

П'ятикоп О.Є., к.т.н.; Шевченко А.Є.;
Левицька Т.О., к.т.н.

ПРОЄКТУВАННЯ ХМАРНОЇ АРХІТЕКТУРИ ТА МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ СИСТЕМИ ІДЕНТИФІКАЦІЇ СІЛЬСЬКОГОСПОДАРСЬКИХ ШКІДНИКІВ

Piatykop O.Ye., Shevchenko A.Ye., Levytska T.O. Designing a cloud architecture and mobile application for an agricultural pest identification system. The article considers approaches to designing a cloud-based architecture for an intelligent agricultural pest identification system using a mobile application. An analysis of modern approaches to building mobile object recognition systems is conducted and the relevance of combining mobile technologies, cloud computing, and machine learning methods for solving computer vision problems in conditions of limited mobile device resources is substantiated. The system architecture based on the distribution of the computational load between the client and server parts is proposed. The mobile application performs the functions of receiving and transmitting images, while data processing is implemented on the server using the YOLOv11-N deep learning model. The architecture includes an application server, a computer vision service, a mobile client, and external cloud services for data storage and user authorization. The technological implementation of the system is described, in particular, the use of Node.js and FastAPI for the server side, Docker containerization and the Google Cloud Run platform for deployment, as well as Firebase for organizing data storage and user management. The proposed solution provides scalability, efficient use of resources and the ability to centrally update machine learning models. The results obtained can be used to create intelligent monitoring systems in the agricultural sector and other industries that require real-time object recognition.

Keywords: cloud computing, mobile application, computer vision, machine learning, pest identification, YOLO, distributed architecture, microservices

П'ятикоп О.Є., Шевченко А.Є., Левицька Т.О. Проєктування хмарної архітектури та мобільного застосунок для системи ідентифікації сільськогосподарських шкідників. У статті розглянуто підходи до проєктування хмарно-орієнтованої системи ідентифікації сільськогосподарських шкідників із використанням мобільного застосунок. Проведено аналіз сучасних рішень у галузі мобільних і хмарних технологій та визначено переваги розподіленої архітектури. Запропоновано модель системи, у якій обробка зображень виконується на сервері з використанням алгоритму YOLO, а мобільний застосунок виступає клієнтським інтерфейсом. Реалізація забезпечує ефективне використання ресурсів, масштабованість та можливість централізованого оновлення моделей.

Ключові слова: хмарні обчислення, мобільний застосунок, комп'ютерний зір, машинне навчання, ідентифікація шкідників, YOLO, розподілена архітектура, мікросервіси

Вступ

У сучасних умовах стрімкого розвитку інформаційних технологій особливої актуальності набувають системи, що поєднують можливості мобільних пристроїв, хмарних обчислень та методів штучного інтелекту. Зростання обсягів даних, потреба в їх оперативній обробці та обмеженість ресурсів мобільних пристроїв обумовлюють необхідність створення ефективних архітектурних рішень, здатних забезпечити високу продуктивність, масштабованість і доступність сервісів. У цьому контексті поєднання мобільних застосунків із хмарною інфраструктурою стає ключовим підходом до побудови сучасних інтелектуальних систем.

Актуальність дослідження визначається кількома чинниками. По-перше, мобільні пристрої виступають основною платформою взаємодії користувачів із цифровими сервісами, однак їх обчислювальні ресурси залишаються обмеженими порівняно з серверними системами. По-друге, сучасні алгоритми обробки даних, зокрема моделі комп'ютерного зору та машинного навчання, є ресурсомісткими та потребують значних обчислювальних потужностей. Це зумовлює необхідність використання хмарних технологій для виконання складних обчислень і централізованого управління моделями. По-третє, зростає потреба у створенні гнучких, масштабованих та адаптивних систем, здатних працювати в умовах змінного навантаження та різних сценаріїв використання.

Додатково важливим аспектом є необхідність оптимального розподілу обчислювальних задач між клієнтською та серверною частинами системи. Неefективне проєктування може

призводити до збільшення затримок, перевитрати енергоресурсів мобільного пристрою та зниження якості користувацького досвіду. Тому виникає потреба у формуванні підходів до побудови гібридних архітектур, які поєднують локальну обробку даних із можливостями хмарних сервісів, забезпечуючи баланс між швидкістю та ресурсною ефективністю.

Таким чином, розроблення узагальнених принципів проектування хмарної архітектури та мобільного застосунок є важливим науково-прикладним завданням, що має міжгалузевий характер і може бути застосоване у широкому спектрі систем розпізнавання об'єктів та підтримки прийняття рішень.

Аналіз останніх досліджень. Багато науковців присвячують свої дослідження питанням проектування архітектури програмних застосунків [1-3], обробки даних у мобільних системах [2, 4-6] та розробці інтелектуальних мобільних систем розпізнавання об'єктів [7-11], в тому числі на пристроях з обмеженими ресурсами [12, 13], а також з використанням хмарних обчислень [7-11]. Важливість проектування архітектури відзначають в роботах [1-3]. В роботі [3] розглянуто метод формування архітектури програмного застосунок на основі бачення (V-VAM), який забезпечує системний підхід до проектування та розвитку архітектури. А роботі [2] аналізують моделі мобільних додатків та відзначають, що комплексна взаємодія всіх складових додатку: інтерфейсу користувача, клієнтської і серверної частин, API, систем безпеки, інструментів аналітики, хмарних сервісів і ретельного тестування, забезпечують успішність та функціональність додатку та відповідність сучасним вимогам користувачів. Тому раціонально спроектована архітектура системи дуже важлива. Також в [3] практично досліджується застосування мікросервісної архітектури для побудови стійких комплексних систем підтримки прийняття рішень із застосуванням моделей машинного навчання. В статті зазначають, що для розробки таких систем доцільно застосувати підхід, де користувацькі застосунки спілкуються з центральним сервером, який в свою чергу на пряму комунікує із внутрішніми мікросервісами.

На сьогодні використання моделей машинного навчання для мобільних застосунків реалізують різними підходами. У статті [4] представлено мобільний застосунок Grape Guard, розроблений для автоматичного виявлення захворювань листя винограду. Застосунок реалізовано для операційної системи Android з використанням моделі глибинного навчання YOLO, конвертовану у формат TensorFlow Lite, що дозволяє виконувати обробку зображень безпосередньо на мобільному пристрої в режимі реального часу. Таким чином, система орієнтована на on-device обробку без необхідності хмарної інфраструктури. Схожий підхід щодо реалізації мобільного застосунок із вбудованою моделлю глибинного навчання описано у статті [5] для виявлення рисових панікул і визначення стадій росту рослин. У дослідженні [6] також пропонується офлайн-орієнтований мобільний застосунок, але вже для класифікації шкідників та із вбудованою моделлю GoogleNet. Уся обробка виконується безпосередньо на пристрої, включаючи захоплення зображень, їх обробку та відображення результатів, проте серверна частина відсутня. В роботі [13] також досліджувались різні бібліотеки машинного навчання для вирішення задачі розпізнавання об'єктів у реальному часі по кадрах на мобільних пристроях. Автори запропонували нову архітектуру для розпізнавання об'єктів на основі машинного навчання на смартфонах, проте самі відзначили, що, хоча їх підхід здатний виконувати попередню обробку в режимі реального часу без втрати жодних кадрів, вузьким місцем є процес розпізнавання.

З іншого боку, значна кількість дослідників пропонують архітектуру мобільної системи розпізнавання зображень, що базується на концепції винесення обчислень у хмару [7-11].

У роботі [7] описано мобільний застосунок для розпізнавання об'єктів на основі великих згорткових нейронних мереж, зокрема ImageNet. Запропонований авторами Android-застосунок виконує зйомку та відправляє зображення на сервер, який містить CNN-модуль і повертає результати розпізнавання додатку. Автори також досліджували питання зменшення часу обробки та енергоспоживання, тому у статті підкреслено необхідність виконання обчислень у хмарному середовищі через обмежені обчислювальні та енергетичні ресурси смартфонів.

Подальший розвиток цього підходу представлено у дослідженні [8], де запропоновано мобільний застосунок, що використовує швидко згорткову нейронну мережу на основі регіонів (Faster R-CNN) для виконання завдання розпізнавання комах-шкідників на основі хмарних обчислень. У цій роботі мобільний клієнт взаємодіє з сервером, на якому розгорнуто модель глибинного навчання, що забезпечує автоматичну класифікацію шкідників за зображенням. Також схожу архітектуру використовують в роботі [9], де створюють мобільний додаток для виявлення дефектів у кавових зернах. Вхідні дані, що обробляються системою, надходять від клієнтського компонента у вигляді зображень, отриманих за допомогою мобільного застосунку шляхом фотографування кавових зерен. Мобільний додаток надсилає запит до сервера, щоб продовжити етап обробки зображень за допомогою моделі машинного навчання та зберігання зображень. Результати виявлення за допомогою моделі повертаються на сторону клієнта у додаток.

У роботі [10] автори пропонують систему з комбінованою архітектурою: хмарну платформу, мобільний застосунок та веб-інтерфейс для користувача. Система спрямована на виявлення та локалізацію щитівок на зображенні з використанням моделі YOLOv4. Схожий підхід застосовано у дослідженні [11], де розроблено мобільну систему для виявлення шкідників у реальному часі з використанням полегшених моделей сімейства YOLO. Мобільний застосунок виконує виявлення шкідників у режимі реального часу, використовуючи камеру пристрою для отримання зображень рослин. Хмарний сервер Firebase виконує функції зберігання, обробки та агрегування даних, а також забезпечує взаємодію з веб-застосунком моніторингу.

Отже, огляд літератури показав, що існують два основні підходи до побудови мобільних застосунків ідентифікації об'єктів – використання хмарних обчислень та виконання обробки безпосередньо на мобільному пристрої. Перший підхід забезпечує високу точність і масштабованість, тоді як другий – автономність і швидкодію. Але хмарна архітектура забезпечує не тільки масштабованість системи, а й централізоване оновлення моделей та зменшення навантаження на пристрій користувача. Це позитивно вплине на швидкодію та енергоспоживання мобільного застосунку.

Метою роботи є проектування інтелектуальної системи ідентифікації з розподіленою (хмарно-орієнтованою) архітектурою та мобільним інтерфейсом для забезпечення інтеграції моделей штучного інтелекту та оптимального розподілу обчислювальних ресурсів між клієнтською та серверною частинами на прикладі задачі ідентифікації сільськогосподарських шкідників з використанням моделі YOLO.

Виклад основного матеріалу дослідження

Інтелектуальну систему ідентифікації сільськогосподарських шкідників вирішено проектувати та розробляти як систему, що матиме розподілену архітектуру, у якій модель глибокого навчання буде виконуватись на хмарному сервері. Водночас взаємодія з користувачем буде організована через мобільний застосунок, що працюватиме як тонкий клієнт. Мобільний додаток буде забезпечувати отримання зображення з камери, передачу його на backend-платформу, отримання відповіді від сервера з результатами та відображення їх користувачу.

Архітектура система пропонується з чотирьох компонентів: серверу, сервісу моделі комп'ютерного зору, мобільного застосунку та зовнішніх сервісів для збереження файлів зображень та бази даних. На рисунку 1 показана діаграма компонентів запропонованої системи.

В якості базової моделі машинного навчання використано YOLOv11-N. Зважаючи на те, що сама модель була розроблена на основі пакету Ultralytics в екосистемі мови Python, доцільно розділити їх за середовищем виконання.

Серверна частина пропонується з двох сервісів:

- серверу – основного компоненту, що відповідає за валідацію вхідних даних, авторизацію, збереження даних тощо;
- сервісу моделі комп'ютерного зору, що передає вхідне зображення моделі та повертає результати її роботи.

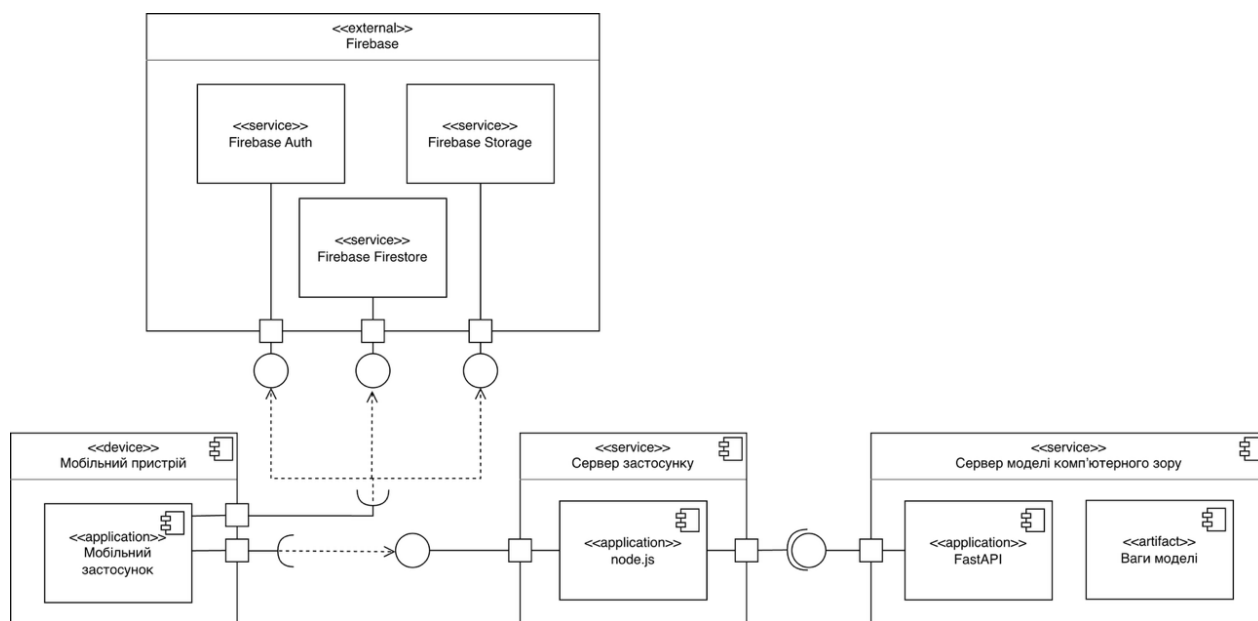


Рис. 1. Діаграма компонентів системи

Для реалізації серверу було обрано середовище node.js з версією двигуна 22, фреймворк серверних застосунків express версії 5, мову програмування TypeScript версії 5.7. Використання саме node.js дозволяє ефективно використовувати ресурси машини, де запускається серверний застосунок, на основі архітектури середовища.

Для обробки мережових запитів та роботи з файлами використані бібліотеки axios та multer. Для сервісу моделі використано мову програмування Python версії 3.12, пакет ultralytics версії 8.3.91 та фреймворк серверних застосунків FastAPI. Для використання моделі її ваги були експортовані в окрему директорію models в кореневій директорії сервісу. Після цього в процесі роботи модель YOLOv11-N ініціалізується на основі цих ваг.

Для серверу та сервісу моделі реалізовані методи валідації вхідних запитів. Розмір файлів зображень обмежений 10-ма мегабайтами, а файл зображення повинен мати розширення jpeg (jpg), webp або png. Після того, як запит пройшов валідацію, зображення зберігається в директорію завантажень, а модель ініціалізується на основі заздалегідь експортованих ваг. Далі в модель передається шлях до файлу, після чого вона обробляє зображення і відповідає набором результатів. Кожен результат містить мітку класу, назву, впевненість у відсотках та координати обмежувальної коробки, у відсотках відносно лівого верхнього кута зображення.

В архітектурі системи також пропонується використання зовнішніх сервісів для збереження файлів зображень та бази даних. Від платформи Firebase використовуються три сервіси: Auth для функцій авторизації та реєстрації, Storage для збереження файлів зображень в хмарному сховищі та Firestore для управління керуємими базами даних. Від платформи GoogleCloudPlatform використовується сервіс хостингу CloudRun, що дозволяє запускати зображення застосунків (англ. container image) серверної частини по запиту в залежності від навантаження на систему. При цьому сервіс, який містить ваги та запускає модель комп'ютерного зору, відділений для того, щоб запобігти можливому зловживанню, наприклад, DDoS-атакам або несанкціонованому доступу до моделі.

Загальна послідовність взаємодії під час обробки зображення між користувачем, мобільним додатком, сховищем (Firebase SDK), сервером додатку та сервером моделі показана на рисунку 2.

Для розгортання серверної частини було використано технологію контейнеризації Docker та платформу Google Cloud Run, що дозволяє запускати сервіси та адаптувати кількість запущених сервісів відповідно до навантаження системи. Технологія контейнеризації Docker, що дозволяє «запакувати» вихідний код застосунку разом з його залежностями та ресурсами у портативне зображення, що можна запускати на вимогу, в залежності від потреби. Для кожного застосунку був створений файл конфігурації Docker, що вказує залежності середовища,

наприклад, операційну систему та версію двигуна або мови, та залежності застосунку, тобто, пакети. Також в цьому файлі вказується механізм збірки та запуску застосунку.

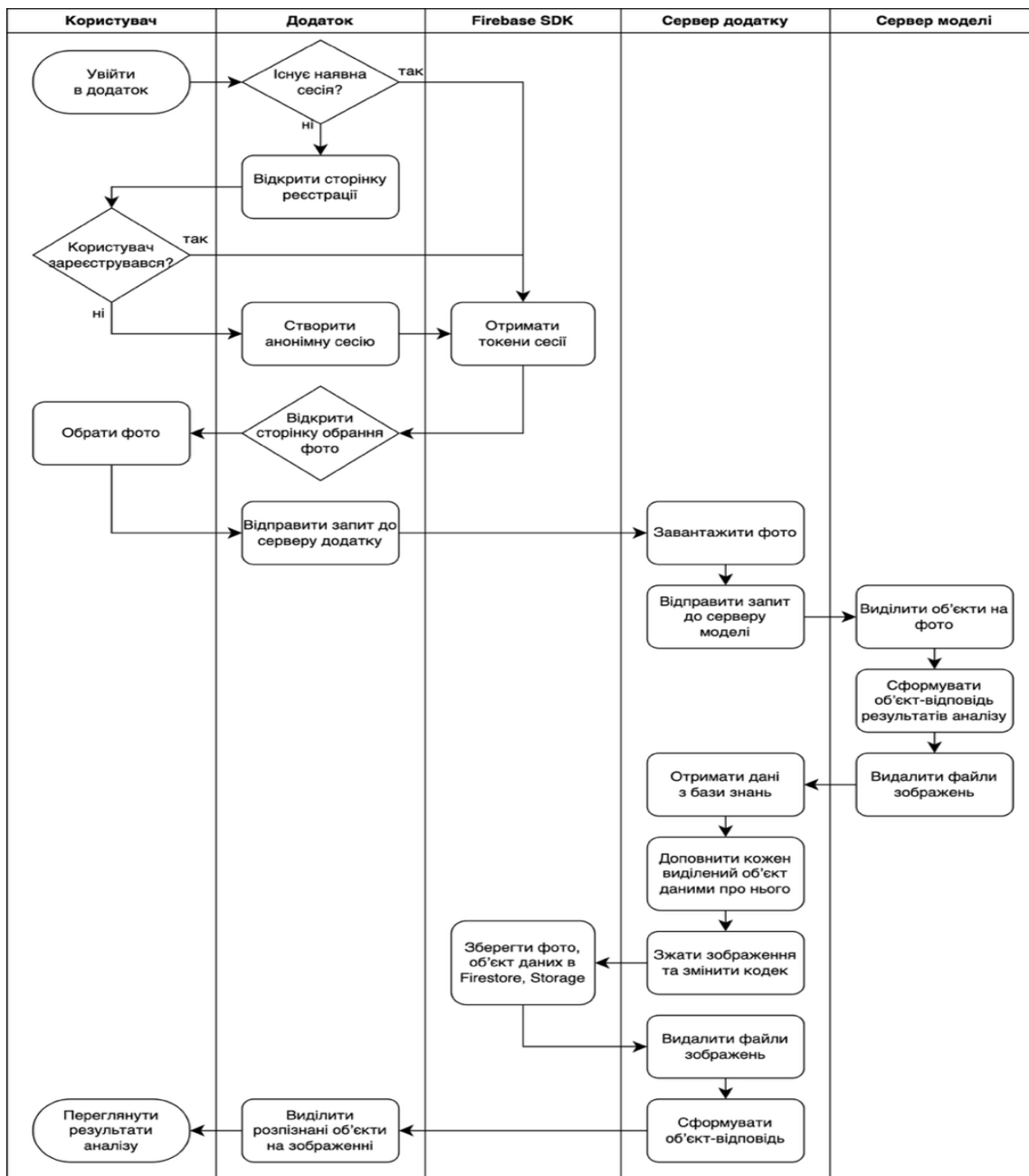


Рис. 3. Послідовність взаємодії елементів системи для ідентифікації об'єкта

Для того, щоб автоматизувати розгортання для сервісів, що розміщуються в мережі інтернет за допомогою сервісу CloudRun, існує декілька способів, а саме:

- самостійне створення Docker-зображень, їх завантаження у реєстр зображень Google (англ. Google Artifact Registry) та публікація змін у Cloud Run;
- автоматизоване рішення з використанням сервісу CloudBuild, що самостійно запускає заздалегідь налаштовані процеси розгортання.

Було прийнято рішення використати другий підхід, оскільки він простіший у налаштуванні і потребує створення меншої кількості файлів конфігурації. Налаштування для створення зображення застосунку описуються в одному файлі разом з ідентифікаторами проєкта, а процес збірки запускається після відправки іменованого тегу у віддалений репозиторій. Після того, як зображення застосунку було успішно зібране і опубліковане в реєстрі контейнерів, запускається окрема дія, що публікує цю версію в відповідному сервісі

CloudRun. Таким чином, досягається автоматизація процесу керування версіями, збірки та розгортання сервісів сервісної частини.

Після реалізації серверної частини та автоматизації її розгортання було проведено проектування та розроблення мобільного застосунку. Для мобільного застосунку було обрано мову програмування Kotlin 2.1.20 та фреймворк Jetpack Compose як найсучасніші інструменти розробки для ОС Android. Для створення мережевих запитів використовується бібліотека ktor, для серіалізації – serialization, для роботи із зображеннями – coil та zoomable. В застосунку реалізовані функції авторизації та реєстрації користувача, збереження та перегляду історії та відображення даних про ідентифікованих шкідників. Користувач має доступ до створення та завантаження фото, проведення аналізу фото, перегляду інформації про ідентифікованих шкідників, а також перегляду історії аналізів та бази шкідників. На рисунку 3 показані приклади результату роботи системи ідентифікації: а) сторінка історії із записами, б) сторінка попереднього перегляду обраного зображення, в) сторінка результатів аналізу.

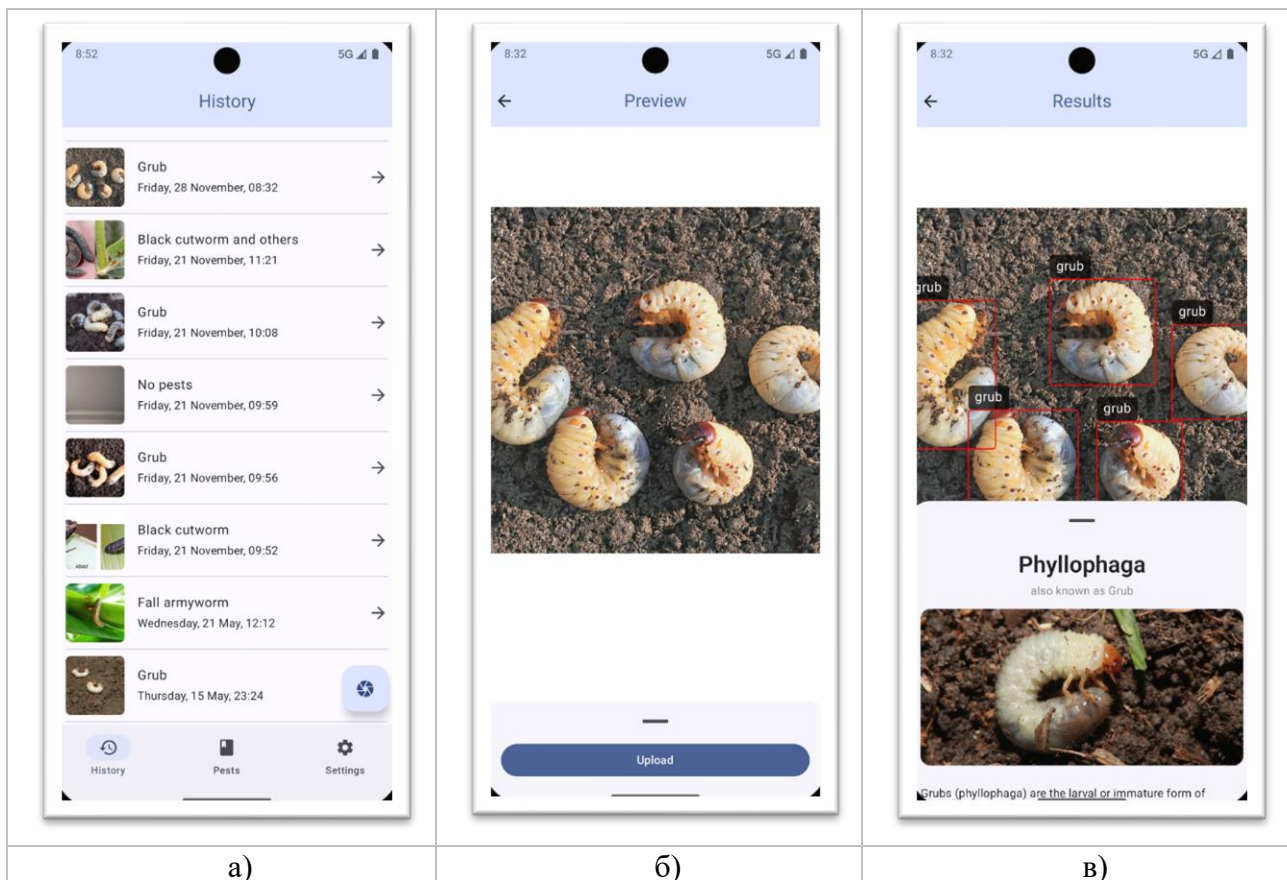


Рис. 4. Приклад екранів розробленого мобільного застосунку.

Висновки

У результаті проведеного дослідження було розглянуто особливості проектування інтелектуальних систем ідентифікації об'єктів із використанням мобільних застосунків та хмарних обчислень. Проведений аналіз сучасних наукових підходів показав, що існують два основні напрями реалізації таких систем – виконання обчислень безпосередньо на мобільному пристрої та винесення обробки даних у хмарну інфраструктуру. Встановлено, що кожен із підходів має свої переваги та обмеження, проте хмарно-орієнтована архітектура забезпечує кращу масштабованість, гнучкість і можливість централізованого оновлення моделей.

У роботі запропоновано архітектуру системи ідентифікації сільськогосподарських шкідників, яка базується на розподіленому підході з використанням мобільного застосунку як клієнтської частини та серверної інфраструктури для виконання обчислювально складних задач. Розроблена структура системи включає сервер додатку, сервіс моделі комп'ютерного

зору, мобільний клієнт і зовнішні хмарні сервіси для збереження даних та керування даними користувачів. Такий підхід дозволяє ефективно розподіляти обчислювальні ресурси, зменшити навантаження на мобільний пристрій і підвищити енергоефективність системи.

Практична реалізація підтвердила доцільність використання моделі глибокого навчання YOLO у поєднанні з хмарною інфраструктурою. Застосування контейнеризації та сервісів автоматизованого розгортання забезпечує адаптивність системи до змін навантаження та спрощує процес її підтримки і масштабування.

Запропоноване рішення може бути використане для створення ефективних систем моніторингу у сільському господарстві, зокрема для оперативного виявлення шкідників і підтримки прийняття рішень. Перспективами подальших досліджень є оптимізація затримок передачі даних, впровадження гібридних підходів із частковою обробкою на пристрої, а також розширення функціональних можливостей системи шляхом інтеграції аналітичних модулів та прогнозування.

Список використаної літератури:

1. Форкун Ю., Мартинюк В., Яшина О. Метод розробки та проєктування архітектурної складової програмного застосунку. *Measuring and Computing Devices in Technological Processes*. – 2023. – №4. – С.87–93. – DOI: <https://doi.org/10.31891/2219-9365-2023-76-11>
2. Бешта В. С., Комаричев А.В., Філімончук Т. В., Бараней Д. І. Модель мобільного додатку, яка орієнтована на обробку даних. Системи управління, навігації та зв'язку. Збірник наукових праць. – 2024. – Том 3, № 77. – С. 80–87. – DOI: <https://doi.org/10.26906/SUNZ.2024.3.080>
3. Semerikov S. et al., Models and Technologies for Autoscaling Based on Machine Learning for Microservices Architecture, *Proceedings of the 8th International Conference on Computational Linguistics and Intelligent Systems. Volume I: Machine Learning Workshop*, – 2024. –P. 316-330. – Електронний ресурс. – Режим доступу: <https://ceur-ws.org/Vol-3664/paper22.pdf>
4. Mamun, S. B., Payel, I. J., Ahad, M. T., Atkins, A. S., Song, B., Li, Y. Grape Guard: A YOLO-based mobile application for detecting grape leaf diseases, *Journal of Electronic Science and Technology*. –2025.– Vol. 23. – Issue 1. 100300. –DOI: <https://doi.org/10.1016/j.jnlest.2025.100300>.
5. Zheng, H., Liu, C., Zhong, L., Yin, X., Zhang, H., Zhu, X., Yang, G., Zhu, Y. An android-smartphone application for rice panicle detection and rice growth stage recognition using a lightweight YOLO network. – *Frontiers in Plant Science*. – 2025. – Vol. 16. – Article 1561632. – DOI: <https://doi.org/10.3389/fpls.2025.1561632>
6. Yulita I.N., Rambe M.F.R., Sholahuddin A., Prabuwno A.S. A Convolutional Neural Network Algorithm for Pest Detection Using GoogleNet. – *AgriEngineering*. – 2023. – Vol. 5, No. 4. – P. 2366–2380. – DOI: <https://doi.org/10.3390/agriengineering5040145>
7. Saeed A.T., Schonfeld D. Cloud-Based ImageNet Object Recognition for Mobile Devices. *Proceedings of IMDC-SDSP 2020*. – 2020. – DOI: <http://dx.doi.org/10.4108/eai.28-6-2020.2297916>
8. Karar M.E., Alsunaydi F., Albusaymi S., Alotaibi S. A new mobile application of agricultural pests recognition using deep learning in cloud computing system. *Alexandria Engineering Journal*. – 2021. – Vol. 60, Is. 4. – P. 4067–4082. – DOI: <https://doi.org/10.1016/j.aej.2021.03.009>
9. Nugroho E.D., Verdiana M., Algifari M.H., Pramudita A.S., Adi S. Development of YOLO-Based Mobile Application for Detection of Defect Types in Robusta Coffee Beans. *Journal of Applied Informatics and Computing*. – 2025. – Vol. 9, No. 1. – P. 153–160. DOI: <https://doi.org/10.30871/jaic.v9i1.8886>
10. Chen, J.-W., Lin, W.-J., Cheng, H.-J., Hung, C.-L., Lin, C.-Y., & Chen, S.-P. A Smartphone-Based Application for Scale Pest Detection Using Multiple-Object Detection Methods. *Electronics*. – 2021. – 10(4). – P. 372. – DOI: <https://doi.org/10.3390/electronics10040372>
11. Wong M.O., Abubacker N.F. YOLO-Driven Lightweight Mobile Real-Time Pest Detection and Web-Based Monitoring for Sustainable Agriculture. *International Journal of Advanced Computer Science and Applications*. – 2024. – Vol. 15, No. 12. – 12 p. DOI: <https://dx.doi.org/10.14569/IJACSA.2024.0151267>

12. Бердник, Ю. М., & Скотаренко, А. О. До проблеми розпізнавання об'єктів на пристроях з обмеженими ресурсами. Проблеми програмування. – 2024. – 4 – С. 14-22. – DOI: <https://doi.org/10.15407/pp2024.04.014>

13. Martinez-Alpiste, I., Golcarenenrenji, G., Wang, Q., & Alcaraz-Calero, J. M. Smartphone-based real-time object recognition architecture for portable and constrained systems. Journal of Real-Time Image Processing. – 2021. – Vol. 19. – P. 103–115 – DOI: <https://doi.org/10.1007/s11554-021-01164-1>

Автори статті

П'ятикоп Олена – кандидат технічних наук, доцент, ДВНЗ «Приазовський державний технічний університет», Дніпро, Україна.

ORCID: 0000-0002-7731-3051

Шевченко Артем – магістр, ДВНЗ «Приазовський державний технічний університет», Дніпро, Україна.

ORCID: 0009-0007-5574-5148

Левицька Тетяна – кандидат технічних наук, доцент, ДВНЗ «Приазовський державний технічний університет», Дніпро, Україна.

ORCID: 0000-0003-3359-1313

Authors of the article

Piatykop Olena – Candidate of Sciences (technical), Associate Professor, SHEI “Pryazovskyi State Technical University”. Dnipro, Ukraine

ORCID: 0000-0002-7731-3051

Shevchenko Artem – master, SHEI “Pryazovskyi State Technical University”. Dnipro, Ukraine

ORCID: 0009-0007-5574-5148

Levytska Tetiana – Candidate of Sciences (technical), Associate Professor, SHEI “Pryazovskyi State Technical University”. Dnipro, Ukraine

ORCID: 0000-0003-3359-1313

Надійшла до редакції: 18.04.2026

Прийнята до друку: 27.04.2026

Опубліковано: 25.05.2026

© 2026 П'ятикоп О.Є., Шевченко А.Є., Левицька Т.О.

Цей матеріал ліцензовано за умовами CC BY 4.0. <https://creativecommons.org/licenses/by/4.0>