

Вінницький національний технічний університет
(повне найменування вищого навчального закладу)

Факультет інтелектуальних інформаційних технологій та
автоматизації
(повне найменування інституту, назва факультету (відділення))

Кафедра комп'ютерних наук
(повна назва кафедри (предметної, циклової комісії))

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

**«Інформаційна технологія розпізнавання
дерматологічних захворювань на зображенні»**

Виконав: студент 2-го курсу, групи
1КН-21м спеціальності 122

«Комп'ютерні науки»

(шифр і назва напрямку підготовки, спеціальності)

Дідківський А.А.

(прізвище та ініціали)

Керівник: к.т.н., доцент каф. КН

Сілагін О.В.

(прізвище та ініціали)

« 15 » 12 2022 р.

Опонент: к.т.н., доцент каф. АІТ

Іванов Ю. Ю.

(прізвище та ініціали)

« 15 » 12 2022 р.

Допущено до захисту

Завідувач кафедри КН

д.т.н., проф. Яровий А.А.


(прізвище та ініціали)

« 16 » 12 2022 р.

Вінниця ВНТУ - 2022 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та
автоматизації Кафедра комп'ютерних наук
Рівень вищої освіти II-й (магістерський)
Галузь знань – 12 «Інформаційні технології»
Спеціальність – 122 «Комп'ютерні науки»
Освітньо-професійна програма – «Системи штучного інтелекту»

ЗАТВЕРДЖУЮ

 Завідувач кафедри КН
д.т.н., проф. Яровий А.А.

(підпис)

“14” вересня 2022 року

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

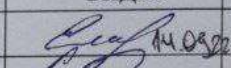



Дідківський Андрій Анатолійович

(прізвище, ім'я, по батькові)

1. Тема роботи Інформаційна технологія розпізнавання дерматологічних захворювань на зображенні
керівник роботи к.т.н., доцент кафедри КН Сілагін О.В., затверджений наказом вищого навчального закладу від “14” 09 2022 року № 203
2. Строк подання студентом роботи 18 листопада 2022 року
3. Вихідні дані до роботи: мова об'єктно-орієнтованого програмування, браузерне середовище роботи додатку, кількість критеріїв класифікації – 7 од., відповідність стандарту ODBS, мінімальна кількість пікселів у зображенні – 1940 од., мінімальна кількість форм на зображенні – 1 од., мінімальна кількість користувачів – 100 чол.
4. Зміст текстової частини: вступ, аналіз сучасного стану розвитку систем розпізнавання об'єктів, обґрунтування методу розв'язання задачі, обґрунтування вибору інструментів технічної реалізації, проектування, програмна реалізація інтелектуальної системи, аналіз результатів тестування, розробка інструкції користувача, економічна частина, висновки, перелік використаних джерел, додатки.

5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень): загальна структурна схема інформаційної технології, схема алгоритму роботи модуля розпізнавання об'єктів, схема процесу інтеграції з системою користувача, діаграма послідовності, діаграма розгортання, загальний вигляд інтерфейсу користувача.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціалита посада консультанта	Підпис, дата	
		завдання видав	виконав прийняв
1-3	Сілагін О.В., к.т.н., доц. каф. КН		
4	Нікіфорова Л. О., к. е. н., доц. каф. ЕПВМ		

7. Дата видачі завдання 14 вересня 2022 року

КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ МКР

№ етапу	Назва етапу	Термін виконання		Очікувані результати
		початок	кінець	
1	Аналіз сучасного стану розвитку систем розпізнавання дерматологічних захворювань на зображенні	14.09.2022	01.10.22	Аналітичний огляд літературних джерел, задачі дослідження розділ 1
2	Обґрунтування методу розв'язання задачі, обґрунтування вибору інструментів технічної реалізації	02.10.2022	16.10.22	Моделі, розділ 2
3	Проектування інформаційної технології. Програмна реалізація інформаційної технології. Аналіз результатів тестування	17.10.22	7.11.22	розділ 2, 3
4	Підготовка економічної частини	08.11.2022	11.11.22	розділ 4
5	Апробація та/або впровадження результатів дослідження	23.11.2022	01.12.22	тези доповідей/звіт про впровадження
6	Оформлення пояснювальної записки, графічного матеріалу та презентації	02.12.22	14.12.22	Пояснювальна записка, графічний матеріал, презентація

Студент

Керівник роботи

(підпис)

Дідківський А.А.

(підпис)

Сілагін О.В.

АНОТАЦІЯ

УДК 004.7

Дідківський А.А. Інформаційна технологія розпізнавання дерматологічних захворювань на зображенні. Магістерська кваліфікаційна робота зі спеціальності 122 – комп'ютерні науки, освітня програма - комп'ютерні науки. Вінниця: ВНТУ, 2022. 105 с.

На укр. мові. Бібліогр.: 21 назв; рис.: 18; табл. 7.

Дана магістерська кваліфікаційна робота присвячена розробці інформаційної технології для розпізнавання дерматологічних захворювань на зображенні. Були розглянуті та проаналізовані існуючі програмні рішення і їх функціональні можливості, та обрано режим функціонування. Було проаналізовано різні підходи до вирішення задачі розпізнавання об'єктів з покращенням за допомогою попередньої обробки зображення. Було спроектовано програму розпізнавання дерматологічних захворювань, написану мовою програмування JavaScript з використанням бібліотеки React для клієнтської частини, Node.js і Express для серверної частини та СУБД MongoDB для управління базами даних.

Графічна частина складається з 7 плакатів.

У економічному розділі розраховано суму витрат на розробку та виготовлення нового технічного рішення, яка складає 603 609 гривень, спрогнозовано орієнтовану величину витрат по кожній з статей витрат, розраховано чистий прибуток, термін окупності витрат для виробника 1,35 роки та економічний ефект для споживача при використанні даної розробки.

Ключові слова: розпізнавання об'єктів, попередній аналіз зображення, веб-клієнт, nodejs, клієнт-серверна архітектура

ABSTRACT

Didkivskyi A.A. Information technology dermatological deceases. Master's thesis in the specialty 122 - computer sciences, educational program - computer science. Vinnytsia: VNTU, 2022. 105 p.

In Ukrainian language. Bibliographer: 21 titles; fig .: 18; table 7.

This master's qualification work is devoted to development of informational technology for recognition of dermatological diseases on the image. The existing software solutions and their functional capabilities were reviewed and analyzed, and the hybrid mode of operation was selected. Various approaches to solving the problem of detection of objects with improvement by means of image processing were analyzed. A program of recognition of dermatological diseases, written in JavaScript programming language with use of the react library for client part, Node.js and Express for server part and the database management database of MongoDB was designed.

The graphic part consists of 7 posters.

The economic section calculates the cost of development and production of a new technical solution, which is 603 609 UAH the estimated cost of each item of expenditure is estimated at uah, the estimated cost of each item is estimated at net profit, the payback period for the manufacturer 1,35 years and the economic effect for consumers using this development.

Keywords: Object recognition, preliminary image analysis, web client, nodejs, client-server architecture.

ЗМІСТ

ВСТУП	6
1 АНАЛІЗ СУЧАСНОГО СТАНУ РОЗВИТКУ СИСТЕМ РОЗПІЗНАВАННЯ ДЕРМАТОЛОГІЧНИХ ЗАХВОРЮВАНЬ НА ЗОБРАЖЕННІ	10
1.1 Огляд проблем в області розпізнавання дерматологічних захворювань на зображенні.....	10
1.2 Аналіз сучасного стану розвитку методів та технологій розпізнавання дерматологічних захворювань	11
1.3 Формулювання задачі і визначення вимог до інформаційної технології розпізнавання дерматологічних захворювань на зображенні	15
1.4 Висновок до розділу 1	16
2 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ РОЗПІЗНАВАННЯ ДЕРМАТОЛОГІЧНИХ ЗАХВОРЮВАНЬ НА ЗОБРАЖЕННІ	17
2.1 Вибір методів організації інформаційних систем розпізнавання дерматологічних захворювань	18
2.2 Моделювання та впровадження методу попередньої обробки зображення	22
2.3 Обґрунтування вибору архітектури інформаційної технології розпізнавання дерматологічних захворювань на зображенні	23
2.4 Моделювання інформаційної технології розпізнавання дерматологічних захворювань на зображенні за допомогою мови UML	24
2.5 Розробка основного алгоритму роботи інформаційної технології розпізнавання дерматологічних захворювань на зображенні	25
2.6 Висновок до розділу 2	26

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ РОЗПІЗНАВАННЯ ДЕРМАТОЛОГІЧНИХ ЗАХВОРЮВАНЬ НА ЗОБРАЖЕННІ	31
3.1 Вибір мови програмування та системи управління базами даних для реалізації інформаційної технології розпізнавання дерматологічних захворювань на зображенні.....	31
3.2 Вибір бібліотек та фреймворків для програмної реалізації інформаційної технології розпізнавання дерматологічних захворювань на зображенні.....	37
3.3 Програмна реалізація компонентів інформаційної технології розпізнавання дерматологічних захворювань на зображенні	40
3.4 Тестування роботи компонентів інформаційної технології розпізнавання дерматологічних захворювань на зображенні.....	44
3.5 Висновок до розділу 3	49
ЕКОНОМІЧНА ЧАСТИНА	50
4.1 Комерційний та технологічний аудит науково-технічної розробки.....	50
4.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи.....	52
4.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором	58
4.4 Висновок до розділу 4	66
ВИСНОВКИ.....	67
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	69
Додаток А Протокол перевірки МКР на наявність текстових запозичень ...	72
Додаток Б Лістинг програми.....	73
Додаток В ІЛЮСТРАТИВНА ЧАСТИНА	74

ВСТУП

Актуальність досліджень. Зі швидким і стабільним розвитком ринку онлайн-послуг компанії часто шукають якісні методи дистанційної діагностики на стороні клієнта, і деякі компанії створюють велику кількість продуктів на основі власних можливостей, власні продукти для покращення внутрішньої роботи. Замість того, щоб використовувати ручні методи розширення рейтингу користувачів за рахунок збільшення кількості співробітників, провідні компанії можуть швидко збирати великі масиви даних, відформатовані за віком, часом і т.д. для подальшої експертної оцінки та зміни ринку. Також існує велика перспектива використання такого програмного забезпечення для автоматичної оцінки проблемних зон для користувачів, які не можуть відвідати медичний заклад.

Впровадження компактного інструменту, здатного з високою точністю автоматично зчитувати зображення, надіслані користувачами, дозволило визначити за допомогою програми ступінь ураження дерматологічних факторів. Довгостроковий аналіз цих факторів дозволяє фахівцям коригувати системи та навантаження на персонал за допомогою найточніших доступних даних.

Зворотною стороною цього нового програмного забезпечення є те, що його неможливо швидко налаштувати на стороні клієнта, за винятком шаблонів дизайну. Крім того, фотографії користувача належать користувачеві на ім'я, і використання даних потребує сповіщення та підтвердження згоди користувача. Таким чином, останні стикаються з незручностями, пов'язаними з питаннями персональних даних. Оскільки завдання ідентифікації об'єктів на шкірі, тобто захворювання, має досить прикладний характер, точність існуючих систем є питанням ефективного використання таких систем, а за допомогою класичних методів веб-

розробки застосування засобів штучного інтелекту є релевантним підходом Інтелектуальний для досягнення кращих показників надійності для ідентифікації захворювань шкіри.

Зв'язок роботи з науковими програмами, планами, темами.

Дана робота виконана згідно з напрямками та планами наукових досліджень факультету комп'ютерних наук Вінницького національного технічного університету 22 К1 «Моделі, методи, методи та пристрої інтелектуальних інформаційних систем управління, економіки, освіти та зв'язку». наукової та навчально-методичної роботи кафедри.

Мета та завдання дослідження.

Метою кваліфікаційної роботи магістра є підвищення точності розпізнавання шкірних захворювань на зображеннях. На відміну від аналогів, що не передбачають механізм децентралізованого, ізольованого доступу до даних, вирішення цієї проблеми пропонується за допомогою ізоляції через консолі управління.

Для досягнення поставленої мети необхідно розв'язати наступні **задачі**:

- провести аналіз сучасного стану розвитку систем розпізнавання дерматологічних захворювань на зображенні;
- обґрунтувати метод розв'язання задачі;
- виконати проектування інтелектуальної системи розпізнавання дерматологічних захворювань на зображенні;
- Виконати програмну реалізацію інтелектуальної системи розпізнавання дерматологічних захворювань на зображенні;
- провести тестування інтелектуальної системи розпізнавання дерматологічних захворювань на зображенні та аналіз результатів тестування;
- здійснити розробку інструкції користувача;

– провести розрахунок економічної доцільності дослідження;

Об'єктом дослідження є процес розпізнавання дерматологічних захворювань на зображенні.

Предметом дослідження є алгоритми та програмне забезпечення, що організують процес розпізнавання шкірних захворювань на зображеннях, їх пріоритизацію та доступ до даних, що зберігаються.

Методи дослідження - методи розпізнавання шкірних захворювань, теорія глибоких нейронних мереж, методи масштабування та інтеграції програмного забезпечення, методи та підходи до розробки систем штучного інтелекту, методи та підходи до розробки веб-орієнтованих програмних додатків, методи об'єктно-орієнтованого програмування.

Наукова новизна одержаних результатів - розроблено нову методику розпізнавання об'єктів на зображеннях. Відрізняється від існуючих аналогів використанням методів передобробки зображень та покращеною структурою згорткових нейронних мереж..

Практичне значення одержаних результатів полягає у наступному:

1. Розроблено алгоритм системи та здійснено програмну реалізацію системи згідно з розробленою структурою.

2. Структура програмного забезпечення вибирається на основі доступу до API.

Достовірність теоретичних положень у магістерській кваліфікаційній роботі залежить від суворості постановки завдань, правильного застосування математичних методів при доказі наукових положень, суворості виведення аналітичних пропорцій та приведення їх у відповідність до відомих, що підтверджується порівнянням результатів. . .

Особистий внесок магістрантів. Усі результати, наведені в магістерській роботі, було отримано самотужки.

Апробація. Основні висновки дослідження було апробовано на III Міжнародній конференції «Наукова практика: молодь у науці: дослідження, проблеми, перспективи». (12.05.2022 – 13.05.2023 року, м. Вінниця, Україна) [1].

1 АНАЛІЗ СУЧАСНОГО СТАНУ РОЗВИТКУ СИСТЕМ РОЗПІЗНАВАННЯ ДЕРМАТОЛОГІЧНИХ ЗАВОРЮВАНЬ НА ЗОБРАЖЕННІ

1.1 Огляд проблем в області розпізнавання дерматологічних захворювань на зображенні

Варто відмітити, що останнім часом розпізнавання образів користується все більшою популярністю, адже розпізнавання зображення, тексту чи мови, а також різноманітних явищ сприяє спрощенню комунікативного зв'язку людини з комп'ютером, допомагає застосовувати різні системи штучного інтелекту, в тому числі в системах попередньої обробки. Здатність сприймати зовнішній світ у вигляді образів полегшує вивчення властивостей великої кількості об'єктів завдяки знайомству з обмеженою кількістю об'єктів. Крім того, системи розпізнавання, що використовують штучний інтелект, на сьогоднішній день активно розвиваються та з успіхом починають застосовуватись на масштабному рівні. Покращується як наукова база штучних нейронних мереж, так і обчислювальні потужності технічного обладнання.

За багато років розвитку технологій розпізнавання предметів на знімках виникла потреба в ширшій аналітиці отриманих даних. Внаслідок довготривалого дослідження технології розпізнавання предметів алгоритми добре навчилися знаходити об'єкти на зображенні та класифікувати їх. Проте одним із суттєвих недоліків таких програмних модулів є нездатність оцінювати об'єкти не тільки в межах плоскості, а ще й визначати їх глибину.

Через високу зацікавленість медичної сфери в роботизації та автоматизації рутинних процесів лікарів, приділяється все більше уваги до

управління процесом аналізу зображення, будь це знімок шкіри. Проте, на даний момент багато проведених досліджень щодо аналітики об'єктів на фото мають вузьку спрямованість [2].

Відомо, що використання методів розпізнавання глибини об'єктів у системах попередньої обробки підвищує ефективність розробки процесів автоматизації, знижує матеріальні та часові витрати, сприяє оперативному отриманню об'єктивних та оперативних даних.

Хоча вже існує велика кількість систем розпізнавання образів, всі вони є частиною більшої системи, тому для вирішення більш конкретних завдань, наприклад, при відстеженні глибини об'єктів у складі системи взаємодії з користувачем, ці системи не підходять.

Відкритою проблемою є те, як знайти дійсні числа для глибини об'єктів у зоні розпізнавання. Тому пропонується усунути цей недолік

Існуючі системи ідентифікують захворювання шкіри на зображеннях за допомогою інформаційних технологій.

1.2 Аналіз сучасного стану розвитку методів та технологій розпізнавання дерматологічних захворювань

В даний момент програмне розпізнавання шкірних захворювань людини є непростим завданням, тому автоматизованих систем, розроблених для автоматизації такого процесу, не так багато. Можна навести приклади програмних продуктів, які використовують розпізнавання плоских об'єктів у вузьких контекстах.

Досить цікавим прикладом є Google Lens. Програма здатна визначити набір об'єктів на площі та класифікувати їх. Сервіс не є вимагає потреби в навчанні або додатковому налаштуванні [4].

Оскільки алгоритм Google Lens звіряється з тисячами зображень пошукача, визначення об'єкта по фото не займає багато часу. Цей метод

дозволив додати значний об'єм даних, однак у нього є свої недоліки. Наприклад, він не враховує старі предмети, які не потрапляють в онлайн-магазини, - на кшталт ретроконсолей або касетних плеєрів. Тому і Google Lens не зможе їх впізнати. Разом з тим фахівці компанії відзначають, що алгоритм досі не завжди працює коректно. Вони пояснюють це сукупністю факторів, серед яких - невідповідність фотографій користувачів картинкам, по яким тренується модель, кут огляду і якість знімків. За їх словами, вони вже працюють над рішенням, пропонуючи штучному інтелекту більше зображень, знятих на камери смартфона. Ще в ранній версії Google Lens могло виводити інформацію про заклад (наприклад, про ресторани) по фотографії його вивіски. Зараз творці пішли далі і навчили додаток читати. Тепер воно здатне скопіювати виділений на фотографії текст. При цьому воно враховує різні шрифти та мову, завдяки зразками з бази даних Google Books, за якими навчалася. Для того щоб розрізняти схожі символи на зразок нуля і букви «о», алгоритм перевіряє орфографію в інтернеті.

Переваги цієї реалізації:

- Середній відсоток помилок (близько 60%). При русі об'єкта, відсоток точності знижується до 40-50%.
- Програма підтримує широкий спектр форматів відео та Кодек MPEG1, MPEG2, DivX4, DivX5, DivX6, DV-AVI [4].

Недоліки цієї реалізації:

- Швидкість розпізнавання об'єктів різко знижується під час роботи з .
Старше 50 років.
- Якість розпізнавання значно погіршується під час носіння користувачем окулярів.



Рисунок 1.1 – Загальний вигляд інтерфейсу сервісу Google Lens

Наступним прикладом є Interactive Health [3] – стартап, який продає програмне забезпечення для розпізнавання висипів на шкірі та таким чином допомагає роздрібним лікарям надавати своїм клієнтам більш персоналізований досвід. Стартапи працюють над різними рішеннями, які покращують взаємодію з клієнтами завдяки інтеграції технології розпізнавання людини у існуючі програми закладів.

Недоліком таких систем є те, що системна інтеграція із бізнес-програмою стає дуже складною. Дискретний підхід збільшив би вартість встановлення такого рішення. В результаті процес затягується та завдає шкоди компанії. Ще одним істотним недоліком є те, що програмне забезпечення обмежене лише платформою операційної системи Лінукс. [5]. SKOUT - комп'ютерний пристрій для виявлення поліпів у режимі реального часу, використовує передову технологію комп'ютерного зору, призначену для розпізнавання підозрілих тканин під час скринінгу та спостереження колоректального раку. SKOUT значно покращує загальне виявлення аденоми порівняно зі стандартною колоноскопією. У своєму реєстраційному

дослідженні, опублікованому в *Gastroenterology*, SKOUT продемонстрував відносне збільшення виявлення аденом на 27% на колоноскопію. ¹

Нарешті, документація штучного інтелекту, яка зараз знаходиться в розробці, може незабаром мати повноваження надавати детальну та точну документацію щодо процедури. Це дозволить гастроентерологам переглядати доказову документацію, а не створювати її з нуля. Цей продукт буде розроблено та продано спільно з *Provation*, провідним постачальником програмного забезпечення та SaaS рішень для клінічної продуктивності та автоматизації робочого процесу.

Стартап підкреслює прагнення розвивати прецизійну медицину та продовжувати підвищувати планку як точності, так і послідовності у прийнятті рішень лікарями. Iterative Health стандартизує високоякісну допомогу за допомогою підходів, розширених штучним інтелектом, зберігаючи здоров'я пацієнтів у центрі своєї місії [5].



Рисунок 1.2 – Загальний вигляд роботи програми SCOUT

Тому програмний продукт, що розглядається, має обмеження користувача, пов'язані із зазначеним завданням або платформою.

1.3 Формулювання задачі і визначення вимог до інформаційної технології розпізнавання дерматологічних захворювань на зображенні

Визначимо задачі і вимоги до інформаційної технології розпізнавання дерматологічних захворювань на зображенні.

Задача: розширення функціоналу розпізнавання дерматологічних захворювань людини шляхом створення адміністративної консолі та систематизації отриманих даних для подальшого аналізу та візуалізації.

Програмний модуль має реалізовувати такі функції:

- розпізнавання поверхні;
- отримання та візуалізація даних про фрагмент зображення;
- систематизація параметрів поверхні;
- періодичне збереження параметрів користувачів;
- візуалізація карти об'єктів на площині;

Також необхідно провести тестування розробленої інформаційної технології на відсутність помилок. Розроблена технологія має працювати хоча б в одному середовищі розробки для коректної демонстрації своєї роботи та тестування.

Початком розвитку інформаційних технологій розпізнавання шкірних захворювань на зображеннях є вибір алгоритмів розпізнавання об'єктів на основі ключових точок зображення. Також необхідно розробити загальну структурну схему модуля, розробити загальну схему алгоритму роботи модуля та створити UML-діаграму послідовності для деталізації компонентів для подальшої розробки. Основними етапами впровадження є вибір технології впровадження та розробка інформаційної технології за

заданими функціями. Завершальним етапом є тестування розробленої програми та розробка інструкції для користувача.

1.4 Висновок до розділу 1

У цьому розділі було розглянуто проблему розпізнавання шкірних захворювань на зображеннях та виявлено основні труднощі у цьому процесі. Проаналізовано існуючі рішення проблеми розпізнавання об'єктів на зображеннях, показано недоліки цих рішень та запропоновано компроміси з використанням штучного інтелекту для аналізу зображень та прийняття рішень експертами-людьми. Тепер, коли завдання інформаційних технологій та вимоги до розпізнавання шкірних захворювань на зображеннях визначені, можна переходити до етапу вибору способу розпізнавання об'єктів на шкірі для реалізації інтелектуальної частини системи.

2 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ РОЗПІЗНАВАННЯ ДЕРМАТОЛОГІЧНИХ ЗАХВОРЮВАНЬ НА ЗОБРАЖЕННІ

2.1 Вибір методів організації інформаційних систем розпізнавання дерматологічних захворювань

На сьогоднішній день розроблено безліч алгоритмів, здатних автоматично розпізнавати захворювання шкіри людини в режимі реального часу. Однак продуктивність систем розпізнавання дерматологічних захворювань страждає від низки проблем, у тому числі:

- Невеликий обсяг високоякісних даних для навчання.
- Етнічна приналежність користувача, вік та стать.
- Неповна вхідна інформація
- Оклюзія — часткове або повне блокування видимості об'єкта, що обробляється іншим об'єктом, або зміна кута повороту об'єкта.
- Недостатнє або надмірне освітлення об'єкту, що обробляється.
- Зміна відсотка оброблених об'єктів.

До існуючих методів розпізнавання шкірних захворювань людини входять такі характерні стадії [6]:

попередня обробка потоку інформації, що складає:

- ключові точки областей шкіри користувача.
- масштабування площі, що купується;
- розміщення знайдених об'єктів обробки;
- отримання підготовлених візуальних даних;
- класифікація знаків за заданими критеріями;

2.2 Моделювання та впровадження методу попередньої обробки зображення

Попередня обробка дозволяє програмам коригувати вхідні дані та мінімізувати ймовірність помилок при читанні оброблених об'єктів. На цьому етапі програмне забезпечення локалізує область оброблюваного об'єкта, виділяє геометричні особливості користувача, обрізає знайдену область і змінює яскравість вхідного потоку [11].

Попередня обробка дозволяє звести до мінімуму відсутні дані, схожі вхідні дані, невеликі зміни у зовнішньому вигляді сайту, обробляти зміни пози об'єктів, значні зміни в освітленні тощо, а також підвищити точність методів розпізнавання об'єктів [11].

Класифікація цих ознак і пам'ять програмного забезпечення таблиць спрямовані на підвищення відсотка відповідності вихідних даних алгоритму класифікації очікуваним результатам під час тестування. Класифікація — це процес осмисленого й систематичного групування та організації інформації в стандартному форматі, який використовується для виявлення подібності в ідеях, подіях, об'єктах, людях, явищах [11].

Основною метою цього огляду є порівняння методів попередньої обробки вхідних об'єктів, максимальний поділ візуальних ознак для алгоритмічної класифікації об'єктів. Це дозволяє визначити найкращий напрямок досліджень для створення альтернативної системи розпізнавання об'єктів шкіри людини. Локалізація області обличчя дозволяє визначити розмір та положення шкіри на зображенні. Найчастіше використовуються такі методи локалізації: метод Віоли-Джонса [11] та `single shot multibox detector`. Для більш детального аналізу методів локалізації особи слід зупинитися на методі Віоли Джонса, який є найбільш популярним методом локалізації. [11].

У вихідній версії алгоритму Віоли-Джонса використовувалися лише примітиви без повороту, а сума інтенсивностей пікселів в одній підобласті вираховувалася із суми інтенсивностей іншої підобласті для обчислення значення ознаки. був. При розробці методу було запропоновано примітиви з нахилом 45 градусів та асиметричною конфігурацією. Також було запропоновано приписувати кожному субрегіону певну політичну цінність замість розрахунку звичайної різниці, а обчислювати цінність ознаки як виважену суму пікселів у різних типах регіонів. [11].

Цей метод ґрунтується на примітивах Хаара. Основна причина в тому, що ми спробували уникнути піксельного бруду, зберігши при цьому швидкість обчислення ознак. Перший етап системи розпізнавання осіб із цілком змістовною інтерпретацією будується з двох ознак Хаара, але отримати значну інформацію для класифікації з значень пар пікселів важко.

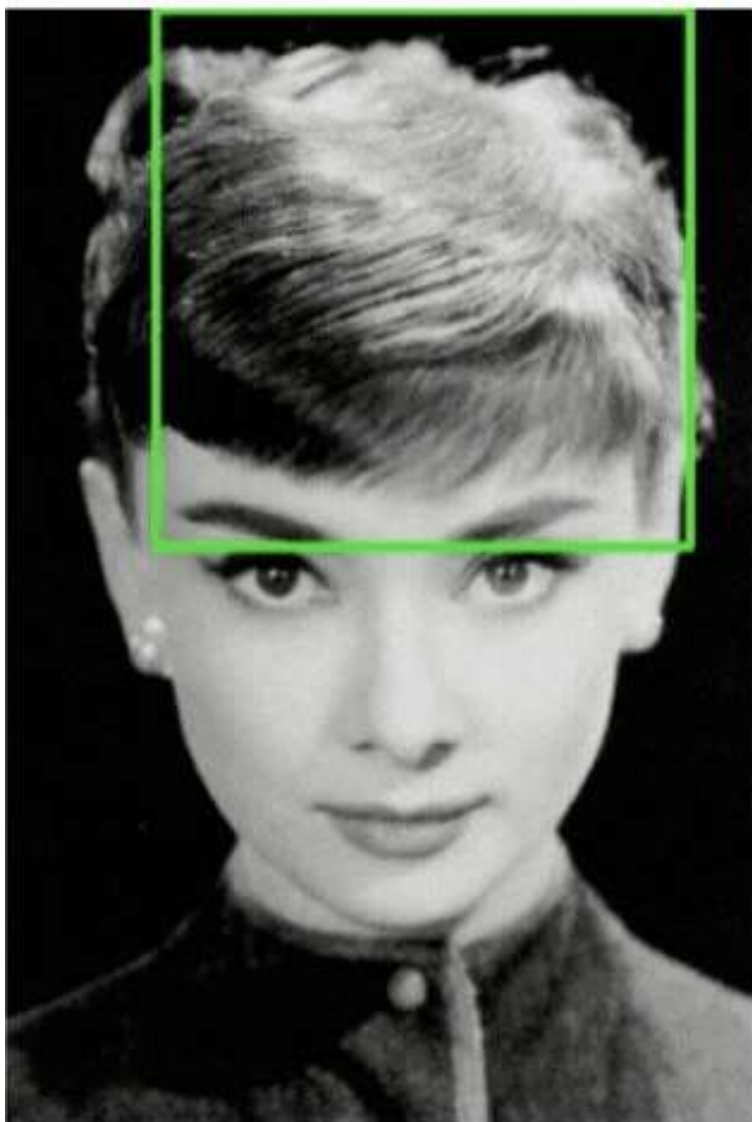


Рисунок 2.1.1 – Візуалізація використання методу каскада Хаара в розпізнаванні об'єктів – крок 1

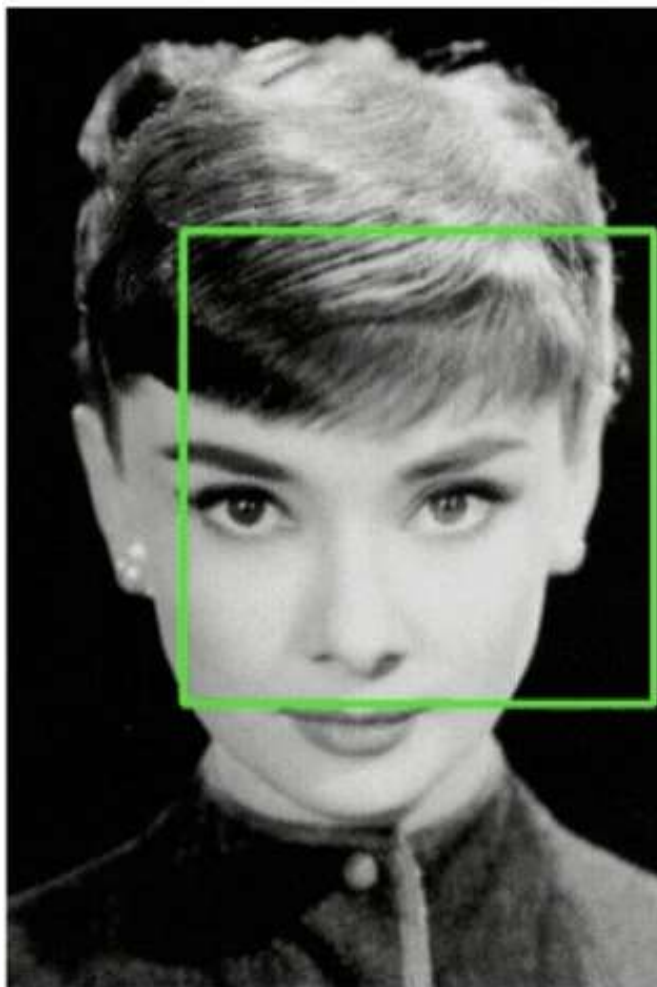


Рисунок 2.1.2 – Візуалізація використання методу каскада Хаара в розпізнаванні об'єктів – крок 2

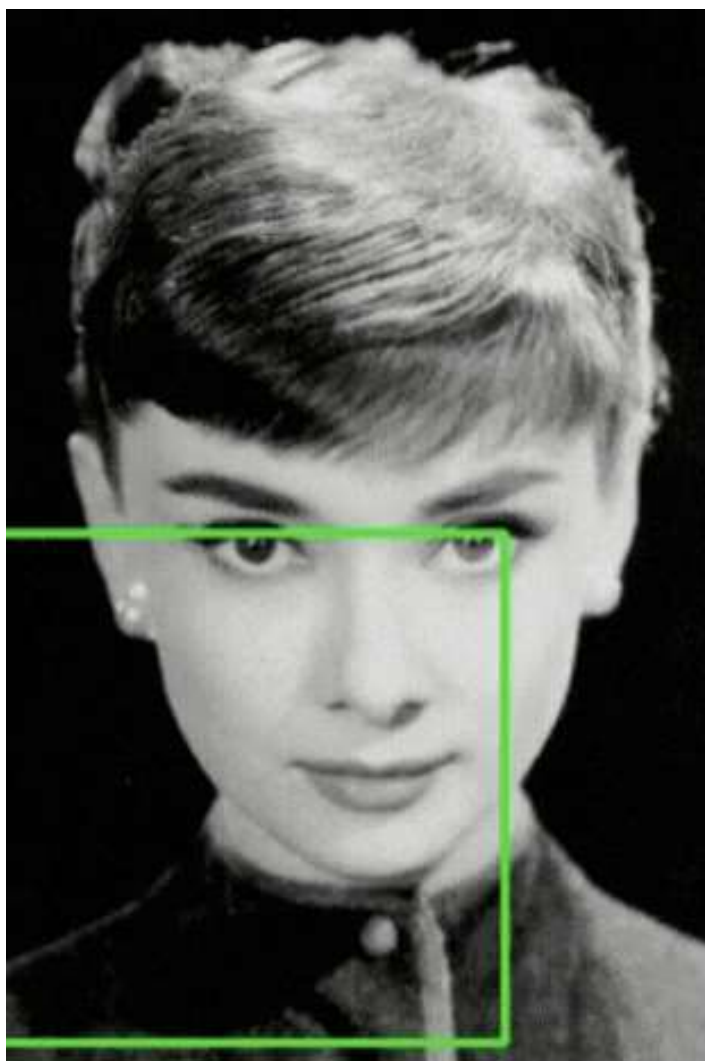


Рисунок 2.1.3 – Візуалізація використання методу каскада Хаара в розпізнаванні об'єктів – крок 3

1. Обрізання та масштабування виявленої області об'єкта провадиться за координатами, отриманими методом локалізації області шкіри. Через різні розміри областей виявлення шкіри зображення необхідно масштабувати, тобто всі зображення мають однаковий коефіцієнт масштабування. [10] можна застосувати до таких завдань - Вибір Безсея (корекція Бесселя) та Гаусовий розподіл
2. Вирівнювання може зменшити відмінності усередині класу. Так, наприклад, для кожного об'єкта на шкірі вибирається еталонне зображення і розбивається на колірні складові або найбільш інформативні області шкіри,

а інше зображення є еталонним зображенням, метод це завдання використовується для [11, 12]:

- широке перетворення предметів (Scale-invariant feature transform, SIFT);
 - області для інтересу (ROI).
3. Регулювання яскравості покращує читаність зображення та зменшує видимий шум, тим самим вирішуючи проблеми з освітленням та багато іншого.

2.3 Вибір архітектури згорткової нейронної мережі

Поява великих обсягів даних та великомасштабних обчислювальних можливостей призвела до активного використання нейронних мереж. Особливої популярності набули згорткові нейронні мережі - архітектура, запропонована Яном Лекуном [13] і спрямована на ефективне розпізнавання зображень. Мережева архітектура отримала свою назву завдяки наявності операцій згортання. Суть його в тому, що кожен фрагмент зображення поелементно множиться на матрицю згортки (ядер), результати підсумовуються та записуються в аналогічному вигляді. Положення вихідного зображення. У мережеву архітектуру закладено апріорні знання з предметної галузі комп'ютерного зору. Пікселі зображення більш тісно пов'язані зі своїми сусідами (локальна кореляція), і об'єкти всередині зображення можна знайти у будь-якій частині зображення. Згорткова мережа - це нейронна мережа, в якій принаймні один шар використовує згортку замість звичайної операції множення матриць. Загалом згортка — це операція над двома функціями допустимих аргументів. Функцію згортки можна подати у вигляді [13]:

$$S(t) = (xw)(t)$$

(2.2)

У термінології згорткових мереж перший аргумент (функція x у цьому прикладі) називається входом, а другий аргумент (функція w) називається ядром. Вихід $S(t)$ називається картою ознак. Для розпізнавання шкірних захворювань на вхід подається зображення, подане у вигляді двомірної матриці, тому рівняння (2.1) має вигляд:

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(i + m, j + n)K(m, n) \quad (2.3)$$

де I – вхідне зображення, K – ядро, i, j – координати елементів вхідного зображення, m, n – координати елементів ядра.

Успішне застосування згорткових нейронних мереж для класифікації зображень призвело до багатьох спроб використати цей метод для інших місій. Останнім часом його активно використовують із класифікації шкірних захворювань. Згорткові нейронні мережі будуються, коли на додаток до вхідних і вихідних шарів вони зазвичай мають згорткові шари, що чергуються, шари субдискретизації і шари, повністю пов'язані з виходом. Всі три типи шарів можуть використовуватися взаємозамінно у будь-якому порядку. [13].

У згортковому шарі нейрони, що використовують однакові ваги, поєднуються в карти ознак, і кожен нейрон у карті ознак з'єднується з деякими нейронами попереднього шару. Обчислюючи мережу, бачимо, кожен нейрон виконує згортку певної області попереднього шару (це визначається набором нейронів, що з цим нейроном).

2.4 Моделювання інформаційної технології розпізнавання дерматологічних захворювань на зображенні за допомогою мови UML

Модель - це абстракція, створена для розуміння чогось перед його створенням. [13].

Абстракція - це вибіркоче вивчення будь-якого аспекту проблеми. Основна мета абстракції полягає в тому, щоб ізолювати аспекти, важливі для мети, та відкинути решту. [13].

UML – це уніфікована мова моделювання. Це невід'ємна частина вбудованого процесу розробки програмного забезпечення. UML — це мова загального призначення та відкритий стандарт, який використовує графічну нотацію для створення абстрактних моделей систем, які називаються моделями UML. [13, 14].

Ви можете використовувати різні діаграми UML для опису основного циклу вашої програми. Зокрема, діаграми діяльності та діаграми послідовності. Діаграми активності аналогічні звичайним схемам програмних алгоритмів. Тобто як потік управління передається від однієї діяльності до іншої.

> Діаграми послідовності також показують життєвий цикл конкретних сутностей, але в той же час показують їх на єдиній часовій шкалі, взаємодія декількох сутностей (акторів) і відносну тривалість процесу. Таким чином, діаграми послідовності дають повніше уявлення про життєвий цикл програмного забезпечення.

Об'єкти на діаграмах послідовності можуть містити компоненти програмного модуля, описані в попередньому розділі. Ми також додамо два об'єкти, User та Admin, щоб показати їх взаємодію. Отже, ми отримуємо сім об'єктів: Веб-клієнт, Програмний інтерфейс, Панель адміністратора, Універсальний кодувальник пропозицій, База даних, Користувач та

Адміністратор. Розроблена діаграма послідовностей зображена на рисунку 2.5.

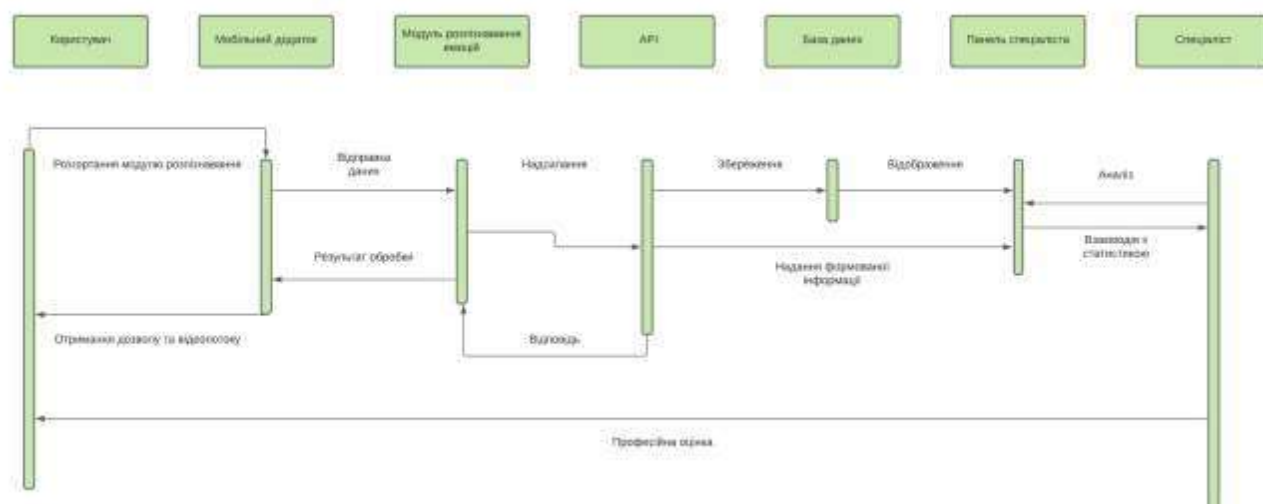


Рисунок 2.4 – Діаграма послідовності програмного модуля розпізнавання дерматологічних захворювань на базі мобільного пристрою

Створити план розробки для проектування конфігурації компонентів програмного модуля розпізнавання шкірних захворювань людини з урахуванням мобільних пристроїв.

Діаграма розгортання — це тип діаграми UML, яка показує виконавчу архітектуру системи, включаючи такі вузли, як апаратні або програмні середовища виконання та проміжне програмне забезпечення, яке їх з'єднує. Схема розгортання програмного модуля текстової аналітики містить такі вузли:

- користувач;
- веб сторінка;
- хостинг бази даних;
- база даних;

- СУБД;
- програмний інтерфейс;
- інформаційна технологія розпізнавання дерматологічних захворювань;

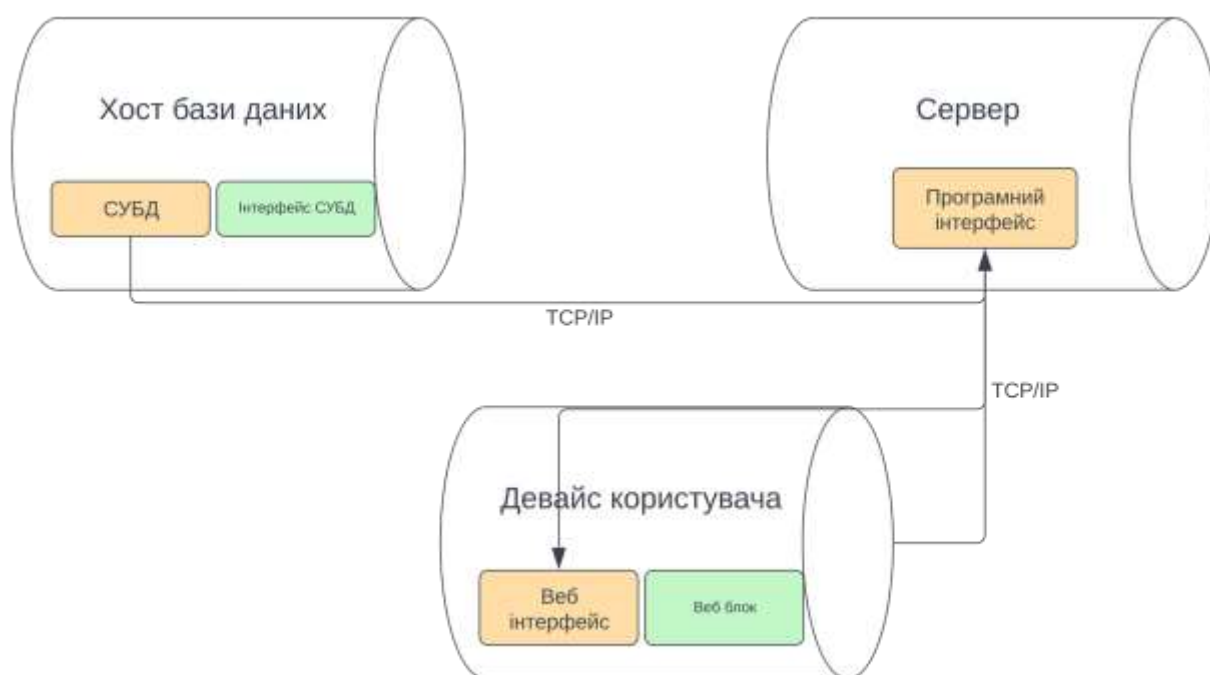


Рисунок 2.5 – діаграма розгортання інформаційної технології розпізнавання дерматологічних захворювань

Усі компоненти пов'язані між собою мережею, зв'язок заснований на протоколі TCP/IP. Покомпонентний вигляд показано на рис. 2.5.

2.5 Розробка основного алгоритму роботи інформаційної технології розпізнавання дерматологічних захворювань на зображенні

Для відображення програмного циклу використовуємо мовний та графічний опис основного алгоритму програмного модуля розпізнавання шкірних захворювань людини на базі мобільного модуля.

Кроки алгоритму:

1. Ідентифікація користувача.
 2. Перевірте статус використання і, якщо користувач є експертом, перейдіть до кроку 3, інакше перейдіть до кроку 8.
 3. Отримайте список клієнтів.
 4. Підбір відповідних клієнтів.
 5. Отримати докладні дані клієнта.
 6. Візуалізація клієнтської статистики.
 7. Оцінка статистичних даних.
 8. Налаштування модуля розпізнавання.
 9. Отримання та обробка зорової інформації у процесі інтервального розпізнавання.
 10. Збереження даних про стан шкіри та синхронізація із сервером.
 11. Оновлення даних у базі даних.
 12. Поновіть дані в експертній панелі.
 13. Виконана робота
- Графічно алгоритм описаний на рисунку 2.4.

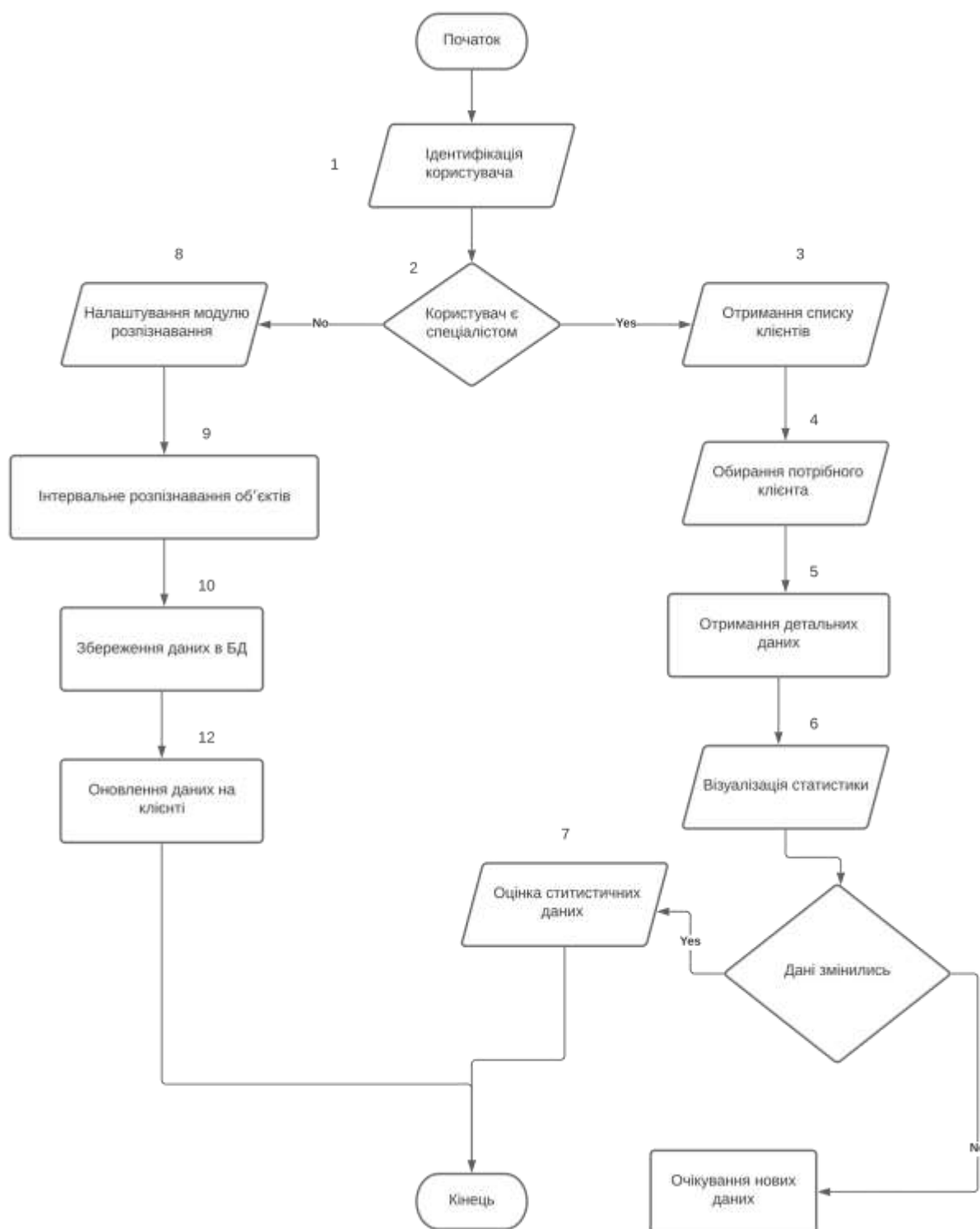


Рисунок 2.6 – Схема основного алгоритму роботи програмного модулю розпізнавання об'єктів

2.6 Висновок до розділу 2

В даному розділі вибрано основні компоненти інформаційної системи розпізнавання шкірних захворювань, складено загальну структурну схему модулів, основні алгоритми роботи модулів та UML-діаграми їх розгортання та послідовності. складено Я був. Розроблені схеми та діаграми докладно описують структуру та принцип роботи програмного забезпечення. Це дозволить перейти до реалізації програмного модуля для розпізнавання шкірних захворювань людини..

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ РОЗПІЗНАВАННЯ ДЕРМАТОЛОГІЧНИХ ЗАХВОРЮВАНЬ НА ЗОБРАЖЕННІ

3.1 Вибір мови програмування та системи управління базами даних

Визначає інструменти, що використовуються для створення реалізацій програмних модулів на основі нейронних згорткових мереж. Зокрема, нам необхідно визначити мову програмування для створення мобільних клієнтів, панелей спеціалістів, мобільних модулів розпізнавання та систем керування базами даних (СУБД) для зберігання даних.

Для аналізу та порівняння було обрано три мови програмування:

C#, JavaScript та Python.

Крім того, варто зазначити, що інтерфейс програми створений з використанням фреймворку розробки React.

C# - це мова програмування, що належить до сімейства мов із C-подібним синтаксисом. Ця мова включає елементи C++ і Java і була створена для розробки додатків на платформі Microsoft .NET Framework. Мова має строгу типізацію, але він також має динамічну типізацію (тип змінної визначається, коли ви надаєте їй значення, тобто тип змінної може змінитися під час виконання програми). Використовується для створення клієнтів та веб-застосунків. Крім того, Visual Studio має вбудований збирач сміття, який відстежує саму пам'ять. C# використовується, тому що він є універсальним.

Python — ще одна швидкозростаюча мова програмування. Мова має простий та зрозумілий синтаксис. Це значно полегшує процес розробки. Він має динамічну типізацію, автоматичний контроль пам'яті та механізм

обробки винятків. Використання бібліотек та низький поріг зробили Python однією з найпопулярніших мов програмування.

JavaScript - мова програмування, що найбільш динамічно розвивається. Спочатку він був орієнтований лише на веб-розробку, клієнтську частину. А ось модифікатори доступу до полів класу, можливість створення інтерфейсів, принципи поліморфізму та навантаження методів. Отже, зараз JavaScript спрощує розробку серверної частини веб-сайту або програми, а платформа Node.js дозволяє створювати звичні програмні програми. У JavaScript взагалі немає статичної типізації, тобто ви не вказуєте тип змінної при її оголошенні, ви просто надаєте його при присвоєнні або зміні значення, тому несумісності типів легко обробити [12]. Гнучкість цієї мови дозволяє використовувати безліч шаблонів програмування залежно від умов. У JavaScript взагалі немає статичної типізації, типи змінних не вказуються під час оголошення, вони просто присвоюються разом із присвоєнням чи модифікацією значень, що спрощує роботу з типами. Несумісність. Прийняття нових стандартів спростило синтаксис JavaScript та додало класи, що зробило JavaScript повноцінною об'єктно-орієнтованою мовою. Для створення програмного забезпечення була обрана мова програмування JavaScript, оскільки вона необхідна під час роботи на клієнтській стороні браузера. Курсова робота написана з використанням однієї мови програмування через її чистий синтаксис з можливістю динамічної зміни перетворення типів, очищення сміття та оголошення анонімних функцій [13].

У таблиці 3.1 наведена порівняльна характеристика мов програмування.

Таблиця 3.1 – Порівняльна характеристика мов програмування

Стандарт/мова	C#	JavaScript	Python
Ручне керування пам'яттю	+	-	-

ООП	+	+	+
Потреба скомпілювати	+	-	-
Необхідність у роботі з браузером	-	+	-
Багатовимірність масивів	+	+	+
Динамічність масивів	+	+	+
Статичність типізації	+	-	-
Динамічність типізації	+	+	+
Виконання на стороні користувача	-	+	-
Пряме використання скрипту	-	+	-

MySQL – це система управління реляційними базами даних. Тому MySQL реалізує модель реляційної бази даних. Реляційні бази даних називаються базами даних, всі дані, доступні користувачам, організовані як таблиць, проте операції над даними зводяться до операцій над цими таблицями. До переваг реляційних баз даних належать: Єдиною інформаційною структурою, що використовується, є «таблиця». Суворі правила проектування, що базуються на математичному апараті. Повна незалежність даних. Недоліком моделі реляційної бази даних є те, що вона використовує багато зовнішньої пам'яті та може призвести до аномалій без нормалізації. Ще одним недоліком є те, що дані мають бути у потрібному форматі. Для завдань розробки на мові JavaScript формат JSON, що потребує використання додаткового ORM (Object-Relational Mapping) для перетворення даних. [14, 15].

MongoDB це система управління базами даних, орієнтована на документи. Особливістю цієї СУБД є те, що дані зберігаються у вигляді

документів у форматі JSON (JavaScript Object Notation). Це полегшує процес роботи з даними, отриманими в результаті запитів до бази даних. Іншими словами, під час використання MongoDB вам не потрібно писати схему таблиці. Коротше кажучи, MongoDB є системою NoSQL. Основною перевагою використання цієї СУБД є її гнучка структура, яка може бути легко змінена та розширена у процесі роботи з базою даних без необхідності зміни існуючих даних. MongoDB також підтримує JavaScript як мову для написання запитів та агрегатних функцій та дозволяє використовувати регулярні вирази. Головний недолік ситуації з MongoDB полягає в тому, що база даних не має суворої схеми даних. Це може призвести до несподіваних результатів під час маніпулювання отриманими даними [16].

Використання моделі SPA веб-програми для реалізації клієнтської частини інтелектуального модуля. SPA (односторінкова програма) — це тип веб-програми, яка не перезавантажує програму при перемиканні сторінок, забезпечуючи кращий інтерфейс користувача. Розгляньте бібліотеку React та фреймворки Vue.js та Angular, щоб вибрати інструмент для реалізації SPA. [17].

React – це бібліотека, розроблена Facebook. React використовує мову шаблонів JSX, яка спрощує розробку об'єктної моделі документа (DOM). React має віртуальний механізм DOM, який визначає, чи повинен рендеруватися наступний екран до фактичного рендерингу. Це пов'язано з тим, що зміна фактичного дерева DOM є одним із найбільш ресурсомістких завдань під час роботи з веб-додатками [18]. React забезпечує односторонню прив'язку даних, що покращує модульність та швидкість. Односторонній потік даних означає, що коли розробники створюють програми React, часто вкладають дочірні компоненти в батьківські компоненти. Таким чином, розробники можуть знати, коли і де виникають помилки, і краще контролювати всі веб-додатки. React - це не фреймворк, що дозволяє адаптувати процес розробки до ваших індивідуальних потреб, тому в ньому

немає чіткої методології розробки. Недоліки React – відсутність механізму навігації та централізованого репозиторію. Тому для створення повноцінної односторінкової програми необхідно встановити додаткові бібліотеки. Vue.js - це прогресивний фреймворк для створення інтерфейсів користувача. На відміну від монолітних фреймворків, Vue розроблений як інкрементальний. Його ядро в основному вирішує завдання рівня представлення та спрощує інтеграцію з іншими бібліотеками та існуючими проектами. З іншого боку, Vue ідеально підходить для створення складних односторінкових додатків при використанні сучасних інструментів та додаткових бібліотек [19]. Переваги Vue включають невеликий розмір пакету, використання віртуальних дерев DOM та реактивну двосторонню прив'язку даних. Двостороння прив'язка даних – це зв'язок між оновленнями даних моделі та поданням (UI). Пов'язані компоненти містять дані, які можуть іноді оновлюватися. Двостороння прив'язка даних полегшує оновлення зв'язаних компонентів та відстеження оновлень даних. Подібно до об'єктів DOM, Vue реактивно оновлює пов'язані дані, що робить його ідеальним для додатків, яким потрібні оновлення в реальному часі. До недоліків Vue можна віднести відсутність підтримки великих проектів, дещо повільну підтримку через невелику команду розробників і великий недолік документації через новизну фреймворку. Angular - це фреймворк, розроблений Google і використовуваний для клієнтських SPA-додатків. Angular використовує механізм двосторонньої прив'язки, описаний вище, щоб дозволити динамічну зміну даних в інтерфейсах. Ключовою відмінністю Angular є використання TypeScript. Це спрощує процес розробки та виправляє помилки на етапі компіляції, а не під час виробництва. Однак TypeScript також можна розглядати як недолік, оскільки для його вивчення потрібний час. Angular — найбільш повний з перерахованих вище фреймворків, що надає інструменти для всього циклу розробки веб-додатків, включаючи управління зберіганням даних, навігацію [21]. Складність фреймворку також

може бути недоліком, оскільки для освоєння принципів роботи Angular потрібно набагато більше часу, ніж для інших фреймворків. Обдумавши плюси та мінуси перерахованих вище інструментів, ми вибираємо бібліотеку React для програмної реалізації клієнтської частини нашого інтелектуального модуля розпізнавання шкірних захворювань на зображеннях. Це з тим, що він має неоднозначну архітектуру і методологію розробки задля забезпечення оптимальної роботи. Є дерево DOM, низькорівневі записи і прості процеси для подальшої підтримки програми, що розробляється. Розглянемо інструменти спрощення написання серверної частини інтелектуальних модулів: бібліотеки Express.js, Koa.js, GraphQL. Express.js – це бібліотека для Node.js, що забезпечує створення традиційних інтерфейсів програмування REST. REST (Representation State Transfer) - тип програмного інтерфейсу, що реалізує модель взаємодії клієнт-сервер. У цій моделі використовуються чотири основні типи запитів: отримання даних (GET), відправлення даних (POST), редагування даних (PUT) та видалення даних (DELETE). Зазначена модель запиту спрощує процес обробки запитів, охоплюючи більшість питань, які необхідно вирішити під час обміну інформацією між клієнтом та сервером [21]. Express - це найпопулярніша бібліотека для створення програмних інтерфейсів на JavaScript з великою документацією та повним пакетом потрібних вам функцій при збереженні невеликого розміру пакета.

Koa.js – це мінімальна бібліотека для створення інтерфейсів REST. Кoa має мінімальний розмір пакету та модульність, що дозволяє вам встановлювати та використовувати лише ті пакети, які вам потрібні. Koa.js розроблений творцями Express, тому ми керуємося тими самими принципами розробки програмних інтерфейсів. Кoa значно покращує та спрощує процес виявлення та обробки помилок під час роботи вашого сервера. Основними недоліками Кoa є його новизна, відсутність великої документації та велика спільнота розробників, що ускладнює процес

розробки. GraphQL – це бібліотека, що забезпечує параметричну взаємодію між клієнтом та сервером. Особливість GraphQL в тому, що в бібліотеці використовується лише одна точка входу (endpoint) для взаємодії та реалізована прикладна схема. Це загальна модель всієї структури даних, яка потрібна клієнту для роботи. GraphQL підтримує потужний механізм кешування даних, повертаючи клієнту лише змінені частини даних, що значно знижує використання трафіку та використання обчислювальних ресурсів для обробки даних на клієнті [22]. Недоліком GraphQL можна назвати нестандартну модель взаємодії клієнт-сервер. Це потребує додаткового досвіду досліджень та розробок. Крім того, механізм взаємодії між клієнтом та сервером не є простим і потребує встановлення на клієнті додаткових пакетів, таких як Apollo. Працюючи з великими наборами даних схема досить сильно масштабується і може навіть «впасти» ваш сервер при неправильному використанні.

Після аналізу сильних та слабких сторін внутрішніх інструментів реалізації для створення об'єктно-орієнтованої інформаційної технології було обрано бібліотеку Express.js. Це пов'язано з простотою архітектури клієнт-серверної взаємодії, великою базою документації та великою спільнотою розробників. Це прискорює та спрощує процес розробки бекенда.

3.2 Програмна реалізація компонентів інформаційної технології розпізнавання дерматологічних захворювань на зображенні

Визначимо основні компоненти програмної реалізації модуля розпізнавання шкірних захворювань. Основна частина модуля є проміжним сервером, що забезпечує взаємодію мобільного клієнта з базою даних для зберігання інформації про шкірні захворювання користувача і здійснює

форматування статистичних даних відповідно до необхідних стандартів. Під час ініціалізації програми виконується підключення до бази даних та розгортання системи інтервальної синхронізації. Користувач повинен встановити програму та без проблем почати користуватися програмою..

Фрагмент коду, що реалізує інтервальну синхронізацію даних:

```

    const processRef = useRef()
  async function syncPictureData() { try {
    await fetch('https://skinapi.herokuapp.com')
  } catch (error) { console.log('failed to sync')
  }
  }
  useEffect(() => {
  syncRef.current = setInterval(() => { syncSkinData()
  }, 2000)
  return () => {
  clearInterval(processRef.current)
  }
  }, [])

```

Як тільки прочитана інформація потрапляє на сервер, отримана інформація форматується і далі розкладається на необхідні формати для зберігання та відображення на панелі експертів. Розглянемо моменти розвитку презентаційної частини програмного модуля розпізнавання об'єктів. Інтерфейс розроблений з використанням мови розмітки HTML-сторінок, стилів CSS та мови програмування Javascript. Розглянемо ключові моменти розробки спеціалізованої панелі програмних модулів розпізнавання об'єктів на шкірі людини. Такі панелі необхідні візуалізації отриманих даних

з використанням як текстового опису статистичних даних, і різних типів діаграм, наприклад діаграм типу «коло».

кількість критеріїв класифікації – 7 од., відповідність стандарту ODBS, мінімальна кількість пікселів у зображенні – 1940 од., мінімальна кількість форм на зображенні – 1 од., мінімальна кількість користувачів – 100 чол

Тестування клієнтської частини проводилося шляхом перевірки відповідності необхідним вимогам:

- Кількість критеріїв 6 та більше.
- Кількість пікселів у зображенні має бути не меншою за 1940.
- Кількість об'єктів на зображенні має бути не менше 1 одиниці.

Щоб перевірити, чи ви відповідаєте вимогам до кількості символів для свого посту, відкрийте форму завантаження зображення. Для цього на головному екрані програми необхідно натиснути кнопку підтвердження (рис. 3.1)

3.3 Тестовий приклад роботи компонентів інформаційної технології розпізнавання дерматологічних захворювань на зображенні

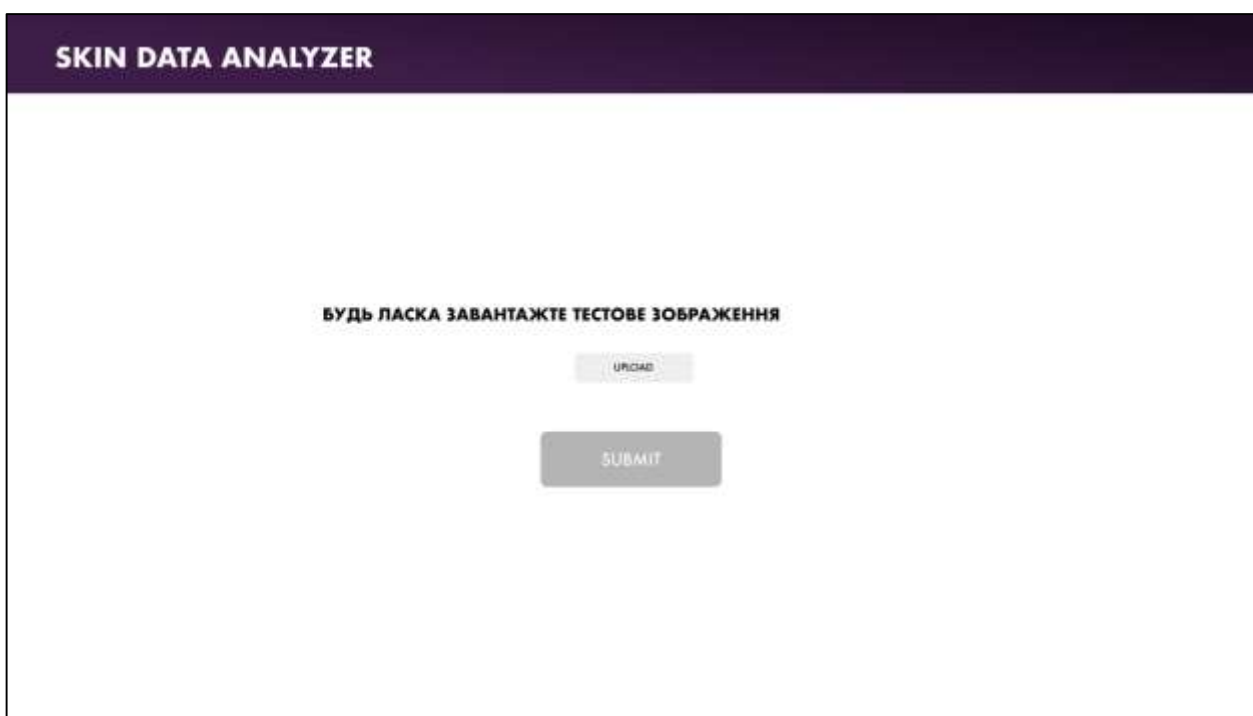


Рисунок 3.1 – Загальний вигляд інтерфейсного вікна

Обираємо тестову картинку та завантажуюємо її за допомогою кнопки upload (рис 3.1)

Після завантаження користувачу доступний інтерфейс із зазначеною назвою файла та кнопкою видалити справа від неї (рис. 3.2)

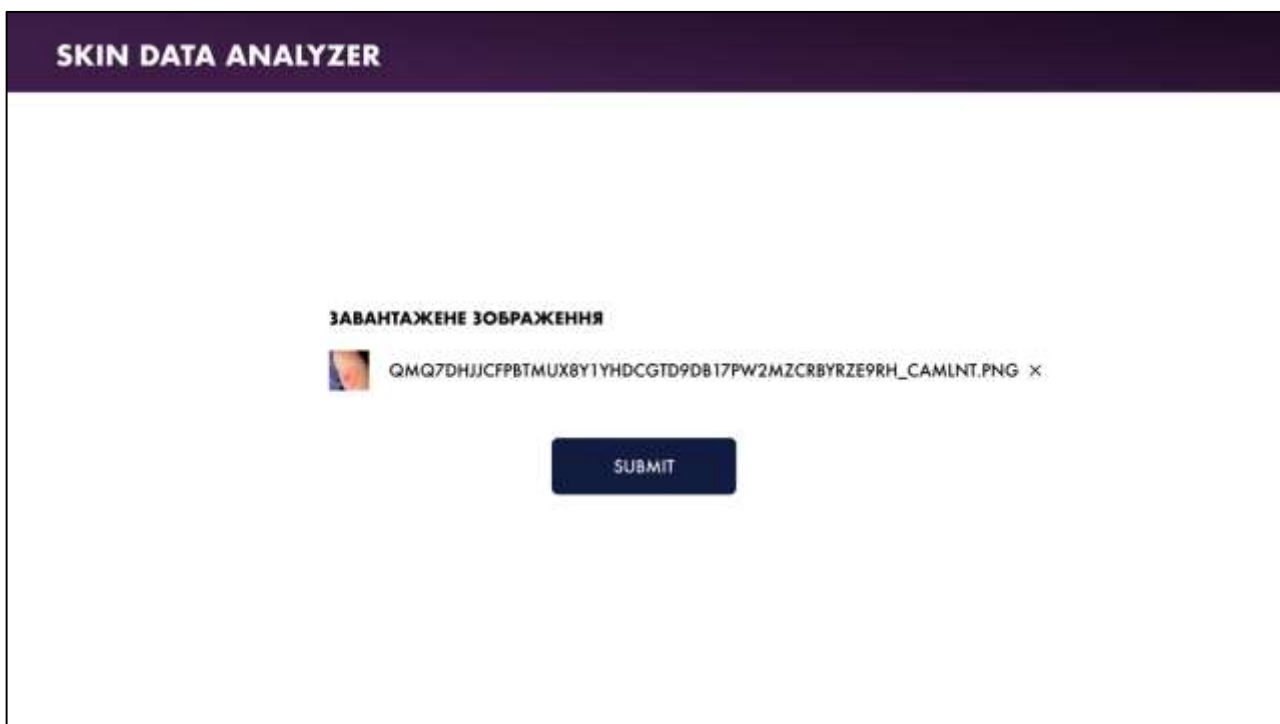


Рисунок 3.2 – Вигляд інтерфейсу користувача з можливістю переглянути назву файла та кнопкою видалення

Після натискання кнопки підтвердження алгоритм розпочинає свою роботу, що відображається у інтерфейсі користувача за допомогою блокування кнопки підтвердження та анімованої іконки завантаження на ній (рис. 3.3)

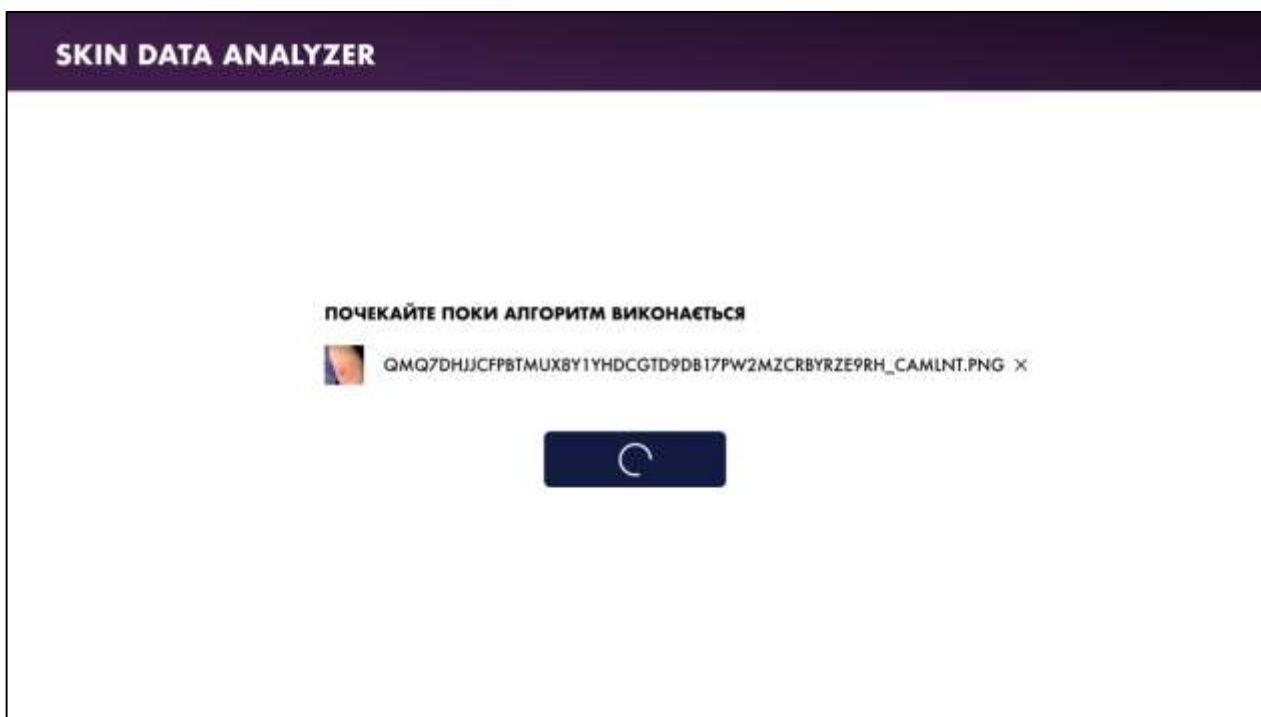


Рисунок 3.3 – Вигляд інтерфейсу користувача під час роботи алгоритму розпізнавання

Після виконання алгоритму відбувається перехід на сторінку результату. Користувач має змогу оглянути критерії оцінки зображення та визначену область із ураженням на початковій картинці (рис 3.4)



Рисунок 3.4 – Вигляд інтерфейсу користувача під час роботи алгоритму розпізнавання

Протягом тестування інтерактивного модулю розпізнавання дерматологічних захворювань на зображенні не було безпосереднього доступу до клінічних даних конкурентів, саме тому було прийнято рішення провести тест в ручному режимі за допомогою відкритих сервісів Google Lens та Interactive Health. Протягом тестування було використано 100 картинок із відкритих джерел, наданих компанією Google та знайдених в мережі Інтернет.

Для того щоб проаналізувати результати інтелектуальної системи модуля «Skin Data Analyzer», було проведено тестування аналогів, а саме «Google Lens» і «Interactive Health». Розмір тестової вибірки дорівнює 100 користувачів.

Google Lens не вірно розпізнав 62 об'єкти, а на вибірці з низькою роздільною здатністю – 80 об'єктів. Interactive Health не вірно розпізнав 80 об'єктів, на вибірці з низькою роздільною здатністю помилок – 85. У Skin

Data Analyzer невірно розпізнано 50, а на вибірці з низькою роздільною здатністю – 70 шкірних об'єктів.

Результати тестування продемонстровані в таблиці 3.1.

Таблиця 3.1 – Порівняння достовірності розпізнавання аналогів та інтелектуальної системи розпізнавання дерматологічних захворювань

Реалізація	Середній відсоток розпізнавання	% вірно розпізнаних об'єктів	% вірно розпізнаних об'єктів на фото низької роздільної здатності	К-сть об'єктів для розпізнавання
Google Lens	27%	38%	20%	100
Interactive Health	17.5%	20%	15%	100
Skin Data Analyzer	40%	50%	30%	100

3.4 Тестування роботи компонентів інформаційної технології розпізнавання дерматологічних захворювань на зображенні

Крім ручного тестування система тестувалася за допомогою модульних тестів.

Модульне тестування – це метод тестування найменшого фрагмента коду, який може бути логічно ізольований у системі. У більшості мов програмування це функція, підпрограма, метод чи властивість. Ізольована частина визначення важлива. У своїй книзі «Робота зі застарілим кодом» автор Майкл Пірс стверджує, що такий тест не є модульним тестом, якщо він залежить від зовнішньої системи. Якщо ви не налаштуєте його, ви не зможете запускати його одночасно з іншими тестами. [23].

The screenshot shows a terminal window titled 'Tilix: Default' with a file path '1: Falcon pusher 22'. The output indicates a test failure in 'src/App.test.js' for the test 'App component > decrements count by 1 when the decrement button is clicked'. The error message states: 'Method "simulate" is only meant to be run on a single node. 0 found instead.' The stack trace includes 'ShallowWrapper.single', 'ShallowWrapper.simulate', 'Object.it', 'new Promise', 'Promise.resolve, then.el', and 'process._tickCallback'. The test results summary shows 1 failed test out of 3 total tests. The terminal also displays 'Watch Usage: Press w to show more.' and a system status bar at the bottom with 'pusher', '1d 9h 13m', '1 nvim 2 2', '70%', '11:13 | 05 Sep', 'ayo', and 'Falcon'.

```

1: Falcon pusher 22
FAIL src/App.test.js
  • App component > decrements count by 1 when the decrement button is clicked

    Method "simulate" is only meant to be run on a single node. 0 found instead.

    at ShallowWrapper.single (node_modules/enzyme/build/ShallowWrapper.js:1815:17)
    at ShallowWrapper.simulate (node_modules/enzyme/build/ShallowWrapper.js:1968:21)
    at Object.it (src/App.test.js:23:18)
      at new Promise (<anonymous>)
    at Promise.resolve, then.el (node_modules/p-map/index.js:46:15)
    at process._tickCallback (internal/process/next_tick.js:68:7)

App component
  ✓ starts with a count of 0 (2ms)
  ✓ increments count by 1 when the increment button is clicked (3ms)
  ✗ decrements count by 1 when the decrement button is clicked (1ms)

Test Suites: 1 failed, 1 total
Tests:       1 failed, 2 passed, 3 total
Snapshots:  0 total
Time:       0.861s, estimated 1s
Ran all test suites.

Watch Usage: Press w to show more.
pusher 1d 9h 13m 1 nvim 2 2 70% | 11:13 | 05 Sep ayo Falcon

```

Рисунок 3.5 – Інтерфейс тестування бібліотекою тестування Jest

Структура Unit тесту блоку програми, що реалізує фільтрацію знайдених об'єктів:

```

describe("Filter function", () => {
  test("it should filter by object", () => {
    expect(filterByObj (input, "link")).toEqual(output);

    expect(filterByObj (input, "LINK")).toEqual(output);
  });
});

function filterByObj (inputArr, searchTerm) {
  const regex = new RegExp(searchTerm, "i");
  return inputArr.filter(function(arrayElement) {
    return arrayElement.url.match(regex);
  });
}

```

Результат перевірки покриття наведено на рис. 3.27.

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	76.92	69.23	68.42	80.65	
src	100	100	100	100	
index.js	100	100	100	100	
src/application	66.67	50	33.33	80	
App.js	66.67	50	33.33	80	50
src/application/About	50	100	0	50	
index.js	50	100	0	50	4
src/application/Users	0	100	0	0	
index.js	0	100	0	0	3-4
src/design-system/Alert	0	100	0	0	
index.js	0	100	0	0	3
src/design-system/GlobalStyles	100	50	100	100	
index.js	100	50	100	100	21
src/design-system/Message	100	100	100	100	
index.js	100	100	100	100	
src/design-system/Toggle	100	83.33	100	100	
index.js	100	83.33	100	100	12
src/utilities	83.33	100	83.33	85.71	
addOne.js	0	100	0	0	1
useToggle.js	100	100	100	100	

Рисунок 3.6 – Результат перевірки покриття тестами програми інтелектуального модуля розпізнавання дерматологічних захворювань

Для того, щоб перевірити сервер на можливість обробляти багато запитів одночасно було проведено навантажувальне тестування шляхом використання спеціалізованого сервісу.

Навантажувальне тестування – це проста форма тестування продуктивності. Зазвичай це робиться для оцінки поведінки програми (додатку) за певного очікуваного навантаження. Це навантаження є, наприклад, кількістю користувачів, що одночасно працюють над програмою. Цей тип тесту дозволяє вам зафіксувати час відгуку всіх ваших найважливіших бізнес-транзакцій. Для тестування використовувався інструмент тестування HTTP/1.1, що підтримує конвеєри HTTP та HTTPS, написаний на Node.js та з відкритим вихідним кодом..

Для тестування використовувалися наступні набори параметрів:

1. Кількість користувачів: 100, запитів на секунду

2. Користувачі: 10; 2. Кількість користувачів: 100, запитів за секунду користувачі: 100;
3. 3. Кількість користувачів: 1000, запитів за секунду Користувачі: 1000;



Рисунок 3.7 – Графік тестування навантаження (1 набір параметрів)

Для загального тестування швидкості застосунку використовувався сервіс Retrace, який показує комплексну аналітику швидкості відповіді сервера на запит, швидкість розгортання інтелектуального модуля, швидкість роботи при низькому інтернеті, час першої взаємодії. Інтерфейс програми зображений на рисунку 3.6



Рисунок 3.8 – Графік тестування навантаження (1 набір параметрів)

Використання Retrace передбачає встановлення програмного модуля безпосередньо на хостинг сайту та сервер, що дозволяє безпосередньо так без перешкод вимірювати показники останніх. Кожна нова перевірка дозволяє сервісу збирати більш якісні дані та точніше визначати проблеми сервера інтелектуального модуля.

3.5 Висновок до розділу 3

В ході тестування програмного модулю було перевірено результати його роботи окремо та порівняно з конкурентами. В ході ручного тестування було перевірено достовірність розпізнавання на зображенні з низькою роздільною здатністю. Інтелектуальна система показала вищу достовірність з високою роздільною здатністю та низькою роздільною здатністю, ніж конкурентні сервіси. Інтелектуальний модуль розпізнавання дерматологічних захворювань на зображенні показав результат достовірності 15% краще серед аналогів.

Таким чином, мети роботи були досягнуті, продемонструвавши підвищену надійність розпізнавання жестів та таку ж ефективну модульність системної інтеграції з іншими веб-додатками. Система також реагувала на жести двома руками під час тестування. Це доводить, що програма може опрацьовувати кілька ходів одночасно.

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Комерційний та технологічний аудит науково-технічної розробки

Метою даного розділу є проведення технологічного аудиту, в даному випадку нового програмного продукту інформаційна технологія розпізнавання дерматологічних захворювань на зображенні. Дана ІТ надає можливість створення програми для автоматизованого розпізнавання та систематизування об'єктів, знайдених на зображенні та інтерактивного доступу до них та має високий рівень інтеграції з іншими програмами.

Аналогами розробки є Interactive Health, ціна 9000 грн, проте програма здатна лише оцінювати стан людини по зображенню, знаходити кінцівки на фото та об'єкти на них. Google Lens, ціна 36000 грн. за бізнес ліцензію. Для проведення комерційного та технологічного аудиту залучають не менше 3-х незалежних експертів. Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням п'ятибальної системи оцінювання за 12-ма критеріями, у відповідності із табл. 4.1.

Таблиця 4.1 – Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Бали (за 5-ти бальною шкалою)					
Кри- терій	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не	Концепція підтверджена	Концепція підтверджена	Концепція перевірена на	Перевірено роботоздат-

Продовження табл. 4.1

Ринкові переваги					
2	Багато аналогів на	Ринкові п Мало аналогів	Кілька аналогів на	Один аналог на великому	Продукт не має аналогів
3	Ціна продукту значно вища за	Ціна продукту дещо вища за	Ціна продукту приблизно до-	Ціна продукту дещо нижче за	Ціна продукту
4	Технічні та споживчі властивості проду-	Технічні та споживчі властивості проду-	Технічні та споживчі властивості проду-	Технічні та споживчі властивості проду-	Технічні та споживчі властивості
5	Експлуатаційні витрати значно вищі, ніж в	Експлуатаційні витрати дещо вищі, ніж в	Експлуатаційні витрати на рівні експлуа-	Експлуатаційні витрати трохи нижчі, ніж в	Експлуатаційні витрати значно
Ринкові перспективи					
6	Ринок малий і не має позити-	Ринок малий, але має пози-	Середній ринок	Великий стабільний	Великий ринок з
7	Активна конкуренція	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практик на здійсненність					
8	Відсутні фахівці як з технічної, так і з ко-	Необхідно наймати фахівців або витратити значні кошти	Необхідне незначне навчання фахівців та збільшення їх	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так із
9	Потрібні значні фінансові ресурси, які	Потрібні незначні фінансові ресурси. Джерела	Потрібні значні фінансові ресурси.	Потрібні незначні фінансові ресурси.	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та

Продовження табл. 4.1

11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років.	Термін реалізації ідеї від 3-х до 5-ти років.	Термін реалізації ідеї менше 3-х років.	Термін реалізації ідеї менше 3-х років.
12	Необхідна розробка регламентних документів та отримання	Необхідно отримання великої кількості дозвільних документів	Процедура отримання дозвільних документів для виробництва	Необхідно тільки повідомлення відповідним органам про виробництво	Відсутні будь-які регламентні обмеження на виробництво та

Усі дані по кожному параметру занесено в таблиці 4.2

Таблиця 4.2 – Результати оцінювання комерційного потенціалу розробки

Критерії оцінювання	ПІБ експертів		
	Експерт 1	Експерт 2	Експерт 3
	Бали		
Технічна здійсненність концепції	3	3	4
Наявність аналогів на ринку	4	3	4
Цінова політика	3	4	3
Технічні та споживчі властивості виробу	4	3	4
Експлуатаційні витрати	3	4	4
Ринок збуту	4	3	4
Конкурентоспроможність	3	4	4

Фахівці з технічної і комерційної реалізації	4	4	4
Фінансування	4	4	3
Матеріально-технічна база	3	3	3
Термін реалізації ідеї	4	3	3
Супровідна документація	4	4	4
Сума	43	42	44
Середньоарифметична сума балів	$(43+42+44) / 3 = 43$		

За даними таблиці 4.2 можна зробити висновок щодо рівня комерційного потенціалу даної розробки. Для цього доцільно скористатись рекомендаціями, наведеними в таблиці 4.3.

Таблиця 4.3 - Рівні комерційного потенціалу розробки

Середньоарифметична сума балів СБ ,	Рівень комерційного потенціалу
0 - 10	Низький
11 - 20	Нижче середнього
21 - 30	Середній
31 - 40	Вище середнього
41 - 48	Високий

Як видно з таблиці, рівень комерційного потенціалу розроблюваного нового програмного продукту є високим, що досягається за рахунок того, що програмний продукт відрізняється від існуючих розширеними функціональними можливостями з метою покращення процесу інтеграції з існуючими системами збору даних про користувача та у можливості задовільнити все більший попит на системи, які дозволяють користувачу не відвідувати медичний заклад фізично, а провести попередню діагностику вдома.

4.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи

4.2.1 Основна заробітна плата розробників, яка розраховується за формулою:

$$Z_o = \frac{M}{T_p} \cdot t, \quad (4.1)$$

де M – місячний посадовий оклад конкретного розробника (дослідника), грн.;

T_p – число робочих днів в місяці, 23 днів;

t – число днів роботи розробника (дослідника).

Результати розрахунків зведемо до таблиці 4.1.

Таблиця 4.1 – Основна заробітна плата розробників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник проекту	28000	1217,39	34	41391,304
Інженер	23000	1000,00	34	34000,000
Всього				75391,30

Так як в даному випадку розробляється програмний продукт, то розробник виступає одночасно і основним робітником, і тестувальником розроблюваного програмного продукту.

4.2.2 Додаткова заробітна плата розробників, які приймали участь в розробці обладнання.

Додаткова заробітна плата прийнято розраховувати як 14 % від основної заробітної плати розробників та робітників:

$$Z_d = Z_o \cdot 14 \% / 100 \% \quad (4.2)$$

$$Z_d = (75391,30 \cdot 14 \% / 100 \%) = 10554,78 \text{ (грн.)}$$

4.2.3 Нарахування на заробітну плату розробників.

Згідно діючого законодавства нарахування на заробітну плату складають 22 % від суми основної та додаткової заробітної плати.

$$H_z = (Z_o + Z_d) \cdot 22 \% / 100\% \quad (4.3)$$

$$H_z = (75391,30 + 10554,78) \cdot 22 \% / 100 \% = 18908,14 \text{ (грн.)}$$

4.2.4. Оскільки для розроблювального пристрою не потрібно витратити матеріали та комплектуючі, то витрати на матеріали і комплектуючі дорівнюють нулю.

4.2.5 Амортизація обладнання, яке використовувалось для проведення розробки.

Амортизація обладнання, що використовувалось для розробки в спрощеному вигляді амортизація обладнання, що використовувалась для розробки розраховується за формулою:

$$A = \frac{Ц}{Tв} \cdot \frac{t_{вик}}{12} \text{ [грн.]} \quad (4.4)$$

де Ц – балансова вартість обладнання, грн.;

T – термін корисного використання обладнання згідно податкового законодавства, років

$t_{\text{вик}}$ – термін використання під час розробки, місяців

Розрахуємо, для прикладу, амортизаційні витрати на комп'ютер балансова вартість якого становить 35000 грн., термін його корисного використання згідно податкового законодавства – 2 роки, а термін його фактичного використання – 1,48 міс.

$$A_{\text{обл}} = \frac{26000}{2} \times \frac{1,73}{12} = 1871,212 \text{ грн.}$$

Аналогічно визначаємо амортизаційні витрати на інше обладнання та приміщення. Розрахунки заносимо до таблиці 4.2. Для розрахунку амортизації нематеріальних ресурсів використовується формула:

$$A_{\text{н.р.}} = C_{\text{н.р.}} * H_a * \frac{t_{\text{вик.}}}{12} \quad (4.5)$$

Але, так як вартість ліцензійної ОС та спеціалізованих ліцензійних нематеріальних ресурсів менше 20000 грн, то даний нематеріальний актив не амортизується, а його вартість включається у вартість розробки повністю, $V_{\text{нем.ак.}} = 9600$ грн.

Таблиця 4.2 – Амортизаційні відрахування матеріальних і нематеріальних ресурсів для розробників

Найменування обладнання	Балансова вартість, грн.	Строк корисного використання, років	Термін використання	Амортизаційні відрахування, грн.
-------------------------	--------------------------	-------------------------------------	---------------------	----------------------------------

			обладнання, місяців	
Комп'ютер та комп'ютерна периферія	35000	2	1,48	2155,797
Офісне обладнання	21000	4	1,48	646,739
Приміщення	1050000	20	1,48	6467,391
Всього				9269,93

5.2.6 Тарифи на електроенергію для непобутових споживачів (промислових підприємств) відрізняються від тарифів на електроенергію для населення. При цьому тарифи на розподіл електроенергії у різних постачальників (енергорозподільних компаній), будуть різними. Крім того, розмір тарифу залежить від класу напруги (1-й або 2-й клас). Тарифи на розподіл електроенергії для всіх енергорозподільних компаній встановлює Національна комісія з регулювання енергетики і комунальних послуг (НКРЕКП). Витрати на силову електроенергію розраховуються за формулою:

$$V_e = V \cdot P \cdot \Phi \cdot K_{\Pi}, \quad (4.6)$$

де V – вартість 1 кВт-години електроенергії для 1 класу підприємства,
 $V = 6,2$ грн./кВт;

P – встановлена потужність обладнання, кВт. $P = 0,6$ кВт;

Φ – фактична кількість годин роботи обладнання, годин.

K_{Π} – коефіцієнт використання потужності, $K_{\Pi} = 0,9$.

$$V_e = 0,9 \cdot 0,6 \cdot 8 \cdot 34 \cdot 6,2 = 910,656 \text{ (грн.)}$$

5.2.7 Інші витрати та загальновиробничі витрати.

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками. Витрати за статтею «Інші витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників:

$$I_e = (Z_o + Z_p) \cdot \frac{H_{ib}}{100\%}, \quad (4.7)$$

де H_{ib} – норма нарахування за статтею «Інші витрати».

$$I_e = 75391,30 * 105\% / 100\% = 79160,87 \text{ (грн.)}$$

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін. Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників:

$$H_{нзв} = (Z_o + Z_p) \cdot \frac{H_{нзв}}{100\%}, \quad (4.8)$$

де $H_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати».

$$H_{нзв} = 75391,30 * 130 \% / 100 \% = 98009 \text{ (грн.)}$$

5.2.9 Витрати на проведення науково-дослідної роботи.

Сума всіх попередніх статей витрат дає загальні витрати на проведення науково-дослідної роботи:

$$B_{заг} = 75391,30 + 10554,78 + 18908,14 + 9269,93 + 9600 + 910,66 + 79160,87 + 98009 = 301804,37 \text{ грн.}$$

5.2.11 Розрахунок загальних витрат на науково-дослідну (науково-технічну) роботу та оформлення її результатів.

Загальні витрати на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються ZB , визначається за формулою:

$$ZB = \frac{B_{заг}}{\eta} \text{ (грн)}, \quad (5.9)$$

де η – коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи.

Так, якщо науково-технічна розробка знаходиться на стадії: науково-дослідних робіт, то $\eta=0,1$; технічного проектування, то $\eta=0,2$; розробки конструкторської документації, то $\eta=0,3$; розробки технологій, то $\eta=0,4$; розробки дослідного зразка, то $\eta=0,5$; розробки промислового зразка, то $\eta=0,7$; впровадження, то $\eta=0,9$. Оберемо $\eta = 0,5$, так як розробка, на даний момент, знаходиться на стадії дослідного зразка:

$$ZB = 301804,37 / 0,5 = 603609 \text{ грн.}$$

4.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

В ринкових умовах загальним позитивним результатом, який потенційний інвестор може отримати від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення чистого прибутку потенційного інвестора, а саме чистий прибуток забезпечить додаткове фінансування. потенційним інвесторам, покращувати фінансові результати його діяльності, підвищувати їхню конкурентоспроможність та позитивно впливати на рішення щодо комерціалізації цієї розробки.

Для розрахунку можливого збільшення чистого прибутку для потенційних інвесторів за рахунок можливого впровадження науково-технічних розробок необхідно::

а) вказати, коли можуть бути впроваджені досягнення науково-технічного розвитку;

б) вказати, через скільки років після впровадження цієї науково-технічної розробки очікуються основні позитивні результати для потенційних інвесторів (наприклад, протягом 3 років після впровадження);

в) кількісно оцінити обсяг поточного та майбутнього попиту на цю чи подібну чи схожу науково-технічну розробку та назвіть основну аудиторію (осіб, які цікавлять) для цього попиту;

г) визначити ринкових цін на науково-технічні розробки зі схожими або подібними функціями.

При розрахунку економічної ефективності необхідно враховувати зміну вартості грошей у часі, оскільки між здійсненням інвестиції та отриманням прибутку минає багато часу. При оцінці ефективності інноваційних проектів розраховуються такі основні показники:

- абсолютного економічного ефекту (чистого дисконтованого доходу);
- внутрішньої економічної дохідності (внутрішньої норми дохідності);

- терміну окупності (дисконтованого терміну окупності).

Аналіз напряму науково-технічного розвитку з урахуванням конкретної ситуації на відповідних умовах економічної ефективності науково-технічного розвитку для можливої комерціалізації потенційними інвесторами розрахунки можуть бути об'єднані.

4.3.1 Розробка чи суттєве вдосконалення програмного засобу (програмного забезпечення, програмного продукту) для використання масовим споживачем.

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

$$\Delta\Pi_i = (\pm\Delta\Pi_o \cdot N + \Pi_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\rho}{100}\right), \quad (4.10)$$

де $\pm\Delta\Pi_o$ – зміна вартості програмного продукту (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу;

N – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки;

Π_o – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки, $\Pi_o = \Pi_o \pm \Delta\Pi_o$;

Π_b – вартість програмного продукту у році до впровадження результатів розробки;

ΔN – збільшення кількості споживачів продукту, в аналізовані періоди часу, від покращення його певних характеристик;

λ – коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт $\lambda = 0,8333$.

p – коефіцієнт, який враховує рентабельність продукту;

ϑ – ставка податку на прибуток, у 2022 році $\vartheta = 18\%$.

Припустимо, що при прогнозованій ціні 7500 грн. за одиницю виробу, термін збільшення прибутку складе 3 роки. Після завершення розробки і її вдосконалення, можна буде підняти її ціну на 250 грн. Кількість одиниць реалізованої продукції також збільшиться: протягом першого року – на 500 шт., протягом другого року – на 250 шт., протягом третього року на 300 шт. До моменту впровадження результатів наукової розробки реалізації продукту не було:

$$\Delta\Pi_1 = (0*250 + (7500 + 250) * 500) * 0,8333 * 0,65 * (1 - 0,18) = 1665624,933 \text{ грн.}$$

$$\Delta\Pi_2 = (0*250 + (7500 + 250) * (500+250)) * 0,8333 * 0,65 * (1 - 0,18) = 2581718,647 \text{ грн.}$$

$$\Delta\Pi_3 = (0*250 + (7500 + 250) * (500+250+300)) * 0,8333 * 0,65 * (1 - 0,18) = 3614406,105 \text{ грн.}$$

Отже, комерційний ефект від реалізації результатів розробки за три роки складе 7861749,69 грн.

4.3.2 Розрахунок ефективності вкладених інвестицій та періоду їх окупності.

Розраховуємо приведену вартість збільшення всіх чистих прибутків $\Pi\Pi$, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$\Pi\Pi = \sum_1^T \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (5.11)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої науково-дослідної (науково-технічної) роботи, грн;

T – період часу, протягом якого виявляються результати впровадженої науково-дослідної (науково-технічної) роботи, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,05 \dots 0,15$;

t – період часу (в роках).

Збільшення прибутку ми отримаємо починаючи з першого року:

$$\begin{aligned} \text{ПП} = & (1665624,933/(1+0,1)^1) + (2581718,647/(1+0,1)^2) + (3614406,105/(1+0,1)^3) = \\ & 1514204,48 + 2133651,774 + 2715556,803 = 6363413,063 \text{ грн.} \end{aligned}$$

Далі розраховують величину початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{инв} * ZB, \quad (4.12)$$

де $k_{инв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{инв} = 2 \dots 5$, але може бути і більшим;

ZB – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

$$PV = 2 * 603609 = 1207217,50 \text{ грн.}$$

Тоді абсолютний економічний ефект E_{abc} або чистий приведений дохід (NPV , *Net Present Value*) для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{abc} = \text{ПП} - \text{PV}, \quad (4.13)$$

$$E_{abc} = 6363413,063 - 1207217,50 = 5156195,56 \text{ грн.}$$

Оскільки $E_{abc} > 0$ то вкладання коштів на виконання та впровадження результатів даної науково-дослідної (науково-технічної) роботи може бути доцільним.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність або показник внутрішньої норми дохідності (IRR , *Internal Rate of Return*) вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій E_g . Для цього використаємо формулу:

$$E_g = \sqrt[T_{жк}]{1 + \frac{E_{abc}}{PV}} - 1, \quad (4.14)$$

$T_{жк}$ – життєвий цикл наукової розробки, роки.

$$E_g = \sqrt[3]{(1 + 5156195,56/1207217,50) - 1} = 0,740$$

Визначимо мінімальну ставку дисконтування, яка у загальному вигляді визначається за формулою:

$$\tau = d + f, \quad (4.15)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2022 році в Україні $d = (0,09 \dots 0,14)$;

f – показник, що характеризує ризикованість вкладень; зазвичай, величина $f = (0,05 \dots 0,5)$.

$$\tau_{\min} = 0,14 + 0,05 = 0,19.$$

Так як $E_b > \tau_{\min}$, то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій за формулою:

$$T_{ок} = \frac{1}{E_g}, \quad (4.16)$$

$$T_{ок} = 1 / 0,740 = 1,35 \text{ р.}$$

Оскільки $T_{ок} < 3$ -х років, а саме термін окупності рівний 1,35 роки, то фінансування даної наукової розробки є доцільним.

4.4 Висновок до розділу 4

В даному розділі розраховано витрати на розробку та виробництво нових технічних рішень на суму 603 609 гривень. Для кожної статті витрат

було розраховано кошторисну вартість. Також розраховується чистий прибуток, який міг би отримати виробник від впровадження нового технологічного рішення, знаходячи термін окупності для виробника та економічний ефект для споживача під час використання цієї розробки. Проаналізувавши розрахунки, можна дійти невтішного висновку, що технологія у виробництві та використанні дешевше і конкурентоспроможніше аналогів. Термін окупності становить приблизно 1,35 року.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи на ступінь магістра вирішено завдання розробки інформаційних технологій та програмного забезпечення для розпізнавання дерматологічних захворювань..

- проведено аналіз сучасного стану розвитку систем розпізнавання дерматологічних захворювань на зображенні;
- обґрунтовано метод розв'язання задачі;
- виконано проектування інтелектуальної системи розпізнавання дерматологічних захворювань на зображенні;
- виконано програмну реалізацію інтелектуальної системи розпізнавання дерматологічних захворювань на зображенні;
- проведено тестування інтелектуальної системи розпізнавання дерматологічних захворювань на зображенні та аналіз результатів тестування;
- здійснено розробку інструкції користувача;
- проведено розрахунок економічної доцільності дослідження;

Мета дослідження були досягнуті шляхом розробки власного веб-додатку з використанням запропонованого шаблону. Це дозволило вирішити проблеми гнучкості та монетизації існуючих аналогів інформаційних технологій, що розробляються, без втрати швидкості програми.

У ході програмної реалізації інформаційних технологій було втілено мови програмування та методики розробки. Розробив та описав алгоритмічну схему програми. Приклади тестів сервера та аналіз наданих результатів, що демонструє здатність витримувати навантаження та перспективи масштабування.

Розвинена інформаційна технологія може бути вдосконалена за рахунок розробки рівневих систем доступу та можливості обробки даних

користувача безпосередньо на пристрої. Це полегшує процес розробки клієнтської частини..

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Інформаційний модуль розпізнавання дерматологічних захворювань на зображенні – [Електронний ресурс]. – Режим доступу: <https://conferences.vntu.edu.ua/index.php/mn/mn2023/paper/view/16872>
2. А. Дідківський. Розробка інтелектуальної системи емоцій в режимі реального часу [Електронний ресурс] – режим доступу: <https://conferences.vntu.edu.ua/index.php/mn/mn2021/paper/view/11793>
3. А. Дідківський. Розробка інтелектуального додатку для аналізу емоцій людини [Електронний ресурс] – режим доступу: <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2021/paper/view/12624>.
4. Google Lens – how it works [Електронний ресурс] – режим доступу: <https://lens.google/howlensworks>
5. Interactive Health [Електронний ресурс] – режим доступу: <https://interactivehealthsolutions.com.au/about-us>
6. Y. Kim. Convolutional Neural Networks for Sentence Classification.
7. АНАЛІЗ МЕТОДІВ РОЗПІЗНАВАННЯ, ЩО ЗАСТОСОВУЮТЬСЯ В КОМП'ЮТЕРНИХ СИСТЕМАХ [Електронний ресурс] – режим доступу: <http://inmad.vntu.edu.ua/portal/static/528B7E7A-0A13-47DF-8C6C-1CF8108A357B.pdf>. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, I. Polosukhin. Attention Is All You Need.
8. Д. Поликовський. Механизмы внимания в нейронных сетях.
9. E. Hoffmann. Standard Statistical Classifications: Basic Principles.
10. A. Kumar. Deep averaging network in Universal sentence encoder [Електронний ресурс] – Режим доступу: <https://medium.com/tech-that-works/deep-averaging-network-in-universal-sentence-encoder-465655874a04>

11. M. Iyyer, V. Manjuhatha, J. Boyd-Graber, Hal Daume III. Deep Unordered Composition Rivals Syntactic Methods for Text Classification.
12. T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean. Distributed Representations of Words and Phrases and their Compositionality
13. Hash Algorithm Comparison: MD5, SHA-1, SHA-2 & SHA-3 [Электронный ресурс] – режим доступа: <https://codesigningstore.com/hash-algorithm-comparison>
14. Загальна характеристика UML [Электронный ресурс] – Режим доступа: <http://www.informicus.ru/default.aspx?SECTION=6&id=73&subdivisionid=2>
15. Д. Фленаган. JavaScript. Подробное руководство.
16. MongoDB [Электронный ресурс] – Режим доступа: <https://www.mongodb.com/>
17. MySQL – Documentation [Электронный ресурс] – Режим доступа: <https://dev.mysql.com/doc/>
18. Redis – Documentation [Электронный ресурс] – Режим доступа: <https://redis.io/documentation>
19. ReactJS - Docs [Электронный ресурс] – Режим доступа: <https://reactjs.org/docs/>
20. Vue.js - Guide [Электронный ресурс] – Режим доступа: <https://vuejs.org/v2/guide/>
21. Angular - Docs [Электронный ресурс] – Режим доступа: <https://angular.io/docs>
22. Express.js - guide [Электронный ресурс] – Режим доступа: <https://expressjs.com/en/guide>
23. Introduction to GraphQL [Электронный ресурс] – Режим доступа: <https://graphql.org/learn>.
24. Jest Documentation [Электронный ресурс] – Режим доступа: <https://jestjs.io/blog>

ДОДАТКИ

Додаток А

Протокол перевірки МКР на наявність текстових запозичень



Имя пользователя:
Озеранський В.С. КН

ID проверки:
1013314585

Дата проверки:
16.12.2022 12:20:49 EET

Тип проверки:
Doc vs Internet + Library

Дата отчета:
16.12.2022 12:33:36 EET

ID пользователя:
62038

Название файла: 122МКР-ДідківськийAA2022

Количество страниц: 45 Количество слов: 5928 Количество символов: 47655 Размер файла: 813.37 KB ID файла: 1013072976

6.93%

Совпадения

Наибольшее совпадение: 2.97% с Интернет-источником (<https://ukr.media/science/381674>)

3.54% Источники из Интернета 7 Страница 47

3.96% Источники из Библиотеки 47 Страница 47

0% Цитат

Исключение цитат выключено

Исключение списка библиографических ссылок выключено

1.38% Исключений

Некоторые источники исключены автоматически (фильтры исключения: количество найденных слов меньш...

1.01% Исключений из Интернета 11 Страница 48

0.71% Исключенного текста из Библиотеки 64 Страница 48

Модификации

Обнаружены модификации текста. Подробная информация доступна в онлайн-отчете.

Замененные символы 1

Додаток Б

Лістинг програми

```
exports.processDermImage = async (req, res) => {
  const { id, data } = req.body;

  try {
    if (!data) {
      throw new Error('No "data" in payload');
    }

    let payload = data;

    const hasNewImages = data.images.some((item) =>
      item.url.includes("base64")
    );

    if (hasNewImages) {
      const old_images = data.images.filter(
        (item) => !item.url.includes("base64")
      );
      const new_images = data.images.filter((item) =>
        item.url.includes("base64")
      );

      const uploaded_images = await Promise.all(
        new_images.map(async (item) => ({
          ...item,
          url: await upload(item.url),
        })))
      );

      const final = [...old_images, ...uploaded_images];

      payload = {
        ...data,
        preview_image: final[0].url,
        images: final,
      };
    }

    const new_item = await Catalogue.update({ _id: id }, payload);

    return res.send(new_item).status(200);
  } catch (err) {
    console.log(err.message);

    return res.send({ error: err.message });
  }
};
```

```
function upload(base64) {
  return new Promise((resolve, reject) => {
    dataSkinAnalyzer.v2.uploader.upload(base64, function (error, result) {
      if (error) {
        reject(error);
      }

      if (result) {
        resolve(result.url);
      }
    });
  });
}
```

Додаток В

ІЛЮСТРАТИВНА ЧАСТИНА

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ РОЗПІЗНАВАННЯ
ДЕРМАТОЛОГІЧНИХ ЗАХВОРЮВАНЬ НА ЗОБРАЖЕННІ

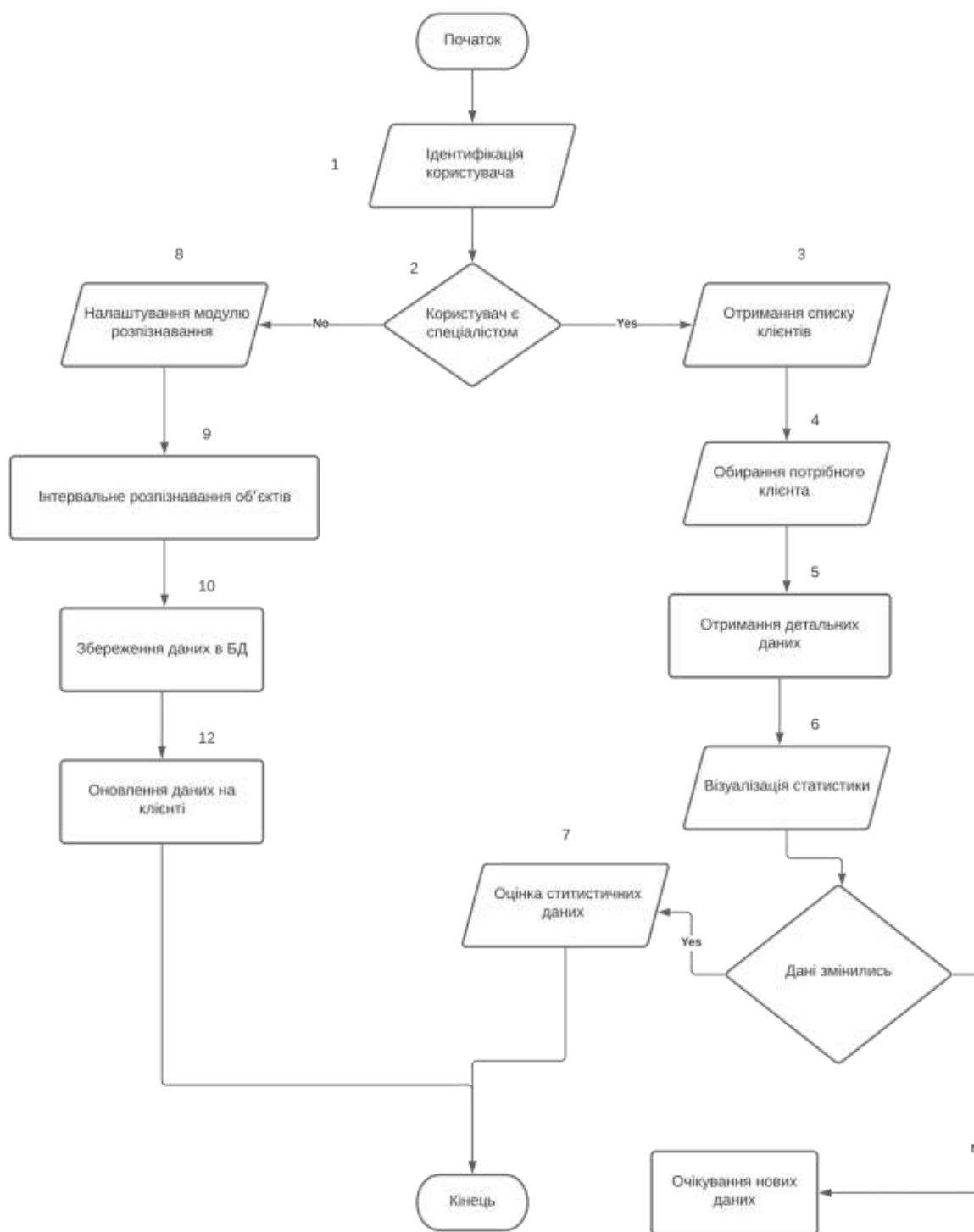
Виконав: студент 1-го курсу,
групи 1КН-21м
спеціальності 122 «Комп'ютерні науки»
(шифр і назва програми підготовки, спеціальності)

Дідківський А.А.
(прізвище та ініціали)

Керівник: д.т.н., професор каф. КН

Сілагін О.В.
(прізвище та ініціали)

«15» 12 2022 р.



11

Рисунок В.1 – Схема основного алгоритму роботи програмного модулю розпізнавання об'єктів

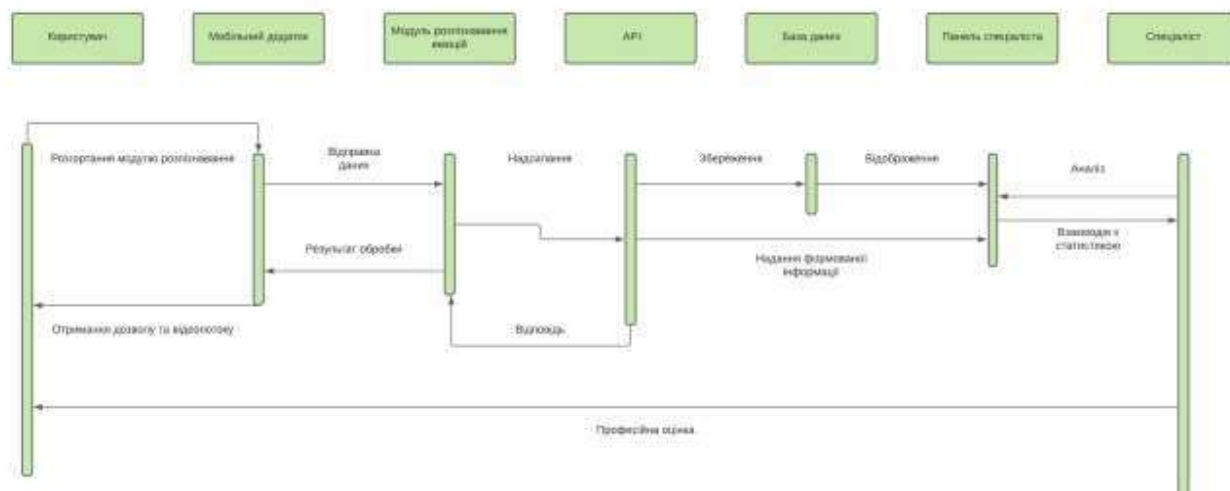


Рисунок В.2 – Діаграма послідовності програмного модуля розпізнавання дерматологічних захворювань

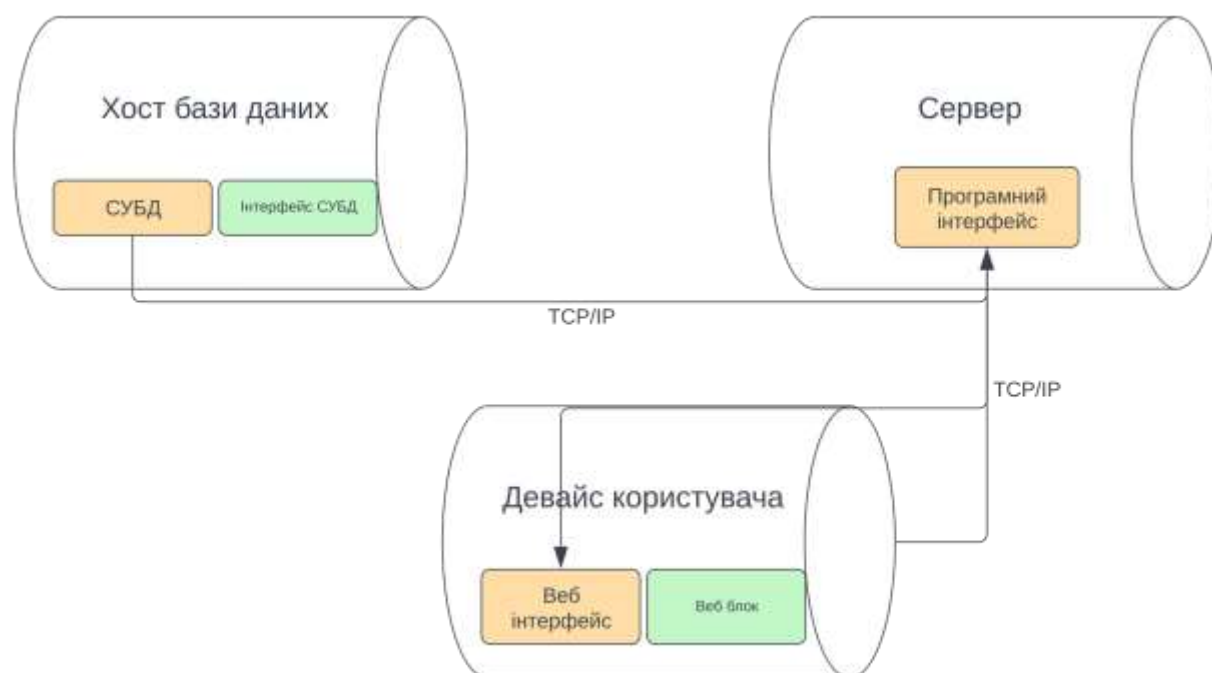


Рисунок В.3 – діаграма розгортання інформаційної технології розпізнавання дерматологічних захворювань



Рисунок В.3 – Загальний вигляд інтерфейсного вікна

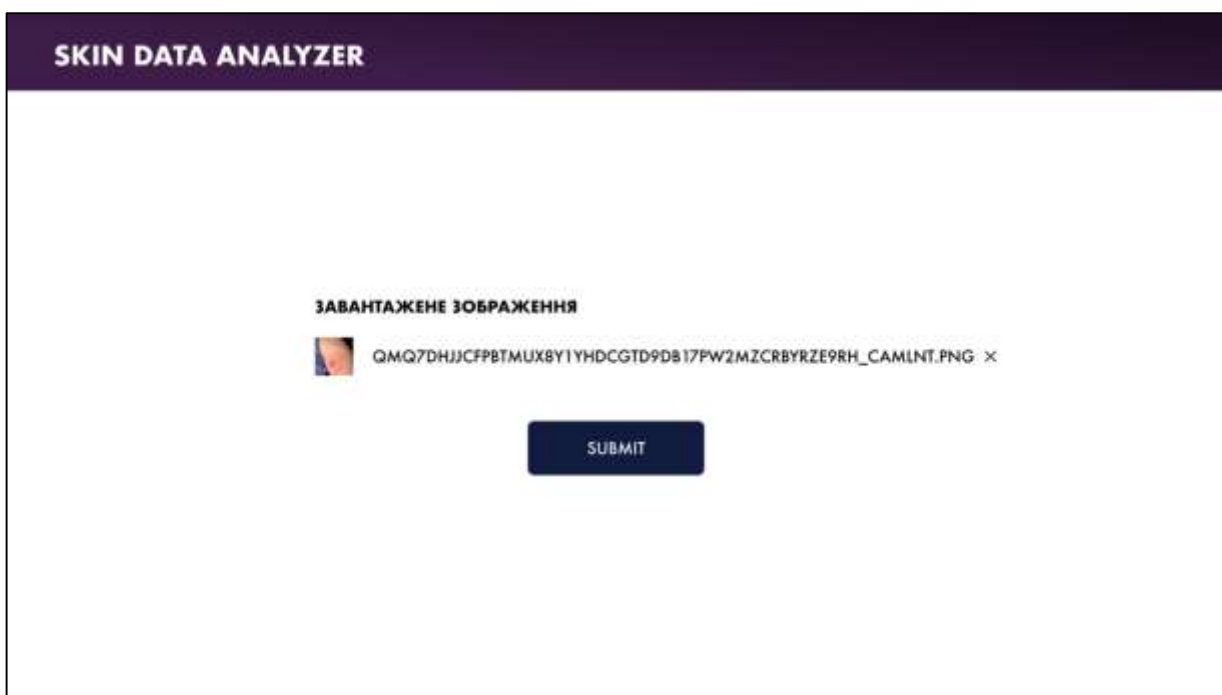


Рисунок В.4 – Вигляд інтерфейсу користувача з можливістю переглянути назву файлу та кнопкою видалення

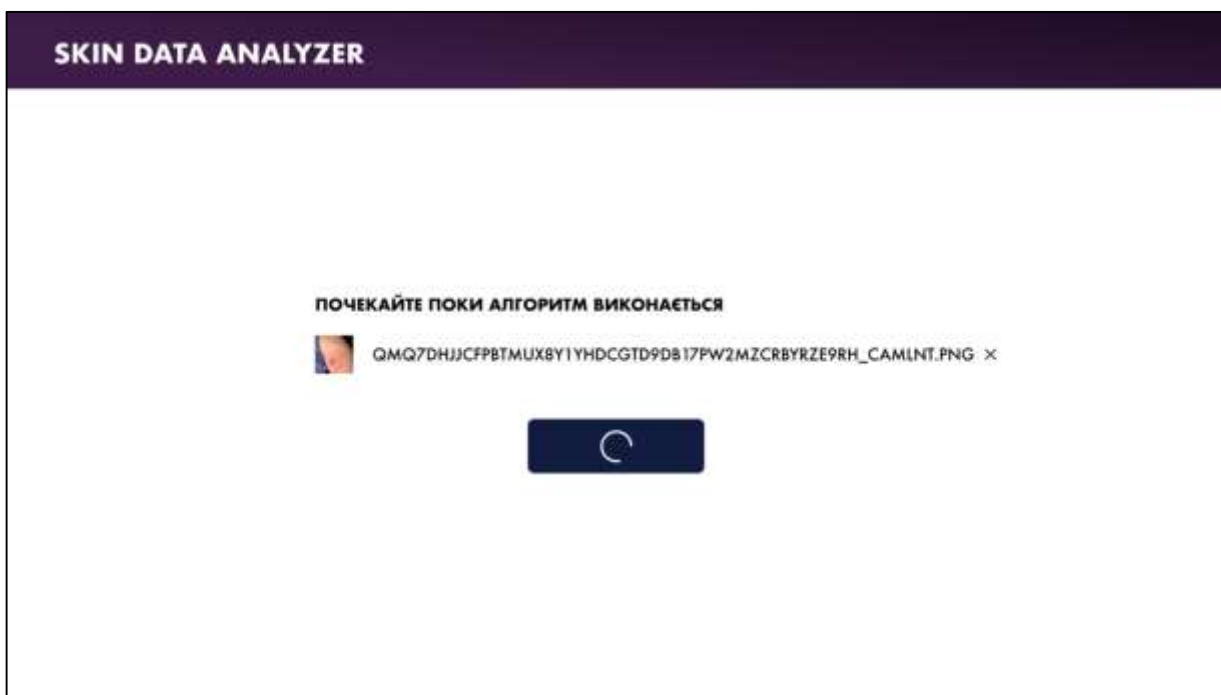


Рисунок В.5 – Вигляд інтерфейсу користувача з можливістю переглянути назву файла та кнопкою видалення

The screenshot shows a terminal window titled "Tilix: Default" with a file path "1: Falcon src/App.test.js". The output indicates a test failure:

```

FAIL src/App.test.js
  • App component › decrements count by 1 when the decrement button is clicked

    Method "simulate" is only meant to be run on a single node. 8 found instead.

    at ShallowWrapper.single (node_modules/enzyme/build/ShallowWrapper.js:1815:17)
    at ShallowWrapper.simulate (node_modules/enzyme/build/ShallowWrapper.js:1068:21)
    at Object.it (src/App.test.js:23:18)
      at new Promise (<anonymous>)
    at Promise.resolve,then,e1 (node_modules/p-map/index.js:46:16)
    at process._tickCallback (internal/process/next_tick.js:68:7)

App component
  ✓ starts with a count of 0 (2ms)
  ✓ increments count by 1 when the increment button is clicked (3ms)
  ✗ decrements count by 1 when the decrement button is clicked (1ms)

Test Suites: 1 failed, 1 total
Tests:       1 failed, 2 passed, 3 total
Snapshots:  0 total
Time:        0.861s, estimated 1s
Ran all test suites.

Match Usage: Press w to show more.
  
```

The terminal window also shows a status bar at the bottom with information such as "pusher", "1 d 9h 13m", "1 nvim", "2 2", "70%", "11:13", "05 Sep", "ayo", and "Falcon".

Рисунок В.6 – Інтерфейс тестування бібліотекою тестування Jest

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #
All files	76.92	69.23	68.42	80.65	
src	100	100	100	100	
index.js	100	100	100	100	
src/application	66.67	50	33.33	80	
App.js	66.67	50	33.33	80	58
src/application/About	50	100	0	50	
index.js	50	100	0	50	4
src/application/Users	0	100	0	0	
index.js	0	100	0	0	3-4
src/design-system/Alert	0	100	0	0	
index.js	0	100	0	0	3
src/design-system/GlobalStyles	100	50	100	100	
index.js	100	50	100	100	21
src/design-system/Message	100	100	100	100	
index.js	100	100	100	100	
src/design-system/Toggle	100	83.33	100	100	
index.js	100	83.33	100	100	12
src/utilities	83.33	100	83.33	85.71	
addOne.js	0	100	0	0	3
useToggle.js	100	100	100	100	

Рисунок В.7 – Результат перевірки покриття тестами програми інтелектуального модуля розпізнавання дерматологічних захворювань