УДК 004.93'1:004.415.2

ПРОЄКТУВАННЯ ВЕБЗАСТОСУНКУ ДЛЯ РОЗПІЗНАВАННЯ ВИДІВ РОСЛИН НА ЗОБРАЖЕННЯХ

Турбаров І. О.[1]; Новікова Т.О.[2]
[1,2] Національний університет «Одеська політехніка»
[1,2] Україна, Одеса
[1] 8088651@stud.op.edu.ua; [2] novikova.t.o@op.edu.ua

Анотація. *Спроєктовано та розроблено вебзастосунок для розпізнавання видів рослин на зображеннях на основі нейромережі ResNet-50, серверного фреймворку FastAPI, клієнтської частини React та хмарних сервісів Firebase. Визначено цілі й вимірювані індикатори, виконано декомпозицію вимог, проаналізовано ризики та заходи їх зниження, обґрунтовано вибір централізованої хмарної архітектури.*
Ключові слова: *розпізнавання зображень; ResNet-50; FastAPI; хмарна архітектура; аналіз ризиків; декомпозиція вимог.*

Вступ. Автоматизоване розпізнавання видів рослин за фотографічними зображеннями є критично важливою задачею для сучасного цифрового сільського господарства, моніторингу біорізноманіття та екологічних досліджень. Зростаючі обсяги даних та потреба в швидкій і точній ідентифікації біологічних об'єктів вимагають створення надійних програмних інструментів, проте широка варіативність формоутворення рослин, подібність міжвидових ознак і мінливість умов зйомки створюють суттєві труднощі для класичних алгоритмів комп'ютерного зору.

Традиційні методи комп'ютерної обробки зображень часто виявляються малоефективними через чутливість до шуму, освітлення та обмеженість вручну сформованих ознак. Фундаментальні підходи до глибокого навчання та побудови залишкових нейронних мереж, закладені у працях K. He, X. Zhang, S. Ren та J. Sun [1], а також масштабні бенчмарки розпізнавання образів, досліджені O. Russakovsky, J. Deng, H. Su [2]. та ін., створили підґрунтя для сучасного розвитку комп'ютерного зору. Водночас питання архітектурного проєктування складних програмних систем і якість взаємодії компонентів вебзастосунків ґрунтовно досліджують L. Bass, P. Clements та R. Kazman [3]., а також українські фахівці у сфері інженерії програмного забезпечення та інформаційних технологій (зокрема, теоретичні й практичні засади розробки вебсистем і хмарних архітектур аналізують В. А. Пасічник, С. Л. Кривий та інші провідні вітчизняні науковці).

Попри значні успіхи в алгоритмічному розпізнаванні, більшість існуючих рішень зосереджені переважно на точності моделей, залишаючи поза увагою питання їх системної інтеграції в адаптивні вебзастосунки. Недостатнє обґрунтування архітектурних рішень щодо взаємодії між клієнтом, сервером і сховищем даних часто призводить до високої затримки відповіді та низької масштабованості. Отже, актуальність дослідження зумовлена необхідністю комплексної розробки та системного обґрунтування хмарної архітектури вебзастосунку, яка поєднує високу точність штучного інтелекту з низьким часом відгуку.

Мета дослідження – проєктування вебзастосунку для розпізнавання видів рослин на зображеннях на основі ResNet-50, FastAPI, React і Firebase та аналіз вимог, ризиків і архітектурних рішень, що забезпечують точність не менше 80 % і середній час відповіді до 3 с.

Основний текст. На першому етапі дослідження було сформульовано організаційні цілі та вимірювані індикатори, а саме вартість використання моделі – 0 грн, частку успішно виконаних запитів – не менше 95 %, середній час розпізнавання – до 3 с, виконання

розпізнавання не більш ніж за три основні дії користувача. Для компонента штучного інтелекту задано цільову точність не менше 80 % на контрольному наборі даних і середній час інференсу до 2 с.

Систему подано як набір компонентів. Так, клієнт на React виконує первинну перевірку формату (JPEG/PNG) і розміру файлу. Сервер на FastAPI [4] повторно валідує дані, виконує масштабування, кадрування та нормалізацію колірних каналів і передає тензор моделі. Попередньо навчена згорткова мережа ResNet-50 [1] повертає назву класу та рівень впевненості. Firebase Authentication і Cloud Firestore забезпечують автентифікацію й зберігання історії розпізнавань, прив'язаної до облікового запису. Ключову вимогу (визначення категорії об'єкта за час, що не перевищує 3 с) декомпозовано на субвимоги до кожного компонента та зафіксовано критичні припущення щодо користувацького, серверного й хмарного середовищ, а також вхідних даних.

У ході дослідження основним ризиком визнано повернення неправильної категорії або перевищення часу відповіді. Його першопричинами є повільне з'єднання чи великий файл, недостатня продуктивність сервера, помилки попередньої обробки, належність об'єкта до класів, на яких модель не навчалась, недоступність FastAPI або помилки запису в Firebase. ResNet-50 у стандартному варіанті навчена на наборі ImageNet [1; 2], тож для предметної області рослин це обмеження потребує окремої перевірки під час вибору й підготовки моделі.

На основі виокремлених першопричин ризику запропоновані наступні заходи його зниження: валідація формату та розміру на клієнті й сервері, контроль часу очікування й обробка винятків, індикатор завантаження та повторення запиту, обмеження навантаження й кешування, а також порогове значення впевненості, за нижчого рівня якого користувач отримує попередження про можливу неточність.

Для вибору архітектури розглянуто чотири кандидати. Це локальне розпізнавання у браузері; мобільний застосунок із моделлю у форматі TensorFlow Lite або ONNX; централізоване хмарне розпізнавання; гібридний підхід із легкою локальною моделлю та резервним сервером. Оцінювання виконано за принципом аналізу компромісів між атрибутами якості [3] (табл. 1).

Таблиця 1 – Порівняння архітектур-кандидатів за характеристиками якості

Характеристика	**Комп'ютер**	**Смартфон**	**Хмара**	**Гібридна**
Затримка відповіді	Низька без мережі	Низька без мережі	Середня, залежить від мережі	Низька для локальних запитів
Спостережуваність	Низька	Низька	Висока	Середня–висока
Масштабованість	Обмежена пристроями	Обмежена пристроями	Висока за належної інфраструктури	Висока, але складніша
Оновлення системи	Складніші	Складніші	Простіші	Середньої складності
Конфіденційність	Висока	Висока	Нижча без додаткових заходів	Висока для локальних запитів
Вартість розробки і підтримки	Середня	Висока	Середня	Висока

Централізовану хмарну архітектуру обрано як найзбалансованішу, тому що вона поєднує передбачувану продуктивність, простоту реалізації, централізований моніторинг і можливість одноразового оновлення моделі на сервері. Її основними компромісами є залежність від мережі, витрати на інфраструктуру та передавання зображень на зовнішній сервер, що потребує шифрування каналу, обмеження доступу до сховищ і визначення строків зберігання

даних. У навчальному форматі ці обмеження зменшуються застосуванням базових серверних потужностей, лімітів на розмір файлів і частоту запитів. Гібридний підхід залишається перспективним напрямом за умови зростання вимог до автономності, конфіденційності чи локальної швидкодії.

Висновки. Сформований комплекс цілей, вимірюваних індикаторів, системних компонентів та декомпозованих субвимог створює чітку концептуальну основу, яка суттєво спрощує подальшу програмну інженерію та проєктування аналогічних інтелектуальних систем. Системний аналіз першопричин ризиків і розробка заходів щодо їх мінімізації дозволяють уникнути критичних архітектурних помилок ще на початкових етапах розробки, гарантуючи високу надійність підсистеми розпізнавання. Обґрунтування централізованої хмарної архітектури на основі стеку React, FastAPI, ResNet-50 та Firebase забезпечує оптимальний баланс між продуктивністю, гнучкістю та масштабованістю. Хоча наведені показники точності (не менше 80 %) та часу відповіді (до 3 с) є цільовими орієнтирами, проведене дослідження закладає якісний фундамент для їх подальшої експериментальної верифікації та донавчання моделі на розширених наборах даних предметної області. Розвиток цього вебресурсу відповідає актуальним IT-тенденціям у галузях штучного інтелекту, агротехнологій та екологічного моніторингу, відкриваючи перспективи для впровадження Edge AI, стиснення моделей (ONNX, TFLite) та мобільної адаптації. У майбутньому інтеграція розробленої підсистеми з мультимодальними LLM та геоінформаційними сервісами дозволить створити комплексний інтелектуальний асистент для автоматизованої аналітики біорізноманіття та точного землеробства.

Використання інструментів ШІ. Використано Claude (Anthropic), модель Claude Sonnet 5.5, для структурування та редагування тексту тез на основі матеріалів курсового проєкту автора. Зміст, результати та висновки перевірено авторами.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 770–778). IEEE. https://doi.org/10.1109/CVPR.2016.90

[2] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., … Fei-Fei, L. (2015). ImageNet large scale visual recognition challenge. *International Journal of Computer Vision, 115*(3), 211–252. https://doi.org/10.1007/s11263-015-0816-y

[3] Bass, L., Clements, P., & Kazman, R. (2021). *Software architecture in practice* (4th ed.). Addison-Wesley. URL: https://www.oreilly.com/library/view/software-architecture-in/9780136885979/

[4] FastAPI. (n.d.). *FastAPI documentation*. Retrieved October 5, 2026, from https://fastapi.tiangolo.com/

Turbarov I. O., Novikova T. O.

Design of a web application for plant species recognition in images

Abstract*. The paper substantiates the choice of a centralized cloud architecture for plant species recognition in images based on the ResNet-50 neural network, the FastAPI server framework, a React client, and Firebase cloud services. Organizational goals and measurable indicators are defined, the key requirement is decomposed, and risks together with mitigation measures are analyzed. Four candidate architectures are compared by quality attributes, and a centralized cloud architecture is justified as the most balanced choice.*

Keywords*: image recognition; ResNet-50; FastAPI; cloud architecture; risk analysis; requirements decomposition.*