

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____
(повна назва)
Кафедра _____ Програмної інженерії _____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка
рівень вищої освіти - другий (магістерський)

Дослідження методів розпізнавання образів для вирішення задач
комп'ютерної діагностики захворювань шкіри
(тема)

Виконав: студент 2 курсу, групи ІПЗм-21-4
Пашаєва С.М.
(прізвище, ініціали)

спеціальності 121- Інженерія програмного забезпечення
(код і повна назва спеціальності)

Освітньо-наукової програми
(тип програми)

Інженерія програмного забезпечення
(повна назва освітньої програми)

Керівник _____ доцент, к.т.н. Мазурова О.О. _____
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри, проф. _____

З.В.Дудар

2023р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Програмної інженерії

Рівень вищої освіти - другий (магістерський)

Спеціальність 121-Інженерія програмного забезпечення

(код і повна назва)

Тип програми освітньо-наукова програма

Освітня програма Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«_____» _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Пашаєвій Сабіні Мамед кизи

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів розпізнавання образів для вирішення задач комп'ютерної діагностики захворювань шкіри

затверджена наказом університету від “10” 04 2023 р. № 340Ст

заповнюється вручну після отримання наказу

2. Термін подання студентом роботи до екзаменаційної комісії

21 травня 2023 р.

3. Вихідні дані до роботи електронні ресурси за обраною тематикою, мінімальні вимоги до функціональності програми, загальні вимоги до архітектури системи

4. Перелік питань, що потрібно опрацювати в роботі аналіз проблемної області і постановка задачі, аналіз існуючих аналогів, UML-моделювання предметної області, дослідження методів розпізнавання об'єктів на зображенні, розробка схеми бази даних, проектування архітектури програмної системи, розробка алгоритму для розпізнавання образів, програмна реалізація системи, тестування.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, слайдів, ілюстрацій 20 слайдів презентації, схема бази даних, діаграми прецедентів, розгортання, загальна діаграма класів.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз проблемної області дослідження	1.10.21 – 1.11.21	Виконано
2	Аналіз аналогів	15.11.21 – 01.12.21	Виконано
3	Розробка постановки задачі	01.12.21-15.12.21	Виконано
4	Дослідження методів розпізнавання образів	15.12.21- 10.01.22	Виконано
5	Планування експериментальної частини дослідження	10.01.22-20.01.22	Виконано
6	Моделювання предметної області та розробка схеми БД	20.01.22-26.01.22	Виконано
7	Розробка алгоритмів розпізнавання образів	26.01.22 - 31.01.22	Виконано
9	Створення коду	7.01.22 – 21.02.22	Виконано
10	Тестування програми	21.02.22 – 28.02.22	Виконано
11	Проведення експериментів	28.02.22 - 7.03.22	Виконано
14	Допрацювання функціоналу	15.03.23 - 1.04.23	Виконано
15	Підготовка пояснювальної записки	1.04.23– 1.05.22	Виконано
16	Підготовка презентації та доповіді	1.05.23 – 15.05.23	Виконано
17	Нормоконтроль	12.05.23	Виконано
18	Рецензування	12.05.23	Виконано
19	Занесення диплома в електронний архів	15.05.23	Виконано
20	Попередній захист	18.05.23	Виконано
21	Допуск до захисту у зав. кафедри	21.05.23	Виконано

Дата видачі завдання _1_ __10__2021 р.

Студент _____

(підпис)

Керівник роботи _____

(підпис)

доц. Мазурова О.О. _____

(посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Пояснювальна записка до кваліфікаційної роботи магістра, 80 стор., 22 рис., 6 табл., 22 джерел.

ДОГЛЯД ЗА ШКІРОЮ, ДОСЛІДЖЕННЯ, КОСМЕТИЧНИЙ ЗАСІБ, МЕТОД РОЗПІЗНАВАННЯ ОБРАЗІВ, ANDROID, ASP.NET CORE WEB API, PYTHON, REACT.

Метою роботи є дослідження методів розпізнавання образів для вирішення задач комп'ютерної діагностики захворювань шкіри.

Метод рішення – порівняння методів розпізнавання образів, проведення експериментів, концептуальне моделювання предметної області, UML-моделювання, розробки програмної системи за допомогою технологій ASP.NET Core WEB API, PostgreSQL, React, Android тощо.

У результаті роботи було проведено експериментальне дослідження та сформульовано рекомендації щодо розробки методу розпізнавання шкірних захворювань на зображенні.

ANDROID, ASP.NET CORE WEB API, COSMETIC, OBJECT RECOGNITION METHOD, PYTHON, REACT, RESEARCH, SKIN CARE.

The aim of the work is the research of object recognition methods for solving the problems of computer diagnosis of skin diseases.

The solution method is a comparison of object recognition methods, conducting experiments, conceptual modeling of the subject area, UML modeling, software system development using ASP.NET Core WEB API, PostgreSQL, React, Android, etc. technologies.

As a result of the work, an experimental study was conducted and recommendations were formulated for the development of a method for recognizing skin diseases on the image.

Умови публікації пояснювальної записки

Я, Пашаєва Сабіна Мамед кизи, студент гр. ПЗМ-21-4, здобувач вищої освіти на другому (магістерському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Дослідження методів розпізнавання образів для вирішення задач комп'ютерної діагностики захворювань шкіри», що буде представлена в екзаменаційну комісію для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату, і вона може бути опублікована в електронному архіві відкритого доступу EIAr KhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомена з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

Вступ.....	8
1 Аналіз проблемної області та постановка задачі	10
1.1 Аналіз проблемної області дослідження	10
1.2 Аналіз існуючих аналогів.....	13
1.3 Постановка задачі.....	15
2 Опис прийнятих проектних рішень.....	17
2.1 Аналіз методів розпізнавання образів на зображенні	17
2.2 Вибір методів розпізнавання образів для подальшого дослідження.....	22
2.3 Аналіз та UML-моделювання предметної області.....	26
2.4 Розробка алгоритму для розпізнавання захворювань шкіри	28
2.5 Вибір метрик для експериментального дослідження.....	33
2.6 Проектування бази даних	34
2.7 Розробка архітектури системи	38
3 Опис програмної реалізації	41
3.1 Вибір засобів програмної реалізації.....	41
3.2 Реалізація серверної частини та бази даних	42
3.3 Опис реалізації розпізнавання дефектів шкіри на зображенні.....	44
3.4 Опис інтерфейсу користувача.....	47
4 Опис експериментальних досліджень.....	54
4.1 Планування експерименту	54
4.2 Експериментальне дослідження	56
5 Аналіз результатів експериментального дослідження та розробка рекомендацій	60
Висновки	62
Перелік джерел посилання	63
Додаток А Перелік джерел посилання за науковими напрямками керівника та науковців кафедри програмної інженерії	66
Додаток Б Результати перевірки роботи на академічну доброчесність	67
Додаток В Слайди презентації.....	68
Додаток Г Апробація результатів.....	78

Додаток Д Результати перевірки роботи на відповідність вимогам оформлення

ВСТУП

Шкіра є важливим показником стану здоров'я людини. Незважаючи на велику кількість косметичних засобів та процедур для догляду за нею, що є доступними на сучасному ринку, люди досі досить часто стикаються з проблемами шкіри через нездоровий спосіб життя, велику кількість стресу у повсякденному житті тощо, та шукають ефективні та доступні методи догляду за шкірою. З появою сучасних комп'ютерних технологій був зроблений значний прогрес у догляді за шкірою. Від діагностики та лікування до індивідуальних схем догляду за шкірою, ці технології змінили підхід до догляду за шкірою та зробили його ефективнішим, ніж будь-коли раніше.

Для вирішення задач діагностування шкірних захворювань на сьогоднішній день можливе використання таких технологій, як комп'ютерний зір та розпізнавання об'єктів на зображеннях. Це дозволить людям, що потребують допомоги у правильному догляді за шкірою, швидко отримати результати діагностики поточного стану шкіри та рекомендацій щодо його покращення.

Метою даної роботи є дослідження існуючих методів розпізнавання образів та порівняння їх ефективності та точності для вирішення задач комп'ютерної діагностики захворювань та дефектів шкіри. За допомогою виявлення проблем шкіри можна буде надавати рекомендації щодо використання певної комбінації косметичних засобів, що можуть допомогти усунути ці недоліки.

Під час виконання кваліфікаційної роботи був проведений аналіз проблемної області використання методів розпізнавання образів на зображенні для комп'ютерної діагностики захворювань шкіри. Були досліджені сучасні та поширені алгоритми розпізнавання на основі публікацій світових та вітчизняних фахівців в проблемній області (див. додаток А).

Також, було спроектовано процес проведення експерименту дослідження, що в свою чергу включає визначення найбільш відповідних методів розпізнавання для діагностики проблем шкіри, метрик для порівняння тощо.

Практичним результатом роботи є спроектована та реалізована програмна система, яка дозволяє людям отримати допомогу у щоденному догляді за шкірою

обличчя. Система складається з наступних частин: серверу, бази даних, сервісу для обробки зображень та розпізнавання об'єктів, веб-сайту та мобільного застосунку.

Робота пройшла успішну перевірку на академічну доброчесність (див. додаток Б).

На основі плану проведення дослідження та його результатів було розроблено презентацію (див. додаток В). За результатами роботи були створені тези доповіді «Використання технології комп'ютерного зору для системи розумного догляду за шкірою» на «Міжнародний молодіжний форум» (див. додаток Г).

Також, робота перевірена на відповідність вимогам оформлення (див. додаток Д).

1 АНАЛІЗ ПРОБЛЕМНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз проблемної області дослідження

На сьогоднішній день комп'ютерні технології все більше й більше набувають популярність у медичній області. Особливо стає поширеним таке поняття, як комп'ютерна діагностика (computer-aided diagnosis, CAD).

Комп'ютерна діагностика – це технологія, яка включає в себе кілька елементів, таких як концепції штучного інтелекту, комп'ютерного зору та обробки медичних зображень. Системи CAD широко застосовуються для виявлення та диференціальної діагностики багатьох різних типів аномалій у медичних зображеннях, отриманих під час різних досліджень із застосуванням різних методів візуалізації [1]. Системи комп'ютерної діагностики обробляють цифрові зображення для типового вигляду та виділяють помітні ділянки, наприклад можливі захворювання, щоб запропонувати вхідні дані для підтримки рішення, прийнятого професіоналом.

Хоча комп'ютерна діагностика використовується в клінічних умовах більше сорока років, вона зазвичай не замінює лікаря чи іншого фахівця, а відіграє допоміжну роль. Професіонал зазвичай відповідає за остаточну інтерпретацію медичного зображення. Однак мета деяких систем CAD полягає в тому, щоб виявити найраніше ознаки відхилень у пацієнтів, яких не можуть спеціалісти.

Через велику кількість доступних даних і необхідності аналізу, «великі дані» (“big data”) є однією з найбільших проблем, з якими стикаються системи комп'ютерної діагностики сьогодні. Кількість даних, що постійно збільшується, є серйозною проблемою. «Великі дані», які надходять сьогодні з безлічі абсолютно нових джерел, відрізняються від традиційних даних складністю, різноманітністю та швидкістю появи. Для аналізу «великих даних» необхідні найдосконаліші ІТ-методи і потужні обчислювальні системи [2]. На сьогоднішній день вже існують технології, що можуть вирішувати проблему аналізу великої кількості даних та зображень. Найяскравішим прикладом є комп'ютерний зір.

Комп'ютерний зір – це область інформатики, яка зосереджується на відтворенні складних частин системи людського зору та наданні комп'ютерам

можливості ідентифікувати й обробляти об'єкти на зображеннях та відео так само, як це роблять люди [3]. Донедавна комп'ютерний зір працював лише в обмежених можливостях.

Завдяки досягненням у сфері штучного інтелекту та інноваціям у глибокому навчанні та нейронних мережах, ця галузь змогла зробити великі стрибки за останні роки та стала здатна перевершити людей у деяких завданнях, пов'язаних із виявленням та маркуванням об'єктів.

Комп'ютерний зір також є важливою частиною досягнень у галузі догляду за здоров'ям. Алгоритми комп'ютерного зору можуть допомогти автоматизувати такі завдання, як виявлення ракових родимок на зображеннях шкіри або виявлення симптомів на рентгенівському та МРТ-скануванні [4]. У дерматології також були досягнуті чудові результати, які можна порівняти з діагностичною ефективністю дерматологів за допомогою клінічної фотографії та дерматологічних зображень.

Системи комп'ютерного зору можуть використовуватися для наступних поширених задач:

- класифікації об'єктів (система аналізує візуальний вміст і класифікує об'єкт на фото/відео до визначеної категорії);
- розпізнавання об'єктів (система аналізує візуальний вміст і визначає певний об'єкт на фото/відео);
- відстеження об'єктів (система обробляє відео, знаходить об'єкт (або об'єкти), які відповідають критеріям пошуку, і відстежує його рух) [5].

Розпізнавання об'єктів – технологія в області комп'ютерного зору для пошуку та ідентифікації об'єктів у зображенні або відеоряді. Люди розпізнають безліч об'єктів на зображеннях з невеликими зусиллями, незважаючи на те, що зображення об'єктів може дещо відрізнятися в різних точках зору, у багатьох різних розмірах і масштабах або навіть коли їх перекладають або повертають. Об'єкти можна розпізнати навіть, якщо вони частково закриті від огляду. Це завдання все ще залишається проблемою для систем комп'ютерного зору. Багато підходів до цієї задачі реалізовувалися протягом кількох десятиліть.

Процес розпізнавання образів можна розділити на наступні кроки:

- подання тестових зображень;
- обробка зображення;
- розуміння образу.

Для першого кроку алгоритмам комп'ютерного зору потрібно подавати великі набори зображень, які можна зібрати з фотографій, відео, 3D-візуалізацій тощо. Хороший набір даних потрібен мати такі характеристики:

- велику кількість, бо це важливо для навчання моделей машинного навчання;
- якість – зображення мають бути у високій роздільній здатності. Крім того, вони не повинні мати неясностей, створених людиною;
- різноманітність зображень, тобто має бути багато варіацій об'єкта інтересу;
- щільність – це означає, що деякі зображення повинні відображати реальні умови – наприклад, інші об'єкти на дорозі.

На другому етапі алгоритми «дивляться» на вхідні дані. Такі алгоритми дозволяють комп'ютеру навчитися відрізнити одне зображення від іншого. Отже, припускаючи, що машина вже навчена, алгоритми розбивають зображення на пікселі, розрізняють жорсткі краї та прості форми та запускають ітерації своїх передбачень. Те ж саме стосується відео – машина розуміє його як серію кадрів.

Нарешті, алгоритми інтерпретують дані та виконують певну задачу комп'ютерного зору, наприклад, розпізнавання зображень. На виході може бути мітка, яка описує те, що на вхідному зображенні, але параметри можна налаштувати.

Для реалізації комп'ютерного зору існують декілька бібліотек та фреймворків для розробників, які допомагають автоматизувати завдання, що включає виявлення та візуалізацію. Мова програмування Python добре відома своєю великою кількістю інструментів і бібліотек, доступних для спільноти, тому є однією з найулюбленіших мов серед розробників для комп'ютерного зору [6]. Найпоширенішими бібліотеками є Keras, OpenCV, TensorFlow, fastai тощо.

Keras – це бібліотека нейронних мереж високого рівня, яка може працювати поверх TensorFlow або Theano. Бібліотека була розроблена з упором на можливість швидкого експериментування. Ця бібліотека глибокого навчання надає кілька функцій, включаючи підтримку як згорткових мереж, так і рекурентних мереж, що, серед іншого, дозволяє легко і швидко створювати прототип.

OpenCV – це популярна бібліотека комп'ютерного зору з відкритим вихідним кодом, яка орієнтована на програми реального часу. Бібліотека має модульну структуру і включає кілька сотень алгоритмів комп'ютерного зору. OpenCV включає в себе ряд модулів, включаючи обробку зображень, аналіз відео, структуру 2D-функцій, виявлення об'єктів, калібрування камери, 3D-реконструкцію тощо.

З використанням таких бібліотек можлива розробка сервісу для розпізнавання образів на зображенні, яка може бути доступною для зовнішніх клієнтів, як звичайне Web API, тобто через програмний інтерфейс, що складається з однієї або кількох загально доступних кінцевих точок, яка відкривається через Інтернет – найчастіше за допомогою протоколу HTTP.

1.2 Аналіз існуючих аналогів

Через стрімкий розвиток можливостей комп'ютерних технологій у області розпізнавання образів на цифрових зображеннях збільшується кількість наявних на ринку програмних систем, що використовують нейронні мережі у різних сферах життя.

На сьогоднішній день розробляються системи для діагностики проблем шкіри. Однією з таких систем є інструмент дерматологічної допомоги на основі штучного інтелекту від компанії Google [7] (див. рис. 1.1).

За допомогою цього інструменту люди можуть використовувати камеру свого телефону, щоб зробити три знімки проблемної зони – наприклад, висипання на руці. Потім вони можуть відповісти на низку запитань про свій тип шкіри та інші симптоми для покращення результатів роботи системи. Потім інструмент

надає список можливих захворювань із набору, які він навчився розпізнавати, та детальну інформації. Інструмент ще не є доступним для звичайних користувачів, тільки демо-версія була презентована компанією Google.

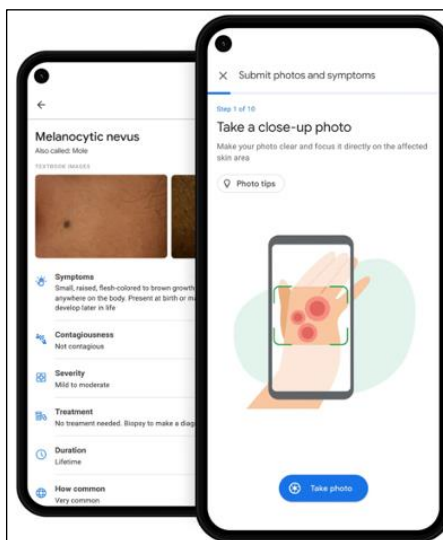


Рисунок 1.1 – Приклад роботи інструменту від Google

Ще одним аналогом є мобільний застосунок «Neutrogena Skin360» (див. рис. 1.2) [8]. Перед початком роботи користувачу потрібно зробити фотографію обличчя за допомогою смартфона. Після цього програма оцінює стан шкіри користувача, аналізуючи зроблене сканування, та ставить певну оцінку, яка використовується для визначення стану здоров'я шкіри та відстеження прогресу з часом. Програмний продукт не є доступним в Україні.



Рисунок 1.2 – Приклад роботи застосунку «Neutrogena Skin360»

Мобільний застосунок “Aysa” (див. рис. 1.3) також був розроблений для діагностики проблем шкіри, а саме, шкірних висипань.

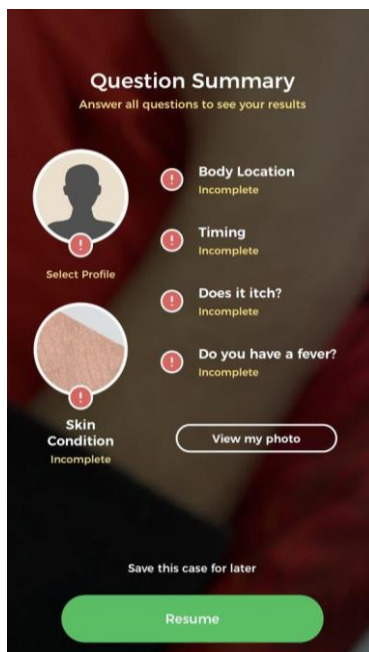


Рисунок 1.3 – Приклад роботи застосунку “Aysa”

Використовуючи знімок, який користувач може зробити за допомогою мобільного пристрою, на якому є та область шкіри, що викликає занепокоєння, “Aysa” надає вказівки щодо того, що робити далі. Однак для отримання рекомендацій користувачу потрібно відповісти на багато питань про висипання, порівняти з типовими зразками. Програмний продукт може рекомендувати косметичні та лікарські засоби для самопомоги, запропонувати почати сеанс зі спеціалістом та допомогти з пошуком найближчого лікаря.

1.3 Постановка задачі

Метою роботи є дослідження методів розпізнавання образів для вирішення задач комп’ютерної діагностики таких проблем шкіри, як акне, лущення, розширені пори тощо.

Практичною задачею роботи є розробка програмної системи, що пропонуватиме користувачам систему для діагностики проблем шкіри за допомогою технології комп’ютерного зору та генерації рекомендацій щодо

правильного догляду за неї, завдяки знаходженню оптимальної комбінації косметичних засобів.

Розроблена програмна система повинна підтримувати наступний функціонал:

- швидке діагностування проблем шкіри за допомогою зображень, зроблених на камеру смартфона або обраних з галереї;
- генерування рекомендацій з оптимальним комплексом засобів, враховуючи результати діагностики та даних від користувача (тип та особливості шкіри, вік тощо), якщо проблема шкіри є косметичною;
- можливість розробки плану використання косметичних засобів та отримання сповіщень на мобільному пристрої;
- можливість занесення даних про засоби догляду за шкірою до системи виробником косметичних продуктів;
- додавання та редагування даних про косметичні засоби користувача, створення коментарів тощо;
- безпечний процес аутентифікації та авторизації користувача в системі.

При розробці програмного продукту потрібно використати можливості хмарних технологій, комп'ютерного зору та глибокого навчання, Big Data технологій.

Таким чином, в ході виконання роботи потрібно виконати такі задачі:

- проаналізувати існуючі методи розпізнавання образів, які використовуються у комп'ютерній діагностиці;
- спланувати та провести експериментальне дослідження обраних методів розпізнавання;
- спроектувати та реалізувати базу даних для збереження даних, програмну систему;
- розробити рекомендації стосовно методів розпізнавання образів для області комп'ютерної діагностики.

2 ОПИС ПРИЙНЯТИХ ПРОЕКТНИХ РІШЕНЬ

2.1 Аналіз методів розпізнавання образів на зображенні

На сьогоднішній день відокремлюють декілька груп методів розпізнавання образів на зображенні, а саме:

- методи на основі зовнішнього вигляду;
- підходи, засновані на об'єктних моделях;
- методи на основі ознак;
- штучні нейронні мережі та глибоке навчання тощо [9].

Методи на основі зовнішнього вигляду використовують зразки зображень (так звані шаблони або зразки) об'єктів для розпізнавання. Основна складність може полягати в тому, що об'єкти можуть виглядати по-різному за різних умов: зміни освітлення або кольору, зміни напрямку перегляду, зміни розміру або форми тощо. Тому єдиний шаблон або зразок не буває надійним.

Методи на основі зовнішнього вигляду зазвичай включають дві фази. На першому етапі з набору опорних зображень будується модель. Набір включає зовнішній вигляд об'єкта з різними орієнтаціями, різними джерелами освітлення та потенційно множинними екземплярами класу об'єктів, наприклад, обличчя. Зображення мають високу кореляцію і їх можна ефективно стискати.

На другому етапі витягуються частини вхідного зображення (підзображення того ж розміру, що й навчальні зображення), можливо, шляхом сегментації (за текстурою, кольором, рухом) або шляхом вичерпного перерахування вікон зображення на всьому зображенні. Потім система розпізнавання порівнює витягнуті частини вхідного зображення з еталонними зображеннями.

Основним обмеженням підходів, заснованих на зовнішньому вигляді, є те, що вони вимагають ізоляції повного об'єкта, що цікавить, від фону.

Приклади методів на основі зовнішнього вигляду:

- метод головних компонентів (PCA);
- пошук «Поділяй і володарюй» («Divide-and-Conquer»);
- гістограми відповідей рецептивного поля;

- узгодження відтінків сірого;
- збіг країв;
- тощо.

У методах, заснованих на моделі, інформація про об'єкти представляється явно. Розпізнавання можна інтерпретувати як рішення про те, чи може дане зображення або його частина бути проекцією відомої (зазвичай 3D) моделі об'єкта.

Основними недоліками методів на основі моделі є: залежність від надійного вилучення геометричних примітивів (лінії, кола тощо), неоднозначність інтерпретації виявлених примітивів (наявність примітивів, які не моделюються), обмежені можливості моделювання лише до класу об'єктів, які складаються з кількох легко виявлених елементів, і потреби створювати моделі вручну.

Ні методи, що базуються на моделях, ні методи на основі зовнішнього вигляду, не відповідають таким вимогам, як загальність, надійність і легкість навчання. Підходи, засновані на моделях, вимагають від користувача вказувати об'єктні моделі і зазвичай можуть обробляти лише об'єкти, що складаються з простих геометричних примітивів. Вони не є загальними і не підтримують легкого навчання. Методи, засновані на зовнішньому вигляді, вимагають вичерпного набору навчальних зображень та такий набір доступний лише тоді, коли об'єкт можна спостерігати в контрольованому середовищі. Ці методи також чутливі до оклюзії об'єктів та до невідомого фону, тому вони не є надійними.

Як спробу вирішення вищезазначених проблем були запропоновані методи, засновані на узгодженні місцевих ознак. Об'єкти представляються набором локальних ознак, які автоматично обчислюються з навчальних зображень та зберігаються в базах даних. Отже, методи на основі ознак використовуються для пошуку можливих збігів між ознаками об'єкта та ознаками зображення. Основне обмеження полягає в тому, що одна позиція об'єкта повинна враховувати всі можливі збіги.

Ця група методів виділяє такі ознаки з об'єктів, як поверхневі плями, кути, лінійні ребра, які потрібно розпізнати, і зображень, які потрібно шукати.

Приклади методів, що базуються на ознаках:

- гістограма напрямлених градієнтів (HOG);
- інтерпретаційні дерева;
- геометричне хешування;
- масштабно-інваріантна трансформація ознак (SIFT);
- прискорені надійні функції;
- узагальнене перетворення Хафа тощо.

Складність цього підходу полягає в тому, що необхідно мануально обрати, які ознаки важливі в кожному даному зображенні. Оскільки кількість класів для класифікації може збільшуватися, вилучення ознак стає все більш громіздким. Вирішення того, які ознаки найкраще описують різні класи об'єктів, залежить від інженера з комп'ютерного зору та тривалого процесу проб і помилок. Більш того, кожне визначення ознак вимагає роботи з безліччю параметрів, усі з яких повинні бути точно налаштовані інженером.

Для вирішення проблеми виявлення ознак для розпізнавання об'єктів експертами-людьми був запропонований метод під назвою еволюційно сконструйовані (Evolution-Constructed, ECO) ознаки на основі генетичних алгоритмів [10].

Генетичний алгоритм – це методика для задач оптимізації на основі природного відбору. Генетичні алгоритми використовують підхід для визначення оптимального набору на основі еволюції. Для вибору ознак першим кроком є створення сукупності на основі підмножин можливих ознак. З цієї сукупності підмножини оцінюються за допомогою прогнозу моделі для цільового завдання. Після розгляду кожного члена популяції проводиться турнір, щоб визначити, які підмножини залишаться в наступному поколінні. Наступне покоління складається з переможців турніру з деякими перехресними (оновлення переможних наборів функцій функціями інших переможців) і мутацією (введення або видалення деяких функцій випадковим чином).

Еволюційні сконструйовані ознаки автоматично створюються шляхом використання стандартного генетичного алгоритму для виявлення серії

трансформацій, які є дуже дискримінаційними. Використання таких ознак дає ряд переваг серед інших алгоритмів виявлення об'єктів, включаючи: відсутність необхідності в експерті-людині для створення наборів функцій або налаштування їх параметрів, можливість створення спеціалізованих наборів функцій для різних об'єктів і відсутність обмежень для певних типів джерел зображення.

Певні прикладні сценарії розпізнавання об'єктів є занадто важкими або виходять за межі можливостей для традиційних алгоритмів машинного навчання. Глибоке навчання (Deep Learning) використовується в області цифрової обробки зображень для вирішення саме таких складних проблем (наприклад, розфарбовування зображень, класифікація, сегментація та виявлення).

Глибоке навчання – це підмножина машинного навчання, який в основному базується на штучних нейронних мережах (ANN), обчислювальній парадигмі, натхненній функціонуванням людського мозку [11]. Як і людський мозок, вона складається з багатьох обчислювальних клітин або «нейронів», кожен з яких виконує просту операцію та взаємодіє один з одним, щоб прийняти рішення.

Такі методи глибокого навчання, як згорткові нейронні мережі (CNN), здебільшого покращили продуктивність прогнозування, використовуючи великі дані та численні обчислювальні ресурси, і розсунули межі можливого.

Згорткова нейронна мережа – це алгоритм глибокого навчання, який може приймати вхідне зображення, призначати важливість (засвоювані ваги та упередження) різним аспектам або об'єктам зображення та мати можливість відрізнити один від іншого [12]. В порівнянні з іншими алгоритмами класифікації, попередня обробка зображення для згорткових нейронних мереж є менш необхідна, бо вони мають можливість вивчати ці фільтри або характеристики. Виявлення об'єктів, розпізнавання облич тощо – це деякі з областей, де згорткові нейронні мережі широко використовуються.

У порівнянні з традиційними методами комп'ютерного зору, глибоке навчання дозволяє інженерам комп'ютерного зору досягти більшої точності. Оскільки нейронні мережі, які використовуються в глибокому навчанні, навчаються, а не програмуються, програми, що використовують цей підхід, часто

вимагають менше експертного аналізу та тонкої настройки та використовують величезну кількість відео- та цифрових даних, доступних у сучасних системах. Глибоке навчання також забезпечує гарну гнучкість, оскільки моделі та фреймворки CNN можна перенавчати за допомогою спеціального набору даних для будь-якого випадку використання, на відміну від алгоритмів комп'ютерного зору, які, як правило, більш специфічні для домену.

Для комп'ютерної діагностики захворювань шкіри використовуються декілька методів розпізнавання образів на зображенні.

Одним з перших методів були підходи, які базовані на експертних системах. Ці системи використовували блок-схеми, теорію ймовірностей, зіставляли статистичні шаблони тощо. Завдяки цьому методу користувач взаємодіє з базою знань системи через механізм висновку. Базу знань створює інженер (як правило, людина, яка має доступ до предметних знань, що стосуються мети експертної системи) [13].

Із розвитком машинного навчання, для розпізнавання образів почали дуже активно використовувати методи на основі ознак, які були розглянуті вище.

Із зростанням кількості даних та обчислювальних можливостей, поширюються використання глибокого навчання, а саме одноетапного та двухетапного виявлення об'єктів, які базуються на згорткових нейронних мережах, які поліпшують якість та швидкість розпізнавання об'єктів на зображеннях [14]. Двоетапне виявлення об'єктів (two stage object detection) використовує дві нейронні мережи: для виявлення регіону зацікавленості та для класифікації, а одноетапне розглядає виявлення об'єктів як просту проблему регресії, беручи вхідне зображення та вивчаючи ймовірності класу та координати рамки. Такі моделі досягають нижчих показників точності, але вони набагато швидше, ніж двоетапні детектори об'єктів. Прикладами двоетапного виявлення об'єктів є такі алгоритми, як Faster R-CNN (Region-based Convolutional Neural Network), Mask R-CNN, які використовують мережу регіональних пропозицій (RPN) для створення регіонів, що представляють інтереси на першому етапі, і передають пропозиції регіону по конвеєру для класифікації об'єктів та регресії

обмежень [15]. З іншого боку найбільш поширеними прикладами одноетапного виявлення об'єктів є YOLO (You Only Look Once) and SSD (Single Shot MultiBox Detector).

2.2 Вибір методів розпізнавання образів для подальшого дослідження

Обрання методів розпізнавання образів, що найбільш підходять поставленій задачі, можна розглядати як задачу прийняття рішень. Серед різноманіття підходів теорії прийняття рішень насамперед можна визначити різноманітні згортки для багатокритеріального аналізу, що дозволяють оцінити корисність того чи іншого рішення, враховуючи різноманітні критерії якості таких рішень.

Найчастіше множину критеріїв зводять до одного глобального та розв'язують класичну однокритерійну задачу, найчастіше за допомогою методів згортання. Найуживанішими методами згортання є лінійне згортання ненормованих та нормованих критеріїв [16]. Метод лінійної згортки критеріїв полягає у призначенні тим чи іншим способом коефіцієнтів у лінійному згортці (лінійній комбінації) вихідних критеріїв та подальшій її екстремізації на безлічі допустимих варіантів. Відповідно до цього методу знайдене у такий спосіб рішення вважається «найкращим».

Інший підхід до розв'язання проблеми багатокритерійності – аксіоматичний – полягає у формулюванні множини аксіом з подальшим формальним виведенням виду функції корисності (глобального критерію), за допомогою якого й здійснюється остаточний вибір. За допомогою цієї згортки можна обрати найбільш оптимальний метод, враховуючи усі недоліки та переваги існуючих альтернатив.

Для порівняння методів було вирішено використання лінійної адитивної згортки, та наступні критерії були визначені:

- незалежність від освітлення та врахування кольору об'єкту при розпізнаванні на зображенні, оскільки колір проблемної часті шкіри може відігравати важливу роль при діагностуванні захворювання;
- нечутливість до обертання об'єкту, що розпізнається;

- рівень автоматизації, а саме наскільки важливе людське втручання при виконанні розпізнавання;
- швидкість обробки зображення;
- вимоги до обчислювальних ресурсів (пам'яті та навантаження процесора).

Задача прийняття рішень у загальному вигляді змодельована за допомогою таблиці 2.1.

За визначеними критеріями була створена шкала оцінок:

а) незалежність від освітлення:

- 1) висока (об'єкт з різним освітленням розпізнається однаково) – 5 балів;
- 2) середня (об'єкт з різним освітленням іноді розпізнається неоднаково) - 3 бали;
- 3) низька - 0 балів;

б) нечутливість до обертання:

- 1) висока (повернутий об'єкт добре розпізнається) – 5 балів;
- 2) середня (повернутий об'єкт не завжди розпізнається) – 3 бали;
- 3) низька – 0 бал;

в) рівень автоматизації:

- 1) повна (система навчається самостійно) – 5 балів;
- 2) середня (потребує налаштування експертом) – 3 балів;
- 3) низька (розпізнавання виконується людьми-експертами) – 0 балів.

г) швидкість обробки зображення:

- 1) задовільна – 3 бали;
- 2) низька – 1 бал.

д) вимоги до обчислювальних ресурсів:

- 1) звичайні – 10 балів;
- 2) середні – 4 балів;
- 3) високі – 1 бал;
- 4) дуже високі (потребують дорогі GPU тощо) – 0 балів.

Таблиця 2.1 – Моделювання задачі прийняття рішень

	Незалежність від освітлення	Нечутливість до обертання	Рівень автоматизації	Швидкість обробки	Вимоги до обчислювальних ресурсів
Методи на основі зовнішнього вигляду	Низька	Середня	Середня	Задовільна	Середні
Методи на основі ознак	Середня	Середня	Середня	Задовільна	Середні
Одноетапні моделі виявлення об'єктів	Висока	Висока	Повна	Низька	Високі
Двухетапні моделі виявлення об'єктів	Висока	Висока	Повна	Низька	Дуже високі

При аналізі медичних зображень найбільш важливими критеріями є точність та правильність результату. Також такі зображення досить рідко обробляються в режимі реального часу, тому швидкість виконання розпізнавання не відіграє велику роль. Отже, наступні вагові коефіцієнти були встановлені:

- незалежність від освітлення: 0,31;
- нечутливість до обертання: 0,31;
- рівень автоматизації: 0,1;

- швидкість обробки зображення: 0,1;
- вимоги до обчислювальних ресурсів: 0,18 (з урахуванням використання можливостей хмарних технологій для розроблюваної системи, а саме Microsoft Azure).

Тепер замінимо усі значення на визначені оцінки та використаємо лінійно адитивну згортку з ваговими коефіцієнтами для знаходження найбільш оптимального методу за формулою 2.1:

$$Z^* = \max \sum_{j=1}^n a_j \beta_j a_{ij} \quad (2.1)$$

де α_j – нормуючі множники,

β_j – вагові коефіцієнти, що відображають відносний внесок окремих критеріїв до загального критерію,

a_{ij} – значення оцінки певного критерію для даного варіанту.

У таблиці 2.2 наведена модель задачі прийняття рішень із встановленими оцінками та ваговими коефіцієнтами.

Таблиця 2.2 – Моделювання задачі прийняття рішень з оцінками та ваговими коефіцієнтами

	Незалежність від освітлення	Нечутливість до обертання	Рівень автоматизації	Швидкість обробки	Вимоги до обчислювальних ресурсів	Z*
Методи на основі зовнішнього вигляду	0	3	3	3	4	0.194

Продовження таблиці 2.2

Методи на основі ознак	3	3	3	3	4	0.266
Одноетапні моделі виявлення об'єктів	5	5	5	1	1	0.28
Двухетапні моделі виявлення об'єктів	5	5	5	1	0	0.26
Вагові коефіцієнти	0,31	0,31	0,1	0,1	0,18	

Отже, за отриманою моделлю потрібно обирати варіант № 3, одноетапні моделі виявлення об'єктів, тобто цей метод буде використаний для розпізнавання об'єктів.

2.3 Аналіз та UML-моделювання предметної області

Для моделювання предметної області було вирішено побудування UML діаграм, які у графічному вигляді дають більш повний опис програмної системи.

UML, скорочення від Unified Modeling Language, – це стандартизована мова моделювання, що складається з інтегрованого набору діаграм, розроблених для допомоги розробникам систем і програмного забезпечення для визначення, візуалізації, конструювання та документування артефактів програмних систем, а також для бізнес-моделювання та інші непрограмні системи.

На основі аналізу концептуальної моделі та постановки задачі була спроектована така UML-діаграми, як діаграма прецедентів (Use Case Diagram) для полегшення проектування інформаційної системи.

Діаграма прецедентів описує функціональні вимоги системи з точки зору варіантів використання. Це модель передбачуваної функціональності системи (варіанти використання) та її середовища (акторів). Діаграми прецедентів дозволяють зв'язати те, що потрібно від системи, і те, як система забезпечує ці потреби.

Діаграма прецедентів наведена на рисунку 2.1.

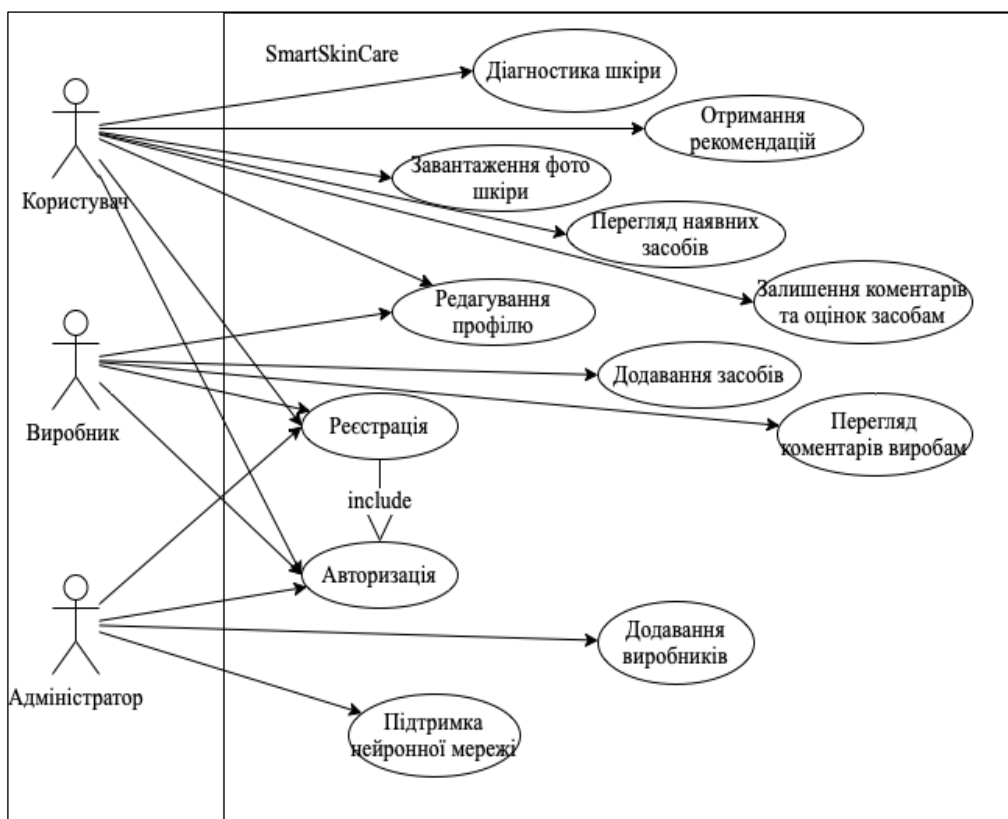


Рисунок 2.1 – Діаграма прецедентів

Вона складається з трьох акторів: користувач, виробник засобів та адміністратор систему. Користувач може завантажити фотографію проблемної зони своєї шкіри, отримати результати діагностики та рекомендації щодо покращення стану шкіри, перегляду та редагувати перелік наявних у нього засобів для догляду за шкірою, залишати коментарі та оцінки тощо.

Виробник засобу може реєструватися у системі як “виробник” та додавати нові засоби для вирішення проблем шкіри, а також переглядати відгуки та оцінки від користувачів. Адміністратори має змогу додавати нових виробників у систему

та підтримувати та керувати нейронною мережею, яка використовується для діагностики проблем шкіри.

2.4 Розробка алгоритму для розпізнавання захворювань шкіри

Алгоритм, що потрібно реалізувати для розпізнавання образів на цифровому зображенні, буде використовувати згорткову нейронну мережу, на чому базуються одноетапні та двухетапні моделі виявлення об'єктів. Розглянемо принцип роботи даної мережі.

Згорткова нейронна мережа складається з шарів входу (Input) та виходу (Output), а також із декількох прихованих шарів. Приховані шари даної мережі зазвичай складаються зі згорткових шарів (Convolution), агрегуювальних шарів (Pooling), повноз'єднаних шарів (Fully-Connected), які відображені на рисунку 2.2.

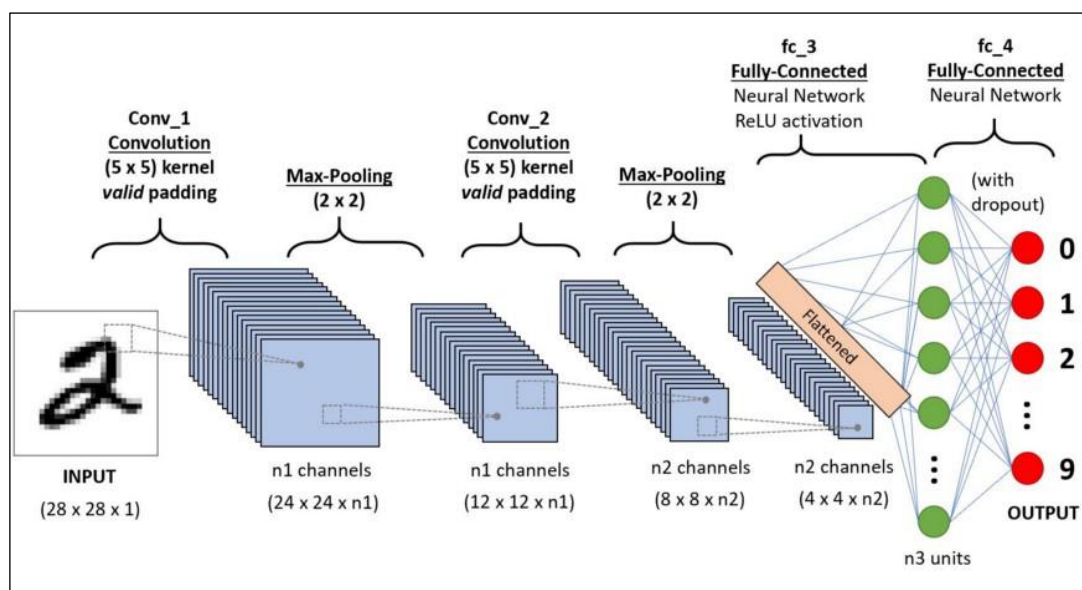


Рисунок 2.2 – Шари згорткової нейронної мережі [17]

Центральними у згортковій нейронній мережі є згорткові шари, які дають мережу назву. Ці шари виконують операцію під назвою «згортка». У контексті згорткової нейронної мережі згортка – це лінійна операція, яка включає множення набору вагових показників на вхідні дані, як і традиційна нейронна мережа. Враховуючи, що методика була розроблена для двовимірного введення, множення виконується між масивом вхідних даних і двовимірним масивом ваг, який називається фільтром або ядром.

Перший згортковий шар зазвичай витягує основні елементи, такі як горизонтальні або діагональні краї. Цей вихідний результат передається до наступного шару, який виявляє більш складні елементи, такі як кути або комбінаційні ребра. У міру того, як ми просуваємося глибше в мережу, вона може ідентифікувати навіть більш складні об'єкти, такі як об'єкти, обличчя тощо.

Шар агрегації відповідає за зменшення просторового розміру згорнутого елемента. Це потрібно для зменшення обчислювальної потужності, необхідної для обробки даних, за рахунок зменшення розмірів. Існує два типи агрегації: середня (Average Pooling) та максимальна (Max Pooling). Max Pooling знаходить максимальне значення пікселя з частини зображення, покритої ядром, а Average Pooling – відповідно середнє.

Максимальна агрегація найбільш частіше використовується у згорткових мережах, адже вона також виконує роль шумозаглушувача. Вона повністю відкидає шумні активації, а також виконує видалення шумів разом із зменшенням розмірності.

Повноз'єднані шари з'єднують кожен нейрон одного шару з кожним нейроном наступного шару. Це, в принципі, є тим же, що й традиційна нейронна мережа багат шарового перцептрон.

При використанні згорткової нейронної мережі потрібно приділити увагу до оптимального налаштування гіперпараметрів. Дана мережа використовує більше гіперпараметрів, ніж стандартний багат шаровий перцептрон. Найважливішими є кількість фільтрів, форма фільтрів, форма максимізаційних агрегувань тощо.

Розглянемо більш детально одноетапні моделі розпізнавання об'єктів на зображенні, а саме модель YOLO. Модель YOLO базується на архітектурі згорткової нейронної мережі та використовує підхід на основі сітки для виявлення об'єктів.

У ході аналізу зображення вхідне зображення ділиться на $N \times N$ клітинок сітки. Для кожного об'єкта на зображенні одна клітинка сітки відповідає за прогнозування об'єкта. Кожна клітинка передбачає ймовірність обмежувальної рамки і класу. Обмежувальна рамка складається з 5 компонентів (x , y , w , h ,

confidence), де (x, y) – координати, що представляють центр обмежувальної рамки, (w, h) – ширина та висота рамки, а confidence представляє наявність/відсутність будь-якого об'єкта.

При аналізі шкіри досить важливо враховувати, що об'єкти, які мають бути розпізнаними, можуть знаходитися близько до одне одного. Тому часто можуть виникати випадки, коли декілька об'єктів є в одній клітинці, і одна обмежувальна рамка не зможе точно відобразити всі об'єкти на зображенні. Для вирішення цієї проблеми використовують якорі (anchors). Якорі – це набір попередньо визначених обмежувальних рамок, які використовуються для представлення об'єктів різних форм і розмірів. Замість того, щоб використовувати одну обмежувальну рамку для представлення кожного об'єкта, алгоритм вибирає рамку прив'язки, яка найкраще відповідає формі та розміру об'єкта. Це дозволяє алгоритму точніше прогнозувати розташування та розмір об'єктів на зображенні.

Визначимо алгоритм підготовки та навчання одноетапної моделі розпізнавання захворювань шкіри на зображенні, який наведений на рисунку 2.3.



Рисунок 2.3 – Блок-схема алгоритму підготовки моделі

Для навчання, по-перше, потрібно створити набір даних виявлення об'єктів для тренування нейронної мережі з анотаціями, в яких визначаються координати рамок образу. Основою набору даних будуть зображення з набору Skin-Deaseses [18].

Щоб підвищити ефективність машинного навчання та класифікації, до вихідних зображень можна застосовувати різні технології попередньої обробки

[19]. Існує кілька методів попередньої обробки, які зазвичай використовуються при виявленні об'єктів для покращення продуктивності та точності моделі:

- зміна розміру зображення до фіксованого розміру;
- нормалізація значень пікселів на зображенні до фіксованого діапазону, що допомагає покращити стабільність і конвергенцію моделі;
- обрізання зображення для видалення нерелевантних областей;
- збільшення даних за рахунок обертання та додавання шуму до зображення, що може допомогти збільшити різноманітність навчальних даних;
- перетворення зображення в інший колірний простір, наприклад у відтінках сірого або HSV, яке допомагає покращити точність моделі та стійкість до змін умов освітлення.

Враховуючи властивості підготовленого набору даних, обрізання зображень не є необхідним, бо нерелевантних областей майже немає. Також для багатьох проблем та захворювань шкіри колір досліджуваної області грає велику роль у визначенні типу, тому будь-які маніпуляції з кольоровим простором можуть призвести до некоректного навчання моделі.

Для поліпшення точності розпізнавання буде використане попереднє навчання. Процес попереднього навчання для YOLO зазвичай передбачає навчання моделі на великому наборі даних, наприклад COCO, щоб вивчити набір загальних характеристик об'єкта. Це початкове навчання, як правило, проводиться з використанням великої кількості немаркованих зображень і може розглядатися як форма самоконтролю. Також було виявлено, що попередньо підготовлені моделі показують кращу продуктивність ніж випадково ініціалізовані [20].

Після попереднього навчання модель YOLO можна налаштувати на меншому наборі даних для конкретного завдання виявлення об'єктів, наприклад виявлення конкретних типів захворювань шкіри.

Перед навчання моделі можна також налаштувати різні гіперпараметри мережі.

Одним із найголовніших гіперпараметрів системи є розмір вхідного зображення (imgsz). У останній версії YOLO, розмір за замовчування є 640x640. Збільшення розміру призводить до повільнішої обробки зображення та збільшенню використання пам'яті та обчислювальних ресурсів. Однак дане значення є досить маленьким для досліджуваної області, бо захворювання шкіри можуть бути не чітко вираженими на шкірі та займати маленьку площу. Отже, для програмної системи потрібно збільшити розмір до 1280x1280.

За замовчуванням у YOLO використовуються 3 якорі різних розмірів: P3 для розпізнавання малих об'єктів на зображенні, P4 – для середніх, та P5 – для великих. Для розроблюваної програмної системи області зображення, що мають бути розпізнані, мають невелику площу, тому можна оптимізувати алгоритм за допомогою видалення останнього якоря.

Після налаштування мережі можна розпочати навчання. Навчання нейронної мережі є ітеративним процесом. На кожній ітерації виконується перехід через шари моделі, щоб обчислити вихідний результат (output) для кожного навчального прикладу у наборі даних. Потім продовжується інший прохід по шару, визначаючи, наскільки кожен параметр впливає на кінцевий результат, та обчислюючи градієнт щодо кожного параметра. Середній градієнт для набору, параметрів і певного стану оптимізації за параметрами передається алгоритмом оптимізації, наприклад оптимізацією «Адам», яка обчислює параметри наступної ітерації, які повинні мати трохи кращу продуктивність. Протягом декількох ітерацій навчання, модель розвивається, щоб видавати точніші результати.

Процес навчання оцінюватиме модель в кінці кожної епохи. Якщо жодних помилок не виникає, а процес навчання проходить гладко, навчальне завдання буде зупинено або через кінець епохи навчання, або якщо зворотний виклик з ранньої зупинки не виявить подальшого покращення моделі та зупинить загальний процес. Якщо точність навчання є низькою, можна додати нові згорткові шари та шари максимальної агрегації, або експериментувати з іншими параметрами моделі.

Отже, після успішного навчання мережі можна запуснути навчену моделі в режимі висновку або валідації.

2.5 Вибір метрик для експериментального дослідження

У задачах виявлення об'єктів із використанням YOLO існує кілька показників, які зазвичай використовуються для оцінки продуктивності моделі. Ці показники використовуються для вимірювання точності моделі при виявленні об'єктів на зображеннях і можуть використовуватися для порівняння продуктивності різних моделей або для відстеження прогресу моделі під час навчання.

Найважливішою метрикою є середня точність (mAP, Mean Average Precision), яка використовується для вимірювання продуктивності моделей комп'ютерного зору. mAP дорівнює середньому значенню середньої точності (AP) для всіх класів у моделі. AP можна розрахувати за допомогою формули 2.2:

$$AP = \sum_{k=0}^{k=n-1} [recall(k) - recall(k + 1)] \times precision(k) \quad (2.2)$$

де n – кількість порогових значень.

Точність (Precision) вимірює відсоток справжніх позитивних (TP, true positive) виявлень серед усіх виявлень (TP і FP), зроблених моделлю, тоді як запам'ятовування (Recall) вимірює відсоток справжніх позитивних виявлень від усіх правдивих об'єктів на зображенні. Отже, точність визначається формулою 2.3:

$$precision = \frac{TP}{TP + FP} \quad (2.3)$$

Тоді як запам'ятовування визначається формулою 2.4:

$$recall = \frac{TP}{TP + FN} \quad (2.4)$$

Також важливе застосування придушення без максимального значення (non-maxima suppression), щоб придушити слабкі обмежувальні рамки, що перекриваються.

Останнім кроком потрібно відобразити остаточні результати виявлення образу, тобто намалювати рамку та відобразити текст класу на зображенні, та зберегти результати у базі даних.

2.6 Проєктування бази даних

До початку проєктування бази даних, потрібно провести концептуальне моделювання предметної області. Концептуальна модель даних – це організоване уявлення про концепції бази даних та їх взаємозв'язки. Метою створення концептуальної моделі даних є встановлення сутностей, їх атрибутів та зв'язків. На цьому рівні моделювання даних майже немає будь-яких деталей щодо фактичної структури бази даних.

Як частина концептуального моделювання предметної області була спроектована схема взаємозв'язку об'єктів, яка зображена на рисунку 2.4 у вигляді загальної діаграми класів. Вона дозволяє розробляти візуалізацію та вивчити взаємозв'язки між різними класами об'єктів в системі.

При аналізі предметної області та концептуального моделювання були виявлені шість сутностей: користувач, діагностика шкіри, проблема шкіри, засіб по догляду за шкірою, виробник та комплекс засобів.

У досліджуваній предметній області можна відокремити такі обмеження цілісності даних, як:

- один користувача може провести багато діагностик шкіри (1:∞);
- одна діагностика може виявити багато проблем шкіри (1:∞);
- одна проблеми шкіри може бути вилікувана декількома засобами (1:∞);
- один засіб для догляду за шкірою може вирішувати багато проблем шкіри (1:∞);

- один засіб для догляду може належати кільком користувачам (1:∞);
- один користувач може мати багато засобів для догляду за шкірою (1:∞);
- один виробник може виробляти багато засобів (1:∞);
- один користувач може мати один рекомендований комплекс засобів (1:1);
- один рекомендований комплекс засобів може складатися з багатьох засобів (1:∞).

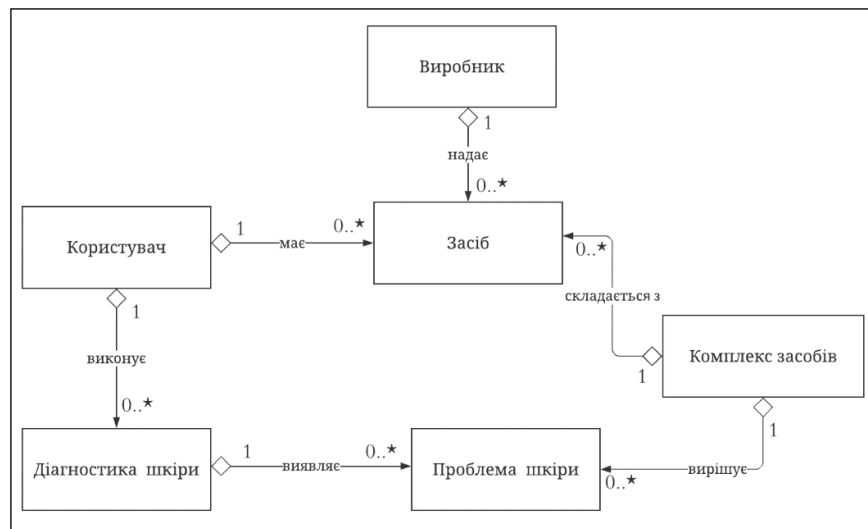


Рисунок 2.4 – Загальна діаграма класів

Отже, після визначення сутностей, обмежень цілісності даних можна перейти до вибору та створення логічної моделі бази даних.

Логічна модель даних використовується для визначення структури елементів даних і встановлення зв'язків між ними. Логічна модель даних додає додаткову інформацію до елементів концептуальної моделі даних. Перевага використання логічної моделі даних полягає в тому, щоб забезпечити основу для формування бази для фізичної моделі. Проте структура моделювання залишається загальною.

При проектуванні була проведена нормалізація бази даних до третьої нормальної форми. Структура отриманої моделі бази даних, наведена на рисунку 2.5.

Решеткою позначені первинні ключі таблиць. Для сутностей, що повинні мати складений ключ, декілька строк позначені такою решеткою. При проектуванні структури таблиць були використані різні види відношень: один до одного, один до багатьох та багато до багатьох.

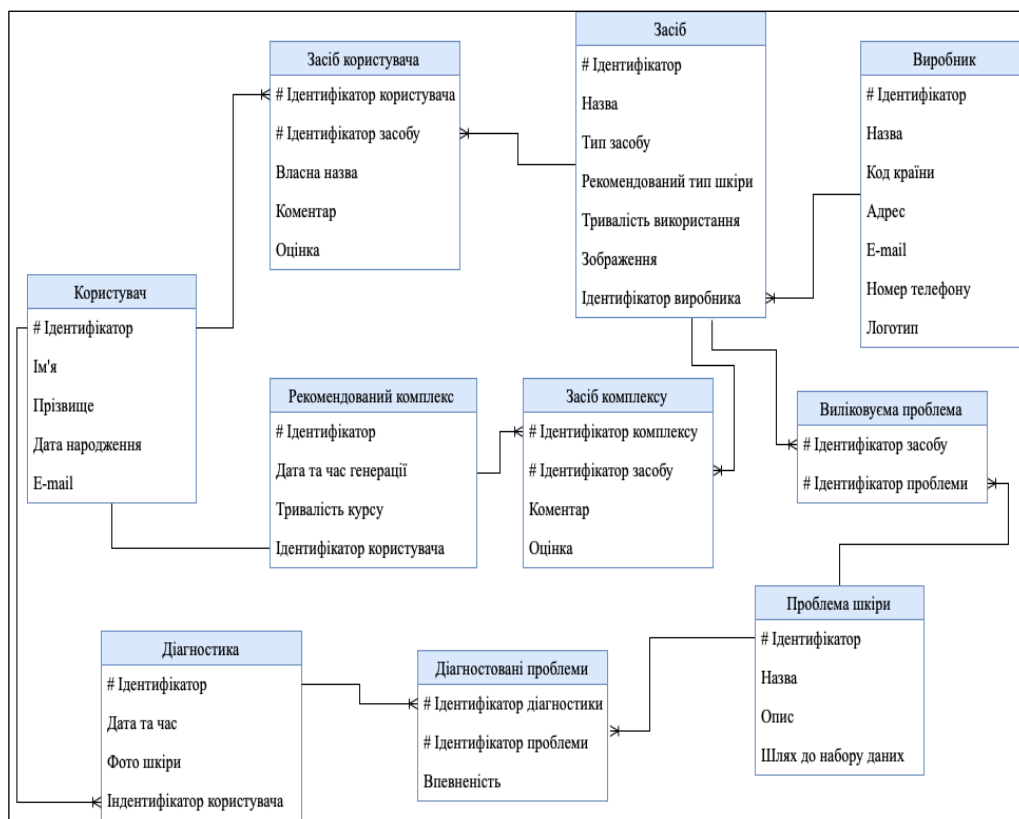


Рисунок 2.5 – Схема бази даних

Отже, після створення логічної моделі даних були відокремлені десять таблиць:

- “користувач”, який має наступні стовпці: ім’я, прізвище, дата народження та поштову адресу;
- “діагностика шкіри” має наступні атрибути: дата та час проведення, шлях до фотографії досліджуваної зони шкіри, ідентифікатор користувача, для якого була проведена діагностика;
- “діагностовані проблеми” є проміжною сутністю між діагностикою та проблемами шкіри, бо у результаті діагностики можуть бути виявлені багато проблем;

- “проблема шкіри”, яка зберігає інформацію про підтримувані системою проблеми шкіри (акне, пігментація тощо). Має наступні стовпці: назва, опис та шлях до набору зразків, що будуть використовуватися для навчання нейронної мережі;
- “відліковуєма проблема” – проблеми шкіри, які можуть бути вилікувані певним засобом для догляду за шкірою. Виступає як проміжна сутність між таблицями “засіб” та “проблема шкіри”;
- “засіб”, який представляє собою засіб, який спроможен вилікувати проблеми шкіри, та має наступні стовпці: назва, тип засобу (косметичний, медичний тощо), рекомендований тип шкіри (нормальний, сухий, жирний або комбінований), рекомендована тривалість використання у днях та ідентифікатор виробника, зображення;
- “виробник” має наступні стовпці: назва, код країни, логотип, адрес та поштовий адрес, телефон контактної особи;
- “засіб користувача” - проміжна сутність для відношення типу “багато до багатьох” між “користувач” та “засіб”. Містить також інформацію про власну назву, оцінку та коментар від користувача;
- “рекомендований комплекс” – комплекс засобів, який має віришити проблеми шкіри користувача. Містить наступну інформацію: ідентифікатор користувача, дата та час генерації рекомендації та тривалість курсу лікування;
- “засіб комплексу” представляє наявність засобу у рекомендованому комплексі.

Відносини у визначеній схемі бази даних знаходяться у третій нормальній формі, бо всі атрибути відношень є простими, всі використовувані домени містять тільки скалярні значення, кожен неключовий атрибут залежить лише від первинного ключа, та відсутні транзитивні функціональні залежності неключових атрибутів від ключових.

Отже, розроблена структура бази даних містить усю інформацію, що було виділено при аналізі предметної області.

2.7 Розробка архітектури системи

Для проектуванні архітектури системи було вирішено розроблення діаграми розгортання, що графічно дозволяє описати архітектуру.

Діаграма розгортання допомагає моделювати фізичний аспект об'єктно-орієнтованої програмної системи. Це структурна діаграма, яка показує архітектуру системи як розгортання (розповсюдження) програмних артефактів до цілей розгортання. Артефакти представляють конкретні елементи у фізичному світі, які є результатом процесу розвитку. У більшості випадків діаграма розгортання включає моделювання апаратних конфігурацій разом із програмними компонентами.

Діаграма розгортання наведена на рисунку 2.6.

Програмна система повинна складатися з серверної частини та декількох клієнтів, тобто реалізувати модель клієнт-сервер. Модель клієнт-сервер – це структура програми, яка розподіляє завдання або робоче навантаження між постачальниками ресурсу або служби, які називаються серверами, і запитувачами послуг, які називаються клієнтами. В архітектурі клієнт-сервер, коли клієнт-комп'ютер надсилає запит на дані серверу через Інтернет, сервер приймає запитуваний процес і доставляє запитані пакети даних назад клієнту. Клієнти не діляться своїми ресурсами. У даному випадку серверна частина буде обробляти запити двох клієнтських застосунків: веб-клієнту та мобільного застосунку, а також звертатися до бази даних та сервісу для розпізнавання образів на зображенні.

Комунікація між клієнтськими застосунками та серверною частиною буде виконуватися за допомогою протоколу HTTPS та архітектурного стилю REST, який полегшує взаємодію систем між собою. В архітектурному стилі REST реалізацію клієнта та реалізацію сервера можна виконувати незалежно один від одного, не знаючи один про одного. Це означає, що код на стороні клієнта можна змінити в будь-який час без впливу на роботу сервера, а код на стороні сервера можна змінити, не впливаючи на роботу клієнта.

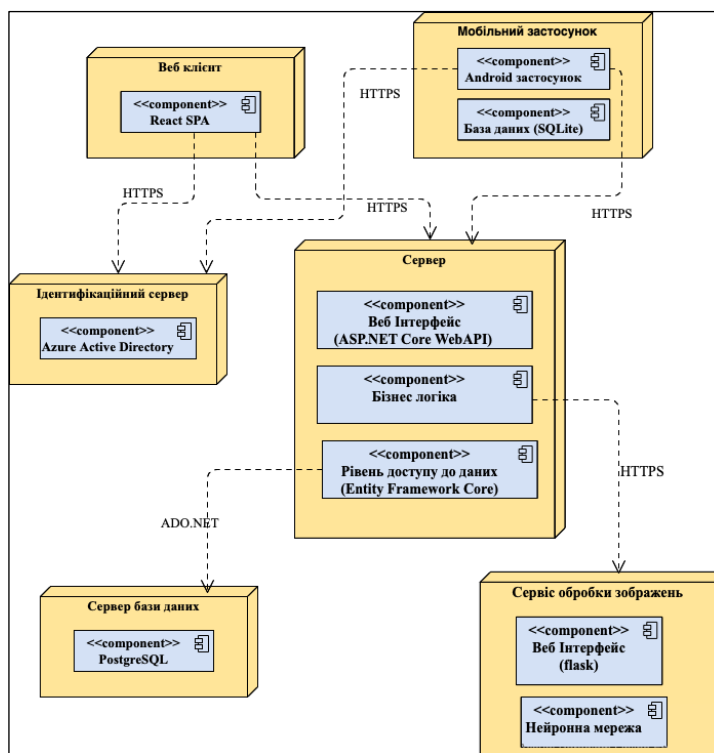


Рисунок 2.6 – Діаграма розгортання

HTTPS – це захищена версія HTTP, який є основним протоколом, який використовується для передачі даних між веб-браузером і веб-сайтом. HTTPS шифрується для підвищення безпеки передачі даних. Це особливо важливо, коли користувачі передають конфіденційні дані, наприклад, увійшовши до банківського рахунку, служби електронної пошти або постачальника медичного страхування.

Для безпечної авторизації та аутентифікації було вирішено використання хмарної технології Azure Active Directory. Azure Active Directory (Azure AD) – це хмарна служба керування ідентифікацією та доступом від Microsoft, яка допомагає користувачам входити в систему та отримувати доступ до ресурсів [21]. Azure AD можна використовувати як підхід на основі стандартів для додавання системи єдиного входу (SSO) до програмного продукту. Azure AD використовує REST API для передачі даних з однієї системи в інші хмарні програми та системи, які підтримують REST (а це більшість хмарних програм).

Для зберігання даних було вирішено використати потужну об'єктно-реляційну базу даних PostgreSQL з відкритим вихідним кодом, яка використовує

та розширює мову SQL у поєднанні з багатьма функціями, які безпечно зберігають і масштабують складні робочі навантаження даних. PostgreSQL може легко керувати найвагомішими запитами найбільших компаній і установ. Його ефективність доведена роками. Крім того, Postgres ніколи не припиняє оптимізувати свою продуктивність, випускаючи все нові й нові версії [22].

Серверна частина буде звертатися до бази даних за допомогою технології ORM (Object Relational Mapping), яка відображає об'єктні сутності (наприклад, властивості та класи) програми у реляційні об'єкти (наприклад, таблиці та стовпці) бази даних. Це дозволяє керування схемою бази даних програми прямо з коду. Існує кілька ORM, таких як Dapper, NHibernate і Microsoft Entity Framework. Для досліджуваної програмної системи було вирішено використання Entity Framework Core, який є легкою та розширеною версією популярної технології доступу до даних Entity Framework.

Обробку зображень було вирішено відокремити у самостійний сервіс, бо вона передбачає запуск нейронної мережі та використання великої кількості обчислювальних ресурсів машини. Сервіс має мати веб інтерфейс для комунікації зі серверною частиною з використанням мікро фреймворку flask.

Мобільний застосунок має орієнтуватися на Android платформу та використовувати вбудовану реляційну базу даних SQLite для зберігання тимчасових даних.

3 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

3.1 Вибір засобів програмної реалізації

Вибір засобів реалізації програмного забезпечення є важливими для успіху будь-якого проекту розробки, оскільки вони можуть впливати на ефективність, якість і вартість процесу розробки. Тому вибір правильних інструментів має вирішальне значення для забезпечення виконання проекту вчасно, у межах бюджету та з бажаною якістю. В даний час існує безліч порівняльних діаграм, таблиць, графіків, які досліджують переваги та недоліки різних технологій і пропонують переконливі аргументи на підтримку або проти їх вибору. Тим не менш, навіть за наявності технічної кваліфікації визначення того, що найкраще підійде для проекту, і вибір правильного варіанту з безлічі рішень, доступних на ринку, може зайняти багато часу.

Коли справа доходить до розробки сервера, розробникам доступні різні технології на вибір. Однією з популярних технологій для побудови серверної частини програмної системи є ASP.NET Core, крос платформна структура з відкритим кодом, розроблена Microsoft. ASP.NET Core використовується для створення надійних і масштабованих веб-застосунків із високою продуктивністю. Та підтримує різні сценарії розгортання, включаючи локальне, хмарне та контейнерне, різні механізми аутентифікації та авторизації, що полегшує розробникам захист веб-застосунків. Для розробки ASP.NET Core рішень існують кілька варіантів інтегрованого середовища розробки (IDE): Visual Studio, Visual Studio Code та JetBrains Rider тощо, серед яких було вирішено використання Visual Studio 2022.

Для реалізації задачі розпізнавання об'єктів на зображенні використовуються кілька мов програмування, таких як Python, C++, Java тощо. Python є однією з найпопулярніших завдяки простоті використання та великій кількості бібліотек, що підтримуються. Деякі з популярних бібліотек, які використовуються для виявлення об'єктів, включають OpenCV, TensorFlow, Keras, PyTorch. Ці бібліотеки забезпечують різні алгоритми виявлення об'єктів, такі як YOLO, R-CNN, SSD тощо, що досліджуються у даній роботі.

Для зберігання та отримання даних використовуються два типи систем керування базами даних: SQL і NoSQL. Вибір між SQL і NoSQL залежить від конкретних потреб проекту. Для проектів з великими обсягами неструктурованих даних і потребою в масштабованості та гнучкості NoSQL є кращим вибором. Але у розроблюваній програмній системі дані мають бути структурованими зі суворими вимогами до цілісності даних, тому SQL є найкращим варіантом. Для розробників .NET доступно декілька популярних фреймворків об'єктно-реляційного відображення, зокрема Entity Framework, NHibernate та Dapper. Кожен фреймворк має власний набір функцій і переваг, але всі вони поділяють спільну мету спрощення доступу до бази даних шляхом абстрагування від деталей SQL.

Через велику кількість технологій, доступних у світі веб-розробки, аналіз існуючих технологій може стати складним. Тенденцією є вибір технологій, яку легко освоїти, яка активно розроблюється та поширюється та надає ефективний продукт. З точки зору фреймворків веб-розробки, Angular є комплексним рішенням, тоді як React є бібліотекою JavaScript. Хоча React можна інтегрувати з іншими бібліотеками програмування, Angular є окремим рішенням, яке охоплює всі аспекти. Однак Angular має крутішу криву навчання порівняно з React, що призвело до рішення використовувати бібліотеку React для розробки веб-частини системи у редакторі коду Visual Studio Code.

Останньою частиною програмної системи є мобільний застосунок. Деякі з основних технологій, які використовуються в розробці програм для Android, включають Java, Kotlin, React Native тощо. Java є основною мовою програмування, яка використовується для розробки Android застосунків, завдяки своїй універсальності та сумісності між платформами, тому саме ця мова була обрана для подальшої розробки з використанням Android Studio.

3.2 Реалізація серверної частини та бази даних

Щоб забезпечити правильну та ефективну роботу системи програмного забезпечення, надзвичайно важливо, щоб база даних зберігала інформацію про

різні сутності, такі як дані користувача, проблеми зі шкірою, результати діагностики, косметичні продукти, їх виробники тощо. Також при цьому необхідно створити фізичну модель даних, яка визначає, як буде реалізована конкретна база даних з врахуванням типів даних, що зберігаються, обмеженням, індексів, тригерів тощо.

Як вже було описано вище, було вирішено використання PostgreSQL як систему керування реляційними базами. Вона забезпечує функції високого рівня, такі як контроль транзакцій, цілісність даних і підтримку складних запитів та широко використовується у продуктах корпоративного рівня. PostgreSQL і Entity Framework Core забезпечують потужну комбінацію для розробки застосунків .NET Core із надійною функціональністю бази даних. Використання цих технологій спрощує процес розробки та підвищує продуктивність операцій з базами даних у програмах .NET Core.

Використовуючи принцип Code First, були визначені класи сутностей, які представляють модель даних, вказані первинні ключі, зв'язки між сутностями та інші обмеження. У результаті Entity Framework Core створило схему бази даних, яка відповідає моделі, включаючи таблиці, стовпці та зв'язки, як це зображено на ER діаграмі у додатку А.

Після створення та налаштування бази даних була розроблена серверна частина програмної системи як ASP.NET Core Web API. На сьогоднішній день зазвичай використовують багаторівневу архітектуру для організації коду та забезпечення чіткої відповідальності кожного рівня. Тому серверна частина складається з трьох основних рівнів: рівня презентації, рівня бізнес-логіки та рівня доступу до даних.

Окрім розробки основних рівнів, було також важливо налагодити наскрізні проблеми, такі як запис та зберігання логів, аутентифікацію, авторизацію за допомогою Azure Active Directory, обробку помилок.

Для зручного відображення та тестування API був використаний інструмент Swagger. На рисунку 3.1 наведений перелік розроблених ендпоінтів.

Diagnoses	
GET	/api/Diagnoses/{userId}/{requestId} Returns state of diagnoses request.
POST	/api/Diagnoses/{userId} Initiates start of image processing.
Manufacturer	
GET	/api/Manufacturer/{manufacturerId} Returns manufacturer information with the list of products.
PUT	/api/Manufacturer/{manufacturerId} Updates information of a manufacturer.
DELETE	/api/Manufacturer/{manufacturerId} Removes manufacturer.
POST	/api/Manufacturer/{manufacturerId} Adds a new skin care product to manufacturer.
POST	/api/Manufacturer Adds a new manufacturer.
POST	/api/Manufacturer/{manufacturerId}/{productId} Removes skin care product and all related information from manufacturer.
Recommendations	
GET	/api/Recommendations/{userId} Returns list of recommended skin products based on diagnoses results.
User	
GET	/api/User/{userId} Returns user information with the list of products.
PUT	/api/User/{userId} Updates information of a user.
DELETE	/api/User/{userId} Removes user.
POST	/api/User/{userId} Adds a new skin care product for a user.
POST	/api/User Adds a new user.
POST	/api/User/{userId}/{productId} Adds user review to skin care product.
DELETE	/api/User/{userId}/{productId} Removes skin care product from a user.

Рисунок 3.1 – Розроблені ендпоінти

Отже, усього було реалізовано 16 захищених REST ендпоінтів, аутентифікацією та авторизацією, які створюють інтерфейс серверної частини, з яким будуть взаємодіяти клієнтські застосунки.

3.3 Опис реалізації розпізнавання дефектів шкіри на зображенні

Першим етапом реалізації розпізнавання об'єктів на зображенні є тренування нейронної мережі, де навчання еквівалентно задачі мінімізації функції втрат. Для цього був підготовлений набір даних, який складається з 800 зображень шкіри з різними захворюваннями та дефектами (див. рис. 3.2). Було

відокремлено 6 класів проблем шкіри для розпізнавання: акне (звичайні та білі вугри), рубці (пост акне), розширені пори, пігментація шкіри тощо.

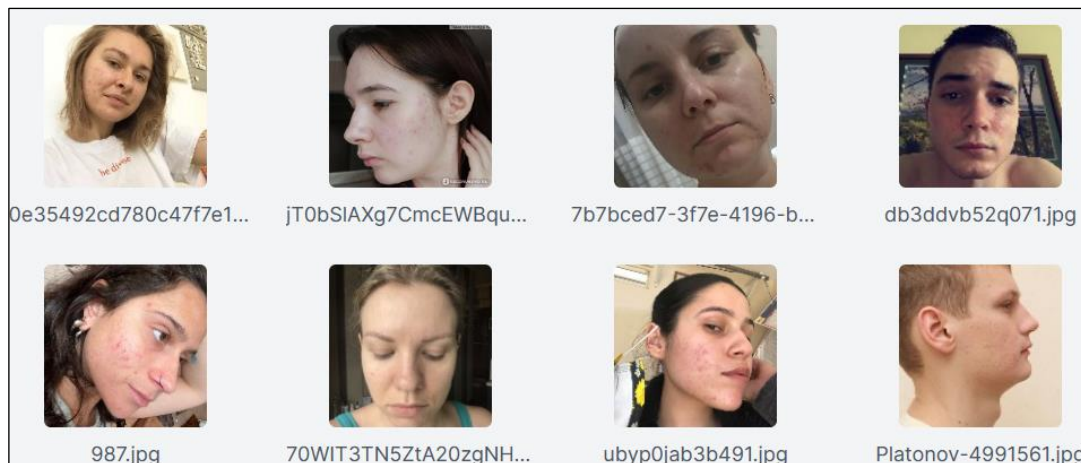


Рисунок 3.2 – Приклад зображень з набору даних

Усі зображення для тренування мають мітки з класом об'єкта, який досліджується. Приклад зображень на рисунку 3.3.



Рисунок 3.3 – Приклад оригінального та анотованого зображення

Для ефективного навчання мережі розміри всіх зображення були зменшені до 1280x1280 пікселів та нормалізовані за допомогою наступного методу з використанням бібліотеки Tensorflow та Keras.

```
resizerescale = tf.keras.Sequential([
    layers.experimental.preprocessing.Resizing(imagesize, imagesize),
    layers.experimental.preprocessing.Rescaling(1.0/255)
])
```

За основу моделі була використана остання версія YOLO. Модель складається з двох основних частин: основних шарів (backbone layers), які служать екстрактором ознак, і головних шарів (head layers), які обчислюють вихідні прогнози. Щоб додатково компенсувати невеликий розмір набору даних, було виконано попереднє навчання на набору даних COCO. Основа YOLOv8 складається з 12 рівнів, які будуть виправлені аргументом «заморозити».

```
python train.py --batch 32 --epochs 150 --data
'data/skin_diseases.yaml' --weights 'yolov5s6.pt' --project
'runs_skin_diseases' --name 'feature_extraction' --cache --freeze 12
```

Наступнім етапом навчання є налаштування, яке складається з розморожування всієї моделі, яка була отримана вище, і повторного навчання на підготовлених даних із дуже низькою швидкістю навчання. Це потенційно може досягти значних покращень шляхом поступової адаптації попередньо навчених функцій до нових даних. Параметр швидкості навчання можна налаштувати у файлі конфігурацій гіперпараметрів. Ваги будуть ініціалізовані даними, збереженими на попередньому кроці.

```
python train.py --hyp 'hyp.finetune.yaml' --img 1280 --batch 16 --
epochs 100 --data 'data/skin_diseases_data.yaml' --weights
'runs_skin_diseases/feature_extraction/weights/best.pt' --project
'runs_skin_diseases' --name 'fine-tuning' --cache
```

Після навчання та збереження моделі можна виконувати передбачення, щодо існуючого на зображенні проблеми шкіри.

```
python detect.py --weights 'runs_skin_diseases/fine-
tuning/weights/best.pt' --conf 0.6 --source
'../datasets/Skin_diseases_data/images/test' --project 'runs_skin_diseases'
--name 'detect_test' --augment --line=3
```

Результати передбачення автоматично зберігаються у визначеній папці. Приклад наведений на рисунку 3.4. За допомогою наступного коду можна переглянути зразки передбачення.

```

detect_path = r'runs_skin_deseases/detect_test'

images = []
for img_path in glob.glob(detect_path + '/*.jpg'):
    images.append(mpimg.imread(img_path))

plt.figure(figsize=(20,25))
columns = 4
for i, image in enumerate(images):
    plt.subplot((int(len(images) / columns + 1)), columns, i + 1)
    plt.tick_params(left = False, right = False , labelleft = False,
labelbottom = False, bottom = False)
    plt.imshow(image)

```



Рисунок 3.4 – Приклад передбачення

Сервіс для розпізнавання захворювань шкіри має бути доступним для головної серверної частини за допомогою REST API. Для цього був реалізований функціонал для обробки POST запитів зі входним зображенням для розпізнавання, використовуючи бібліотеку Flask.

3.4 Опис інтерфейсу користувача

Для ефективної взаємодії з користувачами були розроблені веб- та мобільний застосунки. Найголовнішими задачами цих двох інтерфейсів є надання покрокових інструкцій для того, щоб гарантувати, що користувачі знімають високоякісні зображення, можливість завантаження зроблених зображень у

програму для подальшої діагностики, маніпулювання засобами для догляду за шкірою та отримання рекомендацій за результатами обробки зображень.

Веб-застосування був написаний за допомогою React, а саме JavaScript бібліотеки, що використовується для створення адаптивних і динамічних веб-рішень. У застосунку повинні підтримуватися наступний функціонал:

- реєстрація у систему або вхід із існуючого акаунту;
- перегляд та редагування профілю користувача або виробника косметичних засобів;
- перегляд каталогу засобів для догляду за шкірою, який може розширюватися виробниками, фільтрація та сортування даних;
- додавання або видалення косметичного продукту з набору засобів користувача, можливість залишення коментарів та оцінок;
- завантаження фотографії шкіри користувача та отримання результатів аналізу;
- надання рекомендацій користувачу, а саме комплексу засобів для усунення підтверджених проблем шкіри.

Для роботи із системою користувач повинен увійти у свій акаунт або зареєструватися. Аутентифікація та авторизація була розроблена за допомогою технології Azure Active Directory. На рисунку 3.5 наведено вікно для авторизації.

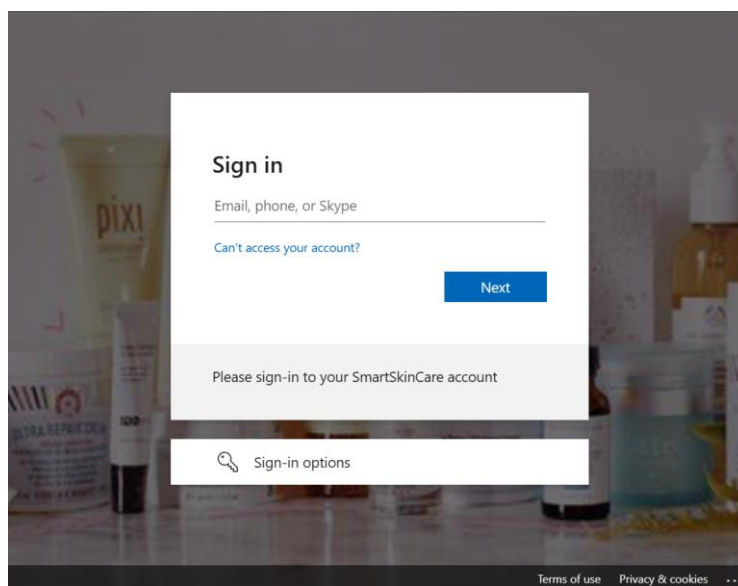
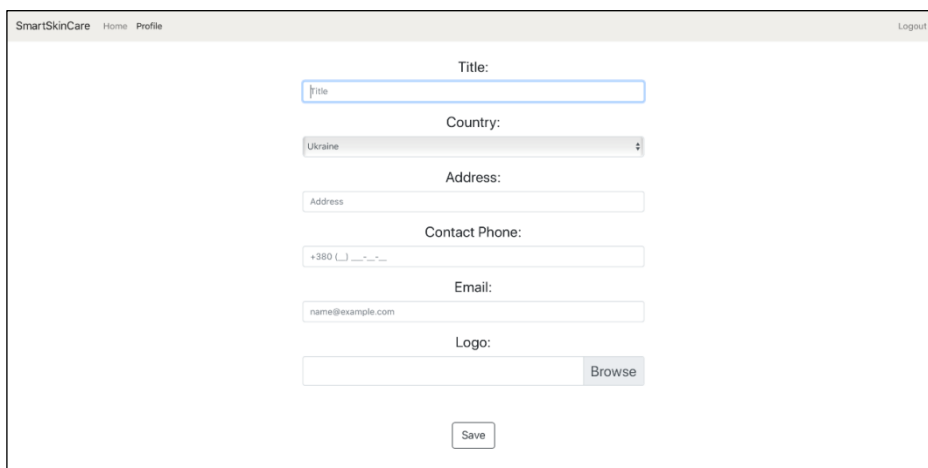


Рисунок 3.5 – Вікно авторизації користувача

Після успішної авторизації користувач має доступ до різних сторінок веб-застосування. Обравши «My Profile», він може оновити свої дані та змінити пароль за необхідності. Якщо авторизований користувач є виробником косметичних засобів, він може відредагувати дані також про свою компанію (див. рис. 3.6).



The screenshot shows a web form titled "SmartSkinCare Home Profile" with a "Logout" link in the top right. The form fields are: "Title:" with a text input containing "Title"; "Country:" with a dropdown menu showing "Ukraine"; "Address:" with a text input containing "Address"; "Contact Phone:" with a text input containing "+380 () ____-__"; "Email:" with a text input containing "name@example.com"; and "Logo:" with a text input and a "Browse" button. A "Save" button is located at the bottom center of the form.

Рисунок 3.6 – Сторінка редагування профілю виробника засобів

Для перегляду каталогу всіх доступних засобів для догляду за шкірою у системі користувач повинен обрати сторінку «Care Product Catalog». На цій сторінці можна також фільтрувати дані за типом та виробником, сортувати за оцінками тощо. Приклад сторінки наведений на рисунку 3.7. На додаток до цього, якщо користувач вже придбав якийсь продукт, надається можливість додавання її до власного набору засобів.

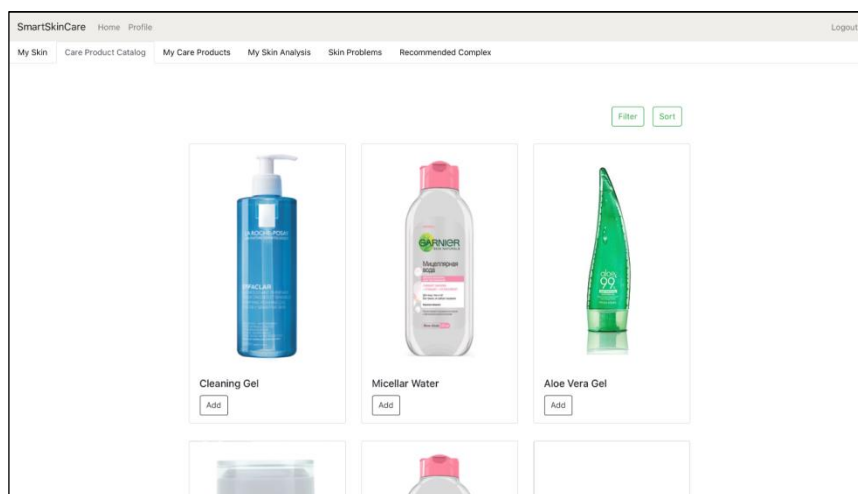
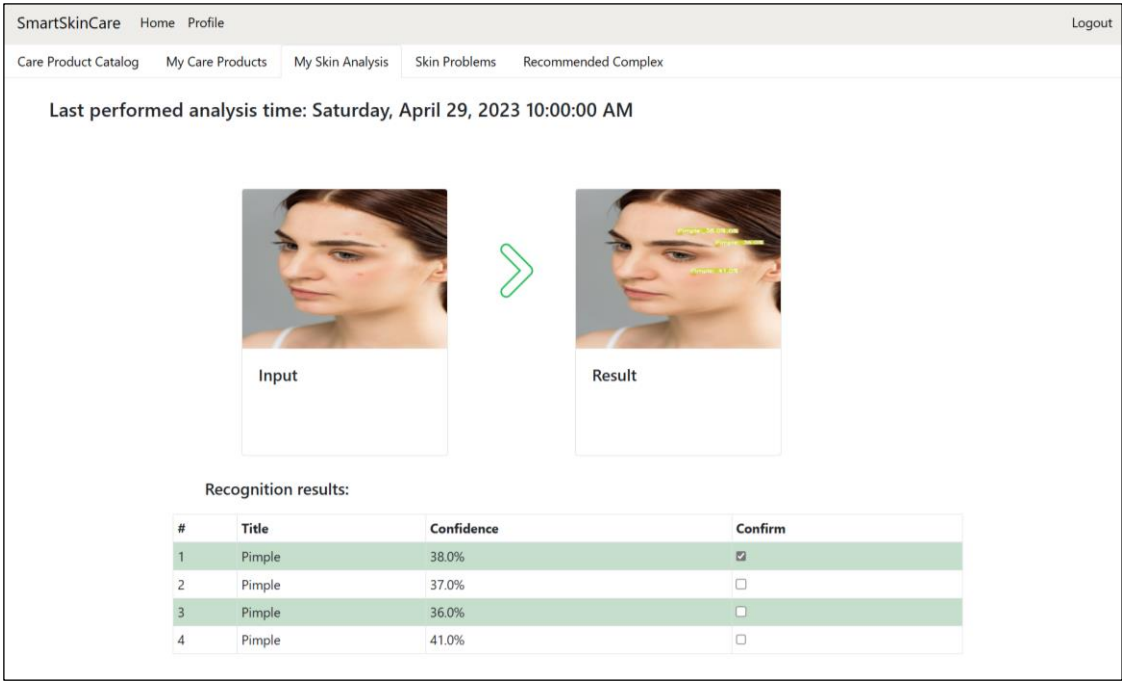


Рисунок 3.7 – Каталог косметичних засобів

Коли користувач натискає кнопку «My Care Products», він переходить на сторінку, на якій відображається каталог доданих косметичних продуктів, де він може залишити коментар та оцінку. Користувач може змінити цей набір, видаливши будь-який продукт. Крім того, якщо користувач включив інструмент у свій набір, він може залишити коментар і оцінити його. Цей набір інструментів буде використано для створення рекомендацій щодо комплексного вирішення проблем зі шкірою, яке є ефективним рішенням певних проблем.

Для виконання діагностики шкіри користувач повинен перейти до сторінки «My Skin Analysis», обрати «Run New» та уважно переглянути наведену інструкцію щодо правильного створення знімку шкіри. Після ознайомлення із інструкцією, користувачу надається можливість завантажити світлинку та почати аналіз шкіри. Після деякого проміжку часу відображаються результати діагностики, а саме оброблене зображення з розпізнаними об'єктами та перелік знайдених проблем шкіри (див. рис. 3.8). Користувач може переглянути значення впевненості у розпізнаванні, опис проблеми та підтвердити наявність.



#	Title	Confidence	Confirm
1	Pimple	38.0%	<input checked="" type="checkbox"/>
2	Pimple	37.0%	<input type="checkbox"/>
3	Pimple	36.0%	<input type="checkbox"/>
4	Pimple	41.0%	<input type="checkbox"/>

Рисунок 3.8 – Результати діагностики шкіри

Після проведення аналізів на сторінці «Skin Problems» з'являться результати розпізнавання, а також недоліки шкіри, що були визначені в минулому.

Якщо користувач виконував хоча б один аналіз шкіри й у результаті підтвердив наявність певної проблеми, він може перейти до сторінки «Recommended Complex» і отримати рекомендації щодо косметичних продуктів, які здатні поліпшити стан шкіри, та рекомендовану тривалість використання засобів. Приклад рекомендації наведений на рисунку 3.9.

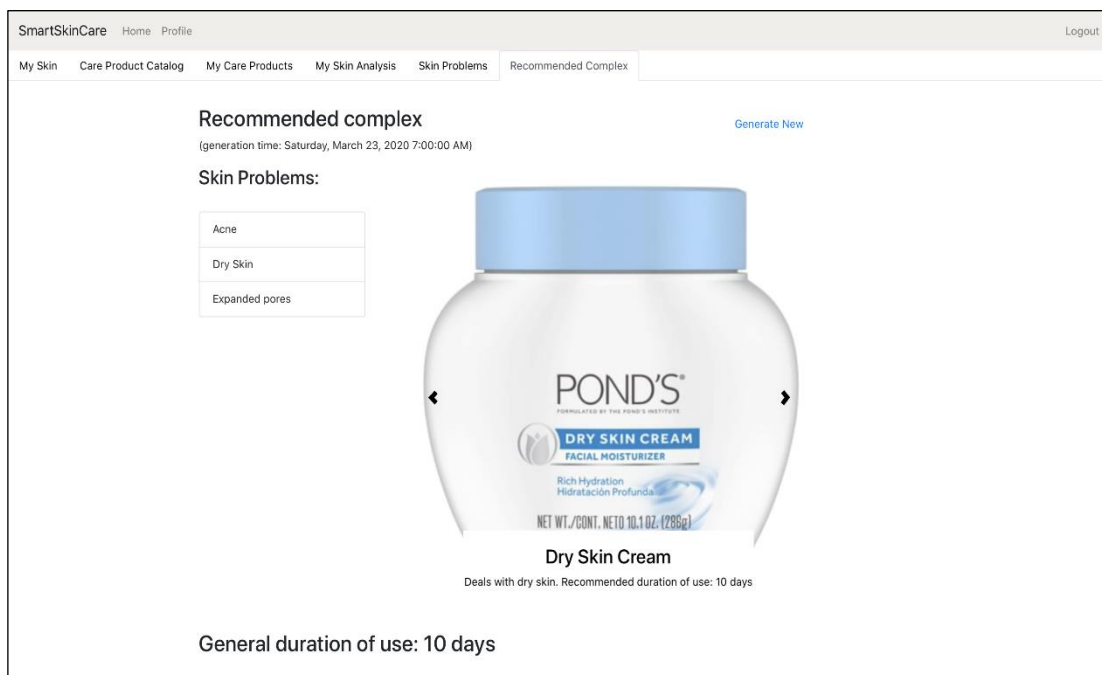


Рисунок 3.9 – Рекомендований комплекс засобів

Для покращення взаємодії з користувачами при щоденному догляді за шкірою був створений мобільний інтерфейс, який надає наступні можливості:

- реєстрація у систему або вхід із існуючого акаунту;
- перегляд та редагування профілю користувача;
- додавання або видалення косметичного продукту з набору засобів, залишення коментарів та оцінок;
- завантаження фотографії шкіри користувача з мобільного пристрою та отримання результатів аналізу;
- отримання рекомендацій та push-сповіщень відповідно до визначеного користувачем плану використання косметичних засобів.

Так само, як у веб-інтерфейсі, користувач має зареєструватися або ввійти у свій акаунт перед початком роботи із системою. У мобільний застосунок не

можливо увійти із роллю «Виробник», тому додавання, редагування та видалення косметичних засобів із повного каталогу не є можливим.

Після входу у систему користувач може перейти до свого профілю (сторінка «Profile») та оновити свої дані, якщо це необхідно. Також є можливість зміни паролю. Також є доступні сторінки «Skin Care Products», «My Skin Analysis», «My Plan», що будуть розглянуті більш детально.

Обравши сторінку про засоби для догляду за шкірою, користувач може переглянути каталог існуючих та доступних косметичних продуктів. Якщо користувач вже прибрав якийсь продукт, він може додати його до свого набору. При натисканні на доданий засіб надається інформація щодо продукту та його виробника. Також є можливість залишення коментарю та оцінки товару, як це зображено на рисунку 3.10.

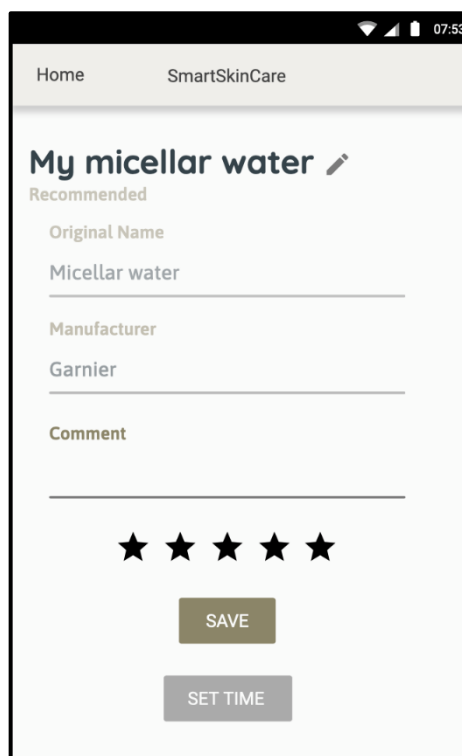


Рисунок 3.10 – Вікно редагування засобу користувача

На сторінці косметичного засобу користувача є можливість встановлення дати використання продукту. Якщо дата є коректною, на сторінці «My Plan» з'явиться новий рядок із засобом та встановленим часом використання, як це показано на рисунку 3.11. Згідно до розкладу, а саме коли настає визначений час

використання продукту, користувачеві надсилається push-сповіщення із нагадуванням.

На сторінці «My Skin Analysis» користувач має можливість створити або завантажити зображення шкіри з мобільного пристрою. Так само, як і веб-застосунку, через декілька секунд після завантаження можна отримати результати аналізу та підтвердити розпізнану проблему шкіри. Після цього система пропонує засоби, що можуть покращити стан шкіри користувача.

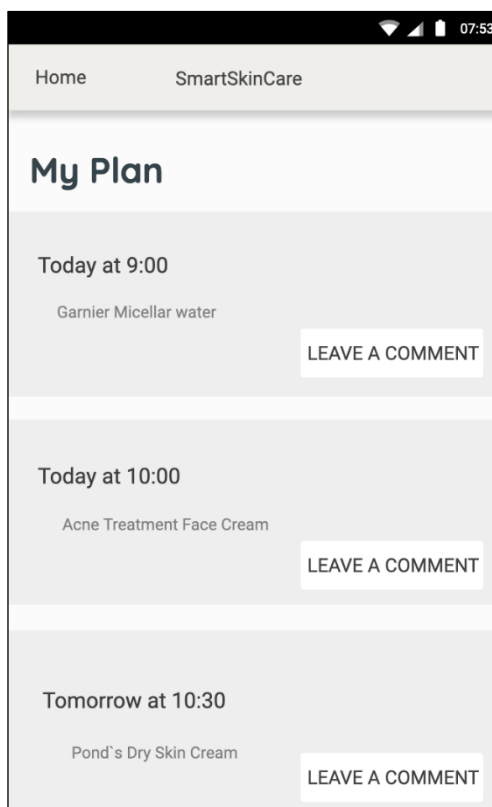


Рисунок 3.11 – Вікно для перегляду створеного плану

Отже, були розроблені веб- та мобільні застосунки для виявлення проблем зі шкірою, які дозволяють легко завантажувати та отримувати результати аналізу шкіри, надають персоналізовані рекомендації щодо лікування та догляду, використовують push-сповіщення у мобільному продукту, щоб нагадувати користувачам відстежувати свій прогрес. Ці застосування є зручними для використання та здатні надавати користувачам приємне враження.

4 ОПИС ЕКСПЕРИМЕНТАЛЬНИХ ДОСЛІДЖЕНЬ

4.1 Планування експерименту

Для створення ефективного модулю розпізнавання образів на зображенні було вирішено провести експериментальне дослідження з використанням різних варіантів алгоритмів одноетапного виявлення об'єктів, що базуються на згорткових нейронних мережах.

Найбільш поширеним та точним одноетапним алгоритмом є YOLO, який виявляє та розпізнає різні об'єкти на зображенні навіть у режимі реального часу. Виявлення об'єктів у YOLO виконується як проблема регресії та надає ймовірності класу виявлених зображень. Однак перший варіант YOLO не ефективний у виявленні близьких об'єктів і невеликих популяцій. У відповідь на ці проблеми в оригінальній мережі були запропоновані нові варіанти моделі, такі як YoloV2, YoloV3 тощо, які запозичили деякі ідеї з інших алгоритмів, як SSD і Faster R-CNN.

YoloV1 використовує одну згорткову нейронну мережу для перетворення проблеми виявлення цілі в задачу регресії, яка витягує рамки (boundary box) і ймовірності класів безпосередньо з зображення.

Мережа YoloV2 була спробою покращити роботу першої версії алгоритму. Основною відмінністю є те, що YoloV2 витягує ознаки на основі моделі Darknet-19, чий останній шар згортки змінено на три шари згортки 3×3 і 1024 канали, додаючи шар згортки 1×1 після кожної згортки для стиснення функцій і прискорення роботи.

YoloV3 включає в себе Darknet-19 від YoloV2, щоб запропонувати нову більш глибоку і широку мережу вилучення ознак під назвою Darknet-53. Основним удосконаленням є застосування мережі піраміди ознак (feature pyramid network, FPN), яка може генерувати піраміду графів характеристик вищої якості.

В YOLOv5 порівняно з попередніми версіями було введено кілька суттєвих змін, таких як нова архітектура мережі під назвою CSPNet, новий метод виявлення об'єктів без прив'язки та вдосконалення процесу навчання.

На сьогоднішній день існує 8 версій YOLO. 10 січня 2023 року була пущена нова версія YOLO, яка називається YOLO8 та стверджує про вдосконалення структури та архітектурні зміни з кращими результатами. На рисунку 4.1 відображене порівняння результатів останніх версій YOLO, які були отримані на набору даних RF100.

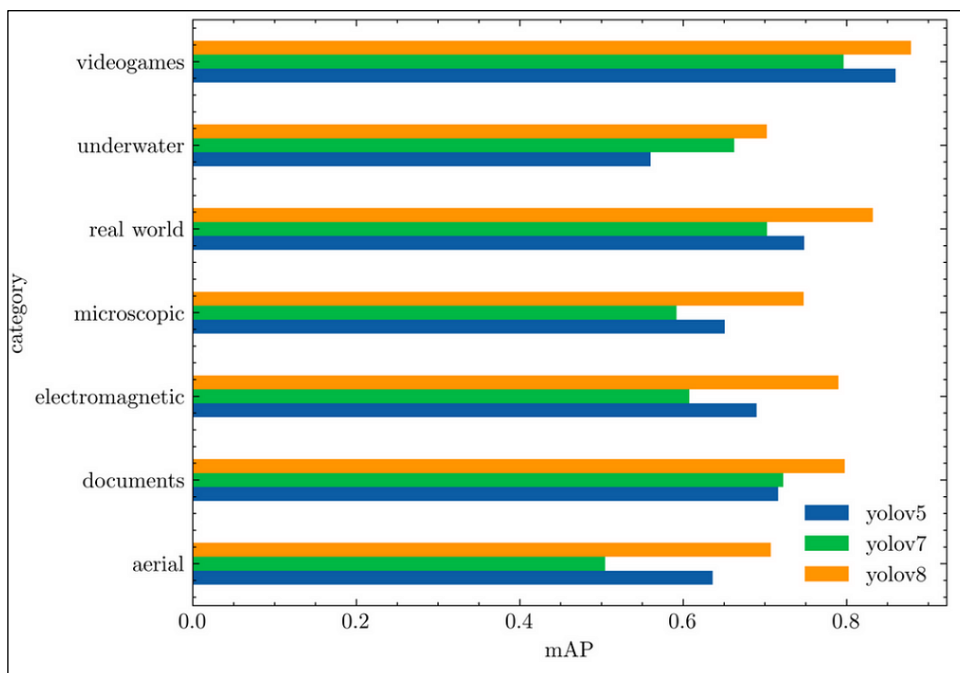


Рисунок 4.1 – Порівняння [mAP@.50](#) для різних категорій зображень

Отже, з наведеного порівняння можна зробити висновок, що для більшості категорій YOLOv5 та YOLOv8 є найбільш точними, тому вони будуть використані при проведенні експерименту для виявлення найбільш точного та швидкого метода розпізнавання проблем шкіри.

Для експерименту також було прийнято рішення розглянути вдосконалений алгоритм, запропонований у другому розділі, який був створений для вирішення конкретної задачі розпізнавання захворювань шкіри на зображенні.

Отже, серія експериментів буде складатися з наступних варіацій:

- розпізнавання проблем шкіри на зображенні за допомогою алгоритму YoloV5;
- розпізнавання проблем шкіри на зображенні за допомогою алгоритму YoloV8;

- розпізнавання проблем шкіри на зображенні за допомогою вдосконаленого алгоритму.

Ефективність методів буде визначатися за допомогою такого критерію, як середня точність (mean average precision, mAP), який є популярним показником для вимірювання точності розпізнавання від 0 до 1. Оцінка mAP визначається шляхом обчислення mAP для всіх класів і всіх порогових значень Intersection over Union (IoU). В експерименту буде врахований mAP з пороговим значення IoU 0.5, яке позначається як mAP@.5. Також важливими критеріями будуть швидкість навчання, використання обчислювальних ресурсів алгоритмами.

Усі експерименти будуть використовувати один і той же набір даних та проводитимуться на одному сервері щоб уникнути різниці у результатах. Сервер повинен мати наступні характеристики:

- 16 ГБ пам'яті GPU;
- 32 ГБ пам'яті CPU.

4.2 Експериментальне дослідження

В ході експерименту був використаний підготовлений набір даних з різними дефекатами шкіри, який складається з 800 зображень для навчання та 165 зображень для валідації мережі та з 6 категорій: акне, пост акне, розширені пори тощо.

Для експерименту кількість епох навчання було визначено 100 через використання слабкої машини. Це може призвести до гірших значень розпізнавання, бо у сучасних системах найчастіше використовують більше, ніж 1000 епох, але в даному випадку цього достатньо для порівняння ефективності різних моделей розпізнавання об'єктів на зображенні.

Спочатку була навчена модель YOLO v5, яка складається з 270 шарів, структура якої наведена на рисунку 4.2.

params	module	arguments
3520	models.common.Conv	[3, 32, 6, 2, 2]
18560	models.common.Conv	[32, 64, 3, 2]
18816	models.common.C3	[64, 64, 1]
73984	models.common.Conv	[64, 128, 3, 2]
115712	models.common.C3	[128, 128, 2]
295424	models.common.Conv	[128, 256, 3, 2]
625152	models.common.C3	[256, 256, 3]
1180672	models.common.Conv	[256, 512, 3, 2]
1182720	models.common.C3	[512, 512, 1]
656896	models.common.SPPF	[512, 512, 5]
131584	models.common.Conv	[512, 256, 1, 1]
0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
0	models.common.Concat	[1]
378624	models.common.BottleneckCSP	[512, 256, 1, False]
33024	models.common.Conv	[256, 128, 1, 1]
0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
0	models.common.Concat	[1]
95104	models.common.BottleneckCSP	[256, 128, 1, False]
147712	models.common.Conv	[128, 128, 3, 2]
0	models.common.Concat	[1]
313088	models.common.BottleneckCSP	[256, 256, 1, False]
590336	models.common.Conv	[256, 256, 3, 2]
0	models.common.Concat	[1]
1248768	models.common.BottleneckCSP	[512, 512, 1, False]
29667	models.yolo.Detect	[6, [[10, 13, 16, 30, 33, 23]]]

Рисунок 4.2 – Структура YOLO v5 за замовчуванням

Навчання моделі зайняло 47 хвилин 50 секунд та було використано 9.13Гб GPU. Результати валідації навченої моделі наведені у таблиці 4.1.

Таблиця 4.1 – Результати валідації моделі YOLO v5

Категорія	mAP@.5	Точність (Precision)	Відкликання (Recall)
Всі	0.081	0.146	0.106
Родимки	0.163	0.267	0.223
Пігментація	0.0203	0.101	0.0044
Акне	0.112	0.1023	0.0452
Пост акне	0.167	0.244	0.29
Розширені пори	0.0224	0.133	0.0269
Веснянки	0.00865	0	0

Наступним методом для дослідження є більш сучасна версія YOLO, а саме YOLO v8 з параметрами за замовчування. Ця модель складається з меншої кількості шарів, а саме 225, але більш вимоглива до обчислювальних ресурсів.

Навчання моделі пройшло майже так само, як і для першого методу, упродовж 50 хвилин 22 секунд та з використанням 8.28Гб GPU. У результаті були отримані дані, наведені у таблиці 4.2.

Таблиця 4.2 – Результати валідації моделі YOLO v8

Категорія	mAP@.5	Точність (Precision)	Відкликання (Recall)
Всі	0.104	0.661	0.125
Родимки	0.249	0.365	0.295
Пігментація	0.0136	1	0
Акне	0.122	0.276	0.117
Пост акне	0.233	0.326	0.34
Розширені пори	0.00374	1	0
Веснянки	0	1	0

Третім методом для дослідження був вдосконалений алгоритм на основі YOLO v8. Навчання зайняло 39 хвилин 47 секунд з використанням 14.53Гб GPU, та результат відображений у таблиці 4.3.

Таблиця 4.3 – Результати валідації вдосконаленої моделі

Категорія	mAP@.5	Точність (Precision)	Відкликання (Recall)
Всі	0.155	0.22	0.223
Родимки	0.289	0.353	0.379
Пігментація	0.0226	0.0601	0.053
Акне	0.282	0.296	0.419
Пост акне	0.232	0.299	0.362
Розширені пори	0.0564	0.143	0.09
Веснянки	0.05	0.17	0.04

За отриманими результатами можна зробити порівнювальну таблицю 4.4 за критеріями, визначеними у підрозділі 4.1.

Таблиця 4.4 – Порівнювання отриманих результатів

Назва моделі	mAP@.5	Швидкість навчання, секунди	Використання GPU, Гб
YOLO v5	0.081	2870	9.13
YOLO v8	0.104	3022	8.28
Вдосконалений YOLO v8	0.155	2387	14.53

Отже, завдяки попередньому навчанню моделі та налаштуванню гіперпараметрів моделі з рахунком особливостей досліджуваної області, модель є більш точною та швидкою, але споживає більше обчислювальних ресурсів.

5 АНАЛІЗ РЕЗУЛЬТАТІВ ЕКСПЕРИМЕНТАЛЬНОГО ДОСЛІДЖЕННЯ ТА РОЗРОБКА РЕКОМЕНДАЦІЙ

В ході виконання кваліфікаційної роботи магістра були досліджені методи розпізнавання захворювань шкіри на зображенні, які подальше можуть бути використані для комп'ютерної діагностики у медицині. Також був проведений експеримент для порівняння ефективності розпізнавання різних моделей.

У результаті експерименту було виявлено, що модель YOLOv5 навчається швидше, ніж остання версія YOLOv8, але розпізнавання взагалі є менш точним та потребує більше ресурсів.

З іншого боку, YOLOv8 має вище значення mAP@.5 для усіх класів, окрім пігментації, розширених пор та веснянок. Однак через особливості архітектури моделі повноцінне навчання потребує більше часу.

При навчанні третьої моделі було використано більше обчислювальних ресурсів, бо розмір оброблюваних зображень був вдвічі більше, ніж для двох інших моделей. Але це сприяло покращенню результатів навчання. На додаток до цього, третя модель була попередньо навчена, тому значення mAP@.5 при перших ітераціях навчання був набагато вище, і це сприяло позитивно на остаточне значення для усіх класів.

Також можна помітити, що усі моделі мають низьке значення точності навчання при виявленні таких особливостей шкіри, як пігментація, розширені пори та веснянки. Ці об'єкти не мають чіткої форми та кольору, тому для покращення результатів потрібно підготувати більше даних для навчання, зробити анотації точніше тощо.

Наступні рекомендації щодо розробки моделей розпізнавання образів, призначених для виявлення проблем шкіри, можуть покращити ефективність та точність розпізнавання:

- підготовка набору даних з великою кількістю зображень шкіри різного типу, кольору, з різноманітними проблемами тощо; зображення також

мають бути точно анотованими, враховуючи навіть маленькі дефекти шкіри;

- використання попередньо навчених моделей, що значно пришвидшує процес навчання та підвищує продуктивність;
- використання зображень більш високої якості для навчання, якщо обчислювальні ресурси цього дозволяють;
- застосування різних методів оптимізації, а саме налаштування гіперпараметрів, оптимізація архітектури моделі тощо;
- збільшення кількості даних (зображень) за допомогою таких методів, як випадкове обертання, масштабування, перевертання зображень та внесення шуму тощо.

Дотримуючись цих рекомендацій, розробники можуть створювати точні та ефективні моделі розпізнавання об'єктів, призначені для виявлення проблем шкіри на зображеннях. Ці моделі можуть значно сприяти ранній діагностиці, покращенню результатів лікування та допомагати дерматологам у процесі прийняття рішень.

Отже, результати дослідження можуть сприяти більш активному застосуванню згорткових нейронних мереж для автоматизації виявлення захворювань шкіри, але варто згадати, що навіть із використанням сучасних технологій, діагностика захворювань не може проводитися самостійно та результати повинні бути проконтрольованими медичними фахівцями.

Ще однією областю застосування є косметична індустрія. За допомогою глибокого навчання можна проаналізувати ефективність засобів по догляду за шкірою шляхом порівняння зображень шкіри до та після використання продукту.

ВИСНОВКИ

В ході виконання кваліфікаційної роботи магістра були досліджені методи розпізнавання захворювань шкіри на зображенні для вирішення задач комп'ютерної діагностики. Були детально розглянуті сучасні методи на основі згорткових нейронних мереж та запропоновані модифікації для існуючих моделей із врахуванням особливостей предметної області.

У результаті були виконані наступні задачі:

- проведений аналіз існуючих методів розпізнавання образів на зображенні, які використовуються у комп'ютерній діагностиці;
- сплановане та проведене експериментальне дослідження обраних методів розпізнавання проблем на шкірі;
- спроектована та розроблена програмна система, що допомагає у щоденному догляді за шкірою із врахуванням результатів розпізнавання шкірних захворювань та проблем;
- розроблені рекомендації стосовно методів розпізнавання образів з врахуванням особливостей шкіри.

У майбутньому планується розширити та покращити набір зображень із різними захворюваннями та проблемами шкіри для навчання нейронної мережі з метою отримання більш точних результатів розпізнавання. Також буде вдосконалюватися програмний продукт за допомогою впровадження порад та відгуків від користувачів і додавання нового функціоналу, наприклад, підтримку платформи iOS для мобільного застосунку.

За результатами роботи були подані тези доповіді «Використання технології комп'ютерного зору для системи розумного догляду за шкірою» на «Міжнародний молодіжний форум» (див. додаток Г).

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Kunio D. Computer-aided diagnosis in medical imaging: Historical review, current status and future potential // Computerized Medical Imaging and Graphics. 2007. С. 198-211.
2. Приборкання «великих даних». Передові методи аналітики та нові можливості для бізнесу / Білл Френкс. – John Wiley & Sons 2012. – 336 с.
3. A Gentle Introduction to Computer Vision / Machine Learning Mastery. URL: <https://machinelearningmastery.com/what-is-computer-vision/> (дата звернення: 26.11.2021).
4. A method of skin disease detection using image processing and machine learning / ScienceDirect. URL: <https://www.sciencedirect.com/science/article/pii/S1877050919321295> (дата звернення: 03.11.2021).
5. Everything You Ever Wanted To Know About Computer Vision / Towards Data Science. URL: <https://towardsdatascience.com/everything-you-ever-wanted-to-know-about-computer-vision-heres-a-look-why-it-s-so-awesome-e8a58dfb641e> (дата звернення: 05.10.2021).
6. Advantages of Using Python for Computer Vision / Full Scale. URL: <https://fullscale.io/blog/advantages-using-python-computer-vision/> (дата звернення: 15.11.2021).
7. Using AI to help find answers to common skin conditions / Google Health. URL: <https://blog.google/technology/health/ai-dermatology-preview-io-2021/> (дата звернення: 12.12.2021).
8. Застосунок Neutrogena Skin360 / Neutrogena – URL: <https://www.neutrogena.com/skin360app.html> (дата звернення: 10.12.2021).
9. Christian Demant, Bernd Streicher-Abel, Peter Waszkewitz. Industrial Image Processing: Visual Quality Control in Manufacturing / Springer Science & Business Media, 1999. – 353 с.
10. K. Lillywhite, B. Tippetts, D.-J. Lee Self-tuned Evolution-Constructed features for general object recognition // Pattern Recogn. 2012. С. 241-251.

11. Goel, R., Sharma, A., Kapoor, R.. State-of-the-Art Object Recognition Techniques: A Comparative Study // Springer. 2020. С. 69-74.

12. A Comprehensive Guide to Convolutional Neural Networks / Towards Data Science. URL: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53> (дата звернення: 15.11.2021).

13. Finlay AY, Hammond P. Expert systems in dermatology: the computer potential. The example of facial tumour diagnosis // Dermatologica. 1986. С. 79-84.

14. The Rebirth of CAD: How Is Modern AI Different from the CAD We Know? / RSNA. URL: <https://pubs.rsna.org/doi/10.1148/ryai.2019180089> (дата звернення: 11.10.2021).

15. A guide to Two-stage Object Detection: R-CNN, FPN, Mask R-CNN / Medium. URL: <https://medium.com/codex/a-guide-to-two-stage-object-detection-r-cnn-fpn-mask-r-cnn-and-more-54c2e168438c> (дата звернення: 10.11.2021).

16. Теорія прийняття рішень: підручник / А. В. Катренко, В. В. Пасічник, В. П. Пасько. – К.: BHV, 2009. – 447 с.

17. Introduction to Convolutional Neural Networks (CNN) / Analytics Vidhya. URL: <https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/> (дата звернення: 12.01.2022).

18. Skin Diseases Atlas / Dermnet. URL: <http://www.dermnet.com> (дата звернення: 10.01.2022).

19. Smelyakov, K., Honchar, Y., Bohomolov, O., Chupryna, A. Machine Learning Models Efficiency Analysis for Image Classification Problem, CEUR Workshop Proceedings, 2022, 3171, pp. 942–959

20. Smelyakov, K., Bohomolov, O., Kizitskyi, M., Chupryna, A. Identification of Modern Facial Emotion Recognition Models CEUR Workshop Proceedings, 2022, 3171, pp. 1267–1281.

21. What is Azure Active Directory? / Microsoft. URL: <https://docs.microsoft.com/en-us/azure/active-directory/fundamentals/active-directory-what-is> (дата звернення: 20.02.2022).

22. Hans-Jürgen Schönig Mastering PostgreSQL 13 / Packt, 2020. – 476 c.