



МОБІЛЬНИЙ ЗАСТОСУНОК НА БАЗІ YOLO ДЛЯ ІДЕНТИФІКАЦІЇ СІЛЬСЬКОГОСПОДАРСЬКИХ ШКІДНИКІВ

MOBILE APPLICATION BASED ON YOLO FOR THE DETECTION OF AGRICULTURAL PESTS

¹П'ятикоп О. Є., ²Шевченко А. Є.¹Piatykor O. Ye., ²Shevchenko A. Ye.^{1,2} ДВНЗ «Приазовський державний технічний університет», Дніпро, Україна^{1,2} SHEI "Pryazovskyi State Technical University", Dnipro, UkraineORCID: ¹<https://orcid.org/0000-0002-7731-3051>, ²<https://orcid.org/0009-0007-5574-5148>E-mail: ¹piatykor_o_ye@pstu.edu, ²shevchenko_a_y@students.pstu.edu

Copyright © 2026 by author and the journal "Automation of technological and business – processes".

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0>

DOI: 10.15673/atbp.v18i2.3454

Анотація. Дослідження спрямоване на розв'язання проблеми ідентифікації сільськогосподарських шкідників шляхом створення мобільного застосунку, що використовує моделі згорткових нейронних мереж.

У роботі проведено аналіз сучасних наукових досліджень, присвячених ідентифікації сільськогосподарських шкідників із використанням згорткових нейронних мереж, візуальних трансформерів та, зокрема, моделей сімейства YOLO. Аналіз літератури показав, що найефективнішим підходом є використання YOLO-моделей завдяки їх високій швидкодії та конкурентній точності, в тому числі в реальному часі. Окрему увагу приділено дослідженням, у яких моделі YOLO інтегруються у мобільні застосунки та системи моніторингу, що підтверджує практичну доцільність такого підходу.

Для навчання моделі використано підмножину набору даних IP102, що містить шість класів поширених шкідників. З метою підвищення якості ідентифікації було застосовано методи розширення тренувальної вибірки шляхом трансформації зображень, зокрема перевертання, зміщення відтинку та розмиття. Проведено порівняльний аналіз двох моделей YOLOv11-N, навчених на початковій та розширеній вибірках, за метриками влучності, повноти, mAP50 та mAP50-95. Результати експериментів показали, що використання розширеної вибірки дозволяє суттєво покращити всі основні показники якості моделі, зокрема підвищити mAP50 з 81,5% до 95,9%. Для практичного використання моделі розроблено розподілену систему з хмарною серверною частиною та мобільним застосунком-клієнтом, що забезпечує зручну взаємодію користувача з моделлю та доступ до результатів ідентифікації. Запропоноване рішення демонструє практичність використання легковагових моделей сімейства YOLO у поєднанні з мобільними інтерфейсами для задач ідентифікації сільськогосподарських шкідників у реальному часі.

Abstract. This study aims to address the problem of agricultural pest detection by developing a mobile app using convolutional neural network models.

This paper analyzes current scientific research on agricultural pest detection using convolutional neural networks, visual transformers, and, in particular, models from the YOLO family. The literature review has shown that the most effective approach is to use YOLO models due to their high performance and competitive accuracy, including in real time. Particular attention is paid to studies integrating YOLO models into mobile apps and monitoring systems, confirming the practical feasibility of this approach.

A subset of the IP102 dataset containing six classes of common pests was used to train the model. To improve identification quality, methods for expanding the training set were applied through image transformations, specifically, flipping, hue shifting, and blurring. A comparative analysis of two YOLOv11-N models trained on the initial and expanded datasets was conducted using the metrics of precision, recall, mAP50, and mAP50-95. The experimental results showed that using the expanded dataset significantly improved all key model performance metrics, specifically increasing mAP50 from 81.5% to 95.9%.

For practical use, a distributed system with a cloud-based server and a mobile client application was developed, ensuring convenient user interaction with the model and access to identification results. The proposed solution



demonstrates the feasibility of using lightweight YOLO family models in combination with mobile interfaces for real-time pest identification tasks.

Ключові слова: ідентифікація шкідників, комп'ютерний зір, машинне навчання, згорткові нейронні мережі, YOLO, мобільний застосунок, Android, обробка зображень, розподілені системи, mAP50, Kotlin, Python.

Keywords: pest detection, computer vision, machine learning, convolutional neural networks, YOLO, mobile application, Android, image processing, distributed systems, mAP50, Kotlin, Python.

Вступ. Однією з важливих цілей сталого розвитку є подолання голоду та розвиток сільського господарства. Беручи до уваги сучасні виклики цієї галузі, зокрема кліматичні зміни та втрату біорізноманіття [1], все більшою загрозою стають сільськогосподарські шкідники. Сучасні дослідження показують, що навіть застосування найновіших засобів контролю шкідників, наприклад, нанопестицидів, є шкідливими в довготривалій перспективі [2]. Натомість, більш ефективним за витратами часу та коштів виявляються превентивні методи ідентифікації шкідників. Актуальним постає вирішення задачі з використанням сучасних інформаційних технологій, методів штучного інтелекту та машинного навчання [3].

Аналіз літературних даних і постановка проблеми. Перед розробкою системи постало питання створення моделі ідентифікації шкідників. Для цієї задачі частіше за все використовують моделі на основі нейронних мереж – залишкових з увагою [4], згорткових [5, 6], глибоких згорткових [7], а також їх комбінацій із зоровими трансформерами [8].

З великою ефективністю застосовується сім'я моделей YOLO. В дослідженні [9] проводиться порівняння різних версій моделі YOLO, де найкращою за параметрами точності виділення об'єктів та кількістю параметрів стала YOLOv8-M. Сплетіння зображень та додавання випадкових точок дозволило покращити точність розпізнавання маленьких шкідників. В роботі [10] описується застосування методів покращення зображень шкідників та виділення ознак для подолання проблем оклюзії та перекриття. В результаті було отримано модель Pest-YOLO, що досягла 77.71% чутливості. В статті [6] використання модулів перехресних часткових мереж CSPResNeXt-50 та VoVGSCSP дозволило досягти чутливості 77.3%, при цьому зберігаючи швидкість ідентифікації.

При використанні моделі важливим є надання інтерфейсу для взаємодії з нею. В [11–13] розглядається інтеграція моделей ідентифікації шкідників в мобільні застосунки із застосуванням хмарних технологій. В роботі [11] запропонований додаток використовує швидку згорткову нейронну мережу на основі регіонів (Faster R-CNN) для виконання завдання розпізнавання комах-шкідників на основі хмарних обчислень. Це дослідження було проведено для п'яти груп шкідників: попелиці, цикадельки, брунькоподібні черви, блішки та червоні павуки. Запропонований Faster R-CNN показав найвищу точність розпізнавання – 99,0% для всіх протестованих зображень шкідників. В роботі [12] використано модель згорткової нейронної мережі (CNN), зокрема GoogleNet, яка показала точність 93,78%. Автори використовували попередню обробку даних: застосування нахилу до зображення, збільшення зображення та горизонтальне відображення зображення. Процес було здійснено для розширення різноманітності зображень. Мобільний застосунок був запропонований для індонезійських фермерів та адаптований для їх регіону.

В роботі [13] для виявлення та локалізації щитівок на зображенні досліджено моделі виявлення об'єктів на основі глибокого навчання, такі як швидші згорткові мережі на основі регіонів (Faster R-CNN), одноразові багатоблокові детектори (SSD) та You Only Look Once v4 (YOLO v4). У своїй роботі автори провели доповнення даних шляхом геометричного перетворення обертання та горизонтального відображення початкових зображень. Автори відзначили, що найкращі результати показала остання модель, яку впровадили у мобільний застосунок, що орієнтований для регіонів Тайвань.

У статті [14] представлено систему виявлення шкідників, яка використовує легку платформу глибокого навчання YOLO та інтегрована з веб-платформою моніторингу. Архітектури виявлення об'єктів YOLO, включаючи YOLOv8n, YOLOv9t та YOLOv10-N, були вивчені та оптимізовані для виявлення шкідників на смартфонах. Моделі були навчені та перевірені за допомогою об'єднання публічних наборів даних, що містять 29 класів шкідників. Серед них YOLOv9t досягає найвищої продуктивності зі значенням mAP@0.5 89,8%, точністю 87,4%, частотою повторення 84,4% та часом виведення 250,6 мс.

Також модель YOLO добре себе показала при розробці мобільних застосунків для розпізнавання та виявлення інших об'єктів аграрної сфери: типів дефектів у кавових зернах робусти [15], хвороби виноградного листя [16], рисових волоток та розпізнавання стадії росту рису [17].

В [18] практично досліджується застосування мікросервісної архітектури для побудови стійких комплексних систем підтримки прийняття рішень із застосуванням моделей машинного навчання. На основі цього для розробки таких систем доцільно застосувати підхід, де користувацькі застосунки спілкуються з центральним сервером, який в свою чергу напруму комунікує із внутрішніми мікросервісами.

На основі розглянутої літератури можна зробити висновок, що частіше за все для задач ідентифікації, визначення та сегментації успішно використовують візуальні трансформери та згорткові нейронні мережі, в тому числі на основі моделі YOLO. Створення моделі на основі YOLO в комбінації з різними методами трансформації даних дозволяє досягати високих результатів точності та затримки обробки кадрів в реальному часі. Дослідження робіт інших авторів показало практичність поєднання легковагових YOLO-моделей із мобільними інтерфейсами, що важливо для ідентифікації сільськогосподарських шкідників в реальному часі. Запропоновані авторами



мобільні додатки орієнтовані на шкідників певних регіонів, тому для ідентифікації інших видів необхідні ще дослідження.

Мета і завдання дослідження. Метою статті є дослідження моделі машинного навчання на базі YOLOv11-N для ідентифікації сільськогосподарських шкідників та розробка мобільного застосунку для взаємодії з нею.

Методи і матеріали досліджень. Всі версії моделей сім'ї YOLO складаються з трьох основних компонентів: хребту (англ. backbone), шиї (англ. neck) та голови (англ. head). Хребет відповідає за виділення ознак із вхідних даних і зазвичай є іншою ЗНН, що натренована на великих вибірках зображень [19]. Часто в якості хребту застосовуються моделі ResNet50, ResNet101 та CSPDarkNet53.

Після проходження хребта ознаки переходять в шию, що обробляє та очищує їх. Часто в шиї застосовуються техніки пірамідальної мережі ознак (англ. Feature Pyramid Network, FPN) та модулів просторової уваги (англ. Spatial Attention Modules, SAM) [20].

Заключним компонентом є голова, де виконується власне передбачення обмежувальних коробок та ймовірностей приналежності до конкретного класу на основі комбінованих ознак шиї. Для цього використовуються якорі (англ. anchor boxes) – набір наперед визначених обмежувальних коробок об'єктів вибірки, що створюється під час тренування моделі – для ефективного визначення об'єктів з різними масштабами та формами.

Під час роботи YOLO розділяє вхідне зображення на таблицю та створює передбачення обмежувальних коробок та класових ймовірностей одночасно. Результати передбачення генеруються на основі згорткових шарів, після чого відбувається виділення ознак за допомогою ЗНН. Фінальні результати обробляються алгоритмом Non-Maximum Suppression (укр. придушення не-максимумів), що має на меті злиття перекриваючих одна одну обмежувальних коробок з наданням пріоритету такій з максимальною впевненістю [21].

Для навчання моделі була підготовлена вибірка на основі існуючої великої вибірки IP102 [22]. В склад вибірки увійшли шість класів: кукурудзяна совка (armyworm), совка-іпсилон (black cutworm), капустянка (mole cricket), червоний павутинний кліщ (red spider), хрущ (grub) та моль персикова (peach borer). Спочатку вибірка налічувала з 779 унікальних зображень.

Щоб дозволити моделі краще вивчити конкретний клас без прив'язки до кольору та симетрії індивідуального об'єкту вибірку було розширено за допомогою попередньої трансформації тренувальних даних. Так для кожного зображення було застосовано три види покращення: перевертання, зміщення відтінку та середнє розмиття. Для визначення значень зміщення відтінку та розмиття використовувались випадкові числа. Значення зміщення відтінку знаходились у проміжку від 2 до 8, розмиття – від 2 до 5. Для виконання трансформацій використовувалась бібліотека OpenCV та мова Python. Таким чином, було створено вибірку 3899 зображень шості видів шкідників.

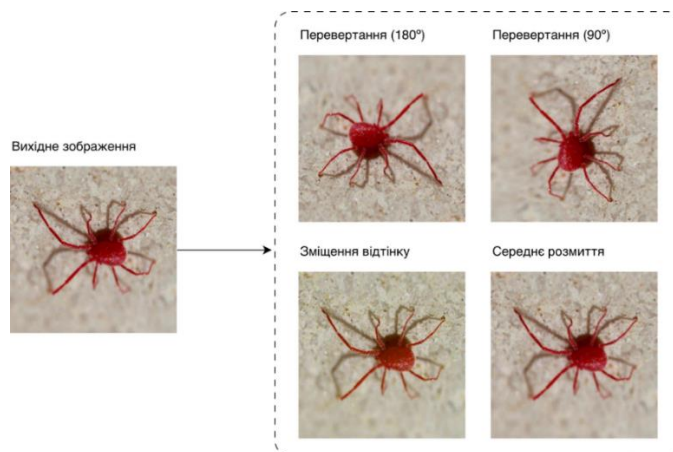


Рис. 1 Трансформації, що застосовуються до зображення

Fig. 1 Transformations applied to the image

Приклад трансформацій, застосованих до одного об'єкту, наведений на рисунку 1.

Для перевірки впливу включення трансформованих зображень на якість ідентифікації було натреновано дві окремих моделі YOLOv11-N. Перша модель тренувалася на повному тренувальному наборі, включаючи зображення з усіма видами трансформацій. Друга модель тренувалася тільки на оригінальних зображеннях.

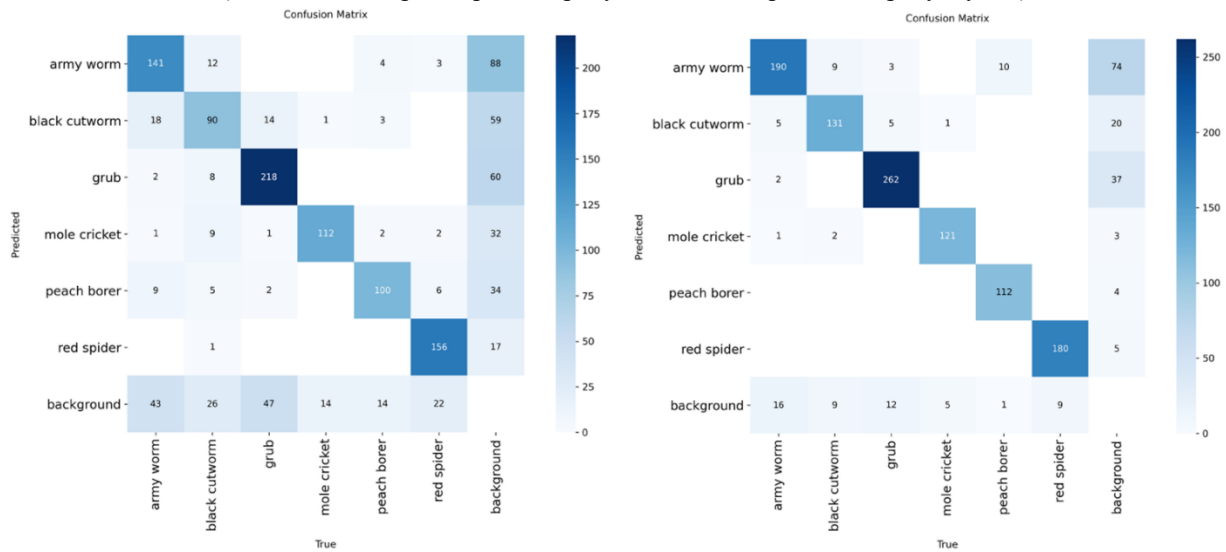
Далі обидві моделі оцінювалися за допомогою метрик влучності, повноти, mAP50 та mAP50-95 на однаковій валідаційній підвбірці для всіх класів. В таблиці 1 наведені результати порівняння. Порівняння показало, що розширення вибірки трансформованими зображеннями дозволило моделі покращити результати роботи відносно усіх основних метрик.

Таблиця 1 – Порівняння різних метрик моделей для всіх класів зі початковою та розширеною тренувальною вибіркою

Клас	Влучність	Повнота	mAP50	mAP50-95
Початкова вибірка	0.813	0.755	0.815	0.535
Розширена вибірка	0.936	0.915	0.959	0.712



Додатково були побудовані матриці невідповідностей для кожної моделі та криві співвідношення влучності-повноти та міри F1-впевненості. Матриця для моделі зі звичайною тренувальною вибіркою наведена на рисунку 2 а), для моделі із розширеною тренувальною вибіркою – на рисунку 2 б).



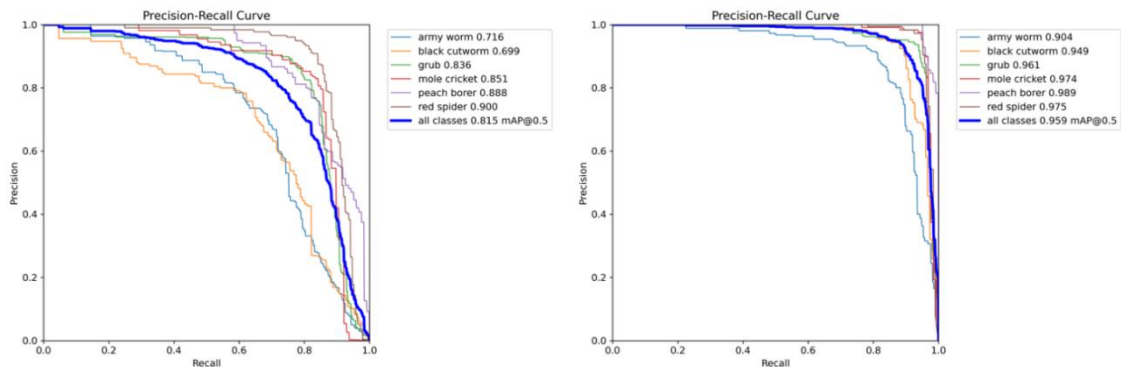
а) матриця невідповідностей моделі зі початковою вибіркою; б) матриця невідповідностей моделі із розширеною вибіркою

Рис. 2 Матриця невідповідностей
Fig. 2 Confusion matrix

Порівнюючи обидві матриці, можна побачити, що модель, яка тренувалася на зображеннях з трансформаціями, менше помиляється у визначенні класу виділеного об'єкту. Для класу armyworm (кукурдзяної совки) похибка зменшилася незначно, що вказує на необхідність збільшення кількості зображень або додавання можливості змінювати відтінок зображення в іншу сторону.

Окремим класом у матриці виділене фонове зображення (background). Матриця на рисунку 2 б) показує суттєве зменшення хибнонегативних випадків, де модель не ідентифікує жоден об'єкт. Також спостерігається менш суттєве зменшення хибнопозитивних випадків, де модель ідентифікує фонове зображення як якийсь з класів. Можна припустити, що додавання у перелік трансформацій зменшення контрастності може допомогти моделі краще відділяти фон і тренувальні класи.

Далі було проаналізовано графіки кривих співвідношення влучності та повноти, що наведено на рисунку 3.



а) результати моделі з початковою вибіркою;

б) результати моделі з розширеною вибіркою

Рис. 3 Графіки кривих співвідношення влучності та повноти
Fig. 3. Graphs of the curves of the relationship between precision and recall

Як видно з рисунку 3, включення зображень з трансформаціями покращило влучність та повноту для всіх класів вибірки. Також це дозволило моделі знизити кількість неправильних передбачень класів armyworm (кукурдзяної совки) та black cutworm (совки-іпсилон), де модель плутала мітки.

Після навчання та дослідження моделі була проведена її інтеграція у систему ідентифікації з клієнтським інтерфейсом у вигляді мобільного застосунку. Інтелектуальну систему ідентифікації було вирішено розробляти, як систему з розподіленою архітектурою. В цьому випадку модель глибокого навчання тренується та виконується на хмарному сервері, а для інтерфейсу користувачу розроблено мобільний застосунок, що працює як тонкий клієнт. Мобільний додаток забезпечує отримання зображення з камери, передачу його на backend-платформу, отримання відповіді від сервера з результатами та відображення їх користувачу. Таким чином, буде можливість доступу до результатів аналізу з декількох інтерфейсів включно з мобільним застосунком.

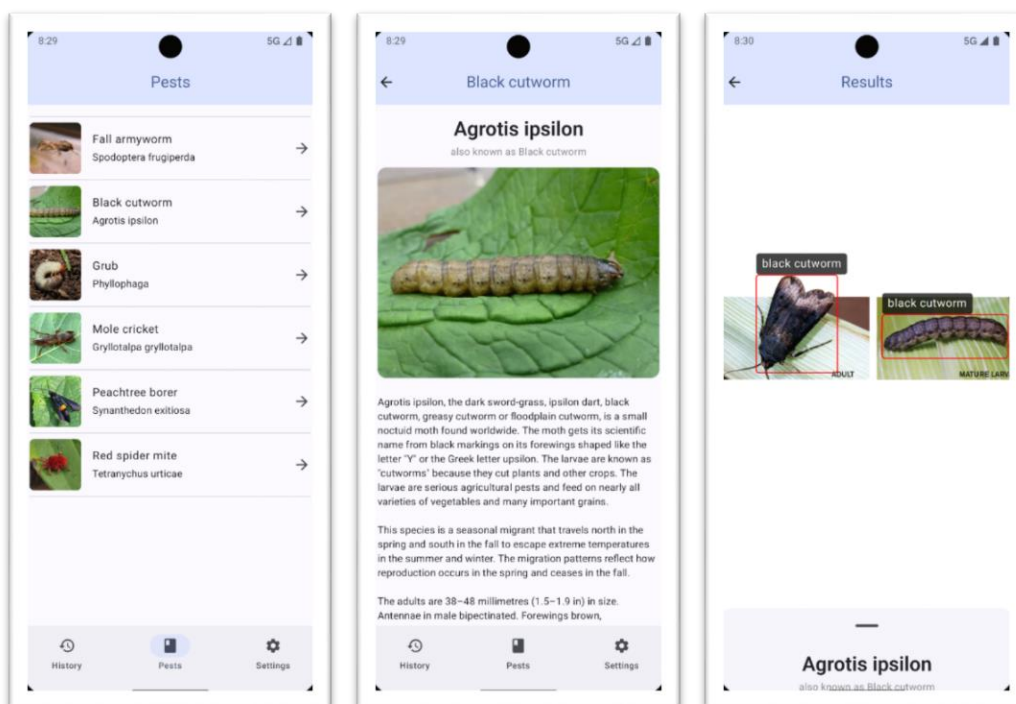


Для розробки серверної частини використано два різних середовища: для серверу застосунку – node.js з бібліотекою express та мовою TypeScript, для сервісу моделі – фреймворк FastAPI та мову Python, оскільки він і запускає модель. Для запитів була реалізована логіка валідації файлового розширення та розміру зображень.

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.

Мобільний застосунок було розроблено для операційної системи Android з використанням фреймворку JetpackCompose та мови програмування Kotlin версії 2.1.20. Для створення мережових запитів використовується бібліотека ktor, для серіалізації – serialization, для роботи із зображеннями – coil та zoomable.

В застосунку реалізовані функції: авторизації та реєстрації користувача, проведення ідентифікації шкідників, відображення результатів, збереження та перегляду історії ідентифікацій, перегляд бази даних шкідників та інформації про них. Для збереження цих даних використано платформу Firebase. Для кожного шкідника була сформована база даних, в якій зберігається його науковий вид, опис, способи виведення, а також відносна шкідливість. На рисунку 5 а) наведено приклад екрану, який виводить усі записи про шкідників. Це ті самі класи, на яких тренувалася модель. При натисканні на плитку шкідника відкривається сторінка опису, на якій наводиться його назва, науковий вид, зразкове зображення, опис його шкоди, життєвого циклу та способів виведення. Приклад екрану наведено на рисунку 5 б).



а)

б)

в)

Рис. 5 Приклад екранів розробленого мобільного застосунку.

Fig. 5 Example of screens of the developed mobile application.

Ідентифікація шкідників в мобільному застосунку виконується в реальному часі по фото, яке можна завантажити з галереї пристрою або зробити за допомоги камери пристрою. Після отримання фото його можна переглянути перед аналізом. Для відображення фото використовується універсальний компонент, що дозволяє наближувати фото. Аналіз виконується натисканням відповідної кнопки в нижній частині екрану. Після натискання на неї виконується мережовий запит, отримуються результати ідентифікації, які виводяться на екрані, як наведено на рисунку 5 в). Всі результати ідентифікації записуються в колекцію Firestore та відображається в історії.

Висновки

Було зібрану вибірку шкідників, що складається з 779 зображень та містить шість класів: кукурудзяну совку та совку-іпсилон, капустянку червоного павутинного кліща, хруща та персикову моль. Кожне зображення було покращено з використанням трьох видів трансформацій: перевертання, зміщення відтинку та розмиття, що дозволило моделі вивчити конкретний клас без прив'язки до кольору та симетрії індивідуального об'єкту та досягти кращих результатів під час тренування. Експериментально перевірено, як модель розпізнає шкідників в різних життєвих стадіях та статях та у випадках часткового перекриття. Кінцева модель завдяки аугментації зображень досягла таких значень основних валідаційних метрик: 93.6% влучності, 91.5% повноти, 95.9% mAP50 та 71.2% mAP50-95 в середньому для всіх класів.

Для надання інтерфейсу для кінцевих користувачів для роботи з моделлю було створено мобільний застосунок для ОС Android та серверні компоненти.



Подальшими кроками в розробці цієї системи є: розширення кількості класів шкідників; оптимізація моделі; додавання ще одного інтерфейсу, наприклад, веб-сайту, для доступу до системи; реалізація механізму обліку та аналітики за конкретні проміжки часу для застосування в реальних умовах, наприклад, на полях або в складських приміщеннях.

Список використаних джерел

1. Vikas, Ranjan R. Agroecological approaches to sustainable development // *Frontiers in Sustainable Food Systems*. 2024. Vol. 8. DOI: 10.3389/fsufs.2024.1405409.
2. Kannan M. et al. Nanopesticides in agricultural pest management and their environmental risks: a review // *International Journal of Environmental Science and Technology*. 2023. Vol. 20, no. 9. P. 10507–10532. DOI: 10.1007/s13762-023-04795-y.
3. Кучмієва Т. С., Мороз Т. О., Шешунова А. В. Використання штучного інтелекту в сільському господарстві // *Modern Economics*. 2023. № 39. С. 69–74. DOI: [https://doi.org/10.31521/modecon.V39\(2023\)-10](https://doi.org/10.31521/modecon.V39(2023)-10).
4. Hassan S. M., Maji A. K. Pest Identification Based on Fusion of Self-Attention With ResNet // *IEEE Access*. 2024. Vol. 12. P. 6036–6050. DOI: 10.1109/ACCESS.2024.3351003.
5. Вакалюк Т., Фуріхата Д., Янчук В., Леут Н. Розробка та оцінка ефективності моделі розпізнавання рукописного тексту на основі згорткових нейронних мереж // *Вісник Херсонського національного технічного університету*. 2025. № 2. С. 67–75. DOI: 10.35546/kntu2078-4481.2025.2.2.8.
6. Yang S. et al. Maize-YOLO: A New High-Precision and Real-Time Method for Maize Pest Detection // *Insects*. 2023. Vol. 14, no. 3. P. 278. DOI: 10.3390/insects14030278.
7. Li H. et al. Plant disease and insect pest identification based on vision transformer // *Proceedings of the International Conference on Internet of Things and Machine Learning (IoTML 2021)*. SPIE, 2022. P. 6. DOI: 10.1117/12.2628467.
8. Xia W. et al. An ensemble learning integration of multiple CNN with improved vision transformer models for pest classification // *Annals of Applied Biology*. 2022. Vol. 182, no. 2. P. 144–158. DOI: 10.1111/aab.12804.
9. Tian Y., Wang S., Li E., Yang G., Liang Z., Tan M. MD-YOLO: Multi-scale Dense YOLO for small target pest detection // *Computers and Electronics in Agriculture*. 2023. Vol. 213. Art. 108233. DOI: 10.1016/j.compag.2023.108233.
10. Wen C. et al. Pest-YOLO: A model for large-scale multi-class dense and tiny pest detection and counting // *Frontiers in Plant Science*. 2022. Vol. 13. DOI: 10.3389/fpls.2022.973985.
11. 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, no. 5. P. 4423–4432. DOI: 10.1016/j.aej.2021.03.009.
12. Yulita I. N. et al. A Convolutional Neural Network Algorithm for Pest Detection Using GoogleNet // *AgriEngineering*. 2023. Vol. 5, no. 4. P. 2366–2380. DOI: 10.3390/agriengineering5040145.
13. 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. Vol. 10, no. 4. P. 372. DOI: 10.3390/electronics10040372.
14. 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. DOI: 10.14569/IJACSA.2024.0151267.
15. Nugroho E. D. 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.
16. 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, no. 1. Art. 100300. DOI: 10.1016/j.jnlest.2025.100300.
17. Zheng H. et al. 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. Art. 1561632. DOI: 10.3389/fpls.2025.1561632.
18. 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*. 2024. Vol. I. P. 316–330. URL: <https://ceur-ws.org/Vol-3664/paper22.pdf> (дата звернення: 2025).
19. Mahasin M., Dewi I. A. Comparison of CSPDarkNet53, CSPResNeXt-50, and EfficientNet-B0 Backbones on YOLO v4 as Object Detector // *International Journal of Engineering Science and Information Technology*. 2022. Vol. 2, no. 3. P. 64–72. DOI: 10.52088/ijesty.v2i3.291.
20. Chen H., Chen Z., Yu H. Enhanced YOLOv5: An Efficient Road Object Detection Method // *Sensors*. 2023. Vol. 23, no. 20. P. 8355. DOI: 10.3390/s23208355.
21. Ali M. L., Zhang Z. The YOLO Framework: A Comprehensive Review of Evolution, Applications, and Benchmarks in Object Detection // *Computers*. 2024. Vol. 13, no. 12. P. 336. DOI: 10.3390/computers13120336.
22. Pest-dataset Computer Vision Dataset [Electronic resource]. Roboflow, 2022. URL: <https://universe.roboflow.com/varshine/pest-dataset-13ph3> (date of access: 2025).

Отримана в редакції 24.04.2026. Прийнята до друку 05.05.2026. Розміщено в інтернеті 31 травня 2026.

Received 24 April 2026. Approved 05 May 2026. Available in Internet 31 May 2026