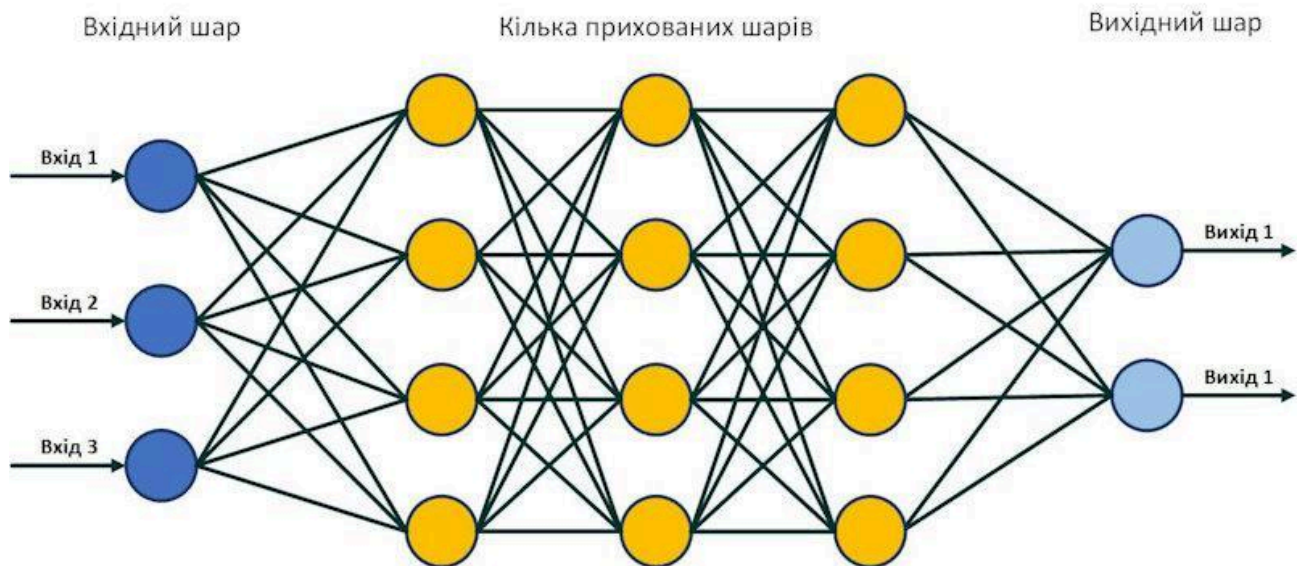


Рисунок 2.8 – Повнозв'язна нейронна мережа

Використання таких рівнів на початкових та прихованих рівнях мережі вважається неефективним, оскільки це може ускладнити модель та втратити раніше виявлені за допомогою згорток та агрегації ознаки та закономірності. Проте, повнозв'язні рівні зазвичай використовуються на передостанньому кроці роботи мережі для підготовки результатів на виході мережі[13].

Для роботи повнозв'язного рівня елементи матриць та шарів з попереднього кроку перетворюються в послідовність значень. Такі рівні, подібно до згорткових, виконують обчислення скалярного добутку даних та параметрів з додаванням зсуву. Тому загальна формула повнозв'язного рівня подібна до формули згорткового:

$$y = f(wT * x + b)$$

за відмінністю того, що w представляє масив параметрів рівня, а не детектор ознак.

На повнозв'язних рівнях застосовують функцію активації, результат якої використовується на наступному рівні або для обчислення результатів мережі на виході. Тому типи активаційних функцій, які використовуються на цьому рівні,

відрізняються від тих, що використовуються на згорткових та агрегаційних рівнях, і можуть бути різними в залежності від положення рівня та загальної мети мережі

2.2.2 Метод зворотнього поширення помилки

Під час навчання багат шарових нейронних мереж виконується процес мінімізації цільової функції, що представляє собою певний критерій якості $E(Q(\varepsilon, s))$. Ця функція відображає інтегральну відстань між виходами мережі $y(M)$ та очікуваними виходами $y^* = \{y_s^*\}$ де s - номер конкретного прикладу в навчальній вибірці, S - загальна кількість прикладів у навчальній вибірці. Функція $Q(\varepsilon, s)$ є миттєвим критерієм якості, який залежить від вектора помилок мережі для i -го вихідного змінного s -го прикладу $\varepsilon(s, i, w) = y(M, i) - y_s^*$, де w - набір ваг мережі, $y(M, i)$ та y_s^* - відповідно, обчислені мережею і бажані значення.

Для кожного вхідного вектора x^s з навчальної множини повинен бути визначений вектор бажаних виходів мережі y^* . Якщо навчальна штучна нейронна мережа використовується в якості класифікатора, зазвичай бажані виходи мають низький рівень (0 або менше 0,1), крім виходу вузла, що відповідає класу, до якого належить x ; цей вихід у даному випадку має високий рівень (1 або більше 0,9). Для початку навчання ваги ШНМ встановлюються рівними випадковим числам.

Для мінімізації цільової функції навчання вирішується завдання багатомірної нелінійної оптимізації, для чого традиційно використовують градієнтні методи. Ці методи є ітеративними, оскільки компоненти градієнта виявляються нелінійними функціями. Градієнтні методи ґрунтуються на ітераційній процедурі корекції значень ваг, яка реалізується за формулою:

$$w_{k+1} = w_k + \alpha k p(w_k), \text{ де } k=0, 1, 2, \dots$$

де w_k та w_{k+1} - поточне та нове наближення значень ваг і порогів ШНМ до оптимального рішення відповідно; αk - крок збіжності; $p(w_k)$ - напрямок пошуку в багатовимірному просторі ваг. Спосіб визначення $p(w_k)$ та αk на кожній ітерації залежить від особливостей конкретного методу [13].

Оскільки градієнтні методи вимагають обчислення похідних цільової функції для визначення напрямку пошуку, для розрахунку часткових похідних цільової функції за вагами мережі використовується метод зворотнього поширення помилки.

Метод зворотного поширення помилки (англ. backpropagation) – метод навчання багатошарового перцептронну.

Це ітеративний градієнтний алгоритм, який використовується з метою мінімізації помилки роботи багатошарового перцептронну та отримання бажаного виходу. Основна ідея цього методу полягає в поширенні сигналів помилки від виходів мережі до її входів, в напрямку, зворотному прямому поширенню сигналів у звичайному режимі роботи. Для можливості застосування методу зворотного поширення помилки функція активації нейронів повинна бути диференційовною.

Метод зворотного поширення помилки для налаштування ваг ШНМ використовує градієнт функції помилки таким чином, щоб мінімізувати помилку на навчальній вибірці. Для цього мережі послідовно надаються вхідні вектори (екземпляри) з навчальної вибірки, і, починаючи з останнього шару, обчислюються градієнти функції помилки для кожного нейрона[20].

У методі зворотного поширення помилки ваги і пороги змінюються у напрямку, протилежному градієнту. Тому базовий метод зворотного поширення часто називають методом градієнтного спуску (gradient descent). Основні кроки базового методу зворотного поширення помилки включають:

- 1) Задати навчальну вибірку x та значення цільового параметру y^* . Ініціалізувати ваги та пороги мережі, встановити рівень помилки ξ , задати $s=1$.
- 2) Подати на вхід s -й екземпляр x та отримати вихідний вектор y .
- 3) Розрахувати виходи нейронів та фактичний вихід мережі при подачі на вхід екземпляра x .
- 4) Розрахувати помилку мережі E . Якщо $E > \xi$, перейти на крок 5; інакше – на крок 6.
- 5) Настроювання ваг і порогів від вихідного до першого шару мережі.
- 6) Збільшити s на одиницю. Якщо $s > S$, перейти на крок 7, інакше – на крок 2.
- 7) Визначити помилку мережі E . Якщо $E \leq \xi$, завершити пошук, в іншому випадку встановити $s=1$ та перейти до кроку 2.

Як додаткові критерії завершення пошуку у методі зворотного поширення помилки можуть використовуватися досягнення ліміту часу або ліміту кількості

циклів корекції ваг (епох навчання) або досягнення заданого мінімального значення градієнта цільової функції[13].

На рисунку 2.9 візуально зображено алгоритм навчання нейронної мережі за допомогою методу зворотнього поширення помилки.

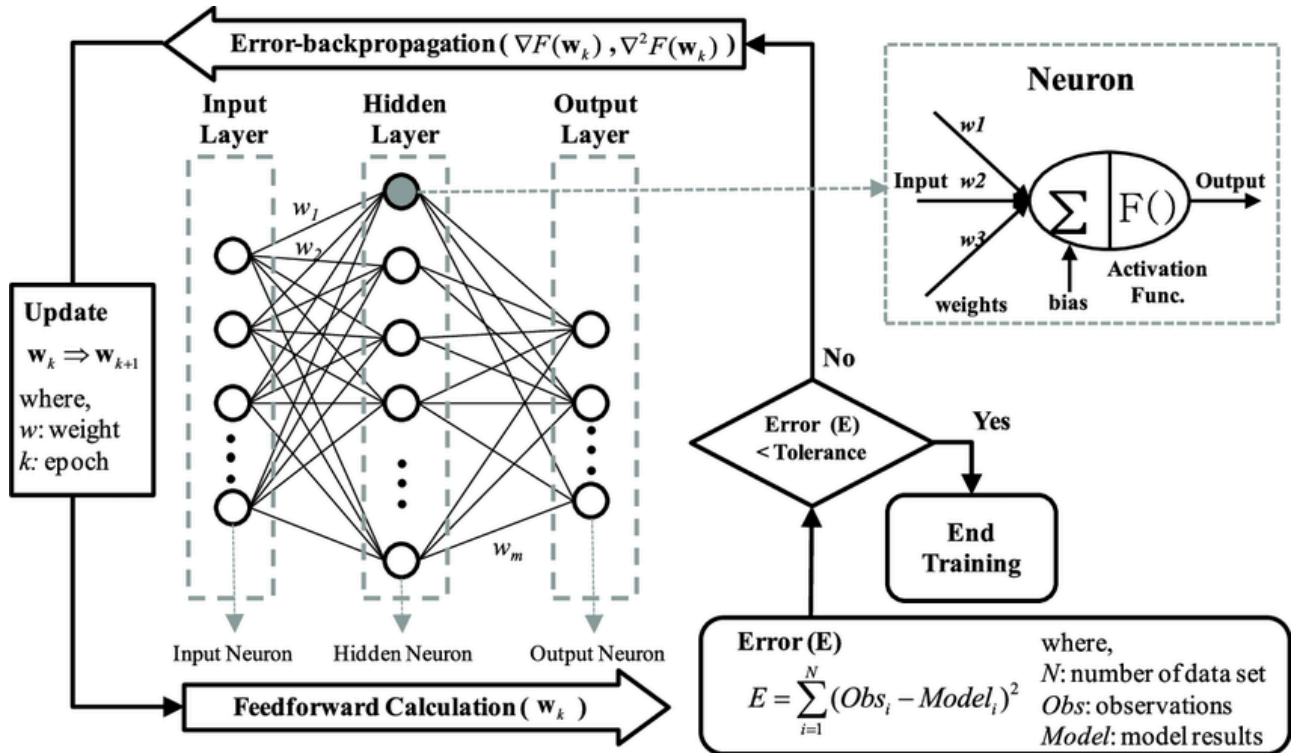


Рисунок 2.8 – Візуальне зображення алгоритму Back propagation

Стандартний метод зворотного поширення часто збігається дуже повільно, рухаючись вздовж плоских ділянок поверхні помилки. Тому на практиці його часто використовують лише для розрахунку приватних похідних цільової функції помилки за вагами ШНМ, а пошук в просторі ваг ШНМ роблять на основі більш швидких градієнтних методів.

Основним недоліком традиційно використовуваних градієнтних методів навчання БНМ є обумовлена ітераційною корекцією ваг низька швидкість збіжності навчання, яка серйозно обмежує практичне застосування нейронного керування. Іншим не менш важливим недоліком даних методів є невизначеність у виборі початкової точки пошуку, параметрів архітектури та топології мережі.

Оптимізатори є ключовими інструментами для розв'язання проблем, зазначених вище. Вони призначені для покращення швидкості збіжності та ефективності навчання нейронних мереж. Одним з найпоширеніших оптимізаторів є

стохастичний градієнтний спуск (SGD). Він використовує градієнт функції втрат для оновлення параметрів мережі, проте може мати проблеми зі збіжністю, особливо на площинах, де градієнт може бути дуже малим.

Щоб подолати ці недоліки, було розроблено багато різних оптимізаторів. Наприклад, оптимізатори на базі методу моменту (наприклад, Momentum, Nesterov Accelerated Gradient) допомагають уникнути стагнації на площинах з малим градієнтом, прискорюючи навчання. Інші оптимізатори, такі як Adam (Adaptive Moment Estimation), комбінують переваги різних методів, використовуючи як інформацію про середні квадрати градієнтів, так і метод моменту.

Однією з основних переваг цих оптимізаторів є їх здатність адаптуватися до різних умов навчання, таких як різка зміна градієнтів або різні масштаби параметрів. Вони також можуть допомогти уникнути попадання в локальні мінімуми, які можуть затримати збіжність.

2.3 Інформаційне забезпечення

2.3.1 Задача виявлення ключових точок

Виявлення ключових точок в комп'ютерному зорі – це завдання, спрямоване на визначення розташування об'єктів (часто людей) (рис. 2.10) та їх ключових точок у певній області (наприклад, ноги, руки, голова). Ця технологія лежить в основі багатьох передових розробок, які дозволяють застосуванням від розпізнавання облич на смартфонах до відстеження об'єктів для автономних транспортних засобів або аналізу медичних зображень.

Виявлення ключових точок є ключем до цього трансформаційного бачення, що дозволяє комп'ютерам ідентифікувати та точно визначати характеристики зображень. Ці ключові точки виступають опорними маркерами, які допомагають машинам розуміти складні візуальні дані, з якими вони працюють.

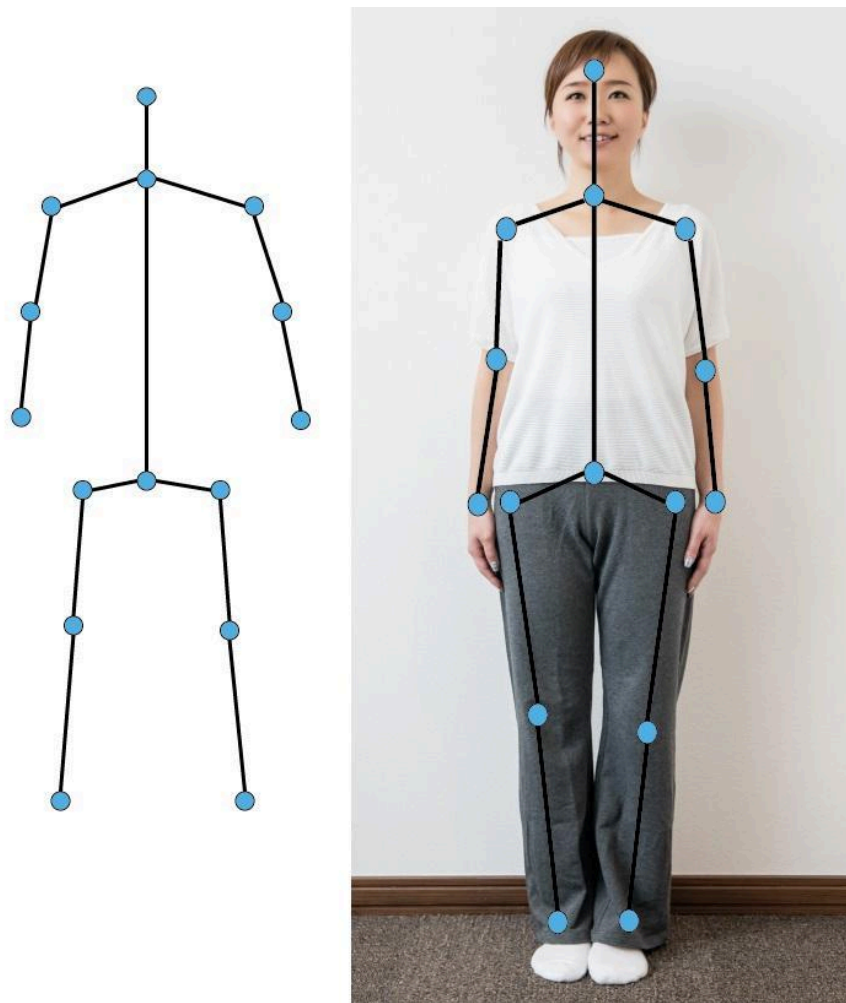


Рисунок 2.10 – Виявлення пози людини

Виявлення ключових точок в комп'ютерному зорі відкриває безліч можливостей у різних сферах. Наприклад, в медицині ця технологія може бути використана для аналізу медичних зображень, допомагаючи лікарям виявляти аномалії або відслідковувати зміни у пацієнтів з хронічними захворюваннями. У сфері безпеки виявлення ключових точок може бути застосоване для відстеження руху людей або об'єктів у великих просторах, що допомагає в підвищенні рівня безпеки на громадських місцях або в об'єктах інфраструктури.

Також ця технологія знайшла широке застосування в індустрії розваг. Вона допомагає в розробці ігор з розширеною реальністю, де комп'ютер може виявляти рухи гравця та інтегрувати їх у віртуальний світ, що підвищує іммерсивність геймплею. Крім того, у сфері робототехніки виявлення ключових точок є важливою складовою автономних систем, де роботи здатні відслідковувати та реагувати на рухи об'єктів навколо них, що робить їх більш ефективними та безпечними для співіснування з людьми.

Виявлення ключових точок означає визначення конкретних відмітних точок або областей на зображенні або відеокадрі. Ці важливі точки, що часто називають "ключовими", виступають як орієнтири або маркери. Вони можуть бути використані машинами для аналізу та інтерпретації візуального змісту зображення.

Існує кілька підтипів задач виявлення ключових точок, включаючи:

- Оцінка пози людини (рис. 2.10): визначення ключових моментів, пов'язаних з людьми.
- Оцінка пози рук: визначення ключових точок, пов'язаних з руками людини.
- Ключові моменти обличчя (рис 2.11): визначення важливих точок на обличчях людей.
- Ключові моменти щодо тварин: виявлення ключових моментів, пов'язаних з тваринами (наприклад, котами, рибами).

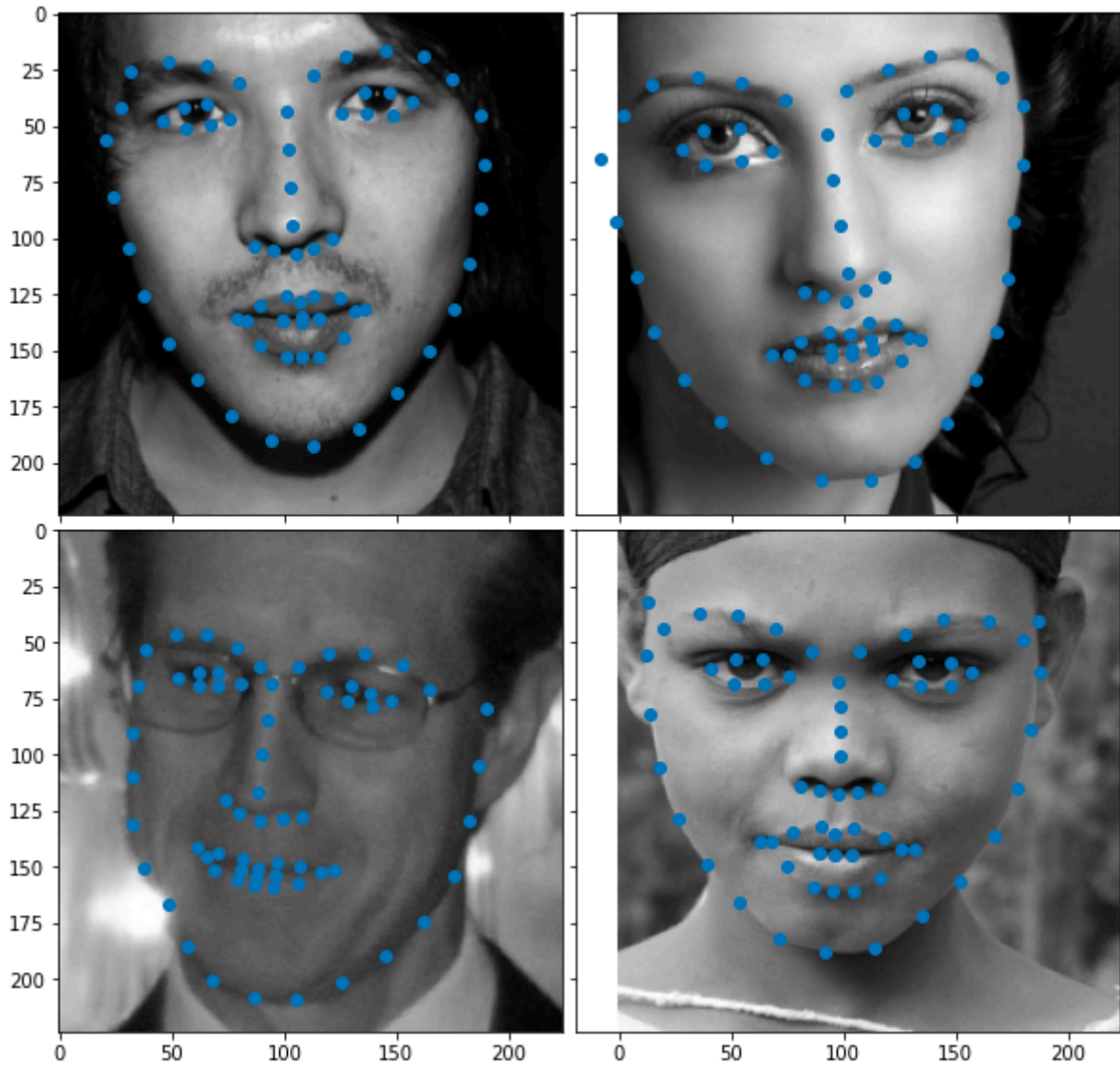


Рисунок 2.11 – Ключові точки на обличчях людей

Ключові точки, як правило, визначаються за допомогою конкретних характеристик, які відрізняють їх від оточуючих пікселів [14].

Ці характеристики:

- **Унікальність:** Ключові точки повинні бути унікальними та легко відрізнятися від інших точок зображення. Вони виділяються завдяки певним візуальним атрибутам, таким як колір, інтенсивність або текстура.
- **Інваріантність:** Ключові точки мають демонструвати певний ступінь інваріантності до типових трансформацій зображення, таких як обертання, масштабування та зміни умов освітлення. Іншими словами, ту саму ключову точку слід виявляти в різних версіях того самого об'єкта чи сцени.

- **Повторюваність:** Ключові моменти мають надійно виявлятися в різних екземплярах того самого об'єкта чи сцени. Ця повторюваність важлива для різних застосувань, включаючи розпізнавання та відстеження об'єктів.

З появою методів глибокого навчання виявлення ключових точок суттєво посилюється. Моделі глибокого навчання проявили вражаючі здібності у різних завданнях комп'ютерного зору, включаючи виявлення ключових точок.

До появи глибокого навчання традиційні методи, такі як Harris Corner Detector і SIFT, були основними в задачах виявлення ключових точок. Однак з розвитком глибокого навчання ці методи втратили свою актуальність, оскільки архітектури глибокого навчання виявилися потужнішими та більш ефективними завдяки кращому використанню даних.

2.3.2 Методи знаходження ключових точок

Методи Harris Corner Detector і SIFT (Scale-Invariant Feature Transform), які згадані раніше, є двома відомими техніками в області комп'ютерного зору, використовуваними для виявлення ключових точок на зображеннях [14].

Harris Corner Detector – це алгоритм, який виявляє точки на зображенні, де зміни в яскравості або інтенсивності змінюються у багатопромінцевому напрямку. Ці точки, відомі як «розгалужені точки» або «кути», є потенційно важливими для розпізнавання об'єктів або структур на зображенні. На рисунку 2.12 зображено результат роботи алгоритму на зображенні дошки для шахів.

SIFT є алгоритмом, який виявляє ключові точки на зображеннях, що є незалежними від масштабу та орієнтації. Цей метод особливо корисний в задачах розпізнавання об'єктів у змінних умовах, таких як різні розміри чи орієнтації об'єктів на зображенні.

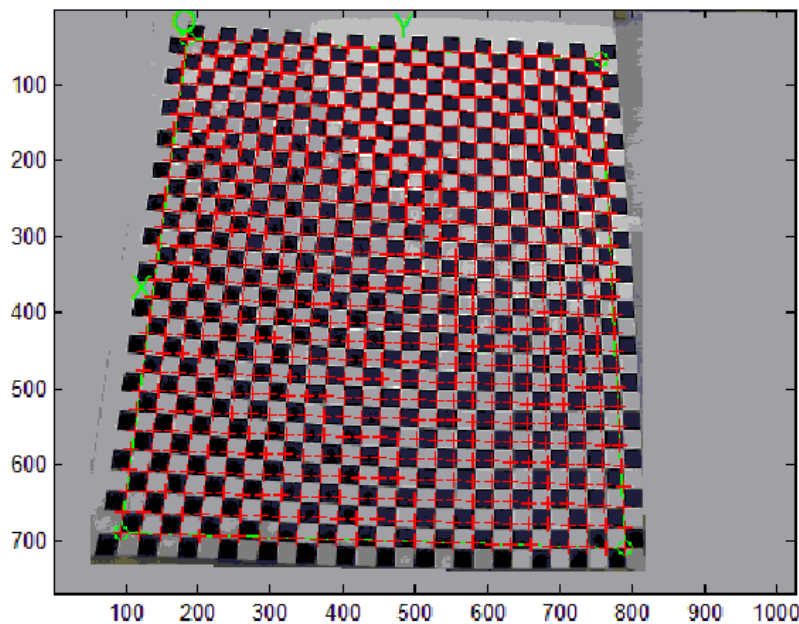


Рисунок 2.12 – Знаходження ключових точок на дошці для шахів

Якщо раніше ці методи були широко використовуваними в комп'ютерному зорі, то з появою методів глибокого навчання вони поступово втрачають свою популярність, оскільки архітектури глибокого навчання надають кращі результати в багатьох випадках та зазвичай не вимагають такого великого обсягу передпроцесингу даних, як це потрібно для Harris Corner Detector і SIFT.

Також недостатком цих методів є те, що вони не передбачають виявлення попередньо визначених точок, тож вони не підходять для вирішення задач нашої системи. Тому варто розглянути методи які базуються на нейронних мережах.

YOLO (You Only Look Once) – це широко відома та потужна архітектура глибокого навчання для виявлення об'єктів на зображеннях та відео. Відома своєю швидкістю та ефективністю, YOLO використовує одну нейронну мережу для прогнозування обмежувальних рамок та класів об'єктів одночасно. Це означає, що зображення проходить через мережу лише один раз, що робить YOLO швидким і придатним для реального часу застосування.

Хоча YOLO переважно використовується для виявлення об'єктів, його можна налаштувати для завдань виявлення ключових точок. Для досягнення цього можна розширити мережу, додавши додаткові вихідні шари, які будуть передбачати координати цих ключових точок. Кожна ключова точка буде відповідати певному набору вихідних каналів у цих додаткових шарах.

Наприклад, у PoseTED (рис. 2.13) використовується YOLO-v4 для виявлення людей, що призводить до локалізації осіб за допомогою обмежувальних рамок. Потім мережа просторового трансформатора (STN) використовується для виділення цікавих областей шляхом обрізання вихідного зображення на основі передбачених обмежувальних рамок.

Магістральна мережа обробляє ці регіони для створення складних представлень функцій. Процес визначення ключових точок відносно відповідних обмежувальних рамок спрощується за допомогою кодера-декодера Transformer з позиційним кодуванням. Нарешті, для передбачення ключових точок і їх подання у вигляді векторів, що відповідають різним частинам тіла, використовується мережа прямого зв'язку на основі прогнозування.

Крім того, архітектура Ultralytics YOLOv8 підтримує визначення ключових точок нативно.

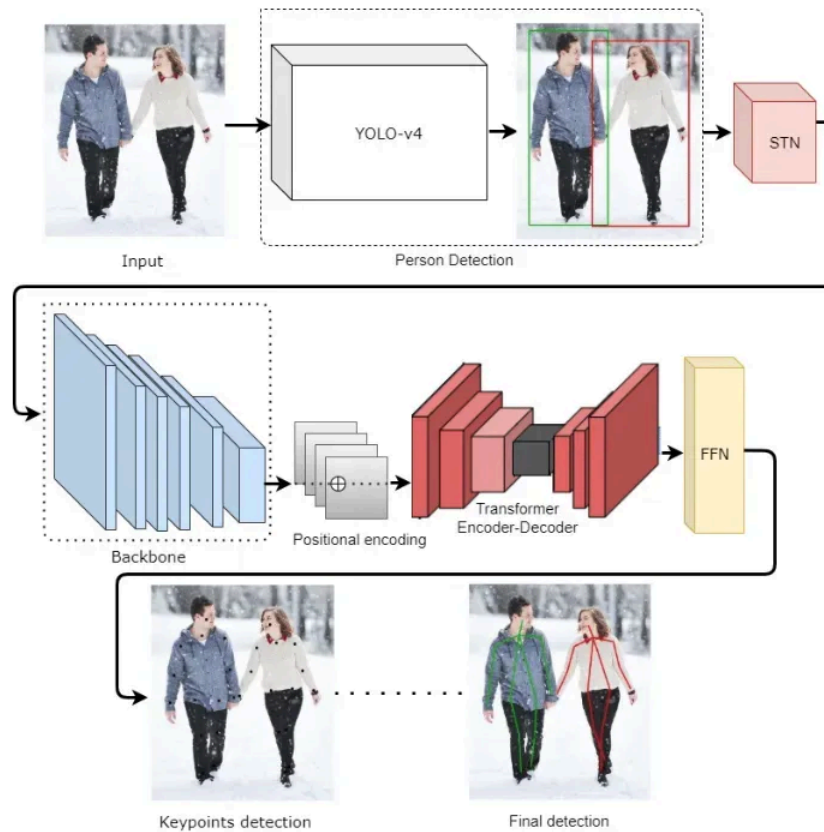


Рисунок 2.13 – Архітектура PoseTed

OpenPose – це система комп'ютерного зору, розроблена для одночасного виявлення та відстеження кількох частин тіла людини на зображеннях або відео. Вона використовує магістральну згорткову нейронну мережу (рис. 2.14) для аналізу вхідних даних та виявлення різних частин тіла, таких як суглоби, руки, ноги та обличчя.

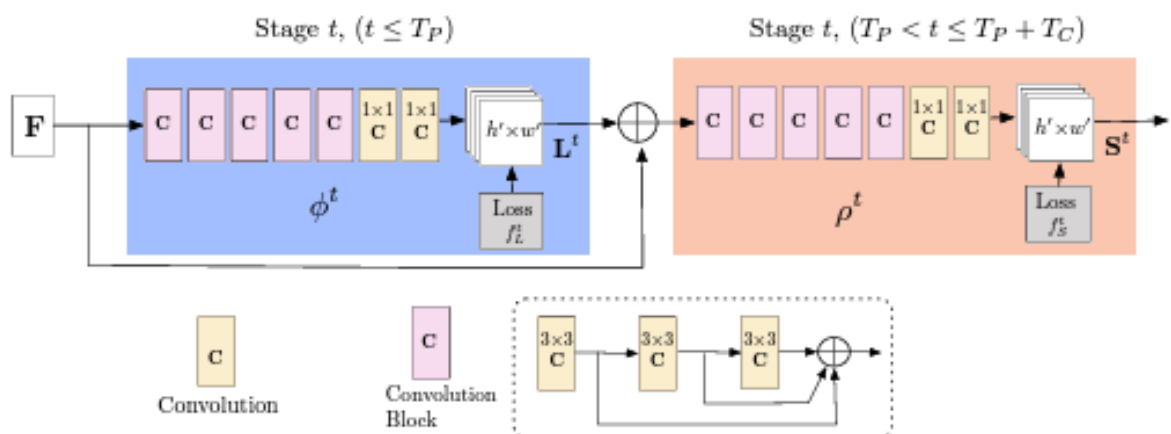


Рисунок 2.14 – Архітектура згорткової нейронної мережі OpenPose

Одним з ключових елементів OpenPose є створення карт достовірності та полів спорідненості частин, що відображає ймовірність присутності частини тіла та її

просторові взаємозв'язки з іншими частинами. На основі цих карт система визначає ключові точки та їх зв'язки для кожної людини на зображенні[15].

Процес ознаходження ключових точок на тілах людей за допомогою OpenPose зображено на рисунку 2.15.

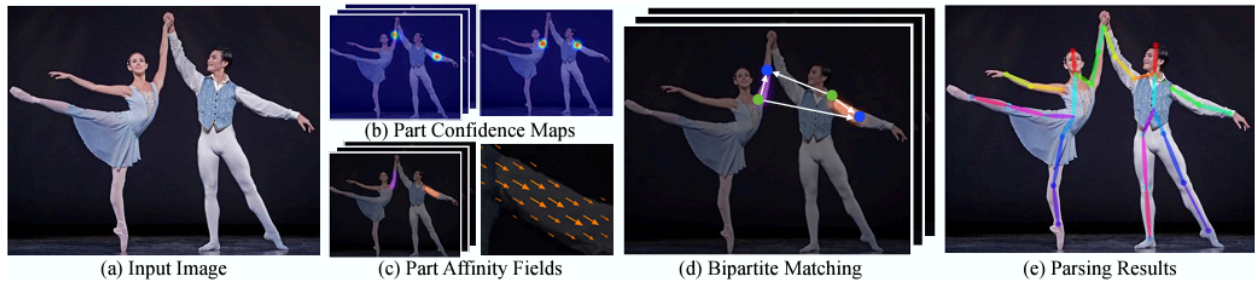


Рисунок 2.15 – Знаходження ключових точок за допомогою OpenPose

Keypoint-RCNN – це модель для виявлення об'єктів, яка розширює архітектуру Faster R-CNN, додавши здатність визначати ключові точки на зображеннях. Починаючи з використання мережі регіональних пропозицій (RPN) для створення обмежувальних рамок, Keypoint-RCNN додатково прогнозує розташування ключових точок у кожній пропозиції

Архітектура Keypoint-RCNN (рис. 2.16) базується на стандартній рамці Faster R-CNN і включає додаткову гілку, присвячену передбаченню ключових точок. Ця гілка складається з серії згорткових шарів, за якими йдуть повністю зв'язані шари, які виводять координати ключових точок для кожного виявленого екземпляра об'єкта.

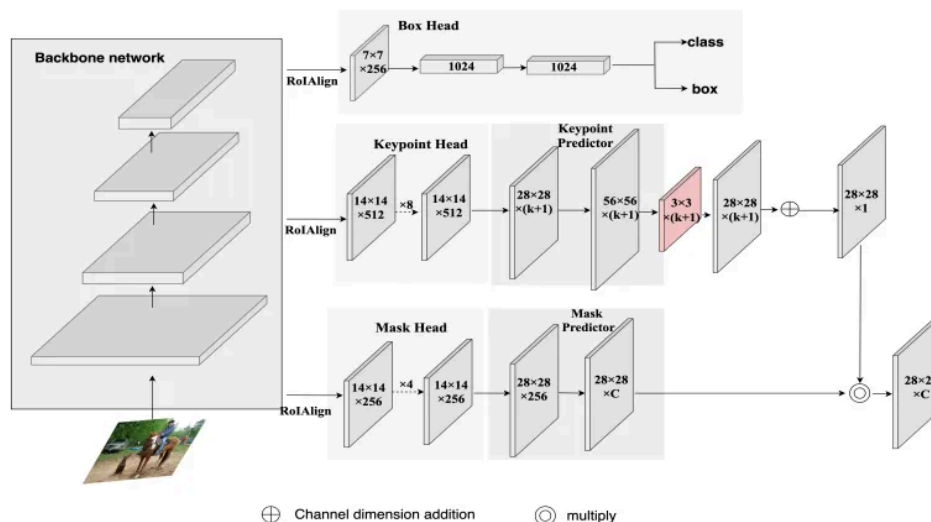


Рисунок 2.16 – Архітектура Keypoint-RCNN

Під час навчання Keypoint-RCNN вчиться одночасно передбачати обмежувальні рамки об'єктів та розташування їх ключових точок. Це досягається за допомогою багатозадачної функції втрат, яка поєднує в собі втрати локалізації для регресії обмежувальних рамок та втрати локалізації ключових точок. Модель навчається на анотованих наборах даних, де для кожного екземпляра об'єкта надаються як анотації обмежувальних рамок, так і анотації ключових точок.

Під час інференції Keypoint-RCNN бере на вхід зображення і генерує обмежувальні рамки навколо виявлених об'єктів, разом з координатами їх ключових точок.

2.4 Проектування системи

Система діагностування захворювань опорно-рухового апарату складається з ряду ключових компонентів, що співпрацюють для швидкого та ефективного аналізу зображень людей.

Система діагностики захворювань опорно-рухового апарату ставить перед собою завдання знаходження ключових точок на тілі людини, що є етапом для подальшого аналізу та діагностики. Цей процес вимагає високої точності та надійності, оскільки правильність визначення цих точок впливає на точність і об'єктивність діагнозу.

Нейромережа, яка входить до складу системи, відіграє ключову роль у визначенні цих ключових точок. Вона навчається на деякій кількості зображень людського тіла, щоб впізнавати різноманітні анатомічні особливості та визначати точки інтересу для подальшого аналізу. Цей процес навчання дозволяє нейромережі адаптуватися до різних типів, забезпечуючи стабільну та ефективну роботу системи в різних клінічних ситуаціях.

Коли ключові точки на тілі людини визначені, інформація передається до інтерфейсу користувача, де медичний фахівець може провести подальший аналіз та встановити діагноз. Інтерфейс користувача надає зручні інструменти для візуалізації та обробки даних, а також для взаємодії з іншими системними компонентами, які сприяють точній та швидкій діагностиці.

Для візуалізації було побудовано діаграму компонентів даної системи, яку зображено на рисунку 2.17. Діаграма компонентів системи діагностики захворювань опорно-рухового апарату відображає взаємозв'язок між основними складовими системи та їх функціональність. Основні компоненти системи включають камеру для знімків, обробник знімків, нейромережу та інтерфейс користувача.

Ця діаграма компонентів ілюструє, як кожен елемент системи взаємодіє між собою для досягнення головної мети системи – визначення ключових точок на тілі людини для подальшого аналізу та діагностики захворювань опорно-рухового апарату.

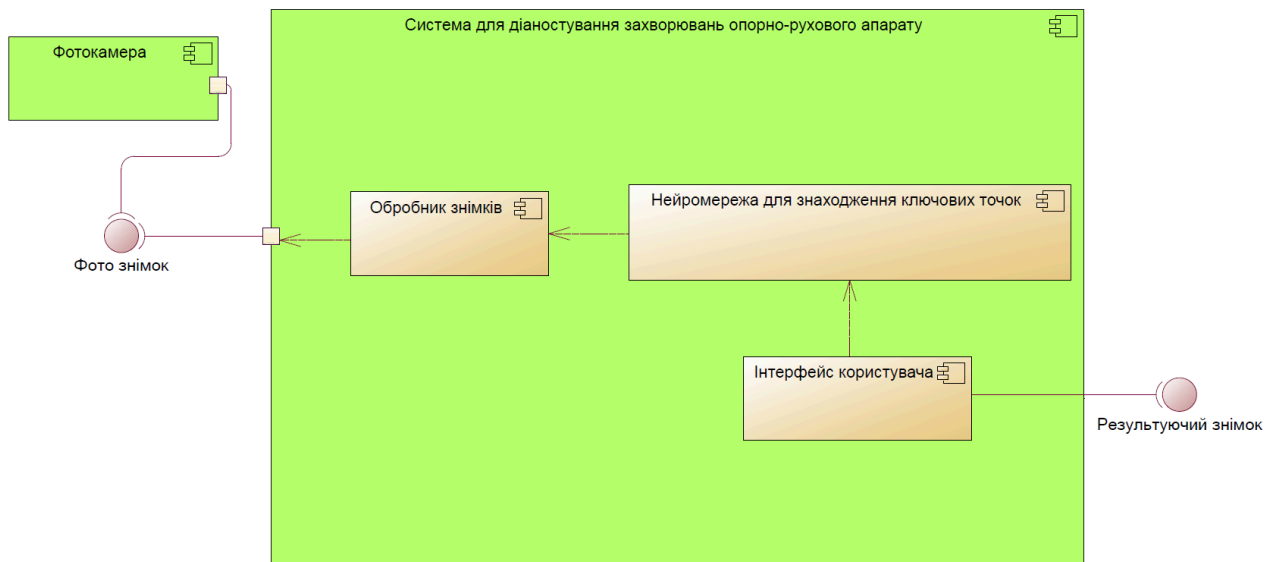


Рисунок 2.17 – Діаграма компонентів для досліджуваної системи

Система діагностики захворювань опорно-рухового апарату має наступні компоненти:

- **Фотокамера.** Цей компонент забезпечує отримання зображень тіла пацієнта. Він може бути у вигляді фотокамери або спеціалізованого обладнання для зйомки з оптимальною роздільною здатністю та якістю зображення.
- **Обробник знімків.** Цей модуль відповідає за обробку отриманих зображень, щоб забезпечити стандартизацію та відокремлення ключових даних. Він може виконувати операції попередньої обробки, такі як вирівнювання, видалення шуму та корекція освітлення.
- **Нейромережа** для визначення ключових точок на тілі людини. Ця складова відповідає за визначення анатомічних ключових точок на зображеннях тіла. Вона може бути навчена розпізнавати такі ключові точки, як суглоби, хребет, кінцівки тощо, що допомагає в подальшому аналізі стану пацієнта.
- **Інтерфейс користувача.** Ця складова надає зручний інтерфейс для взаємодії з системою для медичних фахівців. Він може включати в себе можливості

завантаження зображень, візуалізації результатів, вибору аналітичних параметрів та навігації між різними функціями системи.

Ці компоненти співпрацюють, щоб забезпечити швидку, точну та зручну діагностику захворювань опорно-рухового апарату, сприяючи покращенню медичного догляду та результатів лікування пацієнтів.

Загалом, система діагностики захворювань опорно-рухового апарату інтегрує в собі ряд технологічних рішень, спрямованих на автоматизацію та покращення процесу діагностики, що допомагає медичним фахівцям ефективно виявляти та лікувати захворювання пацієнтів.

Формалізація на теоретико-множинному рівні (морфологічний опис)

Формалізація на теоретико-множинному рівні, також відома як морфологічний опис, є методологією визначення складових системи та їх взаємозв'язків через концепцію множин. У цьому підході кожен елемент або компонент системи представлений як окрема множина, а взаємодія між ними виражена взаємними залежностями між цими множинами.

Модель автоматизованої системи діагностування захворювань опорно-рухового апарату в теоретико-множинному вигляді:

$$DMS = \langle X, P, Y, F_0, \varphi, \psi \rangle,$$

де DMS – Система діагностування захворювань опорно-рухового апарату,

X – вхідні дані про об'єкт, що досліджується, а саме – зображення людей зроблені зі спини;

$P = \{s_i\}_{i=1,3}$ – навчальні вибірки, що регламентують процес визначення

ключових точок нейронною мережею:

- p_1 – зображення людей
- p_2 – ключові точки;
- p_3 – зв'язок зображення і визначених ключових точок;

$Y = \{y_i\}_{i=1,2}$ – дані, що отримуються на виході процесу, де:

- y_1 – ключові точки;
- y_2 – точність визначення.

$Fo = \{fo_i\}_{i=1,2}$ – фактори зовнішнього середовища;

- fo_1 – масштаб зображення на вході;
- fo_2 – правильне визначення ключових точок на навчальних даних.

ϕ – функція корегування навчальної вибірки;

ψ – функція класифікації та формування даних на виході процесу.

Тоді функції корегування і формування вихідних даних можна подати наступним чином:

$$\forall (x, s) \in X \times P \mid \phi: X \times P \rightarrow P$$

$$\forall (x, s) \in X \times P \mid \psi: X \times P \rightarrow Y,$$

що означає що множина P буде змінювати залежності від надходження нових знімків людей, або визначення нових, необхідних для діагностування, ключових точок на тілі. В результаті множина вихідних даних Y також зміниться.

3. ПРОЕКТНІ РІШЕННЯ. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Проектні рішення з програмного забезпечення

Python – високорівнева мова програмування загального призначення, що відзначається простотою і зручністю використання. Вона була створена Гвідо ван Россумом і вперше випущена в 1991 році. Python має простий і читаємий синтаксис, що дозволяє розробникам швидко писати та розуміти код. Це робить мову ідеальною для початківців і дозволяє більш досвідченим програмістам зосередитися на вирішенні проблем, а не на особливостях синтаксису [16].

Python підтримує різні парадигми програмування, включаючи об'єктно-орієнтоване, процедурне і функціональне програмування, що робить його універсальним інструментом для розробки різних типів застосунків. Мова постачається з великою стандартною бібліотекою, яка забезпечує широкий набір функціональних можливостей, включаючи роботу з файлами, мережею, інтерфейсом операційної системи, регулярними виразами та багатьма іншими задачами, що дозволяє розробникам вирішувати більшість задач без необхідності встановлювати додаткові пакети.

Python легко інтегрується з іншими мовами програмування і може бути використаний для написання скриптів, які взаємодіють з кодом на C, C++, Java, та інших мовах, що дозволяє створювати високоефективні додатки, поєднуючи переваги різних мов програмування. Мова працює на більшості сучасних операційних систем, включаючи Windows, macOS, Linux, і навіть мобільні платформи, що забезпечує велику гнучкість у виборі середовища розробки та розгортання додатків.

Python має величезну і активну спільноту розробників, що означає наявність підтримки, документації, навчальних матеріалів, а також численних бібліотек і фреймворків, розроблених спільнотою для розширення функціональності мови. Завдяки своїм характеристикам і широким можливостям, Python став однією з найбільш популярних і затребуваних мов програмування в світі, і його популярність продовжує зростати.

Python широко використовується в різних сферах. Завдяки фреймворкам, таким як Django і Flask, він є популярним інструментом для створення веб-додатків і сайтів. Бібліотеки, такі як NumPy, SciPy, Pandas і Matplotlib, роблять Python ідеальним для наукових досліджень, обробки даних і візуалізації. Python часто використовується для написання скриптів, автоматизації рутинних задач і адміністрування систем. Через свою простоту і читабельність, Python часто використовується як перша мова програмування в освітніх установах.

Python також є однією з провідних мов для побудови і навчання нейронних мереж, завдяки своїй зручності та потужним бібліотекам, такими як TensorFlow, Keras і PyTorch. Одним з важливих напрямків застосування нейронних мереж є комп'ютерний зір, де Python є надзвичайно популярним через свою здатність інтегруватися з бібліотеками для обробки зображень і відео, такими як OpenCV і PIL.

YOLO (You Only Look Once) є однією з найефективніших і найшвидших моделей для виявлення об'єктів у зображеннях і відео. Модель YOLO була вперше представлена Джозефом Редмоном і його колегами, і вона значно відрізняється від традиційних підходів до виявлення об'єктів. Основна ідея YOLO полягає в тому, щоб розглядати виявлення об'єктів як задачу регресії до просторових розташувань об'єктів і класів, і здійснювати це за допомогою єдиної нейронної мережі в один прохід. YOLO здатна обробляти зображення в режимі реального часу, завдяки своїй архітектурі, яка дозволяє моделі робити прогнози для всіх об'єктів у зображенні одночасно, що дозволяє досягати високої продуктивності. Модель забезпечує високу точність виявлення об'єктів навіть на складних зображеннях, завдяки розбиттю зображення на сітку і прогнозуванню обмежувальних рамок і ймовірностей класів, що дозволяє ефективно розпізнавати і локалізувати об'єкти. Використовуючи єдину нейронну мережу для всіх етапів обробки, YOLO спрощує реалізацію і налаштування моделі, роблячи її зручною для інтеграції в різні застосунки [17].

Python є ідеальною мовою для навчання і використання моделей YOLO завдяки своїм бібліотекам і фреймворкам. Бібліотеки, такі як OpenCV, забезпечують інструменти для обробки та анотації зображень, що є необхідним етапом підготовки

даних для навчання моделі YOLO. Бібліотеки TensorFlow і PyTorch забезпечують необхідні інструменти для створення, навчання і тестування моделей YOLO. Зокрема, ці фреймворки підтримують GPU-акселерацію, що значно прискорює процес навчання. Python дозволяє легко налаштувати і тренувати моделі YOLO на великих наборах даних, а завдяки потужним інструментам для обробки даних, таким як Pandas і NumPy, розробники можуть ефективно підготувати дані для навчання моделі. Python також забезпечує просту інтеграцію моделі YOLO у веб-додатки, мобільні додатки або системи реального часу. Крім того, Python забезпечує інструменти для оптимізації та налаштування моделей YOLO, такі як Hyperopt і Optuna, що допомагає розробникам знаходити оптимальні гіперпараметри для підвищення продуктивності моделі.

YOLOv8 – це остання версія алгоритму YOLO, яка перевершує попередні версії, вводячи різні модифікації, такі як просторова увага, об'єднання функцій і модулі агрегації контексту.

Якщо розглядати YOLOv8 у порівнянні зі минулими версіями, то можна виділити деякі ключові особливості:

- Покращена точність : YOLOv8 покращує точність виявлення об'єктів порівняно зі своїми попередниками за рахунок впровадження нових методів та оптимізації.
- Підвищена швидкість : YOLOv8 забезпечує вищу швидкість висновку, ніж інші моделі виявлення об'єктів, зберігаючи високу точність.
- Кілька магістралей : YOLOv8 підтримує різні магістралі, такі як EfficientNet, ResNet і CSPDarknet, що дає користувачам можливість вибрати найкращу модель для конкретного випадку використання.
- Адаптивне навчання : YOLOv8 використовує адаптивне навчання для оптимізації швидкості навчання та збалансування функції втрат під час навчання, що призводить до кращої продуктивності моделі.

- Advanced-Data Augmentation : YOLOv8 використовує вдосконалені методи розширення даних, такі як MixUp і CutMix, щоб покращити надійність і узагальнення моделі.
 - Архітектура, що налаштовується : Архітектура YOLOv8 легко настроюється, що дозволяє користувачам легко змінювати структуру та параметри моделі відповідно до своїх потреб.
 - Попередньо навчені моделі : YOLOv8 надає попередньо навчені моделі для легкого використання та передачі знань на різних наборах даних
- Архітектура YOLOv8 базується на попередніх версіях алгоритмів YOLO .

YOLOv8 використовує згортку нейронної мережі, яку можна розділити на дві основні частини: магістраль і голову. На рисунку 3.1 зображено стурктуру мережі YOLOv8.

в голові мережі, що дозволяє моделі зосереджуватися на різних частинах зображення і регулювати важливість різних ознак залежно від їх відповідності завданню.

Ще однією важливою особливістю YOLOv8 є її здатність виконувати багатомасштабне виявлення об'єктів. Модель використовує мережу піраміди функцій для виявлення об'єктів різних розмірів і масштабів на зображенні. Ця мережа піраміди функцій складається з кількох шарів, що виявляють об'єкти на різних масштабах, що дозволяє моделі розпізнавати як великі, так і малі об'єкти на зображенні.

Завдяки своїй універсальності, потужним бібліотекам і активній спільноті, Python є ідеальним інструментом для розробки і впровадження моделей нейронних мереж YOLO, для різних застосувань у галузі комп'ютерного зору. Це робить Python незамінним інструментом для розробників, які працюють над інноваційними проектами в області штучного інтелекту і машинного навчання.

3.2 Збір даних для навчання

Збір даних є критично важливою складовою будь-якого дослідження або проекту в галузі комп'ютерного зору та машинного навчання. Існують декілька датасетів з ключовими точками на тілі людини, які використовуються для розпізнавання постави, відслідковування руху та розуміння контексту в комп'ютерному зорі та машинному навчанні. Ось кілька найвідоміших:

- MPII Human Pose Dataset: Зберігає понад 25 тисяч зображень з анотованими ключовими точками, представляючи людей у різних позах та сценаріях.
- COCO (Common Objects in Context): Містить зображення людей з анотаціями ключових точок різних частин тіла.
- PoseTrack: Включає відео та зображення людей з анотаціями ключових точок, дозволяючи відслідковувати поставу в реальному часі.

Ці набори даних використовуються в дослідженнях та застосуваннях, таких як аналіз руху, відслідковування осіб на відео та розпізнавання активності.

Проте, для навчання нашої системи діагностики захворювань опорно-рухового апарату необхідно створити власний датасет. Позаяк, представлені датасети містять дійсно велику кількість навчальних даних, мінусом їх використання у сфері оцінки постави людини є обмежена кількість і попередня визначеність ключових точок на тілі людини, що не підходить для задачі оцінки постави. На рисунку 3.2 зображено ключові точки які визначені у датасеті.

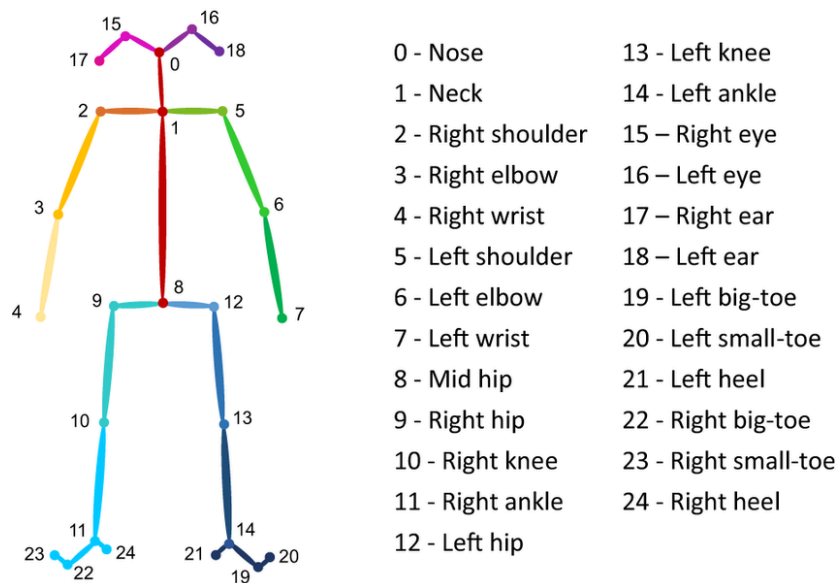


Рисунок 3.2 – Ключові точки у OpenPose датасеті

Створення власного датасету включатиме в себе процес збору зображень та анотацію їх за допомогою кастомних точок, що відповідають ключовим анатомічним місцям на тілі. Через обмежений час та можливості, щодо збору даних датасет буде стантартизований щодо знімків.

Для збору фото було визначено наступний алгоритм.

- 1) Заповнити анкету пацієнта.
- 2) Роздягти пацієнта до нижньої білизни.
- 3) Розмістити телефон на штативі на висоті 2/3 висоти пацієнта, на відстані 1,6 м та паралельно до пацієнта
- 4) Виставити рівень на штативі.
- 5) Виставити горизонту рівень на камері телефона. Допустиме відхилення $\pm 0.2^\circ$.

На рисунку 3.3 схематично зображено положення ключових елементів процесу.



Рисунок 3.3 – Процес збору знімків

Після збору знімків, наступним етапом є анотація зібраних даних. Анотація фото - це процес визначення та маркування об'єктів та їх характеристик на зображенні. У контексті нашого проекту, анотація фото включатиме в себе визначення та позначення кастомних ключових точок на тілі людини, які необхідно використовувати для оцінки постави.

У якості необхідних точок було визначено 3 набори: краї плеч, талія та стегна (рис. 3.4).

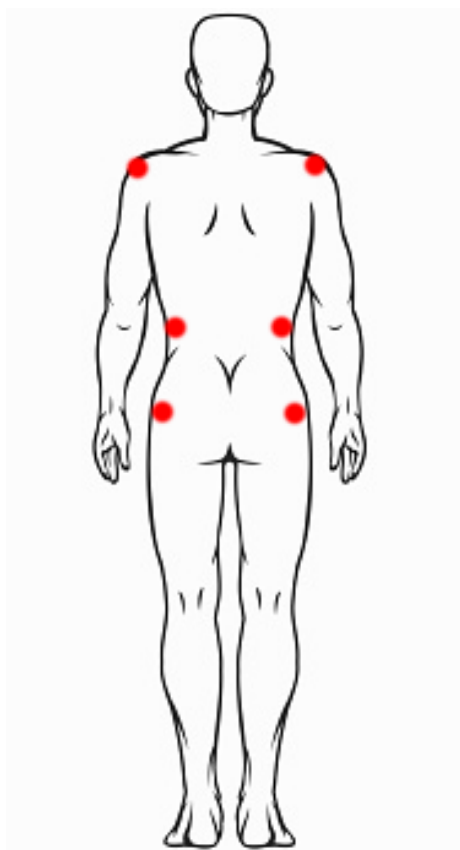


Рисунок 3.4 – Визначені ключові точки на тілі людини

У якості сервісу для анотації даних було обрано онлайн платформу CVAT(рис 3.5). Це платформа для анотації зображень та відео, яка підтримує різні типи анотацій і має великий набір функцій, включаючи можливість спільної роботи над анотаціями декількох користувачів. Також плюсом роботи з цим застосунком є можливість збереження файлів з анотацією одразу у форматі YOLO.

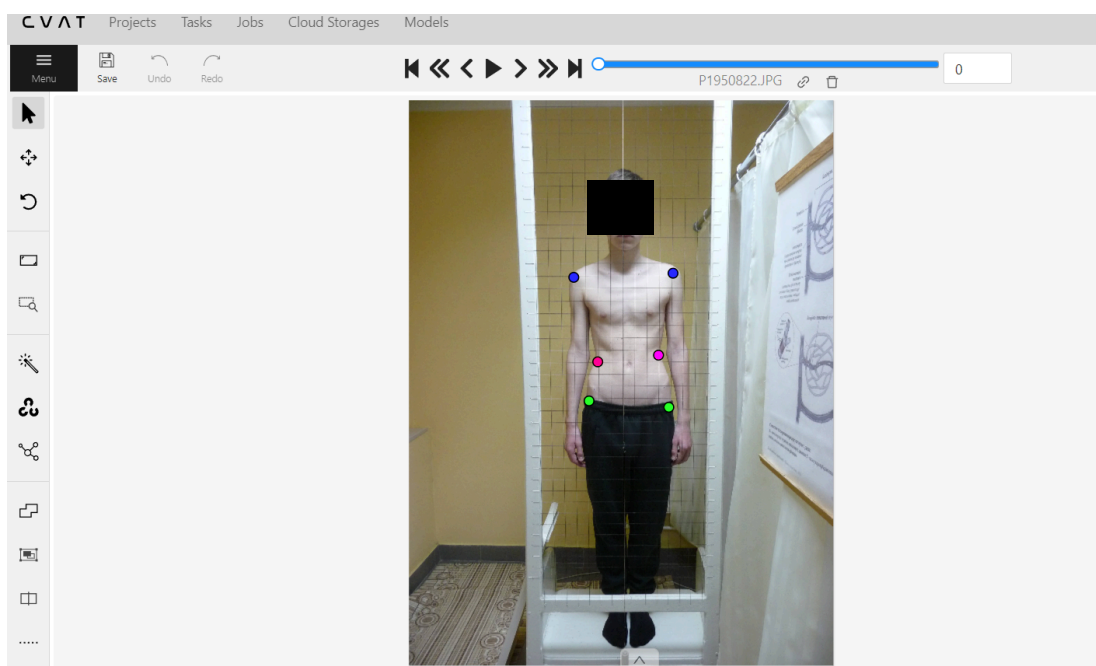


Рисунок 3.5 – Інтерфейс застосунку CVAT

У форматі YOLO для кожного об'єкта у кожному зображенні зберігається наступна інформація:

- Клас об'єкта: Ідентифікатор класу об'єкта (наприклад, "собака", "автомобіль", "людина" тощо).
- Координати верхнього лівого кута: (x, y) координати верхнього лівого кута рамки, що обмежує об'єкт.
- Координати нижнього правого кута: (x, y) координати нижнього правого кута рамки, що обмежує об'єкт.
- Інші параметри (необов'язково): Додаткові параметри, такі як впевненість (confidence) у присутності об'єкта або його розмір.
- Для ключових точок це просто (x, y) та видімість точки.

Структура датасету для навчання YOLO зазвичай включає в себе зображення та відповідні анотації для об'єктів, які потрібно виявляти за допомогою моделі YOLO.

- Зображення: Набір зображень, на яких потрібно виявляти об'єкти. Ці зображення можуть бути у форматі JPEG, PNG або іншому підтримуваному форматі.
- Анотації: Для кожного зображення містяться анотації, які вказують положення та класи об'єктів на зображенні. Формат анотацій може відповідати формату, який я описував раніше для ключових точок у розрізі YOLO.
- Класифікаційний файл: Файл, який містить список класів, які модель повинна розпізнавати, особливості щодо розширення даних та інше.

Загальний набір даних потрібно розбити на 3 вибірки – навчальну, валідаційну, тестову. На рисунку 3.6 зображено структуру датасету.

Ім'я	Дата змінення	Тип	Розмір
test	24.05.2024 16:53	Папка файлів	
train	24.05.2024 16:53	Папка файлів	
valid	24.05.2024 16:53	Папка файлів	
data.yaml	01.05.2024 12:59	Yaml Source File	1 КБ

Рисунок 3.6 – Структура датасету

Кожна з папок містить у собі 2 підпапки – images, у якій зберігаються фото, labels, у якій збережені текстові файли з розміткою ключових точок.

Файл, data.yaml містить інформацію про класи збережені у датасеті а також дані для алгоритмів розширення даних, у намошу випадку нумерацію точок при відзрєккаювання зображення по вертикалі, його вигляд зображено на рисунку 3.7

```

E: > Downloads > dataset > ! data.yaml
1  train: ../train/images
2  val: ../valid/images
3  test: ../test/images
4
5  kpt_shape: [6, 3]
6  flip_idx: [0, 1, 2, 3, 4, 5]
7
8  nc: 1
9  names: ['Person']
10 |

```

Рисунок 3.7 – Вигляд конфігураційного файлу

Отриманий датасет буде використовуватися для навчання моделі YOLOv8 для визначення пози та оцінки постави людини з метою діагностики захворювань опорно-рухового апарату.

3.3 Опис класів

Пакет Ultralytics – це набір інструментів для комп'ютерного зору, який надає широкі можливості для розробки та реалізації проектів з використанням нейронних мереж, зокрема YOLO. Він створений з метою полегшення процесу розробки, тренування та оцінки моделей машинного навчання для завдань виявлення об'єктів.

Найбільш важливий модуль пакету – це YOLOv8, який є підсистемою YOLO, розробленою спеціально для виявлення об'єктів в реальному часі. Ultralytics також

надає інструменти для оцінки моделей, візуалізації результатів, аналізу даних, а також взаємодії з відеопотоками та зображеннями.

YOLOv8 (You Only Look Once, version 8) є останньою версією популярної архітектури нейронних мереж для виявлення об'єктів. Ця архітектура продовжує традиції своїх попередників, забезпечуючи високу швидкість і точність виявлення об'єктів на зображеннях в реальному часі[18]. Класи в YOLOv8 можна розділити на основні категорії залежно від їх функціональних можливостей та використання:

– **Model** (Модель):

YOLOv8n (Nano): Дуже компактна модель, призначена для роботи на пристроях з обмеженими ресурсами.

YOLOv8s (Small): Менша модель з балансом між швидкістю і точністю.

YOLOv8m (Medium): Середня модель, яка пропонує кращу точність, ніж "Small", при трохи більших вимогах до ресурсів.

YOLOv8l (Large): Велика модель, орієнтована на високу точність і продуктивність.

YOLOv8x (Extra Large): Найбільша і найточніша модель у лінійці, яка потребує найбільших ресурсів.

– **Components** (Компоненти):

Backbone (Основна мережа): Відповідає за екстракцію ознак із зображення. У YOLOv8 використовується вдосконалена архітектура, яка може включати CSP (Cross Stage Partial) блоки для покращення продуктивності.

Neck (Шия): Зв'язує основну мережу з головою і включає шари, які допомагають злиттю інформації з різних рівнів ієрархії ознак. Це допомагає виявляти об'єкти різних розмірів.

Head (Голова): Виконує остаточне прогнозування класу об'єкта і його межової коробки. Включає шари для передбачення координат об'єктів і їх класів.

– **Anchors** (Якорі):

Використовуються для допомоги в прогнозуванні обмежувальних рамок. У YOLOv8 можлива адаптація і налаштування якорів для кращої роботи з конкретними наборами даних.

– **Loss Functions** (Функції втрат):

Функції втрат використовуються для тренування моделі і включають в себе компоненти для регресії координат, класифікації та відповідності якорів. Вдосконалені функції втрат в YOLOv8 сприяють кращій точності виявлення об'єктів.

– **Augmentations** (Аугментації):

Різноманітні техніки збільшення даних, що використовуються під час тренування для підвищення загальної продуктивності моделі, такі як випадкові повороти, масштабування, обрізка та зміна яскравості.

– **Post-Processing** (Пост-обробка):

Включає методи, такі як Non-Maximum Suppression (NMS), які використовуються для зменшення кількості перекриття межових коробок і вибору найкращих прогнозів.

– **Training Pipeline** (Тренувальний конвеєр):

Процес тренування моделі включає в себе підготовку даних, налаштування гіперпараметрів, тренування та валідацію. YOLOv8 надає інструменти для полегшення цього процесу та оптимізації моделі.

Загалом, YOLOv8 є потужною і гнучкою архітектурою для виявлення об'єктів, що поєднує високу продуктивність і точність з можливістю налаштування під різні задачі і обмеження апаратних ресурсів.

3.4 Тестовий приклад

У якості середовища виконання було застосовано Google Colab. На рисунку 3.8 зображено основні ресурсні характеристики хмарного середовища.

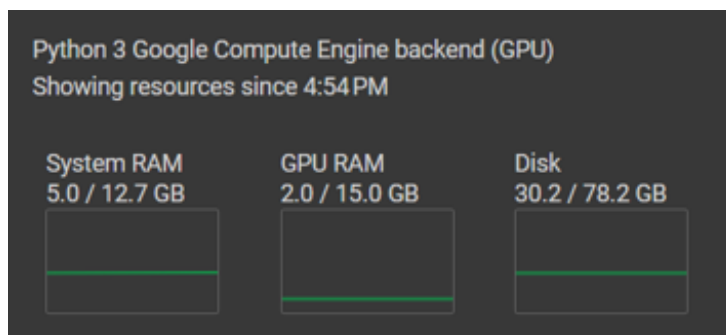


Рисунок 3.8 – Ресурси Google Colab

У якості тренувального датасету використовувався отриманий датасет, який містить 70 зображень та анотаційні файли. Тренувальну і валідаційну вибірки розподілено у 90% та 10%. На рисунку 3.9 зображено приклади тренувальних зображень.

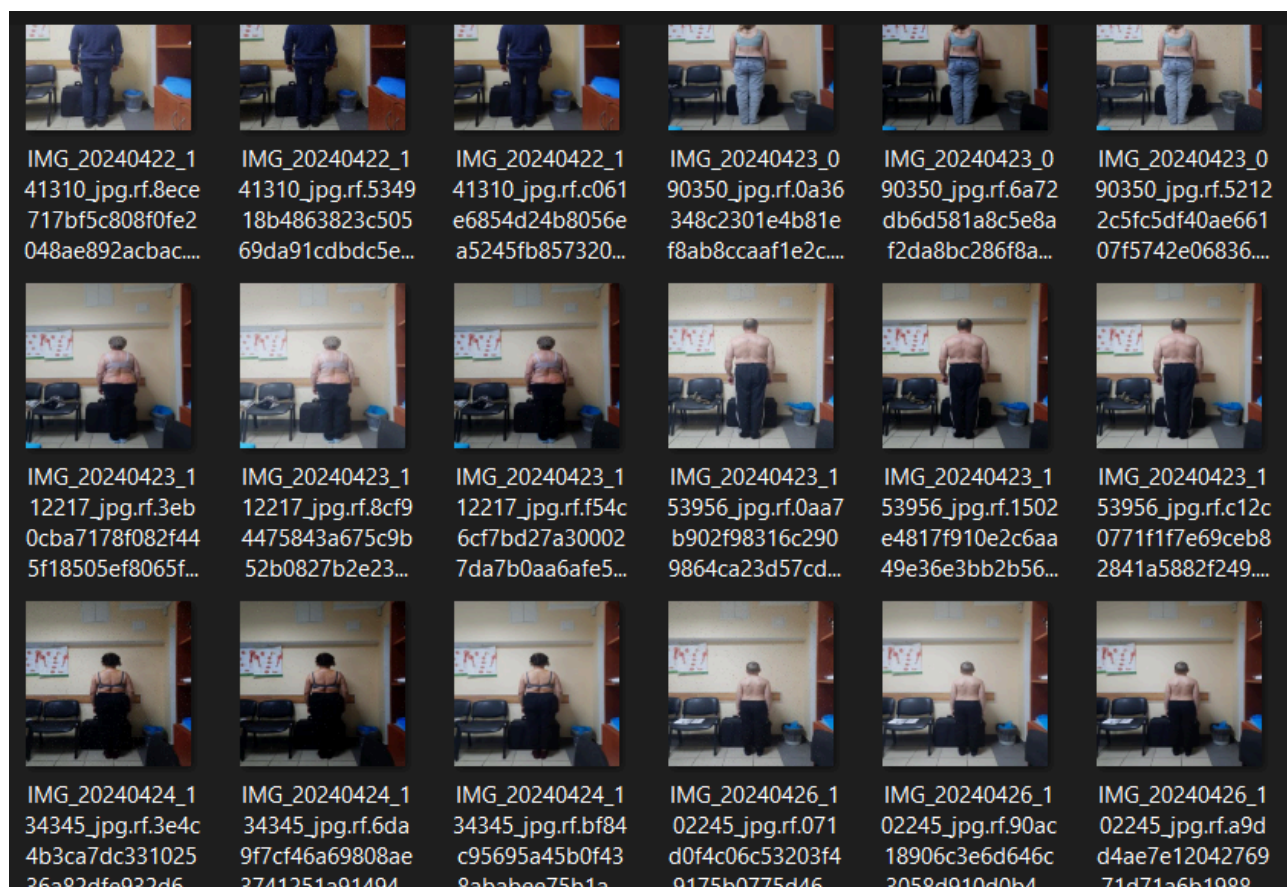


Рисунок 3.9 – Тренувальні зображення

Використовувалась модель YOLOv8l-pose, яка була налаштована з 390 шарами та 44482761 параметрами. Ця модель призначена для виявлення поз людей, і вона була налаштована для роботи з 6 ключовими точками та використовує 169.1 GFLOPs для своєї роботи. На рисунку 3.10 зображено усі складові обраної моделі.

	from	n	params	module	arguments	
0		-1	1	1856	ultralytics.nn.modules.conv.Conv	[3, 64, 3, 2]
1		-1	1	73984	ultralytics.nn.modules.conv.Conv	[64, 128, 3, 2]
2		-1	3	279808	ultralytics.nn.modules.block.C2f	[128, 128, 3, True]
3		-1	1	295424	ultralytics.nn.modules.conv.Conv	[128, 256, 3, 2]
4		-1	6	2101248	ultralytics.nn.modules.block.C2f	[256, 256, 6, True]
5		-1	1	1180672	ultralytics.nn.modules.conv.Conv	[256, 512, 3, 2]
6		-1	6	8396800	ultralytics.nn.modules.block.C2f	[512, 512, 6, True]
7		-1	1	2360320	ultralytics.nn.modules.conv.Conv	[512, 512, 3, 2]
8		-1	3	4461568	ultralytics.nn.modules.block.C2f	[512, 512, 3, True]
9		-1	1	656896	ultralytics.nn.modules.block.SPPF	[512, 512, 5]
10		-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
11		[-1, 6]	1	0	ultralytics.nn.modules.conv.Concat	[1]
12		-1	3	4723712	ultralytics.nn.modules.block.C2f	[1024, 512, 3]
13		-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
14		[-1, 4]	1	0	ultralytics.nn.modules.conv.Concat	[1]
15		-1	3	1247744	ultralytics.nn.modules.block.C2f	[768, 256, 3]
16		-1	1	590336	ultralytics.nn.modules.conv.Conv	[256, 256, 3, 2]
17		[-1, 12]	1	0	ultralytics.nn.modules.conv.Concat	[1]
18		-1	3	4592640	ultralytics.nn.modules.block.C2f	[768, 512, 3]
19		-1	1	2360320	ultralytics.nn.modules.conv.Conv	[512, 512, 3, 2]
20		[-1, 9]	1	0	ultralytics.nn.modules.conv.Concat	[1]
21		-1	3	4723712	ultralytics.nn.modules.block.C2f	[1024, 512, 3]
22		[15, 18, 21]	1	6435721	ultralytics.nn.modules.head.Pose	[1, [6, 3], [256, 512, 512]]

YOLOv8l-pose summary: 390 layers, 44482761 parameters, 44482745 gradients, 169.1 GFLOPs

Рисунок 3.10 – Компоненти архітектури YOLOv8l-pose

Задано налаштування параметрів для навчання моделі з використанням даних з файлу data.yaml, використовуючи задану кількість епох (500), розмір батчів (16), розмір зображення (640x640).

Далі проведено підготовку до навчання. Спочатку проскановано дані навчання та валідації, створюється кеш для навчальних даних, а потім застосовуються методи аугментації зображень, такі як розмивання, медіанне розмивання, перетворення в відтінки сірого та адаптивна гістограма рівномірності контрасту (CLAHE). Після цього визначається оптимальний оптимізатор (AdamW) з параметрами lr=0.002, momentum=0.9 та розпочинається процес навчання моделі на протязі 500 епох. Цей процес зображено на рисунку 3.11.

```

train: Scanning /content/dataset/train/labels... 63 images, 0 backgrounds, 0 corrupt: 100%|██████████| 63/63 [00:00<00:00, 1120.24it/s]train: New ca
albuementations: Blur(p=0.01, blur_limit=(3, 7)), MedianBlur(p=0.01, blur_limit=(3, 7)), ToGray(p=0.01), CLAHE(p=0.01, clip_limit=(1, 4.0), tile_grid
/usr/lib/python3.10/multiprocessing/popen_fork.py:66: RuntimeWarning: os.fork() was called. os.fork() is incompatible with multithreaded code, and I
self.pid = os.fork()
val: Scanning /content/dataset/valid/labels... 7 images, 0 backgrounds, 0 corrupt: 100%|██████████| 7/7 [00:00<00:00, 1398.23it/s]val: New cache cre

Plotting labels to runs/pose/train/labels.jpg...
optimizer: 'optimizer=auto' found, ignoring 'lr0=0.01' and 'momentum=0.937' and determining best 'optimizer', 'lr0' and 'momentum' automatically...
optimizer: AdamW(lr=0.002, momentum=0.9) with parameter groups 103 weight(decay=0.0), 113 weight(decay=0.0005), 112 bias(decay=0.0)
TensorBoard: model graph visualization added ✓
Image sizes 640 train, 640 val
Using 2 dataloader workers
Logging results to runs/pose/train
Starting training for 500 epochs...

```

Риснок 3.11 – Підготовка до навчання

Процес навчання зайняв близько 50 хвилин. Інформація про останні епохи навчання зображено на рисунку 3.12.

Epoch	GPU_mem	box_loss	pose_loss	kobj_loss	cls_loss	dfl_loss	Instances
498/500	11.4G	0.08724	0.04048	0.005137	0.1065	0.7498	15
	Class	Images	Instances	Box(P)	R	mAP50	mAP50-95)
499/500	11.5G	0.1038	0.0406	0.004766	0.1096	0.7495	15
	Class	Images	Instances	Box(P)	R	mAP50	mAP50-95)
500/500	11.4G	0.08677	0.03855	0.00519	0.09911	0.7707	15
	Class	Images	Instances	Box(P)	R	mAP50	mAP50-95)

500 epochs completed in 0.821 hours.
 Optimizer stripped from runs/pose/train/weights/last.pt, 89.5MB
 Optimizer stripped from runs/pose/train/weights/best.pt, 89.4MB

Рисунок 3.12 – Кінець навчання мережі

За попередньою оцінкою середня точність визначення обмежувальних рамок 99.5%, знаходження ключових точок 87.9%. На рисунку 3.13 зображено діаграми з основними метриками навчальної моделі.

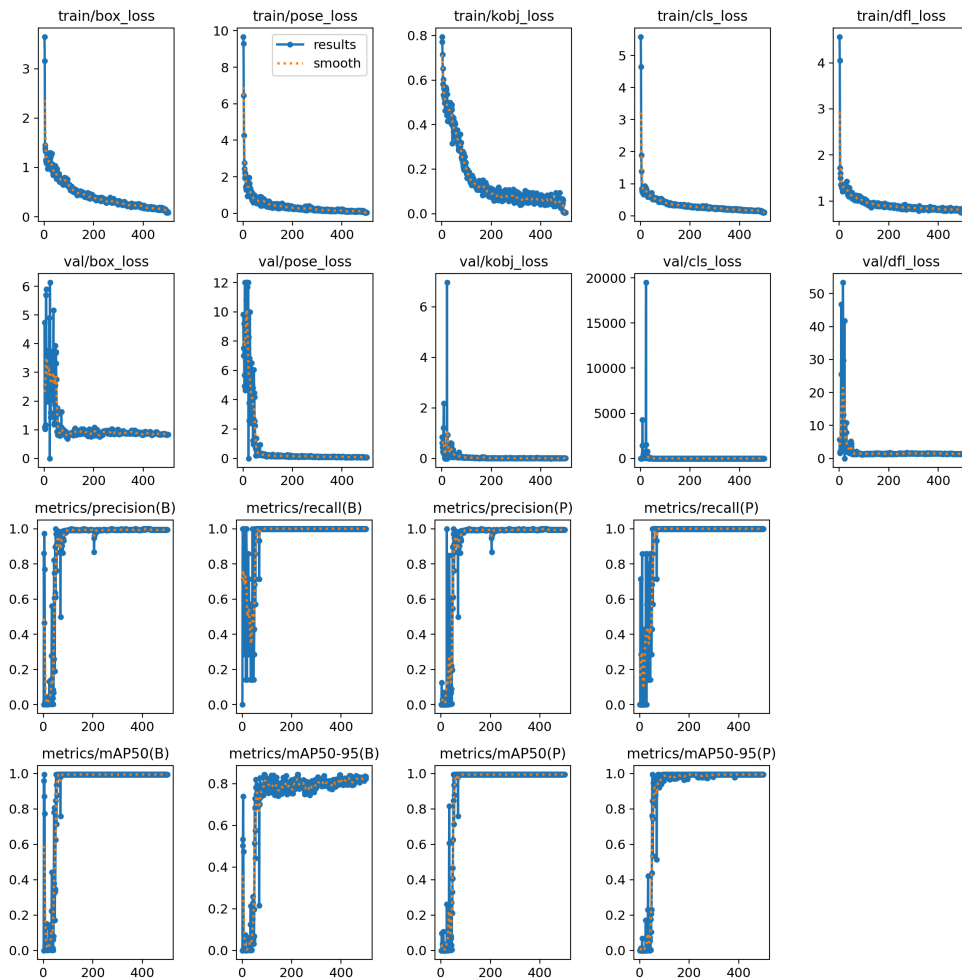


Рисунок 3.13 – Метрики навчальної моделі

На рисунку 3.14 зображено знімки з валідаційної вибірки, а також результат передбачення мережі.



Вхідні зображення

Передбачення мережі

Рисунок 3.14 – Передбачення на валідаційній вибірці

Також було залишено зображення які не приймали участь у навчанні, для переревірки результатів мережі на таких знімках. Результат передбачення та тестовому знімку зображено на рисунку 3.15.

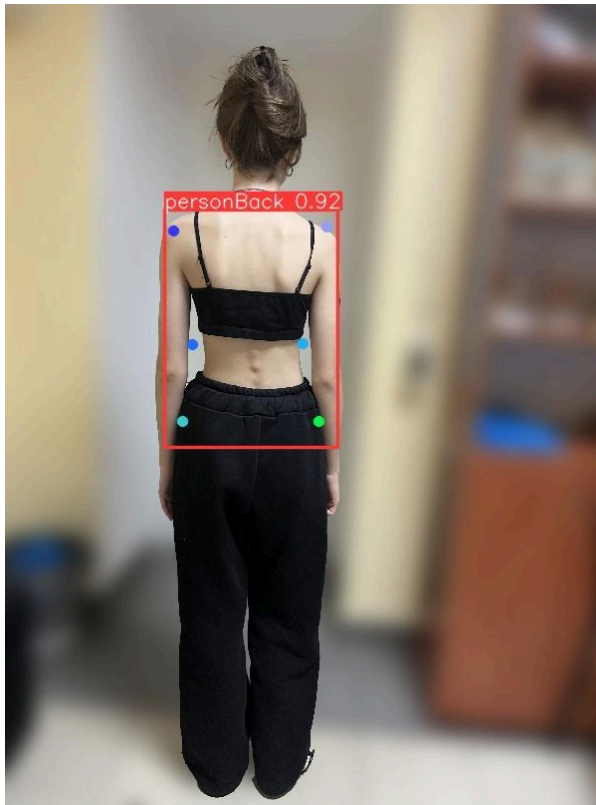


Рисунок 3.15 – Тестовий знімок

Зважаючи на невеликий розмір навчальних даних, можна сказати, що результати навчання мережі задовільні, та даний спосіб знаходження ключових точок та тілі людини ефективний. Даний спосіб має перспективи у разі збільшення розміру навчальної вибірки, тоді можна добитися кращих узагальнювальних здібностей мережі а також підвищити точність знаходження точок.

4. ЕРГОНОМІКА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

4.1 Визначення ергономіки в контексті ІТ

Ергономіка (від грец. ἔργον - робота + νόμος – закон) – в традиційному розумінні – наука про пристосування посадових обов'язків, робочих місць, предметів і об'єктів праці, а також комп'ютерних програм для найбільш безпечного та ефективного праці працівника, виходячи з фізичних і психічних особливостей людського організму[19].

В межах ергономіки існують напрямки, які детально вивчають конкретні особливості людини та її взаємодію з різними системами. На сьогоднішній день можна виділити такі основні області ергономіки:

- **Фізична ергономіка** досліджує анатомічні, антропометричні, фізіологічні та біомеханічні характеристики і їхній вплив на фізичну діяльність людини. Ця галузь включає питання робочих поз, вантажно-розвантажувальних робіт, монотонних рухів, організації робочого місця, а також безпеки та здоров'я, зокрема, ризиків м'язово-скелетних порушень.
- **Когнітивна ергономіка** фокусується на розумових процесах, таких як сприйняття, пам'ять, мислення, моторна реакція і їх роль у взаємодії людини з різними системами. Вона вивчає розумове навантаження, процеси прийняття рішень, високо кваліфіковану роботу, взаємодію людини з комп'ютером, надійність людини, професійний стрес та підготовку.
- **Організаційна ергономіка** зосереджена на оптимізації соціотехнічних систем, включаючи організаційну структуру, політику та процеси. Її питання охоплюють комунікацію, управління трудовими ресурсами, планування діяльності, розклад робочого часу, колективну роботу, нові парадигми організації праці, віртуальні організації, дистанційну роботу та управління якістю.

В контексті інформаційних технологій ергономіка спрямована на розробку програмного забезпечення та апаратних засобів, які є зручними та ефективними для

користувача. Ця галузь охоплює численні аспекти, що стосуються взаємодії людини з комп'ютерами та іншими цифровими пристроями, забезпечуючи комфорт і підвищуючи ефективність роботи. Основні завдання ергономіки в ІТ включають:

1) Поліпшення користувацького досвіду (UX):

- Інтуїтивний інтерфейс: Розробка інтерфейсів, які є легкими для розуміння і використання без попереднього навчання.
- Доступність: Забезпечення можливості користування програмним забезпеченням для людей з різними фізичними можливостями, включаючи адаптацію для людей з порушеннями зору, слуху чи моторики.
- Відповідність очікуванням: Забезпечення того, що програми та системи працюють так, як очікують користувачі, мінімізуючи необхідність для них пристосовуватися до нових або незвичних способів роботи.

2) Зниження ризиків професійних захворювань:

- Ергономічні пристрої: Використання мишей, клавіатур та інших пристроїв, розроблених для зниження навантаження на м'язи та суглоби.
- Розподіл навантаження: Оптимізація робочих процесів, щоб уникнути надмірного використання певних м'язових груп, що може призвести до захворювань, таких як синдром карпального тунелю.
- Інтервали та вправи: Впровадження регулярних перерв і рекомендацій щодо вправ для підтримки фізичного здоров'я користувачів.

3) Підвищення продуктивності праці користувачів:

- Оптимізація робочих процесів: Створення програмного забезпечення, яке дозволяє користувачам виконувати свої завдання швидше і з меншими зусиллями.

- Зменшення помилок: Розробка систем, що мінімізують можливість людських помилок через більш зрозумілий та логічний інтерфейс.
- Навчання і підтримка: Надання користувачам доступних ресурсів для швидкого навчання та отримання підтримки, що дозволяє їм ефективніше використовувати програмне забезпечення.

Загалом, ергономіка в ІТ сприяє створенню робочих середовищ, які підтримують здоров'я, добробут і продуктивність користувачів, забезпечуючи більш зручний та ефективний досвід взаємодії з технологіями.

1.2 Значення ергономіки для медичних інформаційних систем

У медичних інформаційних системах ергономіка має вирішальне значення, оскільки від цього залежить точність і швидкість діагностики. Зручний і зрозумілий інтерфейс дозволяє лікарям швидше знаходити потрібну інформацію і приймати правильні рішення. Ергономічно розроблені системи знижують ймовірність помилок, які можуть мати серйозні наслідки для здоров'я пацієнтів. Крім того, зручні у використанні системи допомагають лікарям краще спілкуватися з пацієнтами, надаючи їм необхідну інформацію більш доступно. Це сприяє покращенню взаємодії з пацієнтами і забезпечує позитивний користувацький досвід, що високо оцінюється як медичним персоналом, так і пацієнтами.

4.2 Основні принципи ергономіки у програмному забезпеченні

Ергономіка у програмному забезпеченні базується на ряді принципів, які спрямовані на створення зручного, ефективного і безпечного продукту для користувачів. Одним з основних принципів є зрозумілість та простота. Інтерфейс має бути інтуїтивно зрозумілим, з мінімальною кількістю складних елементів, використовуючи прості, легко зрозумілі терміни і інструкції. Консистентність також є важливим аспектом, що передбачає єдність стилю та постійність елементів управління на всіх екранах, забезпечуючи однакові функції однаковими елементами управління.

Зворотній зв'язок є ще одним ключовим принципом, що включає негайний відгук на дії користувача та інформативність, тобто зрозумілі повідомлення про помилки та інструкції щодо їх виправлення. Гнучкість та адаптивність забезпечують можливість налаштування інтерфейсу під потреби конкретного користувача та коректну роботу на різних пристроях. Мінімізація навантаження включає зниження когнітивного навантаження, тобто мінімізацію кількості інформації, яку користувач повинен обробляти одночасно, та автоматизацію повторюваних завдань.

Безпека і конфіденційність також є невід'ємною частиною ергономіки програмного забезпечення. Захист даних пацієнтів забезпечується за допомогою шифрування та інших методів, а контроль доступу гарантує відповідний рівень доступу до різних типів інформації для різних користувачів.

Ергономічність у програмному забезпеченні також залежить від доступності, яка полягає у забезпеченні можливості використання продукту людьми з різними фізичними, когнітивними та сенсорними особливостями. Дотримання принципів доступності не лише робить програмне забезпечення інклюзивним, але й покращує загальний користувацький досвід, роблячи продукт зручним для всіх користувачів незалежно від їхніх можливостей та умов використання.

4.3 Підходи до проектування програмного забезпечення з урахуванням потреб користувачів медичних систем

Залучення користувачів на ранніх етапах розробки є ключовим підходом, що включає аналіз потреб через інтерв'ю та опитування кінцевих користувачів, а також створення персони – детальних описів типових користувачів для кращого розуміння їхніх вимог. Прототипування та тестування включає розробку низько- та високофідельних прототипів для попередньої оцінки дизайну та проведення юзабіліті-тестів з реальними користувачами для виявлення проблем на ранніх етапах.

Ітеративний процес дизайну забезпечує внесення змін на основі результатів тестування та зворотного зв'язку користувачів, що дозволяє постійно вдосконалювати інтерфейс. Контекстуальне проектування враховує специфічні умови, в яких працюють медичні працівники, включаючи стресові ситуації, багатозадачність і обмежений час, а також передбачає проектування для мобільності – створення інтерфейсу, який зручно використовувати на різних пристроях.

Підтримка та навчання користувачів є важливим елементом, що включає створення зрозумілих керівництв користувача і навчальних відео, а також забезпечення швидкої технічної підтримки та допомоги користувачам. Врахування специфічних потреб різних категорій користувачів також є критичним. Для лікарів необхідно забезпечити швидкий доступ до медичної інформації та інструменти для прийняття рішень, для медичного персоналу середньої ланки – прості і зрозумілі інструменти для введення і обробки даних, а для пацієнтів – дружній інтерфейс для перегляду своїх даних і комунікації з медичним персоналом.

Масштабованість є підходом проектування програмного забезпечення, що визначає його здатність ефективно обробляти зростаючі обсяги даних і збільшення кількості користувачів без зниження продуктивності. У контексті медичних інформаційних систем масштабованість особливо важлива з кількох причин:

- **Збільшення обсягу даних:** Медичні системи накопичують величезні обсяги даних про пацієнтів, включаючи історію хвороби, результати аналізів,

діагностичні зображення тощо. Здатність системи обробляти і зберігати великі обсяги даних є критичною.

- **Розширення функціональності:** З розвитком технологій і зміною вимог користувачів може виникнути потреба у впровадженні нових функцій і модулів. Система повинна бути гнучкою і легко адаптуватися до таких змін.
- **Інтеграція з іншими медичними системами:** Система повинна мати можливість інтеграції з іншими медичними інформаційними системами, такими як електронні медичні записи або системи управління лікарнею, для забезпечення безшовного обміну даними.

Усі ці підходи разом створюють надійну основу для розробки медичних інформаційних систем, що не тільки задовольняють поточні потреби користувачів, але й готові до майбутнього розвитку і масштабування. Це сприятиме покращенню якості загального медичного обслуговування та підвищенню ефективності роботи медичних установ [19].

4.4 Вимоги до інтерфейсу та проектування його компонентів

4.4.1 Аналіз вимог до проектування

Аналіз потреб користувачів є першим і найважливішим етапом у процесі розробки інтерфейсу. Він включає:

Створення персони: Визначення типових користувачів системи (персони) з детальним описом їхніх характеристик, цілей, задач та умов використання системи. Це допоможе при розробці інтерфейсу зосередитися на реальних потребах користувачів.

Аналіз існуючих рішень: Вивчення існуючих медичних систем для визначення сильних і слабких сторін їхніх інтерфейсів. Це допоможе уникнути поширених помилок і запозичити кращі практики.

Збір вимог: Включає проведення інтерв'ю, опитувань та фокус-груп з майбутніми користувачами системи для розуміння їхніх потреб, очікувань та проблем, з якими вони стикаються у своїй роботі. Особливу увагу слід приділити специфічним вимогам лікарів, медичного персоналу та пацієнтів.

4.4.2 Створення персони для системи ІІІ для діагностування захворювань опорно-рухового апарату

Олександр Петрович

Вік: 45 років

Посада: Лікар-ортопед

Досвід роботи: 20 років

Освіта: Вища медична Ім'я персони: освіта, спеціалізація з ортопедії

Рівень технічної підготовки: Середній. Олександр Петрович добре володіє комп'ютером на рівні користувача, знайомий з базовим медичним програмним забезпеченням.

Основні задачі та обов'язки:

- Діагностика та лікування захворювань опорно-рухового апарату
- Проведення оглядів пацієнтів

- Аналіз клінічних зображень
- Призначення лікування та реабілітаційних заходів
- Ведення медичної документації

Цілі та мотивація:

- Підвищення точності та швидкості діагностики захворювань
- Зменшення часу, що витрачається на аналіз медичних зображень
- Підвищення ефективності роботи за рахунок автоматизації рутинних завдань
- Зниження ризику людських помилок під час діагностики

Проблеми:

- Велика кількість часу, що витрачається на ручний аналіз медичних зображень
- Високий рівень стресу через велику кількість пацієнтів і обмежений час на кожного
- Ризик помилок через втому або людський фактор

Сценарій використання системи:

- Ранкова рутинна робота:
 - 1) Олександр Петрович приходить на роботу і відкриває програму на своєму комп'ютері.
 - 2) Він завантажує фотографії пацієнтів, зроблені під час ранкових оглядів.
 - 3) Програма автоматично аналізує фотографії і надає попередні діагнози.
- Огляд пацієнта:

- 1) Під час прийому пацієнта Олександр Петрович робить фотографії постави за допомогою цифрової камери.
 - 2) Він завантажує ці фотографії в систему для миттєвого аналізу.
 - 3) Програма виводить результати аналізу на екран, що дозволяє лікарю швидко прийняти рішення про подальші дії.
- Підготовка до консилиуму:
- 1) Перед консилиумом Олександр Петрович збирає дані про пацієнтів, включаючи фотографії.
 - 2) Він використовує систему для автоматичного аналізу зображень і генерації звітів.
 - 3) На основі отриманих даних він готує доповідь для колег.

Потреби в інтерфейсі:

- Інтуїтивно зрозумілий інтерфейс, що не потребує тривалого навчання
- Можливість швидкого завантаження та обробки фотографій
- Чіткі та зрозумілі візуальні індикатори результатів аналізу
- Легкий доступ до історії діагнозів і результатів попередніх аналізів
- Функціональність для створення та збереження звітів

Створення персони Олександра Петровича допомагає чітко визначити потреби і вимоги до системи ШІ, що дозволяє зосередитися на розробці функціональності, яка буде максимально корисною і зручною для кінцевого користувача.

4.4.2 Аналіз інтерфейсу існуючої медичної системи FreeMED

FreeMED – це електронна медична система (EHR), яка використовується для управління медичними даними пацієнтів. Розглянемо її інтерфейс, сильні та слабкі сторони, щоб зробити висновки щодо покращення нашого рішення.



Рисунок 4.1 – Інтерфейс FreeMed

Сильні сторони

- Чітка навігація: Інтерфейс має чітко визначені області для навігації зліва (Search, New Patient, Groups, Call In, Rx Refill), що дозволяє швидко знайти необхідну функцію.
- Модульна структура: Система організована в модулі, що дозволяє користувачам легко розширювати та налаштовувати функціонал під свої потреби.
- Інтеграція з різними функціями: FreeMED включає різні модулі для роботи з пацієнтами, фінансовою інформацією та клінічною інформацією, що забезпечує комплексний підхід до управління даними.

- Панель дій: Присутність панелі дій з інструментами (Add, Edit, Delete) дозволяє легко виконувати основні операції з даними пацієнта.

Слабкі сторони

- Складний і застарілий інтерфейс: Інтерфейс виглядає застарілим і перевантаженим інформацією, що може ускладнювати роботу користувачів, особливо новачків.
- Недостатня візуальна естетика: Використання застарілих графічних елементів та кольорових схем робить інтерфейс менш привабливим та менш зручним для тривалого використання.
- Мало місця для клінічної інформації: Відведена область для клінічної інформації є занадто маленькою, що може ускладнювати введення та перегляд медичних даних.
- Недостатня інтерактивність: Відсутність інтерактивних елементів та сучасних технологій, таких як автозаповнення та контекстні підказки, що могли б покращити зручність використання.
- Неоптимізована інформаційна архітектура: Інформація про пацієнта та клінічні дані розташовані в одному вікні без чіткого розмежування, що може створювати плутанину.

Рекомендації для нашої системи

- Сучасний дизайн інтерфейсу: Використання сучасних графічних елементів та кольорових схем, які є більш привабливими та легкими для сприйняття.
- Покращена інформаційна архітектура: Чітке розмежування різних типів даних (фото, діагнози, дані пацієнтів) для зменшення плутанини та полегшення навігації.

- Інтерактивні елементи: Додавання інтерактивних елементів, таких як автозаповнення, контекстні підказки та спливаючі вікна з додатковою інформацією для покращення зручності використання.
- Оптимізація для швидкого доступу до даних: Збільшення розмірів полів для клінічної інформації та забезпечення можливості швидкого доступу до важливих даних пацієнтів.
- Адаптивний дизайн: Забезпечення адаптивності інтерфейсу для різних пристроїв та роздільних здатностей екрану, що покращить зручність роботи на різних платформах.
- Модульність та налаштовуваність: Збереження модульної структури для забезпечення можливості розширення функціоналу та налаштування системи під специфічні потреби користувачів.

Аналізуючи інтерфейс FreeMED, можна визначити як його сильні сторони, так і обмеження. Використовуючи ці знання, ми можемо уникнути поширених помилок і запозичити кращі практики, щоб створити більш зручну, ефективну та привабливу систему для аналізу постави пацієнтів.

4.4.3 Аналіз вимог від користувачів

Для розробки інтерфейсу програми для діагностики захворювань опорно-рухового апарату було проведено аналіз потреб лікарів. На основі зібраних даних, лікарі визначили кілька ключових вимог, які були враховані при проектуванні інтерфейсу.

На рисунку 4.2 зображено схему інтефейсу від лікарів.

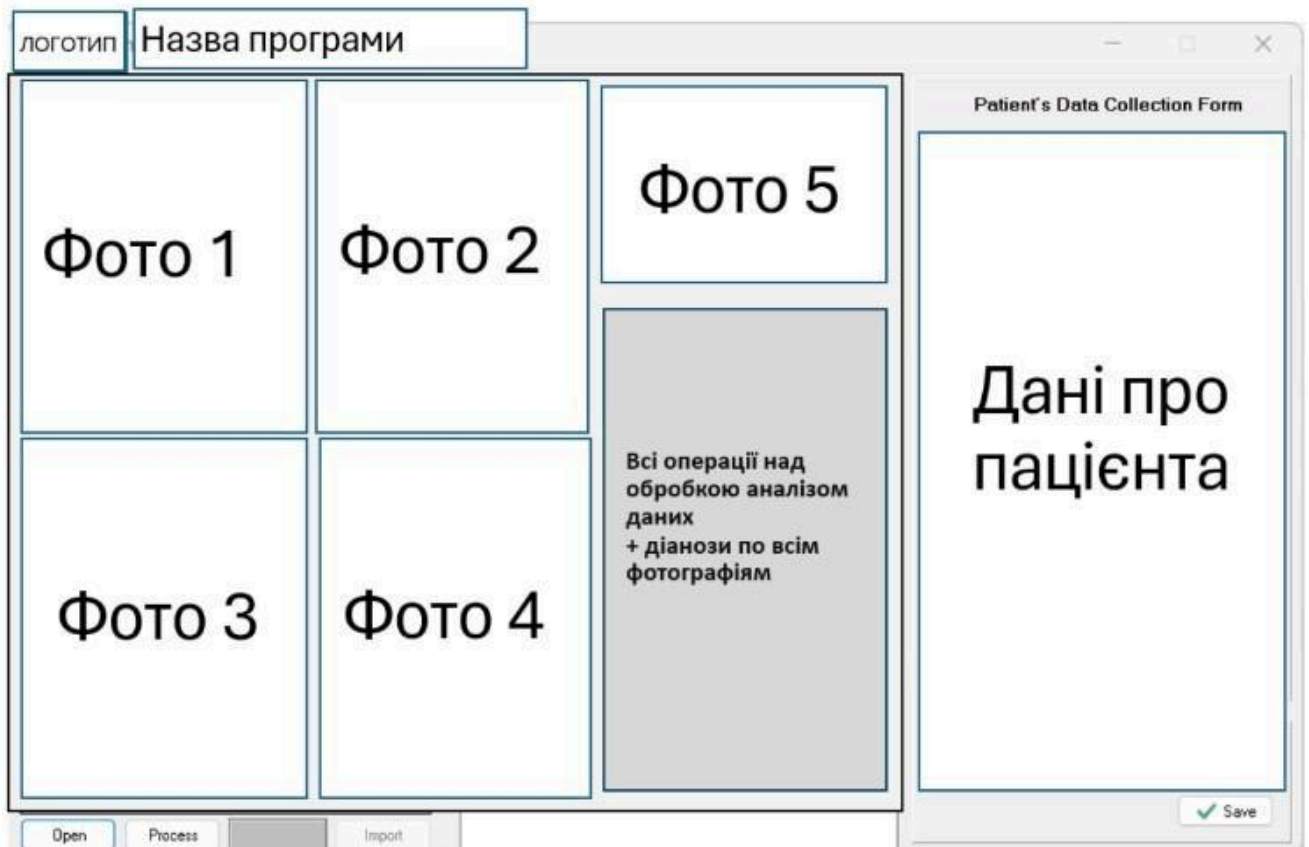


Рисунок 4.2 – Схема інтерфейсу користувача

Легкість і зручність у використанні: Лікарі відзначили, що інтерфейс повинен бути інтуїтивно зрозумілим і простим у використанні. Це зменшує час, необхідний для навчання персоналу і знижує ризик помилок під час роботи.

Можливість роботи з кількома зображеннями: Лікарі підкреслили необхідність одночасного перегляду та аналізу кількох зображень пацієнта. У схемі інтерфейсу передбачено п'ять окремих вікон для відображення фотографій (Фото 1, Фото 2, Фото 3, Фото 4, Фото 5).

Централізоване управління даними: Лікарі зазначили важливість мати доступ до всіх операцій над обробкою аналізом даних та діагнозами в одному місці. У схемі інтерфейсу це відображено у вигляді окремої області, де будуть здійснюватися всі операції над обробкою даних і встановлення діагнозів.

Інформація про пацієнта: Лікарям потрібен швидкий доступ до основних даних про пацієнта під час аналізу зображень. Для цього в інтерфейсі передбачено окремий розділ "Дані про пацієнта", де відобразатиметься вся необхідна інформація.

Інтуїтивні кнопки управління: Для забезпечення зручності роботи передбачені кнопки для відкриття зображень (Open), їх обробки (Process), імпорту даних (Import) та збереження результатів (Save).

Візуальна організація: Інтерфейс передбачає чітке розмежування зон для різних типів інформації та функцій, що дозволяє лікарям швидко орієнтуватися і виконувати необхідні дії без зайвих зусиль.

4.5 Проектування інтерфейсу програмного забезпечення

Під час проектування інтерфейсу для системи ШІ для діагностики захворювань опорно-рухового апарату ми використовували веб-сервіс Figma як ключовий інструмент.

Figma – це веб-сервіс для дизайну і прототипування інтерфейсів, який набув великої популярності серед дизайнерів та розробників програмного забезпечення. Він відрізняється від інших інструментів своєю здатністю до спільної роботи в реальному часі, що дозволяє різним членам команди працювати над проектом одночасно, незалежно від їх місця розташування.

Нижче на рисунку 4.5.1 наведений розроблений інтерфейс, який демонструє основні елементи та функціональність системи.

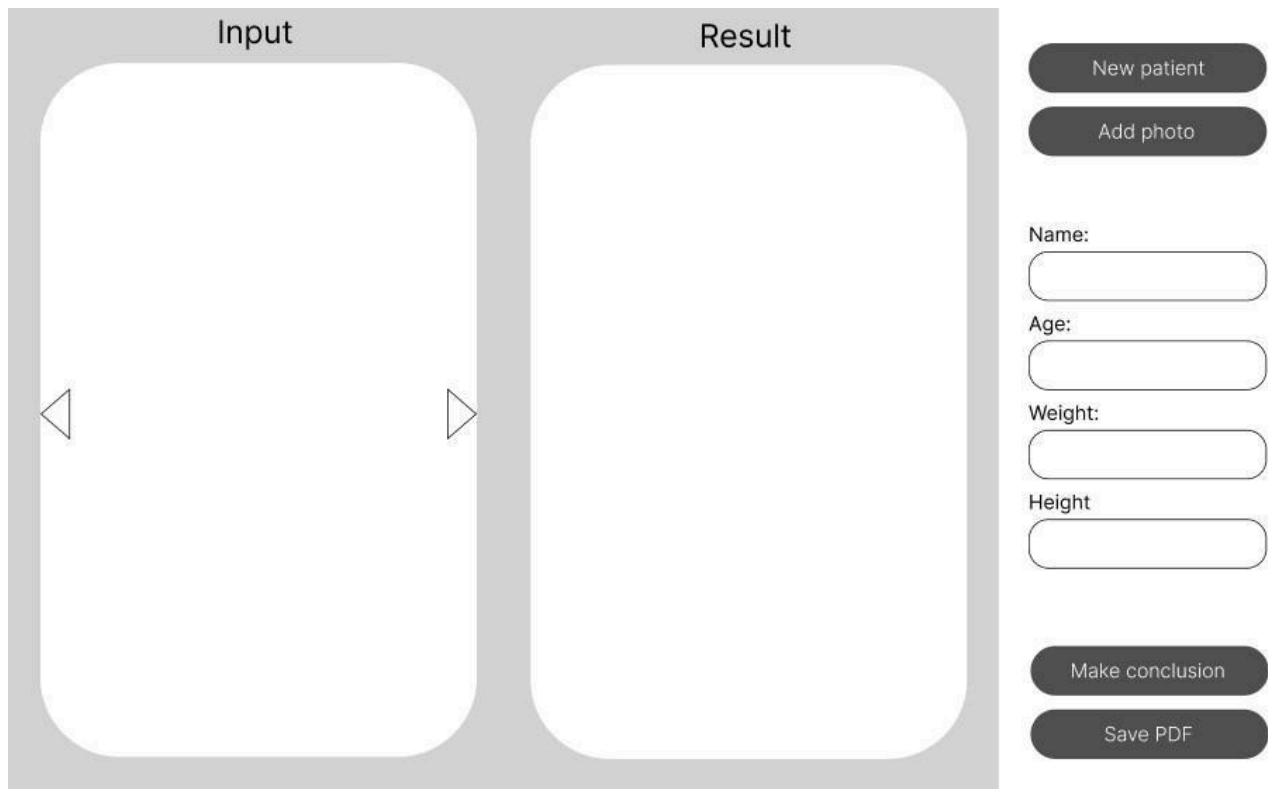


Рисунок 4.3 – Розроблений інтерфейс

На зображенні інтерфейсу (рис. 4.3) можна побачити наступні основні компоненти:

Вкладки Input та Result:

- Ліва частина інтерфейсу (Input) призначена для введення даних. Сюди завантажуються фотографії пацієнтів для аналізу їх постави.
- Права частина інтерфейсу (Result) показує результати аналізу після обробки фотографій системою штучного інтелекту.

Кнопки навігації:

- На обох сторонах (Input і Result) є кнопки навігації (ліворуч і праворуч), які дозволяють переглядати різні фотографії пацієнтів.

Панель керування:

- Розташована справа, панель містить кнопки для основних дій: "New patient", "Add photo", "Make conclusion" та "Save PDF".
- Також на панелі розміщені поля введення для даних пацієнта: Name, Age, Weight, Height.

Важливим аспектом проектування інтерфейсу програмного забезпечення є вибір кольорової палітри. Правильно підібрані кольори не тільки покращують візуальну естетику, але й підвищують зручність використання, знижують втомлюваність очей та забезпечують доступність для користувачів з різними зоровими особливостями. Нижче описана кольорова палітра, використана в інтерфейсі системи штучного інтелекту для діагностування захворювань опорно-рухового апарату.

Основний фон (сірий):

Код кольору: #D3D3D3 (Light Gray)

Основний фон інтерфейсу має світло-сірий відтінок, який забезпечує м'який контраст з основними елементами інтерфейсу. Це допомагає уникнути різких контрастів, які можуть викликати втому очей при тривалому використанні.

Фонові елементи (білий):

Код кольору: #FFFFFF (White)

Білі фоні елементи використовуються для основних робочих областей, таких як поля введення даних пацієнтів та області перегляду фотографій. Білий колір забезпечує максимальну чіткість та зручність для введення та читання текстової інформації.

Акцентні кнопки (темно-сірий):

Код кольору: #4B4B4B (Dark Gray)

Темно-сірі кнопки використовуються для основних дій, таких як "New patient", "Add photo", "Make conclusion" та "Save PDF". Темно-сірий колір надає чіткий контраст та привертає увагу до важливих елементів керування.

Текстові поля (чорний):

Код кольору: #000000 (Black)

Чорний колір використовується для тексту в полях введення, щоб забезпечити максимальну чіткість та контрастність текстової інформації.

Інформаційні повідомлення (зелений):

Код кольору: #008000 (Green)

Зелений колір використовується для відображення позитивних інформаційних повідомлень, таких як успішне збереження даних чи завершення обробки.

Попередження та помилки (червоний):

Код кольору: #FF0000 (Red)

Червоний колір використовується для відображення помилок чи попереджень, що потребують негайної уваги користувача.

Також важливим є забезпечення можливості перемикання між світлим і темним режимами інтерфейсу. Це було зроблено для підвищення комфорту роботи користувачів в різних умовах освітлення, а також для зменшення напруження очей при тривалому використанні системи. Темний інтерфейс було розроблено з урахуванням сучасних тенденцій у дизайні користувацьких інтерфейсів та вимог до ергономіки. Колірна палітра залишилась та сама, але було змінено використання фонових кольорів на протилежні. Темний інтерфейс зображено на рисунку 4.4.

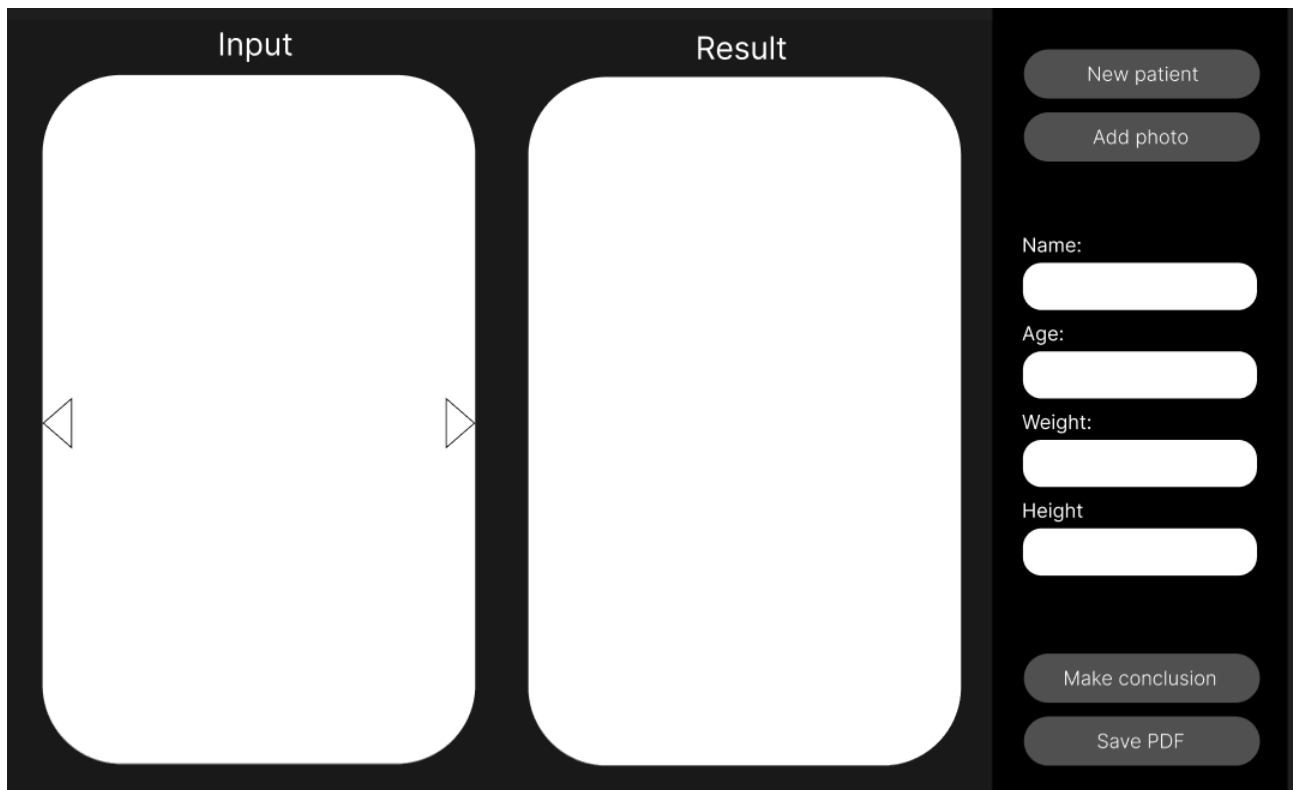


Рисунок 4.4 – Темний інтерфейс

При виборі кольорової палітри особливу увагу приділено контрастності між текстом і фоном для забезпечення доступності. Всі кольорові комбінації були перевірені на відповідність стандартам Web Content Accessibility Guidelines для забезпечення належного рівня контрастності, що дозволяє користувачам з вадами зору комфортно працювати з інтерфейсом.

Обрана кольорова палітра забезпечує зручність та комфорт використання інтерфейсу для лікарів та медичного персоналу. Світло-сірий фон, білий робочий простір, темно-сірі акцентні кнопки та контрастний текст забезпечують хорошу видимість та естетичну привабливість, що сприяє ефективній роботі з системою.

Інтерфейс розроблено таким чином, щоб мінімізувати кількість дій, необхідних для виконання основних завдань. Використано сучасний мінімалістичний дизайн з чіткими лініями та приємними кольорами, що знижує втомлюваність очей і покращує загальне сприйняття інтерфейсу. Всі елементи інтерфейсу розміщені логічно та інтуїтивно зрозуміло, що дозволяє легко орієнтуватися навіть користувачам без технічного досвіду.

ВИСНОВКИ

У даній кваліфікаційній роботі було розроблено систему штучного інтелекту для діагностування захворювань опорно-рухового апарату, яка базується на сучасних методах машинного навчання та нейронних мережах. Основною метою дослідження було створення інструменту, що дозволить значно покращити процес діагностики, зробивши його швидшим, точнішим та доступнішим.

У ході виконання кваліфікаційної роботи було досягнуто важливих наукових та практичних результатів, що підтверджують ефективність розробленої системи штучного інтелекту для діагностування захворювань опорно-рухового апарату. Основним досягненням стало успішне навчання моделі YOLOv8 для знаходження ключових точок на тілі людини, що дозволило досягти високої точності виявлення ключових точок, критичних для подальшого аналізу стану опорно-рухового апарату. Це технічне рішення автоматизує процес аналізу медичних зображень, зменшує навантаження на медичний персонал та підвищує ефективність діагностики.

Розроблена система продемонструвала високу ефективність у виявленні ключових точок, що підтверджує її потенціал для практичного застосування у медичних закладах. Вона може бути інтегрована у існуючі медичні інформаційні системи, забезпечуючи автоматичний аналіз результатів діагностики та допомагаючи лікарям у прийнятті рішень щодо лікування пацієнтів.

Таким чином, результати цієї роботи свідчать про доцільність та перспективність використання систем ШІ для діагностування захворювань опорно-рухового апарату.

Практичне значення отриманих результатів підтверджується можливістю їх впровадження у медичних установах, що дозволить значно покращити процес діагностики захворювань опорно-рухового апарату, забезпечивши швидкість, точність та доступність медичної допомоги.

Список використаних джерел

1. Будова і функції опорно-рухового апарату [Електронний ресурс] // Disted: [сайт]. – URL: <https://disted.edu.vn.ua/courses/learn/2587> (дата звернення: 02.06.2024).
2. LibreTexts. Огляд опорно-рухового апарату [Електронний ресурс] // LibreTexts - Ukrayinska: [сайт]. – URL: <http://surl.li/udkhg> (дата звернення: 02.06.2024).
3. ОПОРНО-РУХОВА СИСТЕМА: опорно-рухова система [Електронний ресурс] // Підготовка до ЗНО - Освітній портал "Академія": [сайт]. – URL: <http://zno.academia.in.ua/mod/book/view.php?id=2353> (дата звернення: 02.06.2024).
4. Порушення опорно-рухової системи [Електронний ресурс] // МійКлас: [сайт]. – URL: <https://www.miyklas.com.ua/p/biologiya/8-klas/opora-i-ruk-327280/khvorobi-oporno-rukhovoyi-sistemi-ta-yikh-profilaktika-327320/re-0940ce76-1629-4a2b-b845-aaccc0099cf8> (дата звернення: 02.06.2024).
5. Опорно-рухові захворювання [Електронний ресурс] // UBI CLINIC: [сайт]. – URL: <https://ubi-clinic.com/oporno-rukho-vi-zakhvoriuvannia/> (дата звернення: 02.06.2024).
6. Machine learning, ML [Електронний ресурс] // IT-Enterprise – your one-stop platform for digital transformation: [сайт]. – URL: <https://www.it.ua/knowledge-base/technology-innovation/machine-learning> (дата звернення: 02.06.2024).
7. В Україні вперше використали ШІ для діагностики хвороб: це допоможе прифронтовим містам [Електронний ресурс] // Zaborona: [сайт]. – URL: <https://zaborona.com/v-ukrayini-vpershe-vykorystaly-shi-dlya-diagnostyky-hvorob-cze-dopomozhe-pryfrontovym-mistam/> (дата звернення: 02.06.2024).
8. Учасники проектів Вікімедіа. Градієнтний спуск [Електронний ресурс] // Вікіпедія: [сайт]. – URL: https://uk.wikipedia.org/wiki/Градiєнтний_спуск (дата звернення: 02.06.2024).
9. Долгополов С. Моделювання задач штучного інтелекту: методичні вказівки до виконання лабораторних робіт: навч. посіб. Київ: КНУБА, 2024. 178 с.

10. Convolutional neural network (cnn) simplified [Електронний ресурс] // Medium: [сайт]. – URL: <https://medium.datadriveninvestor.com/convolutional-neural-network-cnn-simplified-e-cafd4ee52c5> (дата звернення: 02.06.2024).
11. Згорткові нейронні мережі (частина 2) [Електронний ресурс] // IT Master - електроніка та програмування: [сайт]. – URL: <https://itmaster.biz.ua/programming/vision/cnns2.html> (дата звернення: 02.06.2024).
12. Marchuk G. V., Marchuk D. K., Kovalchuk A. M. Application of a convolutional neural network for the recognition of handwritten characters [Електронний ресурс] // Scientific notes of Taurida National V.I. Vernadsky University. Series: Technical Sciences. 2019. Vol. 4, no. 1. P. 68–73. – URL: <https://doi.org/10.32838/2663-5941/2019.4-1/13> (дата звернення: 02.06.2024).
13. Субботін С. Нейронні мережі: теорія та практика: навч. посіб. Житомир: О. О. Євенок, 2023.
14. Potrimba P. What is keypoint detection? [Електронний ресурс] // Roboflow Blog: [сайт]. – URL: <https://blog.roboflow.com/what-is-keypoint-detection/> (дата звернення: 02.06.2024).
15. OpenPose: realtime multi-person 2D pose estimation using part affinity fields / Z. Cao et al. [Електронний ресурс] // IEEE transactions on pattern analysis and machine intelligence. 2019. P. 1. – URL: <https://doi.org/10.1109/tpami.2019.2929257> (дата звернення: 02.06.2024).
16. Учасники проєктів Вікімедіа. Python [Електронний ресурс] // Вікіпедія: [сайт]. – URL: <https://uk.wikipedia.org/wiki/Python> (дата звернення: 02.06.2024).
17. YoloV8 architecture & cow counter with region based dragging using yolov8 [Електронний ресурс] // Medium: [сайт]. – URL: https://medium.com/@VK_Venkatkumar/yolov8-architecture-cow-counter-with-region-based-dragging-using-yolov8-e75b3ac71ed8 (дата звернення: 02.06.2024).
18. Ultralytics. Ultralytics YOLO Docs [Електронний ресурс] // Home - Ultralytics YOLO Docs: [сайт]. – URL: <https://docs.ultralytics.com> (дата звернення: 02.06.2024).

19. Горда О. В. Ергономіка інформаційних технологій: конспект лекцій. Київ: КНУБА, 2024. 83 с.
20. Olah C. Understanding backpropagation [Електронний ресурс] // Colah's blog: [сайт]. – URL: <https://colah.github.io/posts/2015-08-Backprop/#fnref1> (дата звернення: 02.06.2024).

Додаток А

Система ШІ для діагностики захворювань опорно-рухового апарату

ЗДОБУВАЧ: ЙОВЕНКО Д. С.

КЕРІВНИК: ПОПЛАВСЬКИЙ О. А.

КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ 2024

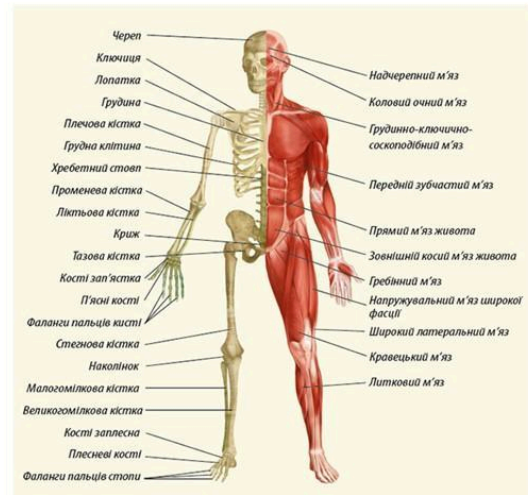


Слайд 1 – Титульний слайд

Предметна область

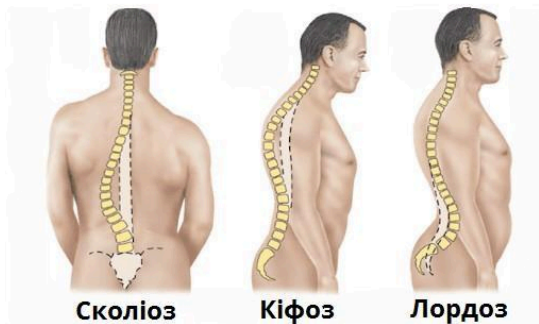
Опорно-руховий апарат - комплекс структур, який утворює каркас, надає форму організму, дає йому опору та забезпечує захист внутрішніх органів і можливість пересування у просторі

Опорно-руховий апарат складається зі скелета та м'язів. Скелет це пасивна частина, а м'язи – активна. Кістки скелета є важелями, які приводяться в рух м'язами.



Слайд 2 – Предметна область

Об'єкт дослідження



Одним з найбільш поширених видів хронічних неінфекційних захворювань є ураження опорно-рухового апарату. Хвороби опорно-рухового апарату є однією з основних причин інвалідності в глобальному масштабі. Існує понад 150 типів захворювань, що вражають опорно-руховий апарат, зазвичай викликаючи біль, обмеження рухливості та зниження функціональних можливостей, що ускладнює виконання роботи.

Слайд 3 – Об'єкт дослідження

Об'єкт дослідження

Основні методи діагностики:

- Клінічний огляд
- Рентгенографія
- Магнітно-резонансна томографія
- Комп'ютерна томографія
- Ультразвукове дослідження
- Лабораторні тести



Слайд 4 – Об'єкт дослідження ч.2

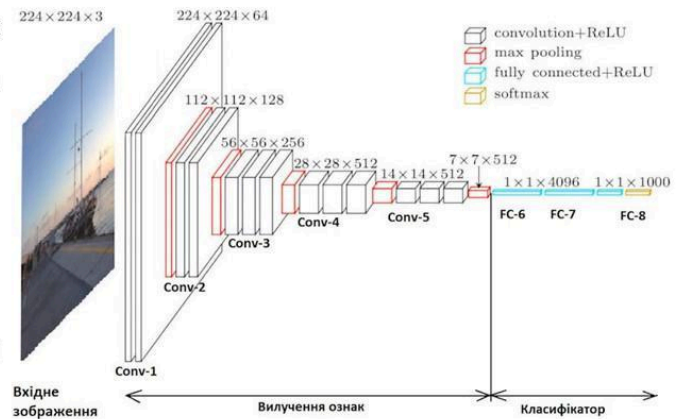
Предмет дослідження

Згорткова нейронна мережа - клас глибоких штучних нейронних мереж прямого поширення, який застосовується для задач аналізу зображень.

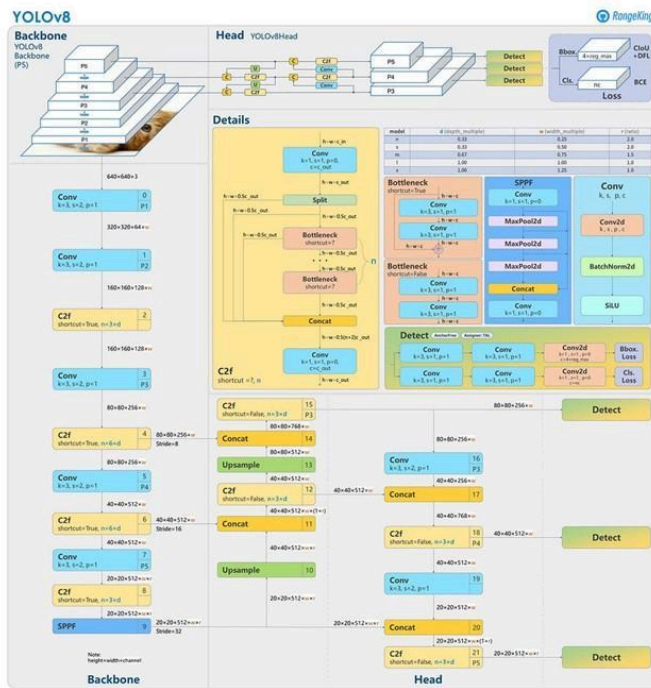
Загальна структура згорткової нейронної мережі має:

- вхідний шар
- приховані шари
- вихідний шар

Приховані шари включають в себе згорткові шари, шари агрегування (пулінгу), шари нормалізації та повнозв'язні шари.



Слайд 5 – Предмет дослідження



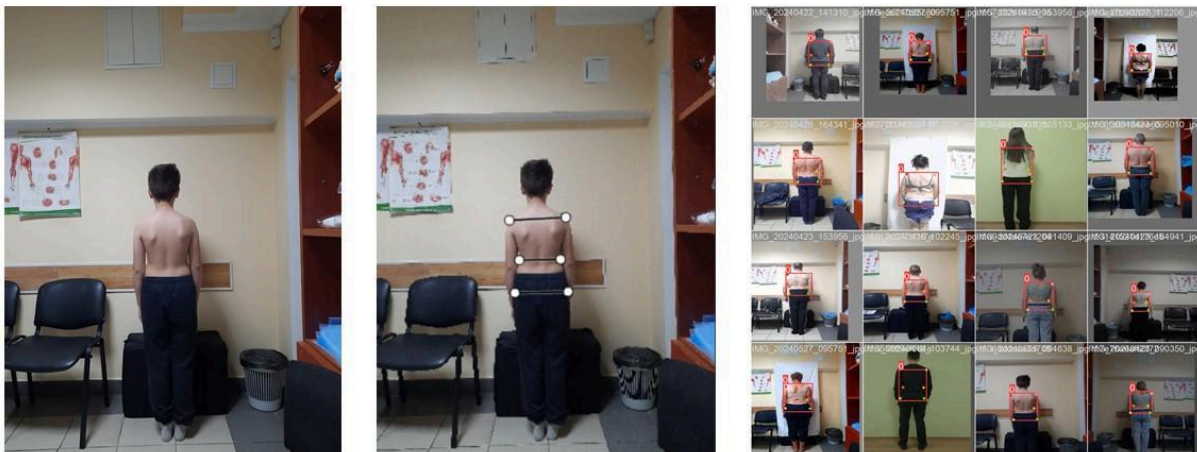
YOLO("You Only Look Once") – модель глибокого навчання, розроблена для швидкої та точної обробки зображень, зокрема для виявлення об'єктів на зображеннях у реальному часі.

YOLOv8 побудована на основі модернізованої архітектури CSPDarknet53, яка використовує міжступеневі часткові з'єднання для покращення потоку інформації між різними рівнями.

Особливістю модифікації цієї мережі є те що вона використовує модуль C2f. Задача цього модулю заключається в тому що він розділяє вхідні карти ознак на дві частини, одна з яких переходить до наступного блоку згорткових шарів, тоді як друга частина проходить через декілька згорткових шарів

Слайд 6 – Предмет дослідження ч.2

Збір даних



Слайд 7 – Збір даних

Тестовий приклад

За попередньою оцінкою середня точність визначення обмежувальних рамок 99.5%, знаходження ключових точок 87.9%.

```

Epoch   GPU_mem  box_loss  pose_loss  kobj_loss  cls_loss  dfl_loss  Instances
498/500  11.46    0.08724  0.04048   0.005137   0.1065   0.7498    15
Class    Images  Instances  Box(P)      R      mAP50  mAP50-95)

Epoch   GPU_mem  box_loss  pose_loss  kobj_loss  cls_loss  dfl_loss  Instances
499/500  11.56    0.1038   0.0406    0.004766   0.1096   0.7495    15
Class    Images  Instances  Box(P)      R      mAP50  mAP50-95)

Epoch   GPU_mem  box_loss  pose_loss  kobj_loss  cls_loss  dfl_loss  Instances
500/500  11.46    0.08077  0.03855   0.00519    0.09911   0.7707    15
Class    Images  Instances  Box(P)      R      mAP50  mAP50-95)

500 epochs completed in 0.821 hours.
Optimizer stripped from runs/pose/train/weights/last.pt, 89.5MB
Optimizer stripped from runs/pose/train/weights/best.pt, 89.4MB

```

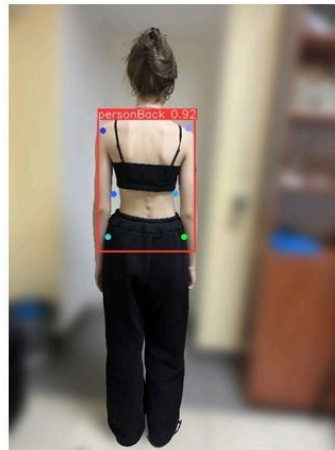


Вхідні зображення

Передбачення мережі

Слайд 8 – Тестовий приклад

Тестовий знімок

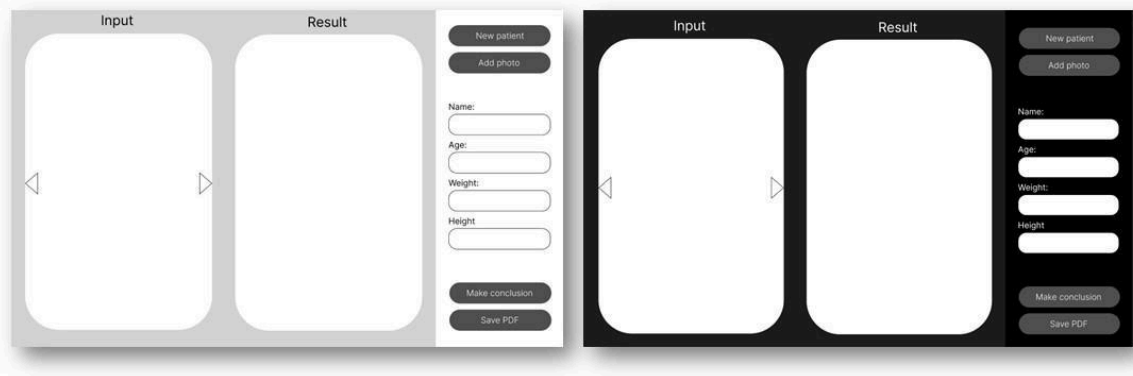


Слайд 9 – Тестовий знімок

Ергономіка в ІТ

ПРОТОТИП СВІТЛОГО ДИЗАЙНУ

ПРОТОТИП ТЕМНОГО ДИЗАЙНУ



Слайд 10 – Ергономіка в ІТ

Висновки

У ході виконання кваліфікаційної роботи було досягнуто важливих наукових та практичних результатів, що підтверджують ефективність методів глибокого навчання для діагностування захворювань опорно-рухового апарату.

Основним досягненням стало успішне навчання моделі YOLOv8 для знаходження ключових точок на тілі людини, що дозволило досягти високої точності виявлення ключових точок, критичних для подальшого аналізу стану опорно-рухового апарату. Це технічне рішення автоматизує процес аналізу медичних зображень, зменшує навантаження на медичний персонал та підвищує ефективність діагностики.

Слайд 11 – Висновки