

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«До захисту допущено»

В.о. завідувача кафедри

Оксана ШОВКОПЛЯС

_____ (підпис)

_____ грудня 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня магістр

зі спеціальності 122 «Комп'ютерні науки»

освітньо-професійної програми «Інформатика»

на тему: Інтелектуальна технологія класифікації проблем постави людини на
основі аналізу візуальних даних

здобувачки групи ІН.м-32 Морквіної Валерії Володимирівни

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

Валерія МОРКВІНА

_____ (підпис)

Керівник

доцент,

кандидат технічних наук, доцент

Борис КУЗІКОВ

_____ (підпис)

Суми – 2024

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«Затверджую»

В.о. завідувача кафедри

Оксана ШОВКОПЛЯС

(підпис)

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня магістра

зі спеціальності 122 «Комп'ютерні науки», освітньо-професійної програми «Інформатика»
здобувачки групи ІН.м-32 Морквіної Валерії Володимирівни

1. Тема роботи: Інтелектуальна технологія класифікації проблем постави людини на основі аналізу візуальних даних

затверджую наказом по СумДУ від «03» грудня 2024 року № 1257-VI

2. Термін здачі здобувачем кваліфікаційної роботи до 06 грудня 2024 року

3. Вхідні дані до кваліфікаційної роботи _____

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)

1) Аналіз проблеми предметної області, постановка й формування завдань дослідження.

2) Огляд вхідних та вихідних даних. 3) Огляд математичної моделі. 4) Вибір засобів розробки,

архітектура програмного забезпечення та огляд програмного продукту. 5) Аналіз

результатів.

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

6. Консультанти до проекту (роботи), із зазначенням розділів проекту, що стосується їх

| Розділ | Консультант | Підпис, дата | |
|--------|-------------|----------------|------------------|
| | | Завдання видав | Завдання прийняв |
| | | | |

7. Дата видачі завдання « ____ » _____ 20 ____ р.

Завдання прийняв до виконання _____
(підпис)

Керівник _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

| № п/п | Назва етапів кваліфікаційної роботи | Термін виконання | Примітка |
|-------|--|------------------|----------|
| 1 | <i>Збір, систематизація й узагальнення матеріалу для використання у кваліфікаційній роботі</i> | 01.09.2024 | |
| 2 | <i>Проведення аналізу актуальності теми та наявних аналогів, постановка задачі</i> | 14.10.2024 | |
| 3 | <i>Визначення вхідних та вихідних даних</i> | 28.10.2024 | |
| 4 | <i>Розробка математичної моделі</i> | 08.11.2024 | |
| 5 | <i>Вибір засобів розробки та планування архітектури програмного забезпечення</i> | 15.11.2024 | |
| 6 | <i>Розробка інтелектуальної технології з класифікації порушень</i> | 25.11.2024 | |

| | | | |
|---|---|------------|--|
| | <i>постави</i> | | |
| 7 | <i>Аналіз отриманих результатів</i> | 29.11.2024 | |
| 8 | <i>Оформлення пояснювальної записки до кваліфікаційної роботи</i> | 06.12.2024 | |

Здобувач вищої освіти

(підпис)

Керівник

(підпис)

АНОТАЦІЯ

Записка: 46 стр., 33 рис., 1 додаток, 21 використаних джерел.

Обґрунтування актуальності теми роботи – тема кваліфікаційної роботи є актуальною, оскільки присвячена розв’язанню важливої практичної задачі прогнозування типів порушення постави на основі візуальних даних шляхом розробки відповідних методів, моделей та інформаційної технології.

Об’єкт дослідження — застосування інтелектуальних технологій у сфері громадського здоров’я.

Предмет дослідження — моделі та методи класифікації різних типів порушень постави на основі даних, отриманих за допомогою візуального аналізу зображень.

Мета роботи — спрощення процесу виявлення ключових анатомічних точок тіла та полегшення аналізу відхилень у поставі

Методи дослідження — моделі машинного навчання, теорія ймовірності та математична статистика.

Результати — розроблено інтелектуальну технологію, яка визначає ключові анатомічні точки постави та обчислює їх кути. Використовуючи існуючі патерни, програмний продукт надає користувачу класифікацію типів порушень постави та рекомендовані вправи для її покращення.

ІНТЕЛЕКТУАЛЬНА ТЕХНОЛОГІЯ, ВИЗНАЧЕННЯ КЛЮЧОВИХ ТОЧОК
ПОСТАВИ, КЛАСИФІКАЦІЯ ПОРУШЕНЬ ПОСТАВИ, НЕЙРОННА
МЕРЕЖА.

ЗМІСТ

| | |
|---|----|
| ВСТУП | 6 |
| 1 АНАЛІТИЧНИЙ ОГЛЯД | 8 |
| 1.1 Опис предметного середовища | 8 |
| 1.1.1 Опис процесу діяльності | 8 |
| 1.1.2 Опис функціональної моделі | 9 |
| 1.2 Огляд наявних аналогів | 12 |
| 1.3 Постановка задачі | 15 |
| 2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ | 17 |
| 2.1 Вхідні дані | 18 |
| 2.2 Вихідні дані | 23 |
| 3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ | 24 |
| 3.1 Змістовна постановка задачі | 24 |
| 3.2 Математична модель | 24 |
| 3.3 Опис методу розв'язання | 30 |
| 3.3.1 Тренування та тестування моделей | 32 |
| 4 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ | 35 |
| 4.1. Вибір засобів розробки | 35 |
| 4.2. Архітектура програмного забезпечення | 37 |
| 4.3. Огляд програмного продукту | 43 |
| ВИСНОВКИ | 49 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ | 51 |
| ДОДАТОК А | 54 |

ВСТУП

Сьогодні проблема неправильної постави є дуже поширеною. Важливо розуміти, що це не лише естетичний недолік, а й потенційна загроза для здоров'я.

Неправильна постава – це тихий вбивця здоров'я, особливо серед молодого покоління. За даними досліджень ВООЗ, понад дві третини українських підлітків мають недостатню фізичну активність[1]. Це пов'язано з тривалим сидінням за комп'ютерами та гаджетами. Ослаблені м'язи спини, нездатні підтримувати хребет у правильному положенні, призводять до розвитку сколіозу та інших порушень постави. В результаті, ми стикаємося з хронічними болями, дискомфортом і навіть обмеженням рухливості.

Неправильна постава не є остаточним діагнозом, а проблемою, яку можна успішно вирішити. Раннє виявлення порушень значно полегшує процес їх корекції. Фахівці, такі як реабілітологи чи фітнес-тренери, можуть за допомогою фотографій аналізувати положення тіла, знаходити основні точки постави та класифікувати порушення. Однак, для точної оцінки потрібні спеціальні знання та досвід, які не завжди є у звичайних людей. Самостійний аналіз фотографій може бути неточним і призвести до неправильних висновків. Сучасні технології надають нові можливості для автоматизації та об'єктивізації процесу оцінки постави, що є особливо актуальним у контексті масового скринінгу та персоналізованого підходу до лікування або створення індивідуальної програми тренувань.

Об'єкт дослідження - застосування інтелектуальних технологій у сфері громадського здоров'я.

Предмет дослідження - моделі та методи класифікації різних типів порушень постави на основі даних, отриманих за допомогою візуального аналізу зображень.

Пропонована технологія поєднує в собі переваги сучасних методів комп'ютерного зору та машинного навчання для вирішення актуальної медичної проблеми. Вона дозволить зменшити час на проведення ручного

аналізу та отримання результатів, а також розширити доступність діагностики постави для широкого кола людей. Ця робота є актуальним внеском у розвиток цифрової медицини та може знайти широке застосування в реабілітації та фітнес-індустрії.

Дана робота складається зі вступу, аналітичного огляду, постановки задачі, математичної моделі розв'язання поставленої задачі, опису програмного забезпечення, висновків, списку використаних джерел та додатків.

1 АНАЛІТИЧНИЙ ОГЛЯД

1.1 Опис предметного середовища

У ритмі сучасного життя, де ми все більше часу проводимо сидячи за комп'ютером, смартфоном чи за кермом, проблема порушень постави набуває все більшої актуальності.

В Україні спостерігається турбуюча тенденція: кожна четверта дитина страждає від порушень постави. Особливо поширений сколіоз, яким хворіють понад 15% дітей та підлітків з проблемами опорно-рухового апарату. За даними МОЗ, у 2019 році сколіоз виявили у майже 100 тисяч дітей[2]. Важливо розуміти, що неправильне положення тіла, яке ми приймаємо протягом дня, поступово деформує наш хребет та м'язовий корсет, призводячи до серйозних наслідків для здоров'я. Наприклад, сколіоз часто є причиною остеохондрозу та радикуліту в дорослому віці.

Аналіз фотографії може бути цінним інструментом у діагностиці порушень постави. Фахівці визначають на знімку ключові точки тіла і з'єднують їх лініями, створюючи своєрідний графік. Він дозволяє оцінити положення тазу, плечей і інших частин тіла. Завдяки такому аналізу можна не тільки виявити проблему, але й розробити індивідуальний план корекції постави, включаючи вправи, мануальну терапію або інші методи лікування.

Даний дипломний проєкт присвячений побудові системи, яка буде робити візуальний аналіз фотографії, визначати основні точки постави та на основі цього надавати фінальний результат (чи є порушення постави та як їх можна вирішити).

1.1.1 Опис процесу діяльності

Процес діяльності в даній роботі спрямований на автоматизацію проведення фотоаналізу та класифікацію порушень постави людини на основі його результатів.

Процес аналізу передбачає два етапи. Спочатку, потрібно провести фотоаналіз, а саме визначати на зображенні ключові точки тіла, такі як плечі,

таз, коліна тощо. При цьому потрібно врахувати, з якого ракурсу зроблено знімок – спереду, ззаду або збоку. Далі, між цими точками будуються лінії, які слугують своєрідними маркерами для оцінки симетрії та вирівняності різних частин тіла.

Існують стандартні патерни цих ліній, за допомогою яких можна класифікувати типові порушення постави, такі як сколіоз, кіфоз, лордоз тощо та запропонувати певні вправи для укріплення м'язів[4,5].

Приклад можна побачити на рисунку 1.1. Лінії, проведені через плечі та таз, не утворюють паралельних прямих ні між собою, ні відносно горизонтальної лінії, що умовно зображає рівень підлоги. Непаралельність цих ліній свідчить про асиметрію тулуба. Таке відхилення від норми є характерною ознакою сколіотичної постави.



Рисунок 1.1 – Сколіотична постава [4]

1.1.2 Опис функціональної моделі

Першим кроком у створенні функціональної моделі є визначення акторів системи та їхні дії. Єдиний актор даної системи - це користувач. Повний перелік дій, які користувач може виконувати, представлено в таблиці 1.1.

Таблиця 1.1 – Варіанти використання

| Актор | Варіант використання | Опис дії варіанта використання |
|------------|----------------------------------|--|
| Користувач | Надання вхідних файлів | Додати фото з розслабленою поставою |
| | Перегляд файлів | Перегляд наданих файлів |
| | Отримання ключових точок постави | Отримання розташування ключових точок постави для кожного фото |
| | Ручне калібрування точок | Можливість перетягнути ключові точки для зменшення похибки аналізу для кожного фото |
| | Отримання результатів аналізу | Отримання результатів аналізу (чи є порушення постави та рекомендації вправ для її виправлення) на основі вхідних даних для кожного фото |

Структурна схема варіантів використання наведена на рисунку 1.2.

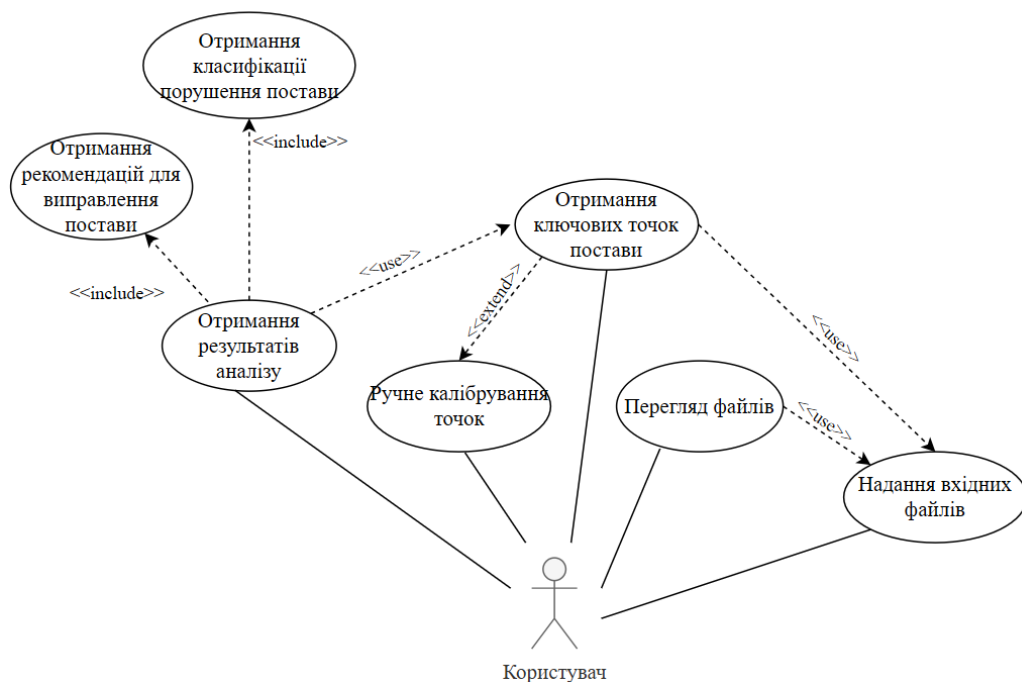


Рисунок 1.2 – Схема варіантів використання

Функціональні вимоги та їх пріоритетність відповідно до визначених варіантів використання наведені у таблиці 1.2.

Таблиця 1.2 – Функціональні вимоги та їх пріоритетність

| Варіант використання | Функціональна вимога | Пріоритет |
|----------------------------------|--|---|
| Надання вхідних файлів | <p>1. Система надає можливість користувачу прикріпити вхідні файли, що містять зображення людини в розслабленій позі.</p> <p>1.1. Система має окреме поле для розміщення файлів.</p> <p>1.2. Система потребує перенесення файлів до цього поля.</p> <p>1.3. Система валідує формат файлу.</p> <p>1.3.1. Система надсилає повідомлення, якщо користувач прикріпив невалідний файл і запрошує повторити дії.</p> | <p>Високий</p> <p>Високий</p> <p>Високий</p> <p>Високий</p> |
| Перегляд файлів | 2. Система надає можливість відображення завантажених файлів перед їх обробкою. | Низький |
| Отримання ключових точок постави | 3. Система відображає ключові точки постави для кожного завантаженого фото. | Високий |
| Ручне калібрування точок | <p>4. Система надає можливість змінити положення знайдених ключових точок</p> <p>4.1 Система дозволяє обрати будь-яку точку.</p> <p>4.2. Система дозволяє перетягнути обрану точку в межах одного фото.</p> | <p>Середній</p> <p>Середній</p> <p>Середній</p> |
| Отримання результатів аналізу | 5. Система надає можливість користувачу отримати результати аналізу на основі виставлених ключових точок постави. | Високий |

Продовження табл. 1.2

| Варіант використання | Функціональна вимога | Пріоритет |
|----------------------|--|--------------------|
| | 5.1. Система надає інформацію щодо класифікації порушення постави. 5.2. Система надає рекомендовані вправи для виправлення постави. | Високий Високий |

1.2 Огляд наявних аналогів

На сьогодні існує чимало різних застосунків, призначених для роботи з поставою людини. Розглянемо 3 популярні програми, які пропонують різні функції для оцінки, моніторингу та корекції постави: *Vodiometer*, *APECS* та *PostureScreen*.

Vodiometer – це застосунок, що надає можливість проводити аналіз постави в реальному часі, одночасно зберігаючи дані для подальшого моніторингу прогресу. Аналіз включає ідентифікацію основних анатомічних орієнтирів та вимірювання кутів між ними, що дозволяє точно оцінити постуральні відхилення. Крім того, цей застосунок пропонує професійну версію, яка призначена для використання в медичних закладах. Ця версія дозволяє лікарям вести облік пацієнтів, відслідковувати їхні результати та зберігати статистику виміряних кутів для подальшого аналізу. Вигляд сторінки з аналізом можна побачити на рисунку 1.3. *Vodiometer* доступний у двох версіях: мобільній, яку можна завантажити з App Store, та десктопній для зручного використання на комп'ютері.

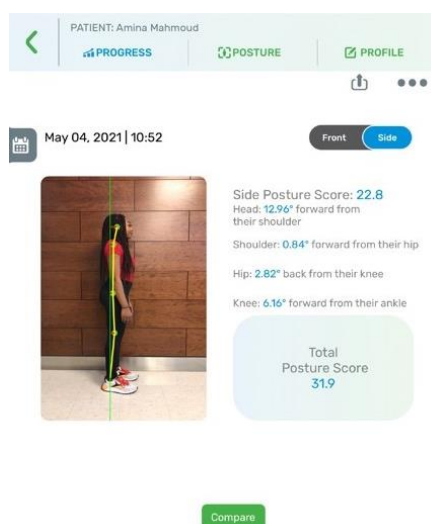


Рисунок 1.3 – Сторінка з аналізом постави в застосунку Vodimeter

APECS – це інструмент, який пропонує оцінку постави з аналізом фронтальної, задньої та бокових проекцій по фото чи відео. Також, користувачі можуть здійснювати оцінку симетрії шиї, плечей і обличчя, а також виконувати спеціальні тести на згинання та вимірювати кути рухливості. Він автоматично розпізнає маркери і створює персоналізовані звіти у форматі PDF. Приклад сторінки з аналізом можна побачити на рисунку 1.4. Додатково, APECS надає щоденні рекомендації та вправи для підтримки правильної постави. Застосунок доступний тільки для завантаження на мобільних пристроях.

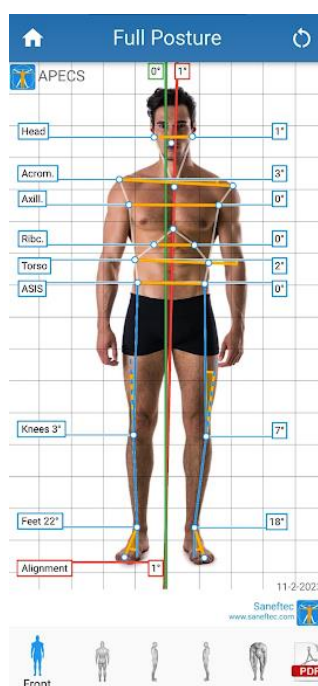


Рисунок 1.4 – Сторінка з аналізом постави в застосунку APECS

PostureScreen – це застосунок для детального аналізу постави та оцінки рухових функцій, таких як присідання або планка. Після проведення аналізу користувачі бачать визначені ключові точки постави та чи є вони симетричними, приклад можна побачити на рисунку 1.5. Цей застосунок дозволяє зберігати результати і легко їх порівняти між собою. Також користувачі можуть отримувати автоматичні рекомендації для корекції дисбалансу м'язів. Застосунок доступний для завантаження в App Store та Google Play.

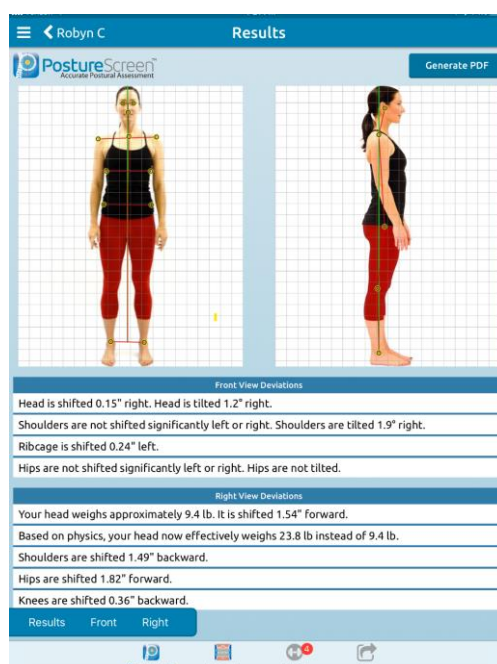


Рисунок 1.5 – Сторінка з аналізом постави в застосунку PostureScreen

Розглянемо функціональні можливості аналогічних систем у порівнянні з системою, що розробляємо (Posture Analyzer). Результати цього аналізу представлені в таблиці 1.3.

Таблиця 1.3 – Порівняльна характеристика аналогів

| Функціональна можливість | Posture Analyzer | Bodiometer | APECS | PostureScreen |
|--|------------------|------------|-------|---------------|
| Аналіз постави по фото | + | - | + | + |
| Аналіз постави по відео | - | - | + | + |
| Аналіз постави в режимі реального часу | - | + | - | - |

Продовження табл. 1.3

| Функціональна можливість | Posture Analyzer | Bodiometer | APECS | PostureScreen |
|--|-------------------------|-------------------|--------------|----------------------|
| Отримання ключових точок постави | + | + | + | + |
| Ручне калібрування точок | + | - | - | - |
| Класифікація порушення постави | + | - | - | - |
| Надання рекомендацій для виправлення постави | + | - | + | + |
| Збереження даних для подальшого моніторингу | - | + | - | + |

Таким чином, ключовими функціональними перевагами розроблюваної системи є можливість ручного калібрування анатомічних точок і автоматизована класифікація порушень постави. Це робить систему більш орієнтованою для звичайних користувачів, які необов'язкові повинні мати спеціальні знання та досвід в цій сфері.

Серед нефункціональних характеристик варто звернути увагу на доступність додатків. Bodiometer має як десктопну, так і мобільну версію, але тільки для iOS. Інші два аналоги доступні виключно на мобільних пристроях з операційними системами iOS та Android. Наша система матиме лише веб-версію, що є її важливою перевагою серед нефункціональних можливостей, адже це забезпечить доступність для користувачів незалежно від платформи.

1.3 Постановка задачі

Метою розробки даної технології є спрощення процесу виявлення ключових анатомічних точок тіла та полегшення аналізу відхилень у поставі.

Для досягнення поставленої мети потрібно вирішити такі задачі:

- створення веб-додатку клієнтської сторони для проведення аналізу, отримання результатів та їх зберігання;

- створення сервісу отримання анатомічних точок постави, використовуючи MediaPipe модель - Pose Landmarker;
- створення моделі нейронної мережі для проведення аналізу та визначення потенційних порушень постави;
- створення веб-інтерфейсу для комунікації веб-додатку клієнтської сторони, сервісу отримання анатомічних точок та нейронної мережі.

2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

За допомогою нотації Йордана-Де Марко відобразимо діаграму потоку даних в даній інформаційній системі. Вона демонструє всі основні компоненти: процеси, зовнішні сутності, сховища даних та потоки даних між ними. Графічна структура потоків даних показана на рисунку 2.1.

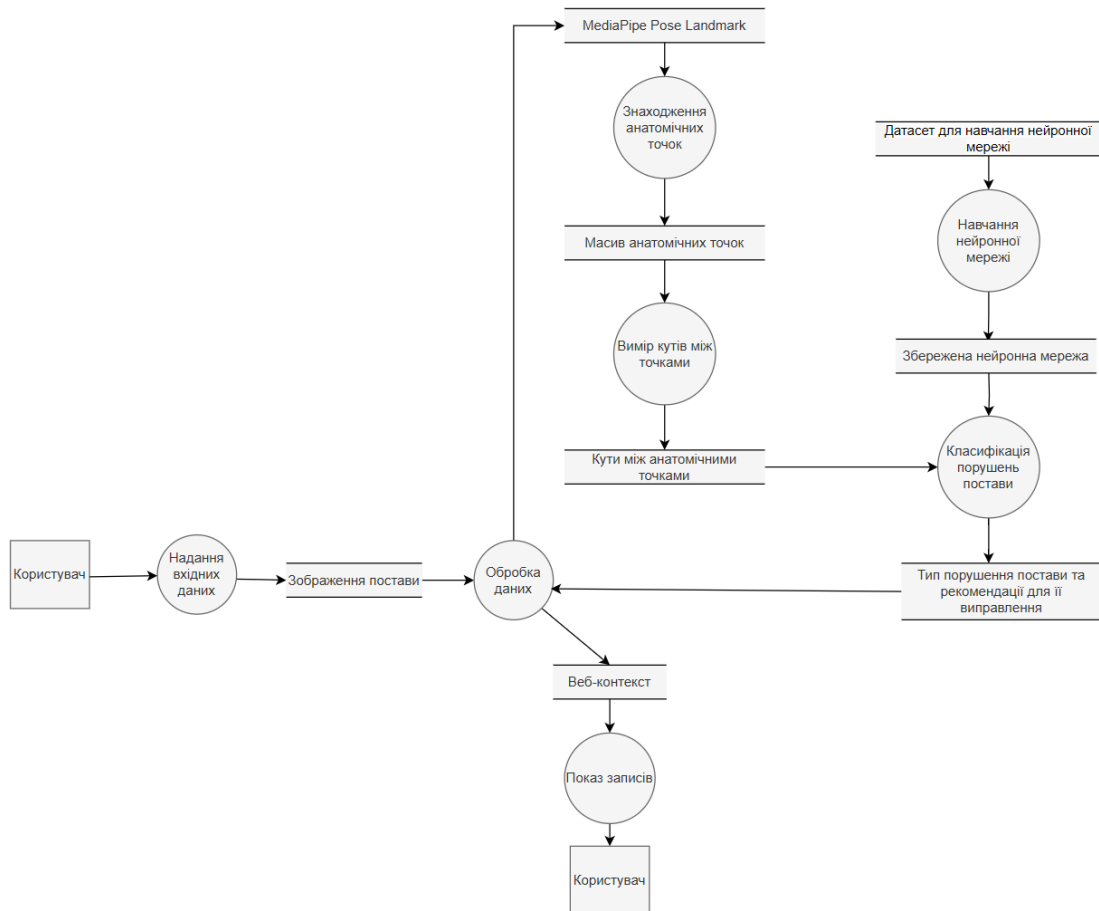


Рисунок 2.1 – Діаграма потоку даних

У цьому розділі буде наведено перелік вхідних та вихідних даних системи. Дивлячись на рисунок 2.1, можна побачити, що одні й ті ж дані можуть бути як вхідними, так і вихідними для різних сутностей системи. Тому вважатимемо, що вхідні дані включають навчальні набори для нейронної мережі, а також інформацію, яку вводить користувач. Вихідними будуть результати, які користувач отримує після проведення аналізу. Усі інші потоки даних, що циркулюють всередині системи, вважатимуться проміжними та забезпечують підтримку основних процесів.

2.1 Вхідні дані

Вхідні дані, які користувач надає для аналізу, – це одна або кілька фотографій, де людина зображена в одній із трьох основних поз: спереду, збоку або ззаду. Ці зображення повинні максимально точно передавати реальну поставу користувача. Важливо, щоб людина на фото була у природному, розслабленому стані, який відображає її справжню поставу в повсякденному житті, без зайвого напруження чи штучної корекції. Це дозволяє отримати більш достовірні результати аналізу.

Зображення можуть бути надані у різних форматах, таких як png або jpg, що забезпечує гнучкість у використанні системи. Це дозволяє користувачам не обмежуватися певним типом файлів і використовувати найпоширеніші формати фотографій для зручного завантаження та обробки зображень. Надання якісних вхідних даних у таких форматах сприяє точнішому аналізу та виявленню відхилень у поставі.

Зовнішній вигляд веб-сторінки для надання вхідних даних для подальшого аналізу представлений на рисунку 2.2.

Рисунок 2.2 – Скріншот веб-сторінки для надання вхідних даних

Вхідні дані, необхідні для навчання нейронної мережі, представлені у вигляді датасету. Оскільки не вдалося знайти вже існуючих датасетів для класифікації поршень постави за знайденими анатомічними точками постави або їх певними характеристиками, було прийнято рішення створити новий власний датасет.

Якщо подавати на вхід нейронній мережі фотографію, це може призвести до великої кількості зайвих даних для обробки, оскільки зображення містить багато пікселів, які не є корисними для вирішення конкретного завдання. З цієї причини було вирішено, що інформаційна система повинна попередньо виділяти анатомічні точки окремо, щоб значно зменшити кількість вхідних параметрів і підвищити ефективність алгоритму.

Існують певні стандартні патерни ліній, які з'єднують ці анатомічні точки, і саме за їх допомогою можна визначати тип постави. Такі лінії та відповідну класифікацію їх використання можна побачити на рисунках 2.3 та 2.4. Однак подання на вхід цих ліній також створює надмірний обсяг інформації для обробки. Тому було прийнято рішення замінити ці лінії на кути між анатомічними точками. Кути дозволяють зберегти всі важливі просторові взаємозв'язки між точками, мінімізуючи кількість оброблюваних даних. Завдяки цьому мережа зосереджується на найрелевантнішій інформації, підвищуючи її здатність виявляти важливі патерни та адаптуватися до різних варіантів вхідних даних без втрати точності.

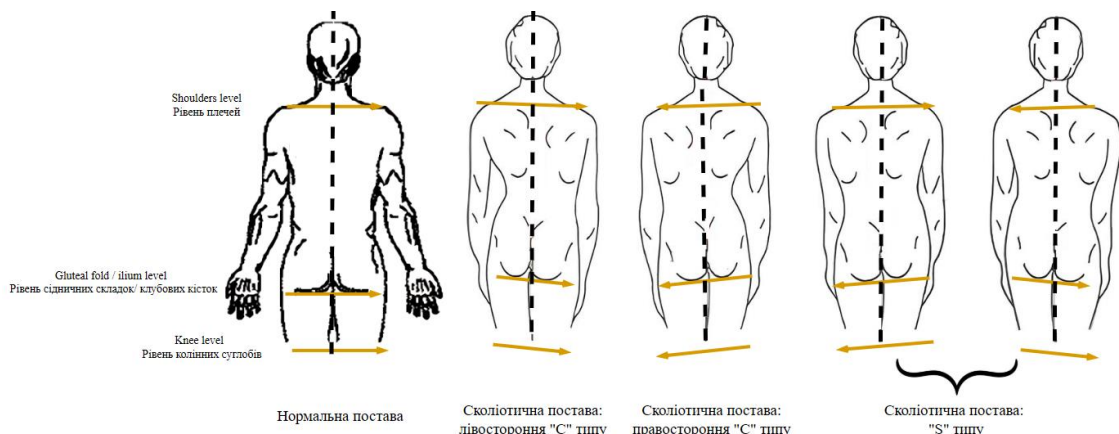


Рисунок 2.3 – Ілюстрація на класів постави, що зображена ззаду на основі анатомічних точок

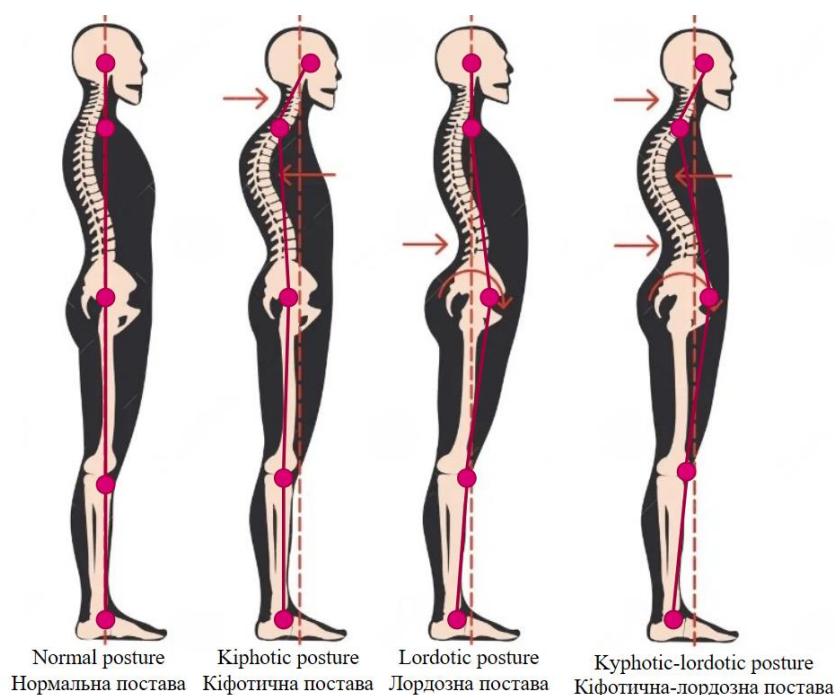


Рисунок 2.4 – Ілюстрація на класів постави, що зображена збоку на основі анатомічних точок

Оскільки важливо враховувати не лише величину кута, а й напрям нахилу прямої, якщо кут не дорівнює 0 градусів, було вирішено запровадити визначення кута відповідно до рисунку 2.5. Таким чином, якщо це вид ззаду чи спереду, і кут знаходиться в діапазоні від 0 до 90 градусів, це означає, що лінії нахилені вправо, а якщо кут від 90 до 180 градусів – вліво. Аналогічний принцип для визначення нахилу ліній для виду збоку. Цей підхід дозволяє забезпечити кращу інтерпретацію для нейронної мережі, щоб вона могла коректно розрізнити різні кути та їхні напрямки.

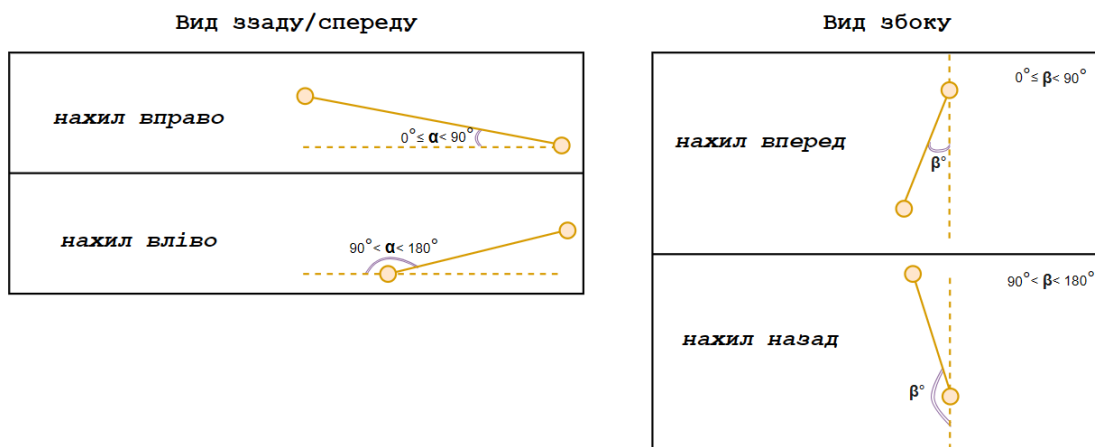


Рисунок 2.5 – Ілюстрація визначення кутів згідно зображення постави

Через те, що кількість кутів, необхідних для класифікації різних ракурсів постави (вид збоку та вид ззаду/спереду), відрізняється, виникла потреба у створенні двох окремих датасетів. Перший датасет, призначений для класифікації порушень постави при огляді збоку, містить 250 записів, розподілених між чотирма різними класами порушень постави. Детальний розподіл записів по кожному з класів представлений у таблиці 2.1. Кожен запис цього датасету складається з п'яти кутів та мітки класу.

Таблиця 2.1 – Розподіл записів по кожному з класів в першому датасеті

| Назва класу | Кількість |
|---------------------------|-----------|
| neutral-posture | 100 |
| kyphotic-posture | 50 |
| lordotic-posture | 50 |
| kyphotic-lordotic-posture | 50 |

Другий датасет, призначений для класифікації порушень постави при огляді ззаду чи спереду, містить 250 записів, розподілених між чотирма різними класами порушень постави. Детальний розподіл записів по класах наведено у таблиці 2.2. Кожен запис цього датасету складається з трьох кутів та мітки класу.

Таблиця 2.2 – Розподіл записів по кожному з класів в першому датасеті

| Назва класу | Кількість |
|---------------------------|-----------|
| neutral-posture | 100 |
| right-c-scoliotic-posture | 50 |
| left-c-scoliotic-posture | 50 |
| s-scoliotic-posture | 50 |

Для створення двох датасетів було зібрано 396 фотографій людей у повний зріст, які демонструють різні типи постави. Знайдені зображення були опубліковані в Kaggle, як два різних датасети: «Posture Types Image Dataset (front/back view)» та «Posture Types Image Dataset (side view)» [16,17]. Після

цього, за допомогою фреймворка Mediapipe та моделі Pose Landmarker Heavy, було визначено ключові анатомічні точки на зображеннях і їх координати. Важливо уточнити, що координати були нормалізовані, оскільки всі зображення були різних розмірів. Наступним кроком, використовуючи Python, були розраховані кути відповідно до схеми, наведеної на рисунку 2.5.

Для збільшення розміру датасету та кількості зразків нормального класу було застосовано аугментацію даних: усі кути нормального класу були відняті від 180. Це дозволило збалансувати обидва датасети. Аналогічний підхід було використано для отримання додаткових 4 записів класу «s-scoliotic-posture». Створені датасети були опубліковані в Kaggle під назвами: «Posture Assessment (Front/Back view)» та «Posture Assessment (Side view)» [18,19]. Структуру датасетів можна побачити на рисунках 2.6-2.7.

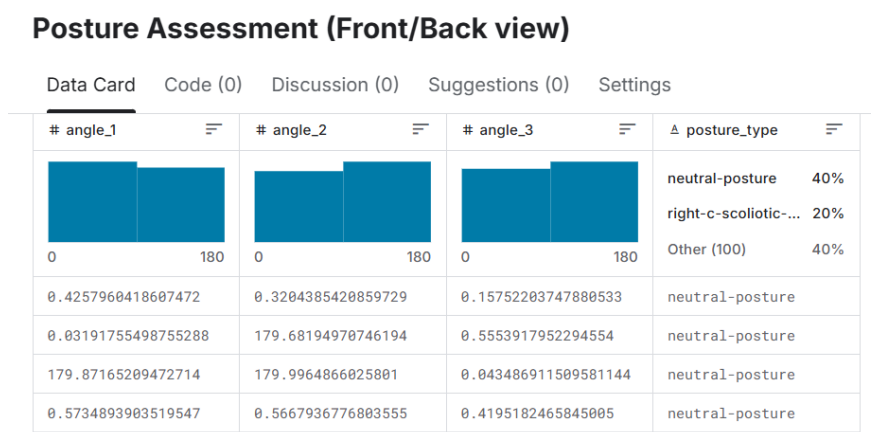


Рисунок 2.6 – Структура датасету «Posture Assessment (Front/Back view)»

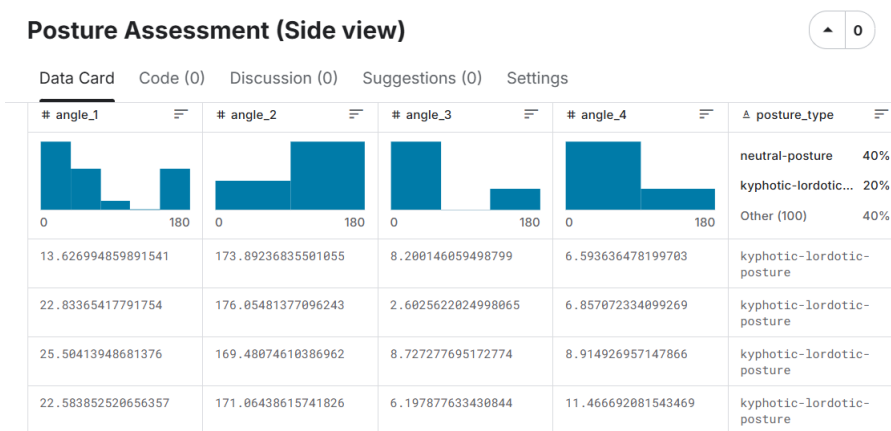


Рисунок 2.7 – Структура датасету «Posture Assessment (Side view)»

2.2 Вихідні дані

Вихідними даними є результат аналізу постави, який містить наступні дані:

- візуальне відображення вхідного фото;
- відображення анатомічних точок на фото та їх з'єднання;
- типи наявних порушень постави та їх ймовірності;
- опис наявних порушень постави та рекомендації для їх виправлення.

Зовнішній вигляд веб-сторінки, що містить вихідні дані, отримані після проведеного аналізу, можна побачити на рисунку 2.8.

The screenshot displays the 'Posture Analyzer' interface. At the top, there's a header with the logo and the title 'Posture Analyzer'. Below it, a progress bar indicates the current step is 'Assessment Results'. The main content area is titled 'Assessment Results' and includes a thank-you message and a brief explanation of the analysis. On the left, a side-view photo of a woman is shown with red lines connecting key anatomical points (ear, shoulder, hip, knee, ankle). On the right, the results are displayed under the heading 'SINGLE'. It states that the assessment was completed using a single photo. Under 'Calculated Angles', four specific angles are listed with their values in degrees. Under 'Posture Classification', it explains that the posture is categorized into specific types (neutral, scoliotic, kyphotic) and provides a classification for the user's posture. A summary box indicates the posture is 'Probably a balanced and ideal posture' with a 52% probability and that no exercises are recommended. Navigation buttons for 'Back' and 'Finish' are located at the bottom.

Posture Analyzer

Analyze your posture

Assessment Results

Thank you for completing your posture assessment! Below, you will find the results generated by our advanced neural network analysis, which evaluates key angles and classifies your posture. Based on these findings, we've also provided tailored recommendations for exercises to help improve your alignment and overall well-being.

SINGLE

★ Your posture assessment has been completed using single photo you provided.

Calculated Angles

Precise angles measured between key anatomical points to identify alignment deviations or asymmetries. These metrics provide a detailed understanding of your posture and areas for improvement.

- Ear-shoulder angle = 0.8951737102110744°
- Shoulder-hip angle = 0.5700884449888735°
- Hip-knee angle = 3.3109415146114434°
- Knee-ankle angle = 0.447614170860561°

Posture Classification

Using neural network algorithms, your posture is categorized into specific classifications (e.g., neutral, scoliotic, kyphotic). This classification helps identify patterns that may contribute to discomfort or movement inefficiencies.

Probably a balanced and ideal posture where the head, shoulders, hips, and ankles are aligned, minimizing strain on muscles and joints. This posture promotes efficiency in movement and reduces the risk of discomfort or injury.

Probability: 52%

Recommended Exercises:

🚫 Not required.

Back Finish

Рисунок 2.8 – Скріншот веб-сторінки для перегляду вихідних даних

3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Змістовна постановка задачі

Маємо вибірку невеликого розміру з точками $(x_1, y_1, l_1) \dots (x_n, y_n, l_n)$, що представляють анатомічні точки постави, позначені множиною X , з урахуванням ракурсу постави R та задана множина типів порушення постави Y .

Якщо доступні кілька вибірок точок, упорядкованих за часом, які мають однаковий ракурс R , необхідно обчислити їх математичне сподівання для кожної координати. Це дозволить зменшити статистичну похибку, пов'язану з індивідуальними відхиленнями у вибірках. На основі отриманих узагальнених значень можна продовжувати вирішення задачі класифікації типів порушення постави.

Необхідно за допомогою вхідних точок системи координат та ракурсу постави знайти кути нахилу $\alpha_1 \dots \alpha_m$. Якщо ракурс постави R - це вид збоку, то $m = n$, в іншому випадку - $m = \frac{n}{2}$.

Далі розробити архітектуру нейронної мережі на основі створеної навчальної вибірки та натренувати її, щоб вона могла класифікувати вхідні кути $\alpha_1 \dots \alpha_m$ за типом порушення постави з множини Y та обчислити ймовірність їх належності до кожного з класів.

3.2 Математична модель

Позначимо X – множина точок системи координат (x, y) з міткою назви анатомічної точки l , R – ракурс постави, Y – множина типів порушення постави.

Припустимо, що було задано декілька множин X : X_1, X_2, \dots, X_z , де $z \geq 4$. Необхідно знайти математичне сподівання точок для кожної координати (x_{ij}, y_{ij}) , які належать кожній множині X_i ($i=1 \dots z$ та $j=1 \dots n$). Це дозволить сформулювати нову агреговану множину X .

Щоб врахувати різні впливи множин X_1, X_2, \dots, X_z побудуємо таблицю ваг w_i для кожної множини X_i . На основі досліджень, опублікованих у журналі

«BMC Sports Science, Medicine and Rehabilitation», відомо, що якість постави погіршується у вечірній час [14]. Крім того, існує факт, що під час сну м'язи та зв'язки спини розслабляються та відновлюються, тому після сну постава зазвичай здоровіша, а самопочуття людини краще [15].

Виходячи з цього, можна стверджувати, що наявні проблеми постави є більш помітними у вечірній час, через те, що ціллю даної технології є саме виявлення можливих проблем, тобто сконцентрованість на негативних результатах. Множині X_z (останній вибірці, яка відповідає вечірнім даним) присвоюється найбільша вага $w_z = 0.4$. Для інших множин ваги розподіляються рівномірно, щоб їх сума дорівнювала одиниці: $\frac{0.6}{z-1}$. Варто зазначити, що статистичних даних, які б точно обґрунтовували саме ці значення ваг, не знайдено. Проте такий вибір зроблено для акценту на вечірніх даних, оскільки саме вони важливі для аналізу ранніх проявів проблем із поставою. Ваги занесені у таблицю 3.1.

Таблиця 3.1 – Таблиця ваг для множин X_i

| Множина X_i | X_1 | X_2 | ... | X_z |
|---------------|-------------------|-------------------|-------------------|-------|
| Вага w_i | $\frac{0.6}{z-1}$ | $\frac{0.6}{z-1}$ | $\frac{0.6}{z-1}$ | 0.4 |

Дивлячись на цей опис можна побачити, що дану модель можна представити за допомогою теорії ймовірностей та математичної статистики, розглядаючи множини X_i як події певної випадкової величини E – положення ключових точок постави протягом дня, а ваги w_i , відповідно, ймовірностями цих подій $p_i = P(E_i)$, розглядаючи наявний простір подій як повний ($\sum p_i = 1$). Таким чином, в таблиці 3.2 можна побачити представлення випадкової множини E .

Таблиця 3.2 – Таблиця випадкової множини E

| Подія E_i | E_1 | E_2 | ... | E_z |
|-------------------|-------------------|-------------------|-------------------|-------|
| Ймовірність p_i | $\frac{0.6}{z-1}$ | $\frac{0.6}{z-1}$ | $\frac{0.6}{z-1}$ | 0.4 |

Математичне сподівання для кожної координати (x_{ij}, y_{ij}) з однаковою міткою l обчислюється за зваженою сумою:

$$\bar{x}_j = \sum_{i=1}^Z p_i * x_{ij}, \quad \bar{y}_j = \sum_{i=1}^Z p_i * y_{ij} \quad (3.1)$$

Отримані координати $(\bar{x}_j, \bar{y}_j, l)$ утворюють нову множину X , яка враховує всі задані вибірки з їх вагами.

Далі потрібно знайти кути нахилу постави за допомогою множини X . Для цього візьмемо за основу рисунок 2.5 та проведемо перпендикуляри, щоб отримати прямокутні трикутники. Нові зображення трикутників зі знайденими катетами можна побачити на рисунку 3.1.

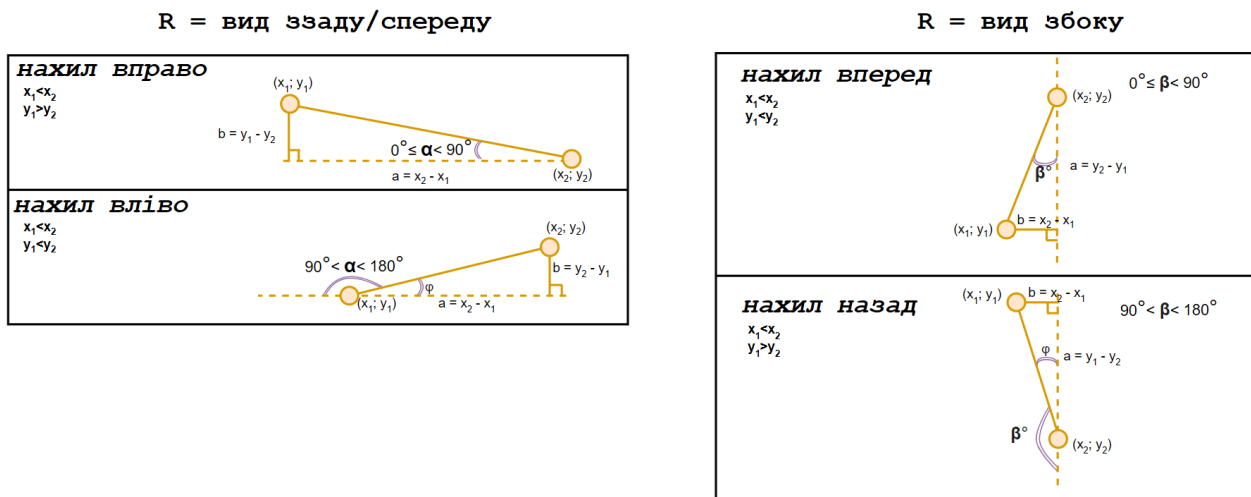


Рисунок 3.1 – Ілюстрація визначення кутів нахилу постави $\alpha_1 \dots \alpha_m$

Розглянемо випадок $R = \text{вид ззаду/спереду}$ та обиремо дві точки з спільною назвою l . Якщо $x_1 < x_2$ та $y_1 > y_2$, то знайти бажаний кут α_1 можна за наступною формулою, яка впливає з тригонометричних функцій гострого кута в прямокутному трикутнику:

$$\alpha_n = \tan^{-1} \frac{b}{a} = \tan^{-1} \frac{y_n - y_{n+1}}{x_{n+1} - x_n} \quad (3.2)$$

Якщо $x_1 < x_2$ та $y_1 < y_2$, то тоді обираємо наступну формулу:

$$\alpha_n = 180^\circ - \varphi = \tan^{-1} \frac{b}{a} = 180^\circ - \tan^{-1} \frac{y_{n+1} - y_n}{x_{n+1} - x_n} \quad (3.3)$$

Аналогічний розв’язок для другого випадку, коли $R = \text{вид збоку}$. Фінальні формули наступні:

$$\alpha_n = \begin{cases} \tan^{-1} \frac{x_{n+1} - x_n}{y_{n+1} - y_n}, \text{ якщо } x_{n+1} > x_n \text{ та } y_{n+1} > y_n \\ 180^\circ - \tan^{-1} \frac{x_{n+1} - x_n}{y_n - y_{n+1}}, \text{ якщо } x_{n+1} > x_n \text{ та } y_{n+1} < y_n \end{cases} \quad (3.4)$$

Використовуючи формули 3.1 – 3.3, отримуємо кути нахилу $\{\alpha_1 \dots \alpha_m\}$, які є елементом множини A . Присутня невідома цільова залежність – визначення $\hat{y}: A \rightarrow Y$. Її значення є відомими лише на елементах множини навчального датасету:

$$A^v = \{(\alpha_1, y_1), \dots, (\alpha_m, y_m)\}. \quad (3.5)$$

Введемо додаткову змінну $k = P(Y)$ – потужність множини типів порушення постави.

Наступним кроком слід визначити необхідні шари нейронної мережі та провести її тренування, щоб вона могла класифікувати належність будь-якого елемента з множини A до множини Y .

Штучні нейрони були спроектовані за аналогією з біологічними, тому їхня структура має багато спільного і показана на рисунку 3.2. Вони включають вхідні нейрони для отримання даних, зв'язки між ними та центральним штучним нейроном, механізм обробки інформації на основі вхідних даних для формування результату, а також передачу вихідного значення.

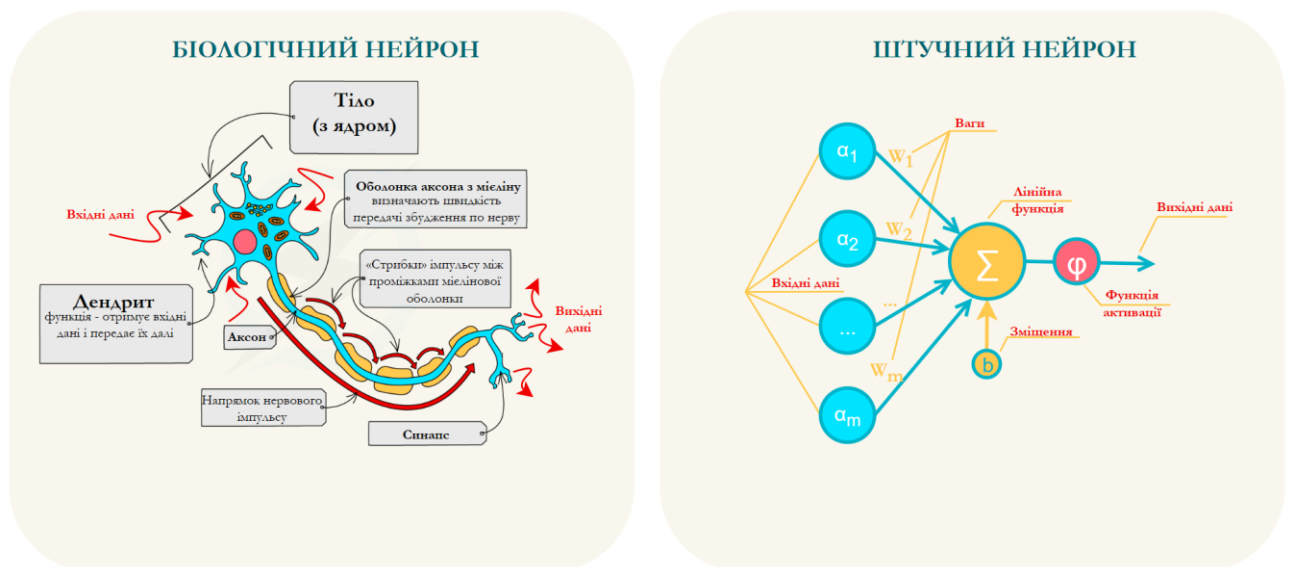


Рисунок 3.2 – Будова біологічного та штучного нейронів

Процес прямого поширення, тобто передача даних через нейронну

мережу, складається з трьох кроків.

Перший крок – це обчислення суми кожного елементу множини A (α_i) помноженого на відповідну вагу w_i для кожного класу k :

$$\sum \alpha_i w_{ik} = (\alpha_1 \times w_{1k}) + (\alpha_2 \times w_{2k}) + \dots + (\alpha_m \times w_{mk}). \quad (3.6)$$

Вагові коефіцієнти w_{ik} показують ступінь впливу вхідних даних множини A на вихідні значення для класу k . Чим більше значення коефіцієнту, тим суттєвіший вплив відповідного елементу α_i на кінцевий результат.

На другому кроці вводиться параметр зміщення b , який коригує зважену суму, обчислену на попередньому кроці за формулою 3.6. Цей параметр є індивідуальним для кожного класу, що дозволяє адаптувати модель для точнішого відображення характеристик кожної категорії даних.

$$z_k = \sum \alpha_i w_{ik} + b_k. \quad (3.7)$$

Фінальний крок процесу прямого поширення передбачає застосування нелінійної функції активації до значень z_k . Оскільки задача стосується мультикласифікації, обирається функція softmax, яка визначається за формулою:

$$\hat{y}_k = S(z_1, \dots, z_k)_k = \frac{\exp(z_i)}{\sum_j^k \exp(z_j)}. \quad (3.8)$$

Результатом є вектор, сума елементів якого дорівнює одиниці. Тобто кожен його елемент - це ймовірність належності до відповідного класу k .

Алгоритм навчання нейронної мережі складається з двох основних етапів: зворотнього поширення та оптимізації. Зворотне поширення є ключовим етапом, що включає обчислення градієнтів функції витрат відносно вагів і параметрів мережі.

Перший крок – це обчислення функції витрат, яка оцінює відхилення між прогнозованими значеннями для кожного класу та цільовими результатами. Для мультикласових задач використовується функція Categorical Cross-Entropy, яка вимірює різницю між фактичними і прогнозованими ймовірностями. Вона має логарифмічну форму, що дозволяє

зменшити вплив помилок при великих відхиленнях і зберігати точність при малих помилках:

$$C_{CE} = - \sum_{i=1}^k t_i \cdot \log(\hat{y}_i), \quad (3.9)$$

де $i=1, \dots, k$ – кількість класів, t_i - прогнозована ймовірність для класу i та y_i – фактична ймовірність для класу i .

На другому кроці обчислюють градієнти, які визначають напрямок і величину змін вхідних параметрів для мінімізації втрат. Для цього використовується диференціювання функції витрат [16, 18], після обчислення якого маємо формули 3.10-3.12, де s - це число зразків у навчальному наборі даних.

Градієнт функції витрат відносно z :

$$dz = \hat{y} - y. \quad (3.10)$$

Градієнт функції витрат відносно ваг:

$$dw = \frac{1}{s} \times A^T \times dz. \quad (3.11)$$

Градієнт функції витрат відносно зміщення:

$$db = \frac{1}{s} \times \sum dz. \quad (3.12)$$

Заключним кроком є оптимізація, яка спрямована на підбір найоптимальніших значень параметрів (ваг і зміщень) на основі розрахованих градієнтів. Серед найбільш популярних алгоритмів для цього використовується Adam (Adaptive Moment Estimation). Цей метод поєднує сильні сторони двох інших алгоритмів: градієнтного спуску з імпульсом (Momentum) і RMSProp. Adam здійснює коригування швидкості навчання для кожного параметра, використовуючи оцінки першого та другого моментів градієнта [19].

Формула для оновлення ваг:

$$W_t = W_{t-1} - \alpha \times \frac{\frac{\beta_1 \times v_{t-1} + (1-\beta_1) \times dw}{1-\beta_1^t}}{\sqrt{\frac{\beta_2 \times s_{t-1} + (1-\beta_2) \times dw^2}{1-\beta_2^t} + \epsilon}}. \quad (3.13)$$

Формула для оновлення зміщень:

$$b_t = b_{t-1} - \alpha \times \frac{\frac{\beta_1 \times v_{t-1} + (1 - \beta_1) \times dw}{1 - \beta_1^t}}{\sqrt{\frac{\beta_2 \times s_{t-1} + (1 - \beta_2) \times dw^2}{1 - \beta_2^t} + \varepsilon}}. \quad (3.14)$$

Типові значення гіперпараметрів: $\beta_1 = 0.9, \beta_2 = 0.999, \varepsilon = 1e - 8$ [19].

3.3 Опис методу розв'язання

Для вирішення задачі класифікації кутів між анатомічними точками з різних ракурсів було розроблено дві моделі нейронних мереж. Це обумовлено тим, що кількість кутів варіюється залежно від ракурсу. Обидві моделі мають однакову архітектуру шарів з різним гіперпараметром, що відповідає за розмірність вихідного простору. Реалізація була виконана за допомогою бібліотеки Keras, яка забезпечує зручність у створенні та налаштуванні нейронних мереж.

Вхідний шар – це щільний шар (Dense) із функцією активації ReLU. Кількість вхідних даних залежить від ракурсу: 3 (вид ззаду/спереду) або 4 (вид збоку) параметри. Принцип роботи ReLU полягає у тому, що кожен нейрон обчислює зважену суму вхідних даних, додаючи зміщення. Якщо результат сумування від'ємний, то ReLU переводить його в нуль. Якщо додатний – значення залишається без змін. Формулу цієї функції активації та її графічне відображення проілюстровано на рисунку 3.3. Активоване значення передається до наступного шару, що забезпечує глибше розуміння моделі даних в процесі тренування моделі.

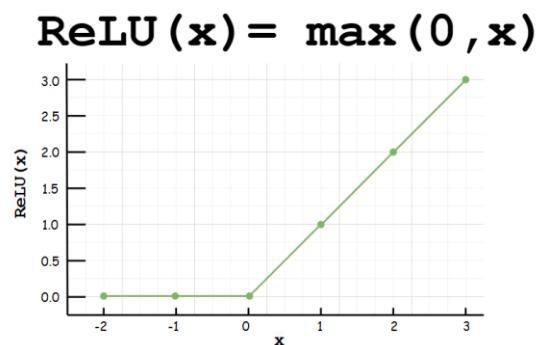


Рисунок 3.3 – Графічне відображення функції ReLU

Прихованні шари – це кілька Dense-шарів із функцією активації ReLU. Між ними було додано Dropout-шари для запобігання перенавчанню, шляхом

випадкового виключення заданого відсотка значень нейронів. Оскільки датасети маленькі, то відсоток був обраний у межах 10–30%, щоб зберегти баланс між точністю та регуляризацією.

Вихідний шар – це також щільний шар (Dense), але із функцією активації softmax. Цей шар обчислює ймовірності належності кожного зразка до одного з 4 класів.

Для мультикласової задачі була обрана Categorical Cross-Entropy функція витрат та алгоритм Adam для оптимізації з стандартними налаштуваннями. Створені моделі можна побачити на рисунках 3.4, 3.5.

| Layer (type) | Output Shape | Param # |
|---------------------|--------------|---------|
| dense (Dense) | (None, 300) | 1200 |
| dense_1 (Dense) | (None, 200) | 60200 |
| dropout (Dropout) | (None, 200) | 0 |
| dense_2 (Dense) | (None, 150) | 30150 |
| dropout_1 (Dropout) | (None, 150) | 0 |
| dense_3 (Dense) | (None, 100) | 15100 |
| dropout_2 (Dropout) | (None, 100) | 0 |
| dense_4 (Dense) | (None, 70) | 7070 |
| dropout_3 (Dropout) | (None, 70) | 0 |
| dense_5 (Dense) | (None, 20) | 1420 |
| dense_6 (Dense) | (None, 4) | 84 |

Total params: 115224 (450.09 KB)
 Trainable params: 115224 (450.09 KB)
 Non-trainable params: 0 (0.00 Byte)

Рисунок 3.4 – Побудована модель нейромережі для класифікації кутів постави з ракурсу ззаду/спереду

| Layer (type) | Output Shape | Param # |
|---------------------|--------------|---------|
| dense (Dense) | (None, 350) | 1750 |
| dense_1 (Dense) | (None, 250) | 87750 |
| dropout (Dropout) | (None, 250) | 0 |
| dense_2 (Dense) | (None, 200) | 50200 |
| dropout_1 (Dropout) | (None, 200) | 0 |
| dense_3 (Dense) | (None, 150) | 30150 |
| dropout_2 (Dropout) | (None, 150) | 0 |
| dense_4 (Dense) | (None, 90) | 13590 |
| dropout_3 (Dropout) | (None, 90) | 0 |
| dense_5 (Dense) | (None, 40) | 3640 |
| dense_6 (Dense) | (None, 4) | 164 |

Total params: 187244 (731.42 KB)
 Trainable params: 187244 (731.42 KB)
 Non-trainable params: 0 (0.00 Byte)

Рисунок 3.5 – Побудована модель нейромережі для класифікації кутів постави з ракурсу збоку

3.3.1 Тренування та тестування моделей

Навчання моделей здійснювалося на повному обсязі датасету через його невеликий обсяг із використанням `batch_size=10` протягом 25–30 епох. Це забезпечувало наявність 2–3 зразків одного класу в кожній ітерації. Результати навчання та тестування представлені на рисунках 3.6 і 3.7.

```

25/25 [=====] - 0s 4ms/step - loss: 0.9397 - acc: 0.6080
Epoch 12/25
25/25 [=====] - 0s 4ms/step - loss: 0.8217 - acc: 0.6480
Epoch 13/25
25/25 [=====] - 0s 4ms/step - loss: 0.8107 - acc: 0.6680
Epoch 14/25
25/25 [=====] - 0s 4ms/step - loss: 0.8352 - acc: 0.6360
Epoch 15/25
25/25 [=====] - 0s 4ms/step - loss: 0.8576 - acc: 0.6000
Epoch 16/25
25/25 [=====] - 0s 4ms/step - loss: 0.7618 - acc: 0.6480
Epoch 17/25
25/25 [=====] - 0s 4ms/step - loss: 0.9370 - acc: 0.6520
Epoch 18/25
25/25 [=====] - 0s 4ms/step - loss: 0.8371 - acc: 0.6280
Epoch 19/25
25/25 [=====] - 0s 4ms/step - loss: 0.8062 - acc: 0.6480
Epoch 20/25
25/25 [=====] - 0s 4ms/step - loss: 0.7689 - acc: 0.7040
Epoch 21/25
25/25 [=====] - 0s 4ms/step - loss: 0.6819 - acc: 0.7360
Epoch 22/25
25/25 [=====] - 0s 4ms/step - loss: 0.7196 - acc: 0.7040
Epoch 23/25
25/25 [=====] - 0s 4ms/step - loss: 0.7272 - acc: 0.7200
Epoch 24/25
25/25 [=====] - 0s 4ms/step - loss: 0.7691 - acc: 0.6760
Epoch 25/25
25/25 [=====] - 0s 4ms/step - loss: 0.7109 - acc: 0.7240
Accuracy: 78.48%

```

Рисунок 3.6 – Результати тренування та тестування побудованої моделі для класифікації кутів постави з ракурсу ззаду/спереду

```

25/25 [=====] - 0s 4ms/step - loss: 1.2032 - acc: 0.4900
Epoch 5/30
25/25 [=====] - 0s 3ms/step - loss: 1.1011 - acc: 0.5200
Epoch 6/30
25/25 [=====] - 0s 5ms/step - loss: 1.1508 - acc: 0.4800
Epoch 7/30
25/25 [=====] - 0s 4ms/step - loss: 1.0420 - acc: 0.5280
Epoch 8/30
25/25 [=====] - 0s 3ms/step - loss: 1.0434 - acc: 0.5400
Epoch 9/30
25/25 [=====] - 0s 3ms/step - loss: 1.0878 - acc: 0.5200
Epoch 10/30
25/25 [=====] - 0s 3ms/step - loss: 0.9104 - acc: 0.5880
Epoch 11/30
25/25 [=====] - 0s 3ms/step - loss: 0.8950 - acc: 0.6360
Epoch 12/30
25/25 [=====] - 0s 3ms/step - loss: 0.8681 - acc: 0.6200
Epoch 13/30
25/25 [=====] - 0s 3ms/step - loss: 0.7440 - acc: 0.6880
Epoch 14/30
25/25 [=====] - 0s 3ms/step - loss: 0.8093 - acc: 0.6280
Epoch 15/30
25/25 [=====] - 0s 3ms/step - loss: 0.7437 - acc: 0.6680
Epoch 16/30
25/25 [=====] - 0s 3ms/step - loss: 0.6999 - acc: 0.7080
Epoch 17/30
25/25 [=====] - 0s 3ms/step - loss: 0.7877 - acc: 0.6560
Epoch 18/30
25/25 [=====] - 0s 3ms/step - loss: 0.7065 - acc: 0.7160
Epoch 19/30
25/25 [=====] - 0s 3ms/step - loss: 0.7494 - acc: 0.6920
Epoch 20/30
25/25 [=====] - 0s 3ms/step - loss: 0.7300 - acc: 0.6840
Epoch 21/30
25/25 [=====] - 0s 3ms/step - loss: 0.6599 - acc: 0.7160
Epoch 22/30
25/25 [=====] - 0s 3ms/step - loss: 0.6558 - acc: 0.7480
Epoch 23/30
25/25 [=====] - 0s 3ms/step - loss: 0.6856 - acc: 0.6720
Epoch 24/30
25/25 [=====] - 0s 3ms/step - loss: 0.5962 - acc: 0.7560
Epoch 25/30
25/25 [=====] - 0s 3ms/step - loss: 0.5895 - acc: 0.7480
Epoch 26/30
25/25 [=====] - 0s 3ms/step - loss: 0.5532 - acc: 0.7760
Epoch 27/30
25/25 [=====] - 0s 3ms/step - loss: 0.6547 - acc: 0.7120
Epoch 28/30
25/25 [=====] - 0s 3ms/step - loss: 0.6060 - acc: 0.7080
Epoch 29/30
25/25 [=====] - 0s 3ms/step - loss: 0.5941 - acc: 0.7560
Epoch 30/30
25/25 [=====] - 0s 3ms/step - loss: 0.6112 - acc: 0.7080
Accuracy: 80.80%

```

Рисунок 3.7 – Результати тренування та тестування побудованої моделі для класифікації кутів постави з ракурсу збоку

Як видно з процесів тренування, під час навчання з кожною епохою втрати поступово знижуються, а точність моделей збільшується. Після тестування було отримано точність 78.40% для класифікації кутів із ракурсу спереду/ззаду та 80.80% для ракурсу збоку. Однак це не гарантує аналогічної точності на нових даних. Для оцінки реальної ефективності моделі проводилася валідація.

Для перевірки здатності моделей працювати з незнайомими даними використовувалися нові зразки для валідації. З цією метою було обрано по два зразки для кожного класу. Результати валідації наведені на рисунках 3.8 і 3.9. На їх основі можна зробити висновок, що моделі здатні розрізняти всі 4 класи з імовірністю 49–76% для ракурсу ззаду/спереду та 39–95% для ракурсу збоку. Найменшу точність демонструє нормальний клас, оскільки його патерни частково присутні в інших класах і через невеликий розмір датасетів моделі не можуть повністю зафіксувати відмінності між цими класами.

```
{'left-c-scoliotic-posture': 0, 'neutral-posture': 1, 'right-c-scoliotic-posture': 2, 's-scoliotic-posture': 3}

item#0: 0.7333422303199768 - right-c-scoliotic-posture, but expected - right-c-scoliotic-posture
[9.937501488366252e-08, 0.266306072473526, 0.7333422303199768, 0.0003515348944347352]

item#1: 0.7309714555740356 - right-c-scoliotic-posture, but expected - right-c-scoliotic-posture
[1.1453413861772788e-07, 0.2686540484428406, 0.7309714555740356, 0.00037444028217624873]

item#2: 0.7600162029266357 - left-c-scoliotic-posture, but expected - left-c-scoliotic-posture
[0.7600162029266357, 0.2290724664926529, 0.0079876147210598, 0.002923672553151846]

item#3: 0.74659264087677 - left-c-scoliotic-posture, but expected - left-c-scoliotic-posture
[0.74659264087677, 0.22840559482574463, 0.017671357840299606, 0.007330387830734253]

item#4: 0.7415964603424072 - s-scoliotic-posture, but expected - s-scoliotic-posture
[0.03561054915189743, 0.2078063189983368, 0.01498666126281023, 0.7415964603424072]

item#5: 0.592257559299469 - s-scoliotic-posture, but expected - s-scoliotic-posture
[0.026696693152189255, 0.28605058789253235, 0.09499520063400269, 0.592257559299469]

item#6: 0.717292845249176 - neutral-posture, but expected - neutral-posture
[0.006731148809194565, 0.717292845249176, 0.22953589260578156, 0.04644003137946129]

item#7: 0.4949176609516144 - neutral-posture, but expected - neutral-posture
[0.49410557746887207, 0.4949176609516144, 0.0075353956781327724, 0.0034413624089211226]
```

Рисунок 3.8 – Результати валідації побудованої моделі для класифікації кутів постави з ракурсу ззаду/спереду

```

{'kyphotic-lordotic-posture': 0, 'kyphotic-posture': 1, 'lordotic-posture': 2, 'neutral-posture': 3}

item#0: 0.3944835364818573 - neutral-posture, but expected - neutral-posture
[0.10999041795730591, 0.3117409646511078, 0.1837850660085678, 0.3944835364818573]

item#1: 0.4752182066440582 - neutral-posture, but expected - neutral-posture
[0.09647475928068161, 0.24114352464675903, 0.18716351687908173, 0.4752182066440582]

item#2: 0.5088921785354614 - lordotic-posture, but expected - lordotic-posture
[0.043964728713035583, 0.018681805580854416, 0.5088921785354614, 0.4284612834453583]

item#3: 0.9541523456573486 - lordotic-posture, but expected - lordotic-posture
[0.0008869990124367177, 0.0005209141527302563, 0.9541523456573486, 0.04443976283073425]

item#4: 0.9339762926101685 - kyphotic-posture, but expected - kyphotic-posture
[2.8519659736048197e-06, 0.9339762926101685, 0.0003227521665394306, 0.06569813191890717]

item#5: 0.6130033135414124 - kyphotic-posture, but expected - kyphotic-posture
[0.010897825472056866, 0.6130033135414124, 0.04724566265940666, 0.3288532495498657]

item#6: 0.6385295391082764 - kyphotic-lordotic-posture, but expected - kyphotic-lordotic-posture
[0.6385295391082764, 0.35518962144851685, 0.004806295968592167, 0.0014745668740943074]

item#7: 0.659355878829956 - kyphotic-lordotic-posture, but expected - kyphotic-lordotic-posture
[0.659355878829956, 0.3372158110141754, 0.002681941259652376, 0.0007464119698852301]

```

Рисунок 3.9 – Результати валідації побудованої моделі для класифікації кутів постави з ракурсу збоку

4 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

4.1. Вибір засобів розробки

Програмне забезпечення включає такі основні компоненти: клієнтську частину, REST API для отримання координат анатомічних точок, дві розроблені нейронні мережі для класифікації постави за кутами та REST API для отримання результатів класифікації створених нейронних мереж.

Для створення всіх частин використовувалося середовище розробки Visual Studio Code, яке забезпечує зручність написання та редагування коду на різних мовах програмування.

Тестування двох API проводилося за допомогою Postman — інструмента, що дозволяє виконувати HTTP-запити, тестувати роботу сервісів і отримувати їхні відповіді.

Для реалізації API для визначення анатомічних точок постави, а також для створення двох моделей нейронних мереж і API для прогнозування типу постави на основі цих моделей, була обрана мова програмування Python. Її вибір зумовлений широким спектром бібліотек і фреймворків, зокрема для роботи з нейронними мережами.

Основні бібліотеки та фреймворки, використані для створення нейронних мереж і API для класифікації постави:

- Keras;
- Pandas;
- NumPy;
- FastAPI.

Keras – це високорівневий фреймворк для роботи з нейронними мережами. Він надає широкий набір попередньо налаштованих моделей нейронних шарів, функцій активації, оптимізаторів та функцій втрат, які легко адаптувати під потреби конкретного завдання.

Pandas – це потужний інструмент для аналізу даних, що дозволяє ефективно працювати з таблицями й обробляти великі обсяги інформації.

NumPy – це бібліотека, яка надає інструменти для роботи з

багатовимірними масивами, а також містить широкий набір математичних функцій для реалізації складних обчислень.

FastAPI – це сучасний фреймворк для створення RESTful API з підтримкою інтегрованої документації, валідації даних і механізмів авторизації та аутентифікації.

Бібліотеки та фреймворки, використані для створення API для отримання анатомічних точок постави:

- MediaPipe;
- NumPy;
- FastAPI.

MediaPipe – це фреймворк для обробки даних комп'ютерного зору, що дозволяє в реальному часі або по фотографії/відео визначати координати ключових анатомічних точок людського тіла.

Для розробки клієнтської частини було обрано мову програмування TypeScript, яка є надбудовою над JavaScript. Вона широко використовується для створення веб-додатків завдяки можливості статичної типізації, що дозволяє виявляти потенційні помилки ще на етапі написання коду. Це значно підвищує якість та надійність розробки. Для запуску коду застосовувалася платформа Node.js, яка забезпечує виконання JavaScript-коду на стороні сервера.

Основні бібліотеки, використані для реалізації клієнтської частини:

- React;
- React-dom;
- React-router-dom;
- React-draggable;
- Bootstrap;
- React-bootstrap;
- Antd;
- Axios;
- Rfdc.

React – популярна бібліотека для створення динамічних інтерфейсів користувача на основі компонентного підходу.

React-dom – це бібліотека, яка є ключовим компонентом екосистеми React і забезпечує інтеграцію React-компонентів із DOM-деревом веб-додатка.

React-router-dom – інструмент для створення маршрутизації та управління переходами між сторінками у веб-додатках.

React-draggable – це бібліотека для роботи з перетягуванням елементів інтерфейсу. Вона дозволяє реалізовувати функціонал drag-and-drop для різних елементів, таких як вікна, блоки або інтерактивні компоненти.

Bootstrap – CSS-фреймворк, що забезпечує набір стилів та готових компонентів для швидкої розробки адаптивних інтерфейсів.

React-bootstrap – це версія Bootstrap, адаптована для роботи з React-компонентами.

Antd (Ant Design) – це набір компонентів інтерфейсу, розроблений для створення професійних та стильних веб-додатків. В цьому проєкті було використано компонент Upload для завантаження файлів.

Axios – це бібліотека для здійснення HTTP-запитів, зручна для обміну даними між клієнтською та серверною частинами.

Rfdc – це швидка та ефективна бібліотека для глибокого клонування об'єктів.

4.2. Архітектура програмного забезпечення

Для даного програмного забезпечення було обрано мікросервісну архітектуру. Цей підхід базується на розподілі програми на невеликі, автономні модулі (мікросервіси), які працюють незалежно один від одного. Кожен мікросервіс виконує конкретну бізнес-логіку і може розроблятися, розгортатися та масштабуватися окремо.

Ключова ідея мікросервісної архітектури полягає у створенні слабо пов'язаних компонентів, що взаємодіють між собою через прості протоколи, наприклад, HTTP або інші стандарти інтерфейсів. Це дозволяє реалізувати

гнучке управління системою, знижуючи залежності між її частинами. Завдяки такому підходу, кожен компонент може бути реалізований на різних мовах програмування або з використанням різноманітних технологій, що відповідають його функціональним вимогам.

До основних переваг мікросервісної архітектури можна віднести автономність, масштабованість, гнучкість у виборі технологій, стійкість до збоїв та повторне використання.

Автономність означає, що кожен мікросервіс є незалежним модулем із власною кодовою базою, базою даних і середовищем виконання. Це забезпечує можливість паралельної розробки, тестування та розгортання різних частин програми.

Друга перевага це те, що мікросервіси можна масштабувати окремо, виділяючи додаткові ресурси для тих компонентів, які мають підвищене навантаження. Це дозволяє ефективно використовувати ресурси без необхідності масштабувати всю систему.

Під гнучкістю у виборі технологій мається на увазі можливість використовувати той технологічний стек, який найкраще підходить до його задач. Наприклад, одні компоненти можуть бути написані на Python, інші – на Javascript, залежно від їхніх потреб.

Стійкість до збоїв передбачає, що помилки в одному мікросервісі не впливають на роботу інших компонентів. Це підвищує надійність системи в цілому, оскільки навіть при відмові одного сервісу решта програми продовжує функціонувати.

Остання перевага – повторне використання. Вона полягає у тому, що окремі мікросервіси можна інтегрувати в інші програми або повторно використовувати в нових проєктах. Це знижує витрати на розробку та сприяє створенню модульних систем.

Архітектура програмного забезпечення включає такі основні компоненти: клієнтська частина, API для визначення анатомічних точок постави, а також API для отримання результатів класифікації порушень

постави.

Клієнтська частина реалізована за допомогою SPA архітектури, яка забезпечує динамічне оновлення контенту в межах однієї HTML-сторінки. Це дозволяє зменшити обсяг даних, які передаються між клієнтом і сервером, та підвищує швидкість роботи додатка. Завдяки SPA користувач може взаємодіяти з інтерфейсом без повного перезавантаження сторінки, що робить перехід між розділами плавним і зручним. Такий підхід також сприяє оптимізації продуктивності та підвищенню загальної якості взаємодії з додатком.

API-компоненти реалізовані з використанням принципів REST-архітектури. REST (Representational State Transfer) не є формалізованим стандартом, але забезпечує високу гнучкість у розробці. Основні принципи, яких дотримуються під час створення RESTful API:

- ресурсно-орієнтована структура (всі дані й функції представлені у вигляді ресурсів, доступ до яких здійснюється за допомогою унікальних URL);
- уніфікований інтерфейс (операції над ресурсами виконуються за допомогою стандартних HTTP-методів);
- представлення ресурсів (дані, що повертаються сервером, формуються у стандартизованому форматі, наприклад, JSON, що забезпечує їхню сумісність із широким спектром клієнтів);
- автономність відповіді (відповідь сервера містить повну інформацію для обробки запиту, незалежно від інших операцій).

Завдяки обраному підходу вийшла масштабована, надійна система, яку легко можна адаптувати до додаткових бізнес-вимог і технологічних тенденцій. Діаграма компонентів даного програмного забезпечення зображена на рисунку 4.1.

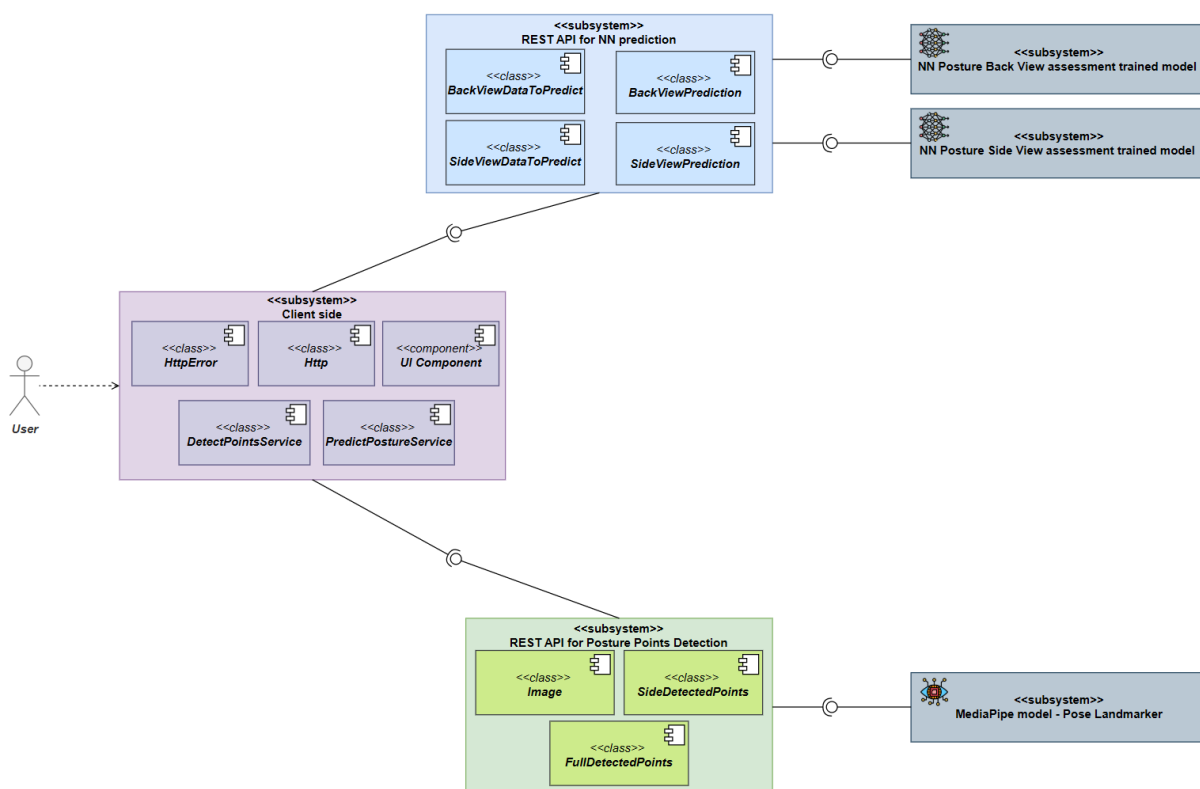


Рисунок 4.1 – Діаграма компонентів програмного забезпечення

Опис класів програмного забезпечення подано у таблицях 4.1–4.3. Вони відповідно описують структуру та функціонал API для визначення анатомічних точок постави, API для отримання результатів класифікації порушень постави за допомогою нейронних мереж, а також клієнтської частини.

Таблиця 4.1 – Опис класів API для визначення анатомічних точок постави

| Назва класу | Опис |
|---------------|--|
| Image | DTO (Data Transfer Object) клас, що є похідним від класу BaseModel з бібліотеки Pydantic, надає можливість створення моделі даних для валідації вхідних даних (представлення фото у форматі base64). |
| Point | DTO клас-модель даних для зберігання координат визначеної анатомічної точки. |
| SideLandmarks | DTO клас-модель даних для зберігання знайдених анатомічних точок (посилання на клас Point) з їх назвою з |

Продовження табл. 4.1

| Назва класу | Опис |
|--------------------|--|
| | ракурсу збоку. |
| SideLandmarks | DTO клас-модель даних для зберігання знайдених анатомічних точок (посилання на клас Point) з їх назвою з ракурсу збоку. |
| FullLandmarks | DTO клас-модель даних для зберігання знайдених анатомічних точок (посилання на клас Point) з їх назвою з ракурсу ззаду/спереду. |
| SideDetectedPoints | DTO клас-модель даних для валідації вихідних даних (знайдені анатомічні точки з їх назвою – SideLandmarks та визначена сторона з ракурсу збоку). |
| FullDetectedPoints | DTO клас-модель даних для валідації вихідних даних (знайдені анатомічні точки з їх назвою – FullLandmarks з ракурсу ззаду/спереду). |

Таблиця 4.2 – Опис класів API для отримання результатів класифікації порушень постави за допомогою нейронних мереж

| Назва класу | Опис |
|-----------------------|---|
| BackViewDataToPredict | DTO клас-модель даних для валідації вхідних даних (кути постави з ракурсу ззаду/спереду). |
| SideViewDataToPredict | DTO клас-модель даних для валідації вхідних даних (кути постави з ракурсу збоку). |
| BackViewPrediction | DTO клас-модель даних для валідації вихідних даних (ймовірності належності кутів з ракурсу ззаду/спереду до певних порушень постави). |
| SideViewPrediction | DTO клас-модель даних для валідації вихідних даних (ймовірності належності кутів з ракурсу збоку до певних порушень постави). |

Таблиця 4.3 – Опис класів клієнтської частини

| Назва класу | Опис |
|-----------------------|---|
| HttpError | Цей клас є похідним від класу Error і додає до нього властивість statusCode, що дозволяє генерувати помилки з конкретним кодом статусу. |
| Http | Цей клас використовується для взаємодії клієнтської частини з іншими сервісами(API), використовуючи HTTP-протокол. |
| DetectPointsService | Клас-сервіс, який відповідає за відправку запитів до Detect Points API для отримання анатомічних точок постави на певній фотографії. |
| PredictPostureService | Клас-сервіс, який відповідає за відправку запитів до Predict Posture API для отримання результатів класифікації постави на основі кутів основних анатомічних точок. |

Діаграма класів даного програмного забезпечення зображена на рисунку 4.2. Посилання на лістинг коду усіх компонент можна знайти у додатку А.

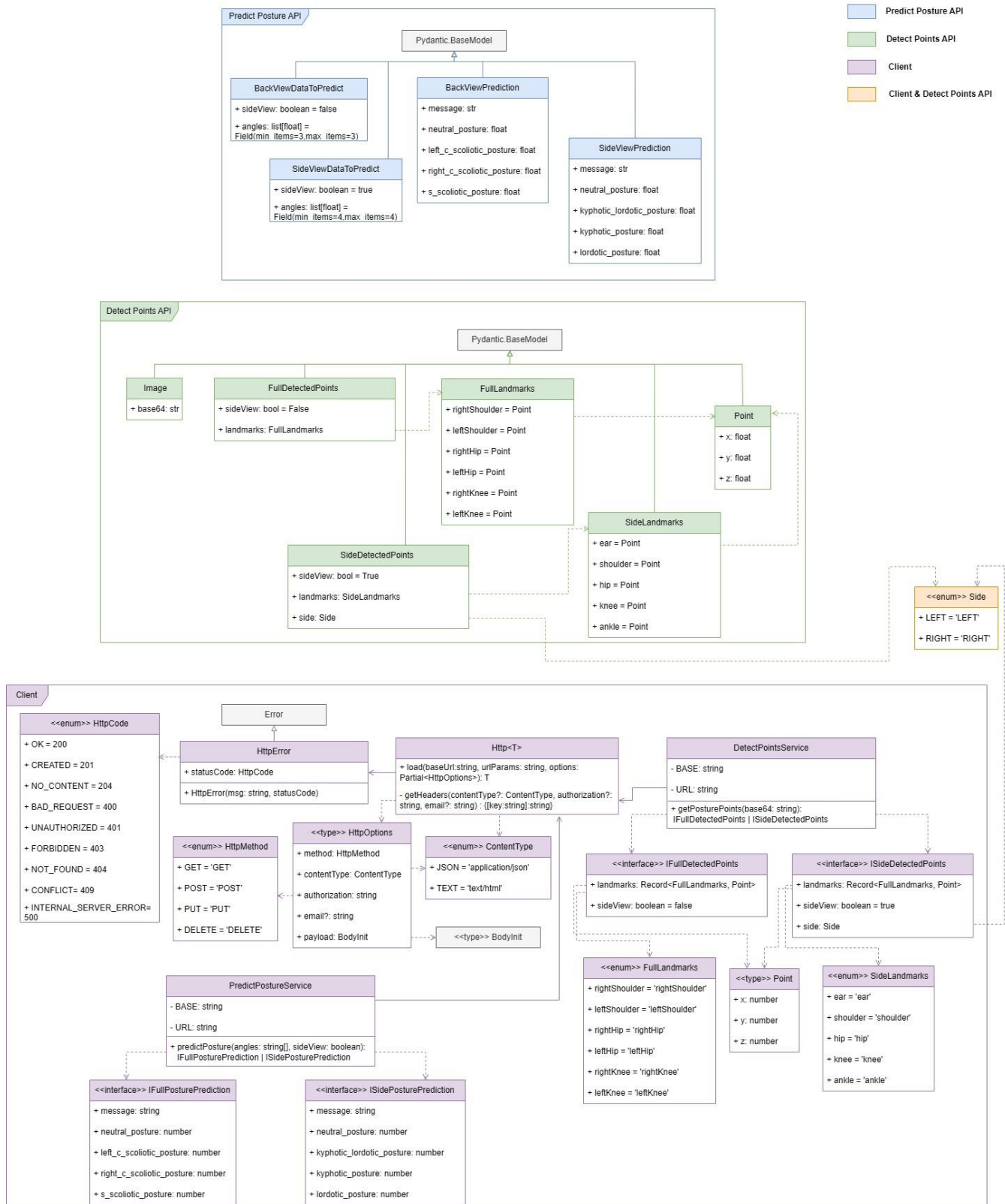


Рисунок 4.2 – Діаграма класів програмного забезпечення

4.3. Огляд програмного продукту

Для початку роботи з проектом користувач повинен зклонувати Git репозиторії з додатку А та запустити кожну підсистему.

Команди для запуску Predict Posture API:

- pip install -r requirements.txt;
- uvicorn src.main:app --port 8086.

Команди для запуску Detect Points API:

- pip install -r requirements.txt;
- uvicorn src.main:app.

Команди для запуску клієнтської частини:

- npm install;
- npm run start.

Після виконання цих кроків усі компоненти будуть запуснені на локальному сервері. Потім користувачеві необхідно перейти на адресу <http://localhost:3000/> в браузері, і він побачить головну сторінку додатку, як показано на рисунку 4.3.

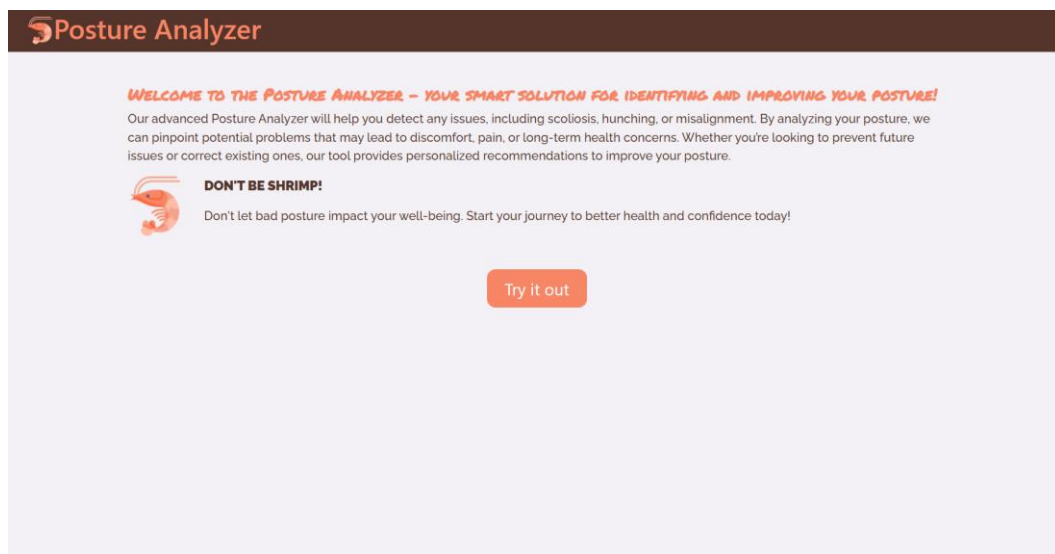


Рисунок 4.3 – Головна сторінка програмного продукту

Після натискання кнопки «Try it out» відкриється сторінка для завантаження фотографій із вказаними вимогами. Користувач може вибрати режим аналізу: Single або Statistical. Інструкції змінюються залежно від обраного режиму.

У режимі Single можна завантажити від 1 до 3 фотографій, які не мають бути пов'язані між собою (одна людина з різних ракурсів чи можуть бути різні люди). В такому випадку аналіз робиться окремо для кожної завантаженої

фотографії. Приклад сторінки можна побачити на рисунку 4.4.

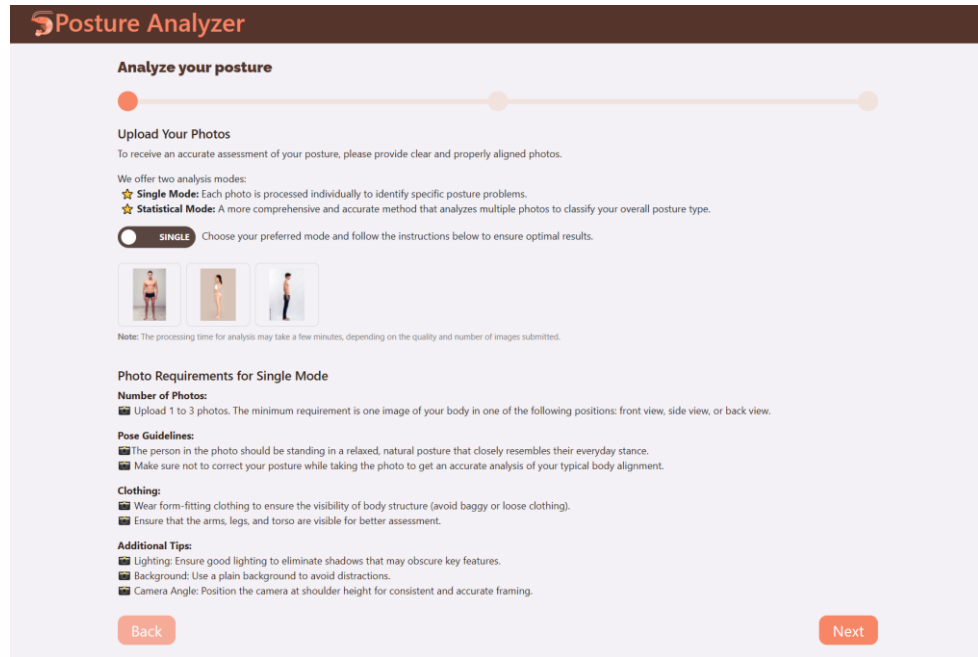


Рисунок 4.4 – Сторінка для завантаження фото в режимі «Single»

У режимі Statistical користувач має завантажити від 4 до 6 фотографій, зроблених з одного ракурсу в різний час протягом дня. У цьому випадку аналіз здійснюється на основі всіх завантажених зображень, і результат буде об'єднаним. Приклад сторінки для цього режиму наведено на рисунку 4.5.

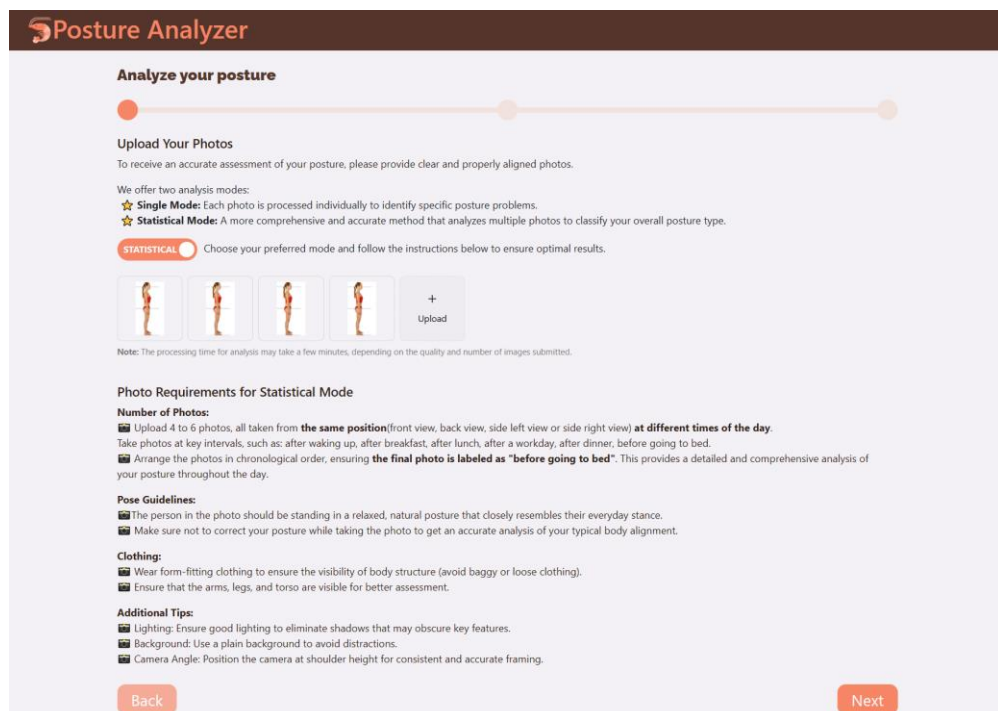


Рисунок 4.5 – Сторінка для завантаження фото в режимі «Statistical»

Програмний продукт підтримує два формати файлів: PNG та JPEG. Спроба завантажити файл іншого формату спричинить появу повідомлення про помилку, і файл не буде завантажено, як показано на рисунку 4.6.

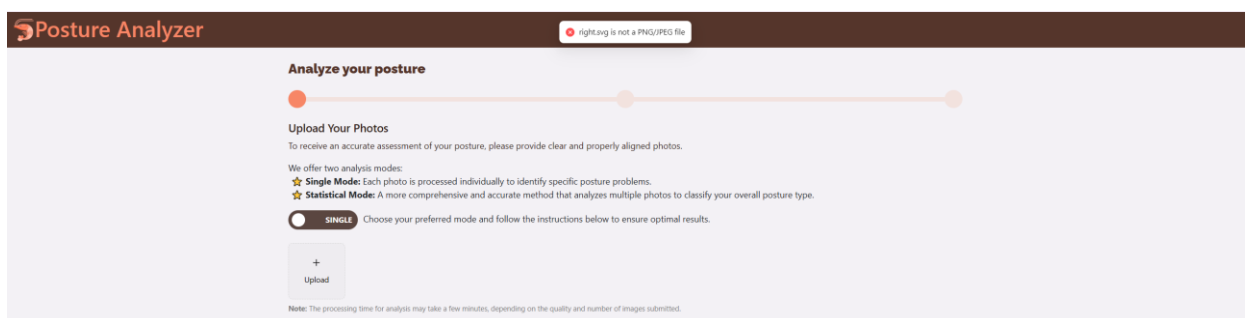


Рисунок 4.6 – Повідомлення про помилку через невалідний формат файлу

Після завантаження фото і натискання кнопки «Next» відкривається сторінка з відображенням анатомічних точок, виявлених на завантажених фотографіях. У разі помилок користувач має можливість вручну скоригувати розташування точок. Інструкція щодо правильного розташування точок оновлюється відповідно до ракурсу фотографії.

Для фотографій з ракурсу збоку необхідно перевірити 5 ключових точок: вухо, плече, стегно (великий вертлюг), коліно та щиколотку. Приклад цієї сторінки наведено на рисунку 4.7.

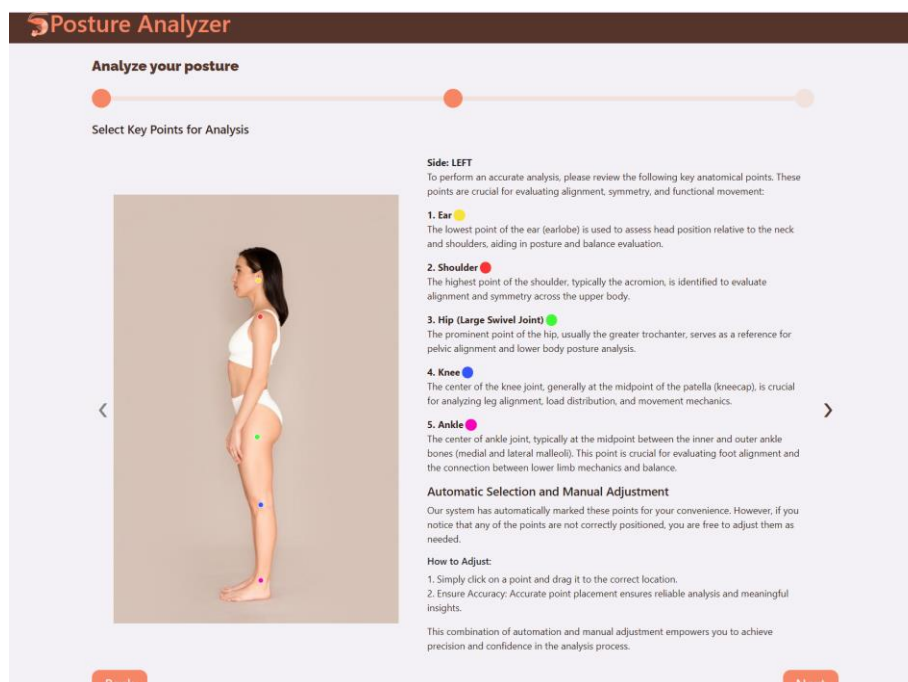


Рисунок 4.7 – Сторінка перегляду та редагування анатомічних точок для фотографії з ракурсу збоку

Для фотографій з ракурсу спереду або ззаду потрібно перевірити 6 точок: ліве та праве плече, ліве та праве стегно, ліве та праве коліно. Важливо визначати сторони щодо людини, зображеної на фотографії. Приклад сторінки для цього ракурсу наведено на рисунку 4.8.

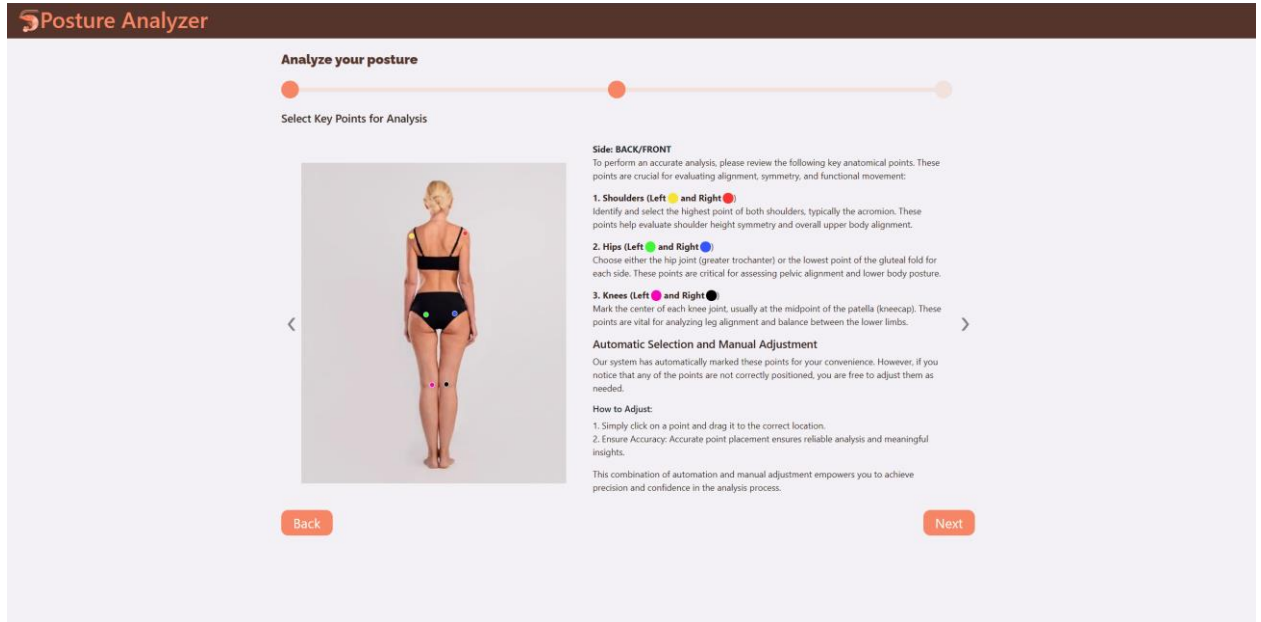


Рисунок 4.8 – Сторінка перегляду та редагування анатомічних точок для фотографії з ракурсу ззаду або спереду

Після перевірки та підтвердження розташування анатомічних точок кнопкою «Next» відкривається сторінка з результатами аналізу. На цій сторінці відображаються: знайдені кути між анатомічними точками, патерн постави із візуалізованими лініями та результат класифікації постави, включно з рекомендованими вправами та ймовірностями належності до типів, якщо ці ймовірності складають 25% і більше. Приклад сторінки з результатами зображено на рисунку 4.9.

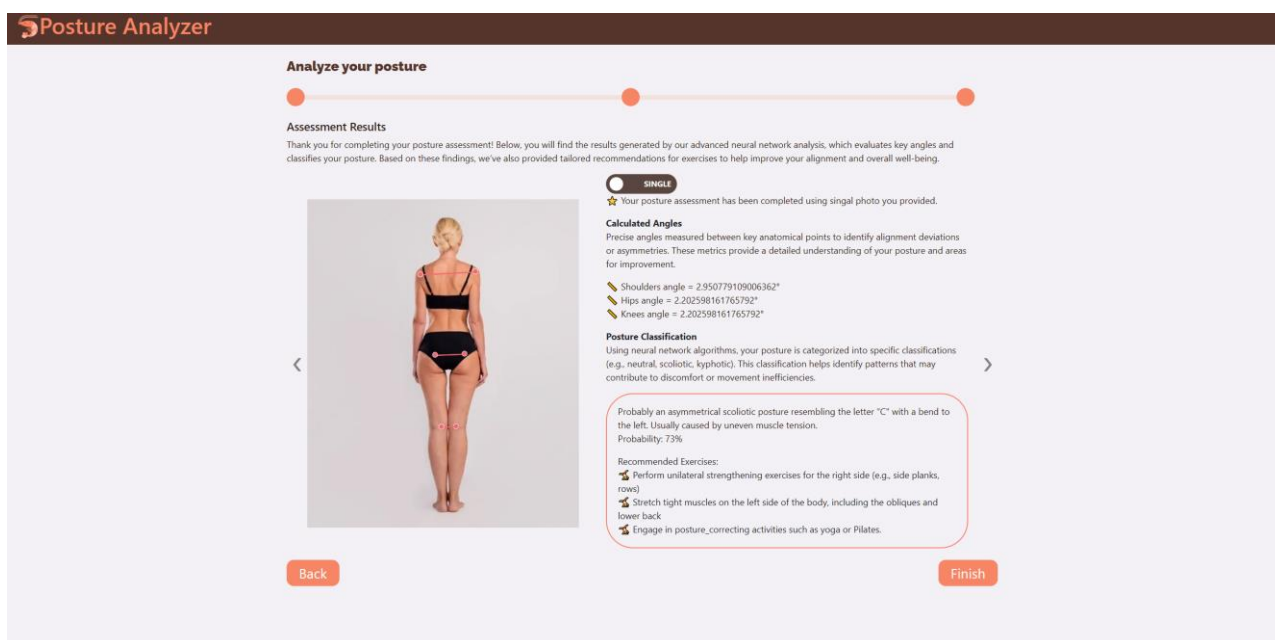


Рисунок 4.9 – Сторінка перегляду результатів аналізу у режимі «Single»

Для режиму «Statistical» сторінка з результатами має додаткову логіку. На зображенні будуть зображені дві лінії: рожева та фіолетова. Рожева лінія відповідає анатомічним точкам для останнього завантаженого фото, а синя – зважена лінія, яка була обчислена на основі усіх завантажених фотографій. Важливо зазначити, що всі кути та результати класифікації у цьому режимі базуються на зважених даних. Приклад сторінки з результатами для «Statistical» режиму наведено на рисунку 5.8.

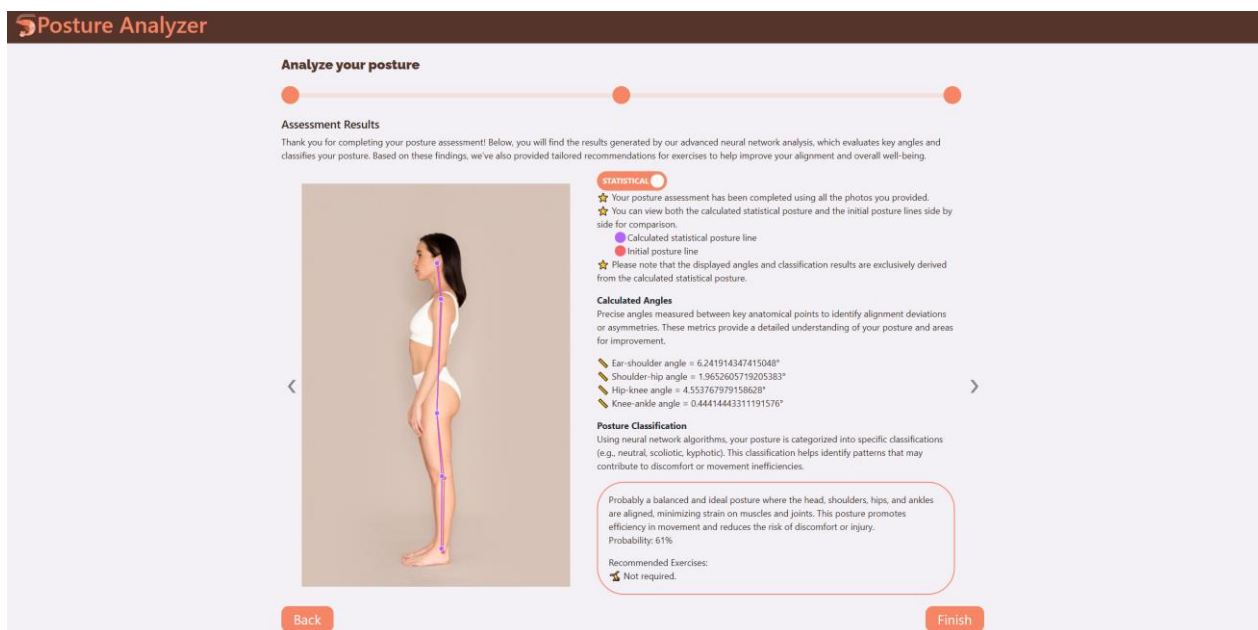


Рисунок 4.10 – Сторінка перегляду результатів аналізу у режимі «Statistical»

ВИСНОВКИ

Дана кваліфікаційна робота присвячена розробці інтелектуальної технології для класифікації порушень постави людини на основі аналізу візуальних даних. Усі аспекти реалізації цієї технології детально описані у пояснювальній записці.

Метою створення цього програмного забезпечення було спрощення процесу виявлення ключових анатомічних точок тіла та полегшення аналізу відхилень у поставі.

У розділі аналітичного огляду досліджено предметну область, визначено акторів системи та їхні можливості, а також сформульовано функціональні вимоги. Крім цього, проведено аналіз наявних аналогів із подальшим порівнянням їх з розробленим програмним продуктом. Окремо була сформована постановка задачі та визначено ключові цілі для її виконання. Для наочності також представлено діаграму варіантів використання.

Розділ, присвячений інформаційному забезпеченню, включає опис вхідних і вихідних даних системи. Для демонстрації передачі даних у системі створено діаграму потоку даних. Оскільки готового датасету для навчання нейронної мережі не існувало, було розроблено власний набір даних, розділений на дві частини залежно від ракурсів: збоку та спереду/ззаду. Два датасети, кожен з яких складається з 250 прикладів, є збалансованими і опубліковані на платформі Kaggle для подальшого використання. Хоча їх розмір поки обмежений, це забезпечує основу для подальшого розширення.

Математичне забезпечення описує, як координати анатомічних точок перетворюються у кути між ними, і розглядає модель для зменшення статистичної похибки при обробці множинних вибірок точок. Також представлено загальну математичну модель для задачі мультикласової класифікації та детально описано архітектуру двох нейронних мереж, їхні гіперпараметри та отримані результати. Точність моделей варіюється: для ракурсу спереду/ззаду від 49% до 76%, а для ракурсу збоку — від 39% до 95%. Найменшу точність показує нормальний клас через його часткову схожість із

іншими класами, що можна покращити шляхом збільшення розміру датасету. Для цього можна залучити групи людей з необхідними порушеннями постави та медичних фахівців, які правильно розставляють анатомічні точки на фото респондентів.

Розділ програмного забезпечення охоплює опис клієнтської частини, API для визначення анатомічних точок (Detect Points API) та API для класифікації (Predict Posture API). Розглянуто архітектурні рішення для кожного компоненту, представлено діаграми компонентів і класів. Окрім технічного опису, у цьому розділі наведено інструкції щодо запуску системи, детальний опис сторінок програми та їх функціональних можливостей.

Загальні висновки роботи підтверджують, що розроблена система відповідає поставленим функціональним вимогам. Тестування продемонструвало, що створені нейронні мережі показують задовільні результати, які можна суттєво покращити за умови розширення навчального датасету. Програмний продукт має перспективу для подальшого вдосконалення і може бути використаний у медичній практиці.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Guthold R. Global trends in insufficient physical activity among adolescents: a pooled analysis of 298 population-based surveys with 1.6 million participants / R. Guthold, G.A. Stevens, L.M. Riley, F.C. Bull // The Lancet Child & Adolescent Health. – 2020. – Vol. 4. – № 1. – P. 23–35.
2. Центр громадського здоров'я України. 90% дорослих хворих на остеохондроз і радикуліт – це люди, які перенесли в дитинстві сколіоз. Facebook.URL: <https://www.facebook.com/phc.org.ua/posts/2907217089403000> (дата звернення: 03.09.2024)
3. Williams R. S. The Role of Posture and Health: Good vs Bad. Coastal Orthopedics. URL: <https://www.coastalorthoteam.com/blog/the-role-of-posture-and-health-good-vs-bad> (дата звернення: 03.09.2024)
4. Jockey Club Children's Spine Care Community Project. Adolescent Idiopathic Scoliosis (AIS): Assessment. The Hong Kong Polytechnic University. URL: <https://www.polyu.edu.hk/bme/spinecare/en/adolescent-idiopathic-scoliosis-ais/assessment/> (дата звернення: 04.09.2024).
5. Posture: Is there an ideal posture?. ProActive Health. URL: <https://proactivehealth.ca/posture-is-there-an-ideal-posture/> (дата звернення: 04.09.2024).
6. Muscolino J. Scoliosis. Learn Muscles. URL: <https://learnmuscles.com/blog/2017/09/25/scoliosis/> (дата звернення: 10.09.2024).
7. Schell P. Posture Types: Painful Power Leaks You Should Know About. Activ8 Posture. URL: <https://activ8posture.com/posture/posture-types/> (дата звернення: 10.09.2024).
8. Schafer R. C. Clinical Biomechanics: Musculoskeletal Actions and Reactions // 2-ге вид. Williams & Wilkins. – 1987. – Т. 13 : Scoliosis. URL: <https://chiro.org/ACAPress/Scoliosis.html> (дата звернення: 30.09.2024).
9. Трифанова Т. Прикладна екологія людини. – 2018.
10. Morkvina V. Posture Types Image Dataset (front/back view). Kaggle.

URL: <https://www.kaggle.com/datasets/valerri/posture-types-image-dataset-frontback-view> (дата звернення: 04.12.2024).

11. Morkvina V. Posture Types Image Dataset (side view). Kaggle. URL: <https://www.kaggle.com/datasets/valerri/posture-types-image-dataset-side-view> (дата звернення: 04.12.2024).

12. Morkvina V. Posture Assessment (Front/Back view). Kaggle. URL: <https://www.kaggle.com/datasets/valerri/posture-assessment-frontback-view> (дата звернення: 04.12.2024).

13. Morkvina V. Posture Assessment (Side view). Kaggle. URL: <https://www.kaggle.com/datasets/valerri/posture-assessment-side-view> (дата звернення: 04.12.2024).

14. Noé F. Does the time of day differently impact the effects of an exercise program on postural control in older subjects? A pilot study / Korchi K., Bru N., Paillard T.// BMC Sports Science, Medicine and Rehabilitation. – 2022. URL: <https://bmcsportsscimedrehabil.biomedcentral.com/articles/10.1186/s13102-022-00467-5#citeas> (дата звернення: 01.10.2024).

15. Good Sleeping Posture Helps Your Back. University of Rochester Medical Center Rochester, NY URL: <https://www.urmc.rochester.edu/encyclopedia/content.aspx?ContentTypeID=1&ContentID=4460> (дата звернення: 01.10.2024)

16. Perrotta P. Killer Combo: Softmax and Cross Entropy. URL: <https://levelup.gitconnected.com/killer-combo-softmax-and-cross-entropy-5907442f60ba> (дата звернення: 10.10.2024)

17. Pramoditha R. The Concept of Artificial Neurons (Perceptrons) in Neural Networks. URL: <https://towardsdatascience.com/the-concept-of-artificial-neurons-perceptrons-in-neural-networks-fab22249cbfc> (дата звернення: 15.10.2024)

18. Klaas J. Machine Learning for Finance. Measuring model loss. URL: <https://subscription.packtpub.com/book/data/9781789136364/1/ch011v11sec21/measuring-model-loss> (дата звернення: 20.10.2024)

19. Kumar D.K. Getting to Know Adam Optimization: A Comprehensive Guide. URL: <https://www.linkedin.com/pulse/getting-know-adam-optimization-comprehensive-guide-kiran-kumar> (дата звернення: 20.10.2024)
20. Dasaradh S. K.. A Gentle Introduction To Math Behind Neural Networks. URL: <https://towardsdatascience.com/introduction-to-math-behind-neural-networks-e8b60dbbdeba> (дата звернення: 22.10.2024)
21. Patel M. Understanding the Rectified Linear Unit (ReLU): A Key Activation Function in Neural Networks. URL: <https://medium.com/@meetkp/understanding-the-rectified-linear-unit-relu-a-key-activation-function-in-neural-networks-28108fba8f07> (дата звернення: 22.10.2024)

ДОДАТОК А

Посилання на програмний код трьох компонент:

1. Клієнтська частина: <https://github.com/morkvinaValeria/analyze-posture-frontend>
2. API для отримання анатомічних точок постави (Detect Points API):
<https://github.com/morkvinaValeria/detect-posture-points>
3. API для класифікації порушень постави (Predict Posture API):
<https://github.com/morkvinaValeria/predict-posture>