

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерних наук _____
(повна назва)

Кафедра _____ програмної інженерії _____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти _____ другий (магістерський) _____

Дослідження методів розпізнавання зображень для ідентифікації та класифікації
рослин
(тема)

Виконав:
здобувач _____ 2 _____ року навчання
групи ПЗм-23-1
_____ Дмитро КОШИЛЬ _____
(власне ім'я, прізвище)

Спеціальність 121 – Інженерія програмного
забезпечення
(код і повна назва спеціальності)

Тип програми _____ освітньо-наукова _____

Освітня програма Інженерія програмного забезпечення
(повна назва освітньої програми)

Керівник _____ доц. Ілона РЕВЕНЧУК _____
(посада, власне ім'я, прізвище)

Допускається до захисту
Зав. кафедри

_____ Кирило СМЕЛЯКОВ _____
(підпис) (власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерних наук
 Кафедра _____ програмної інженерії
 Рівень вищої освіти _____ другий (магістерський)
 Спеціальність _____ 121 – Інженерія програмного забезпечення
 Тип програми _____ освітньо-наукова програма
 Освітня програма _____ Інженерія програмного забезпечення
 (шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«____» _____ 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові _____ Кошилю Дмитру Володимировичу
 (прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів розпізнавання зображень для ідентифікації та класифікації рослин.

Затверджена наказом по університету від 15 квітня 2025 р. № 290 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 17 червня 2025 р.

3. Вихідні дані до роботи: дослідження дата-сетів для детекції рослин та порівняння їх на моделях YOLO8, YOLO11, технології PyCharm, YOLO8, YOLO11, VilagePlant, DocPlant

4. Перелік питань, що потрібно опрацювати в роботі: порівняти існуючі дата-сети, провести аналіз алгоритмів розпізнавання об'єктів, провести тестування на моделях, порівняти результати.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз предметної галузі і постановка задачі	19.05.2025	<i>виконано</i>
2	Назва за розділами теоретичного і практичного дослідження	20.05.2025	<i>виконано</i>
3	Назва за розділами теоретичного і практичного дослідження	21.05.2025	<i>виконано</i>
4	Підготовка до апробації результатів дослідження. Публікація матеріалів	22.05.2025	<i>виконано</i>
5	Підготовка пояснювальної записки	26.05.2025	<i>виконано</i>
6	Підготовка презентації та доповіді	27.05.2025	<i>виконано</i>
7	Перевірка на плагіат	30.05.2025	<i>виконано</i>
8	Нормоконтроль	1.06.2025	<i>виконано</i>
9	Рецензування	3.06.2025	<i>виконано</i>
10	Попередній захист	14.06.2025	<i>виконано</i>
11	Занесення диплома в електронний архів	17.06.2025	<i>виконано</i>
12	Допуск до захисту у зав. кафедри	18.06.2025	<i>виконано</i>

Дата видачі завдання: 15 квітня 2025 р.

Студент _____
(підпис)

Дмитро КОШИЛЬ
(власне ім'я, прізвище)

Керівник роботи _____
(підпис)

доц. Ілона РЕВЕНЧУК
(посада, власне ім'я, прізвище)

РЕФЕРАТ / ABSTRACT

Робота містить: 56 стор., 10 рис., 16 джерел, 5 додатків.

ГЛИБОКЕ НАВЧАННЯ, ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ, КЛАСИФІКАЦІЯ РОСЛИН, КОМП'ЮТЕРНИЙ ЗІР, МОБІЛЬНІ ДОДАТКИ, РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ, СЕГМЕНТАЦІЯ ЗОБРАЖЕНЬ, ТЕКСТУРНИЙ АНАЛІЗ.

Об'єктом дослідження є системи розпізнавання зображень для ідентифікації та класифікації рослин у реальних умовах.

Метою роботи є створення автоматизованої системи, яка дозволяє визначати вид рослини за зображенням її частин з використанням сучасних методів машинного навчання.

Методами розробки та проектування є застосування згорткових нейронних мереж (CNN), алгоритмів сегментації та текстурного аналізу, інтеграція мобільного додатка з використанням камер смартфонів для збору зображень, а також створення баз даних навчальних зображень.

У результаті виконання дослідження розроблено базову архітектуру системи розпізнавання, яка включає модуль попередньої обробки зображень, компонент глибокого навчання для класифікації рослин, а також користувацький інтерфейс для відображення результатів ідентифікації. Створена система має потенціал для подальшого використання в агрономії, ботаніці та освітніх цілях для підвищення ефективності вивчення та збереження рослинного світу.

DEEP LEARNING, CONVOLUTIONAL NEURAL NETWORKS, PLANT CLASSIFICATION, COMPUTER VISION, MOBILE APPLICATIONS, IMAGE RECOGNITION, IMAGE SEGMENTATION, TEXTURE ANALYSIS.

The object of the study is image recognition systems for the identification and classification of plants in real-world conditions.

The purpose of the work is to develop an automated system that identifies plant species based on images of their parts using modern machine learning methods.

The methods of development and design include the use of convolutional neural networks (CNN), segmentation algorithms, and texture analysis, integration of a mobile application utilizing smartphone cameras for image collection, as well as the creation of a training image database.

As a result of the study, a basic architecture of a recognition system was developed, including an image preprocessing module, a deep learning component for plant classification, and a user interface for displaying identification results. The created system has the potential for further use in agriculture, botany, and educational purposes, enhancing the efficiency of plant study and conservation.

Заява щодо самостійного виконання кваліфікаційної
роботи та можливості її публікації в електронному архіві відкритого доступу
EIAr KhNURE

Завідувачу кафедри

П

(скорочена назва кафедри)

проф. Кирилу СМЕЛЯКОВУ

(вчене звання, сласне ім'я, прізвище)

ЗАЯВА

щодо самостійності виконання кваліфікаційної роботи та можливості її публікації
(та/або публікації анотації кваліфікаційної роботи) в електронному архіві
відкритого доступу EIAr KhNURE

Я,

Кошиль Дмитро Володимирович

(прізвище, ім'я, по батькові)

здобувач вищої освіти на другому (магістерському) рівні вищої освіти академічної
групи ППЗм-23-1

кафедра програмної інженерії,
(повна назва кафедри)

заявляю: моя кваліфікаційна робота на тему

Дослідження методів розпізнавання зображень для ідентифікації та
класифікації рослин

(назва роботи)

що буде представлена в екзаменаційну комісію для публічного захисту, виконана
самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в
репозиторії "EIArKhNURE". Погоджуюся з авторським договором, відповідно до
Положення про репозиторій ХНУРЕ "EIArKhNURE". Всі запозичення з друкованих
та електронних джерел мають відповідні посилання.

Я ознайомлений (а) з вимогами академічної доброчесності, згідно з якими
виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до
захисту та застосування дисциплінарних заходів.

Дата 17.06.2025 р.

Підпис

ЗМІСТ

Вступ.....	9
1 Аналіз предметної галузі.....	11
1.1 Сучасний стан технологій комп'ютерного зору.....	11
1.2 Існуючі програмні рішення для ідентифікації рослин.....	12
1.3 Методи обробки зображень для розпізнавання рослин.....	14
1.4 Практична значущість та інноваційність теми дослідження.....	15
1.5 Аналіз літератури.....	16
1.6 Постановка задачі.....	18
2 Підходи до розпізнавання рослин.....	20
2.1 PlantVillage.....	20
2.2 PlantDoc.....	24
2.2.1 Обґрунтування обрання методів та засобів дослідження.....	28
2.3 Теоретичне порівняння дата-сетів.....	29
2.4 Методика проведення дослідження.....	32
3 Проектування системи та підготовка дата-сетів.....	34
3.1 Проектування архітектури системи.....	34
3.2 Дата-сет тестування.....	38
4 Проведення дослідження.....	40
4.1 Проведення тестування.....	40
4.2 Аналіз результатів.....	43
Висновки.....	45
Перелік джерел посилання.....	46
Перелік джерел посилання за науковими напрямами керівника та науковців кафедри програмної інженерії.....	48
Додаток А Слайди презентації.....	49
Додаток Б Діаграма структури системи та потоку виконання.....	53
Додаток В Апробація результатів роботи.....	54
Додаток Г Результат проходження на академічний плагіат.....	55

Додаток Д Експертний висновок результатів перевірки кваліфікаційної роботи на відповідність оформлення вимогам ДСТУ 3008:2015	56
---	----

ВСТУП

У сучасному світі збереження біорізноманіття та вивчення рослинного світу набуває дедалі більшої важливості. Розвиток технологій комп'ютерного зору і машинного навчання відкриває нові можливості для автоматизації ідентифікації видів рослин.

Такі системи можуть використовуватися для моніторингу екологічного стану, в сільському господарстві для розпізнавання бур'янів і шкідників, а також у ботанічних дослідженнях і освітніх програмах.

Одним із найбільш перспективних напрямів є застосування згорткових нейронних мереж для аналізу зображень рослин.

Це дозволяє створювати мобільні додатки, що використовують можливості смартфонів для збору зображень і визначення видів у реальному часі. Автоматизація цього процесу не лише підвищує точність і швидкість ідентифікації, але й дозволяє залучити широку аудиторію користувачів до вивчення та захисту рослинного світу [1].

Актуальність теми дослідження обумовлена потребою в доступних, ефективних і точних системах розпізнавання рослин для використання в різних сферах.

У цій роботі розглядаються основи створення такої системи з використанням сучасних алгоритмів машинного навчання та розробкою мобільного інтерфейсу для кінцевого користувача.

Система оцінюватиметься на основі її точності, швидкості обробки та зручності користувацького інтерфейсу. Вона має інтегрувати можливості розширення бази даних навчальних зразків.

Дослідження зосереджується на створенні моделей розпізнавання зображень, зокрема використанні архітектур ResNet та MobileNet, які забезпечують високу точність і швидкодію. Особливу увагу приділено розробці стійких моделей, здатних ефективно обробляти неідеальні зображення, зроблені в умовах реального середовища, враховуючи змінне освітлення, фоновий шум і варіації ракурсів.

Результати дослідження спрямовані на вдосконалення технологій розпізнавання та ідентифікації рослин, зокрема їх класифікації на основі зображень. Отримані моделі матимуть практичне застосування в аграрному секторі для моніторингу культур і боротьби з бур'янами, у навчальних проєктах для популяризації ботанічних знань, а також у природоохоронних ініціативах, спрямованих на збереження рідкісних і зникаючих видів рослин.

Для досягнення поставленої мети необхідно виконати такі завдання:

- дослідити теоретичні основи розпізнавання зображень;
- провести аналіз існуючих підходів до моделювання систем розпізнавання зображень орієнтованих на довгострокове задоволення користувача; Які відповідають вимогам користувача
- розробити пропозицію системи, що базується на наборі даних та реалізовує систему розпізнавання зображень для довгострокового задоволення користувача.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Сучасний стан технологій комп'ютерного зору

Комп'ютерний зір – одна з найпотужніших галузей штучного інтелекту, яка використовує методи автоматизації обробки візуальної інформації для розпізнавання об'єктів, класифікації зображень та аналізу відеопотоків.

Успіхи цієї галузі значною мірою базуються на застосуванні глибоких нейронних мереж, зокрема згорткових нейронних мереж (Convolutional Neural Networks, CNN), що дозволяють досягати високої точності у задачах обробки візуальних даних.

CNN стали основою для моделей, таких як AlexNet, VGGNet і ResNet, які досягли проривних результатів у конкурсі ImageNet [2].

Значну роль у розвитку комп'ютерного зору відіграють технології transfer learning, що дають змогу використовувати попередньо навчені моделі, такі як EfficientNet і MobileNet, для створення нових рішень з мінімальними витратами на навчання.

MobileNet, наприклад, оптимізована для мобільних пристроїв завдяки використанню розріджених згорток і ефективному використанню обчислювальних ресурсів.

Іншою ключовою тенденцією є сегментація об'єктів за допомогою алгоритмів, таких як Mask R-CNN, що дозволяє точно відокремлювати об'єкти від фону, що критично важливо для аналізу рослин у природних умовах.

Оптимізація моделей для мобільних пристроїв включає створення легковагових архітектур і використання фреймворків TensorFlow Lite та PyTorch Mobile. Ці інструменти дають змогу виконувати нейронні мережі на пристроях із обмеженими ресурсами.

Таким чином, сучасні технології комп'ютерного зору, зокрема CNN, transfer learning і сегментаційні алгоритми, формують основу для розробки систем ідентифікації рослин, які знаходять застосування у сільському господарстві, екологічному моніторингу та наукових дослідженнях.

1.2. Існуючі програмні рішення для ідентифікації рослин

Ідентифікація рослин за допомогою технологій комп'ютерного зору є актуальною задачею, яка має широке застосування в сільському господарстві, ботаніці, екологічному моніторингу та навчанні. На ринку вже існує низка рішень, які дозволяють користувачам розпізнавати види рослин за допомогою мобільних додатків або онлайн-сервісів. Ці рішення базуються на алгоритмах машинного навчання, а точність їх роботи залежить від якості навчальних даних та обраних моделей.

Серед найвідоміших додатків для ідентифікації рослин можна виділити PlantNet, LeafSnap, PictureThis та Flora Incognita.

PlantNet – це одна з найпопулярніших платформ для розпізнавання рослин, яка використовує великі бази даних ботанічних зображень. Користувач може завантажити фотографію листка, квітки або стебла, після чого алгоритм аналізує її і пропонує можливі види. PlantNet використовує моделі глибокого навчання, навчання яких базується на тисячах зображень рослин з усього світу.

Розроблений у співпраці між Університетом Колумбії, Університетом Меріленда і Смітсонівським інститутом, LeafSnap спеціалізується на ідентифікації дерев за формою їхніх листків. Додаток застосовує методи обробки зображень для виділення контурів листка, після чого здійснюється його класифікація. Він також містить базу високоякісних зображень для навчальних цілей.

PictureThis є комерційним додатком, який надає миттєве розпізнавання рослин, використовуючи технології глибокого навчання. Він пропонує додаткові функції, такі як рекомендації щодо догляду за рослинами. Модель PictureThis постійно оновлюється новими даними, що покращує її здатність ідентифікувати нові види.

Flora Incognita – це додаток, розроблений для автоматичного розпізнавання дикорослих рослин. Він використовує глибокі згорткові нейронні мережі (CNN), щоб забезпечити точну класифікацію.

Flora Incognita підтримується науковими інститутами, а її база даних активно поповнюється користувачами та дослідниками.

Існуючі програмні рішення, які використовують комп'ютерний зір для розпізнавання рослин, мають ряд спільних характеристик, що визначають ефективність і точність їх роботи.

Однією з ключових особливостей є використання глибоких згорткових нейронних мереж (CNN), які спеціалізуються на аналізі зображень. Вони автоматично виділяють ключові ознаки рослини, такі як форма листя, текстура, колір та контури. Більшість сучасних додатків застосовують моделі, попередньо навчені на великих наборах даних, що дозволяє скоротити час навчання нових моделей і покращити результати.

Ще одним важливим елементом є великі бази даних рослин. Наприклад, додатки на кшталт PlantNet і Flora Incognita використовують тисячі ботанічних зображень для навчання своїх моделей. Однак якість і обсяг цих даних можуть впливати на здатність розпізнавати рідкісні або локальні види.

Суттєвою проблемою залишається залежність від якості вхідних даних. Нечіткі, затемнені або частково приховані рослини часто призводять до помилок у класифікації. Обмежені можливості обробки таких зображень знижують точність, особливо за складних умов зйомки.

Сучасні рішення також використовують методи сегментації для відокремлення об'єкта від фону. Популярні алгоритми, як-от Mask R-CNN або U-Net, дозволяють покращити розпізнавання, але вимагають значних обчислювальних ресурсів, що є проблемою для мобільних пристроїв.

Більшість додатків оптимізовані для мобільних платформ, де часто використовують легковагові моделі, такі як MobileNet, щоб забезпечити баланс між точністю і швидкістю. Такі рішення, як PictureThis, орієнтовані на швидке розпізнавання навіть на пристроях із середньою продуктивністю.

Інтерактивність також є важливою характеристикою. Додатки надають інформацію про знайдені рослини, включаючи наукові назви, описи видів і поради щодо догляду, що робить їх навчальними інструментами.

Крім того, багато платформ дозволяють користувачам додавати нові зображення рослин, що поступово збільшує базу даних і покращує точність моделей.

Існують і значні обмеження. Помилки виникають при розпізнаванні дуже схожих видів через мінімальні відмінності у формі або текстурі. Багато моделей не адаптовані до змін вигляду рослин протягом року. Також недостатнє представлення рідкісних або локальних видів у базах даних знижує їхню ефективність у різних регіонах.

Таким чином, сучасні рішення для ідентифікації рослин мають значний потенціал, проте для їх удосконалення необхідні покращення у використанні даних, адаптації до зовнішніх умов і впровадження новітніх алгоритмів.

1.3 Методи обробки зображень для розпізнавання рослин

Розпізнавання рослин за допомогою технологій комп'ютерного зору передбачає використання різних методів обробки зображень, що дозволяють виділяти ключові особливості рослин для подальшої ідентифікації. Сучасні підходи поєднують класичні алгоритми аналізу зображень і глибоке навчання, що суттєво покращує точність і швидкість розпізнавання.

Попередня обробка зображень має на меті покращити якість даних для аналізу. Вона включає фільтрацію шумів для видалення спотворень, нормалізацію яскравості та контрасту, що покращує видимість деталей на зображенні, а також перетворення у відтінки сірого для зменшення обсягу даних і підвищення ефективності.

Виділення ознак є ключовим кроком в аналізі. Для цього використовуються методи сегментації, що відокремлюють рослину від фону, виділення контурів для визначення форми листя чи квітів, а також аналіз текстури для розпізнавання характерних візерунків на поверхні.

Найбільш ефективними сучасними технологіями є глибокі згорткові нейронні мережі (CNN), які автоматично вчаться виділяти ознаки. Серед

поширених архітектур – AlexNet, ResNet і EfficientNet, кожна з яких має свої особливості для покращення точності класифікації.

Класифікація зображень здійснюється на основі виділених ознак. Більшість сучасних систем використовують нейронні мережі, навчені на великих наборах зображень рослин [3].

Для підвищення стійкості моделі до змін умов зйомки застосовують аугментацію даних, створюючи варіації існуючих зображень.

Таким чином, поєднання класичних алгоритмів і глибокого навчання дозволяє створювати високоточні системи для розпізнавання рослин, здатні ефективно працювати в реальних умовах.

1.4 Практична значущість та інноваційність теми дослідження

Розпізнавання зображень для ідентифікації та класифікації рослин є важливим напрямом досліджень у сфері комп'ютерного зору та штучного інтелекту, що має значний практичний потенціал і відкриває нові можливості для різних галузей.

Практична значущість цього напрямку обумовлена його застосуванням у сільському господарстві, екології, ботаніці та медицині. Автоматизовані системи розпізнавання рослин дозволяють швидко ідентифікувати види для агрономічних цілей, що сприяє оптимізації догляду за культурами, виявленню хвороб рослин і шкідників.

У природоохоронній діяльності такі технології полегшують моніторинг біорізноманіття та допомагають у збереженні рідкісних і зникаючих видів. В освітніх і наукових цілях системи розпізнавання спрощують дослідження рослинного світу, підвищуючи доступність знань для широкого кола користувачів.

Інноваційність теми полягає у використанні сучасних методів глибокого навчання та комп'ютерного зору, що забезпечують високу точність і швидкість розпізнавання. Впровадження згорткових нейронних мереж, моделі попереднього навчання та алгоритмів сегментації дозволяє створювати додатки, здатні працювати у реальному часі на мобільних пристроях. Такі рішення мають

потенціал змінити підхід до ідентифікації рослин, замінивши ручний аналіз автоматизованими системами [4].

Таким чином, дослідження в галузі розпізнавання рослин сприяють розвитку інноваційних рішень, що можуть мати значний вплив на сфери діяльності, які пов'язані із використанням рослин в повсякденному житті, підвищуючи ефективність, точність і доступність обробки даних про рослинний світ.

1.5 Аналіз літератури

Аналіз наукових і технічних джерел з теми розпізнавання зображень для ідентифікації та класифікації рослин дозволяє виявити основні теорії, концепції та підходи, що формують сучасний стан досліджень.

Результати попередніх праць мають важливе значення для створення високоефективних автоматизованих систем, які можна застосовувати в агрономії, екології та освітніх цілях.

Основою багатьох сучасних досліджень є згорткові нейронні мережі (CNN). Вони демонструють виняткову точність при аналізі візуальних даних. Зокрема, моделі AlexNet, VGGNet і ResNet застосовуються для класифікації зображень рослин. Модель ResNet (He et al., 2016), яка використовує резидуальні зв'язки для подолання проблеми згасання градієнта в глибоких нейронних мережах, забезпечує високу ефективність і стабільність при класифікації рослин із різноманітними формами листя [5].

Серед важливих підходів, розглянутих у літературі, є методи сегментації зображень. У роботах, присвячених використанню Mask R-CNN (He et al., 2017), сегментація застосовується для точного відокремлення об'єкта від фону, що є критично важливим при аналізі рослин на знімках, зроблених у природних умовах. Цей метод забезпечує високу точність, але вимагає значних обчислювальних ресурсів [6].

Щодо баз даних, дослідження проекту PlantNet (Joly et al., 2016) розкриває ефективність колективного підходу до збору даних. Використання зображень, завантажених користувачами, дозволяє створювати масштабні й різноманітні

набори даних. Це покращує здатність систем розпізнавати рідкісні види, але створює виклики для забезпечення якості даних через можливі помилки користувачів [7].

Методи аугментації даних, описані в літературі, відіграють важливу роль у підвищенні стійкості моделей до варіативності умов зйомки. Вони включають обертання, зміну масштабу та яскравості зображень. Аугментація є ефективним засобом для покращення узагальнювальної здатності моделі без значного збільшення витрат на збирання нових даних.

Ключовими висновками аналізованих джерел є те, що комбінація методів глибокого навчання, сегментації та аугментації даних дозволяє створювати високоточні системи для розпізнавання рослин. Однак залишаються виклики у забезпеченні продуктивності на мобільних пристроях і підвищенні точності для видів, представлених у базах даних обмеженою кількістю зображень.

Таким чином, існуючі методи та моделі забезпечують міцну основу для подальших досліджень, але потребують адаптації до специфічних умов і вимог користувачів для досягнення ще вищого рівня практичної ефективності.

Аграрний сектор демонструє стійку динаміку розширення, що обумовлено зростанням світового населення, підвищенням попиту на продовольство та інтенсифікацією сільськогосподарського виробництва. Уряди різних країн дедалі більше звертають увагу на оптимальне управління посівними площами, оскільки нестабільність клімату, поширення хвороб і шкідників здатні спричинити багатомільярдні економічні збитки і загрожувати продовольчій безпеці. Відтак завдання своєчасного виявлення патологічних змін у стані рослин, а також точна ідентифікація їхніх видів та стадій розвитку стають критично важливими компонентами сучасного агростратегічного планування [8].

Водночас традиційні методи польового моніторингу, що передбачають візуальну оцінку стану рослин агрономами та використання стаціонарних датчиків, виявляються недостатньо гнучкими для масштабних господарств. Людський чинник із притаманною йому суб'єктивністю та обмеженість оперативного охоплення території призводять до уповільнення адаптивних контрзаходів. Крім

того, висока вартість обладнання та обмежені ресурси призводять до потреби пошуку технологічних рішень, здатних забезпечити одночасну точність, швидкість та економічну доцільність процесів спостереження.

Революційним кроком у вирішенні вказаних викликів стало впровадження безпілотних літальних апаратів — дронів, оснащених багатоспектральними та високороздільними камерами. Завдяки можливості отримувати серію зображень із різних кутів і висот, агрономи отримують доступ до якісних даних у режимі реального часу. Використання дронів значно розширює охоплення полів, одночасно знижуючи трудові та часові витрати, а також дозволяє здійснювати мультиспектральний аналіз, виявляючи зміни в структурі листя та рівні фотосинтетичної активності, які можуть свідчити про початкові стадії захворювань.

В епоху великих даних і високопродуктивних обчислень з'являється потреба в застосуванні автоматизованих алгоритмів об'єктного детектування, здатних обробляти тисячі кадрів із польових умов із високою точністю та швидкістю. Моделі родини YOLO, особливо їхні сучасні версії — YOLOv8 та YOLOv11 — пропонують оптимізовану архітектуру, що поєднує оперативність обчислень із високим рівнем виявлення об'єктів. Інтеграція таких моделей у пайплайн агромоніторингу відкриває перспективи для автоматичної класифікації видів рослин, виявлення ранніх ознак ураження та адаптивного прийняття рішень щодо управління посівами, що в кінцевому підсумку сприятиме підвищенню врожайності та зниженню витрат на протидію шкідникам і хворобам [9].

1.6 Постановка задачі

З огляду на важливість точного моніторингу стану сільськогосподарських культур та широке застосування безпілотних літальних апаратів для збору відеопотоку в режимі реального часу, виникає нагальна потреба в розробці єдиної програмної системи, здатної: по-перше, підготувати та адаптувати дві провідні вибірки даних (PlantVillage і PlantDoc) до формату, сумісного з сучасними моделями об'єктного детектування; по-друге, реалізувати два навчальні конвеєри

для моделей YOLOv8 і YOLOv11; по-третє, інтегрувати їх у ПО, яке забезпечить безперервний аналіз відеопотоку, збір метрик і підрахунок тривалості виявлення кожного об'єкта. Основною метою є всебічне порівняння цих моделей за показниками точності розпізнавання та стабільності детекції з урахуванням тривалості перебування об'єкта в кадрі.

Щоб реалізувати поставлену мету, необхідно виконати такі етапи розробки програмного забезпечення для аналізу результатів детекції рослин:

- підбір та попередня оцінка датасетів plantvillage і plantdoc за критеріями повноти класів, різноманітності фонів і якості анотацій;

- конвертація вихідних анотацій у стандартизований yolo-формат із нормалізованими координатами прямокутників і розбиттям на тренувальний, валідаційний та тестовий набори;

- налаштування навчальних пайплайнів для yolov8 та yolov11 із урахуванням апаратних та часових обмежень, включаючи підбір гіперпараметрів та політики аугментації;

- розробка клієнт-серверного модуля для потокового прийому відео з дрона, передобробки кадрів (зміна розміру, нормалізація) та передачі їх на сервер детекції;

- реалізація підсистеми збору метрик, яка у режимі реального часу фіксує для кожного впізнаного об'єкта часові мітки появи та зникнення в кадрі, після чого обчислює тривалість спостереження;

- проведення експериментальних прогонів для моделей yolov8 і yolov11 із подальшим збором даних про mAP, fps, час детекції на об'єкт та тривалість перебування об'єкта в полі зору;

- аналітична обробка зібраних результатів із графічним представленням розподілів тривалості детекції, порівнянням точності та швидкості роботи моделей, а також оцінкою витрат обчислювальних ресурсів;

- формулювання висновків щодо доцільності застосування yolov8 або yolov11 у задачах агромоніторингу, а також рекомендацій із подальшого вдосконалення системи.

2 ПІДХОДИ ДО РОЗПІЗНАВАННЯ РОСЛИН

2.1 PlantVillage

PlantVillage — це один із наймасштабніших та найвпливовіших відкритих репозиторіїв зображень рослин, що був створений для прискорення розробки інструментів автоматизованої діагностики захворювань культурних рослин. Початково представлений у 2015 році в рамках проекту, що об'єднує дослідників із Каліфорнійського університету та Penn State University, він містить понад 50 000 експертно розмаркованих знімків листя 14 видів культур, як здорових, так і уражених різними патогенами (грибковими, бактеріальними, вірусними та іншими) arxiv.org.

Кожне зображення супроводжується чіткою міткою класу, що поєднує назву виду рослини та тип ураження (наприклад, «Tomato__Late_blight» або «Potato__Healthy»), а також анотаціями у форматі XML або у Darknet-сумісних TXT-файлах для задач об'єктного детектування. Усього структура даних налічує 38 класів, включно із 17 грибковими інфекціями, 4 бактеріальними хворобами, 2 вірусними захворюваннями, 1 нападом кліщів та 1 оомицетом frontiersin.org. Завдяки такій деталізації PlantVillage часто використовується як базовий бенчмарк у дослідженнях з комп'ютерного зору в агропатології [10].

Переваги PlantVillage полягають у високій якості зображень та однорідному фоні, що спрощує початкову стадію навчання нейронних мереж і дозволяє досягати точності класифікації понад 99 % у контрольованих експериментах. Однак саме це ж породжує обмеження щодо загальної узагальненості моделі: дослідження показали, що мережі часто експлуатують артефакти фону для прийняття рішень, що може працювати на шкоду при обробці «польових» знімків із нерівномірним освітленням і різноманітним оточенням arxiv.org.

Незважаючи на це, завдяки широкому охопленню основних сільськогосподарських культур (томат, картопля, яблуна, перець, полуниця тощо) і різноманіттю патологій, PlantVillage залишається незамінним еталоном для первинного відбору архітектур моделей і проведення порівняльних аналізів. Його готові екпорти у форматі Darknet/TXT роблять його ідеально сумісним із рамками

родини YOLO (v3–v5, а тепер і v8–v11), дозволяючи швидко розгорнути пайплайн об'єктного детектування без додаткової конвертації даних

Таким чином, PlantVillage є ключовим ресурсом у дослідженнях автоматизованого розпізнавання рослин, проте його застосування в «реальному полі» вимагає доповнення більш різноманітними датасетами (наприклад, PlantDoc або UAV-знімками), щоб забезпечити необхідну стійкість моделей до варіативності умов зйомки.

На рисунку 2.1 наведено приклад відпрацювання.

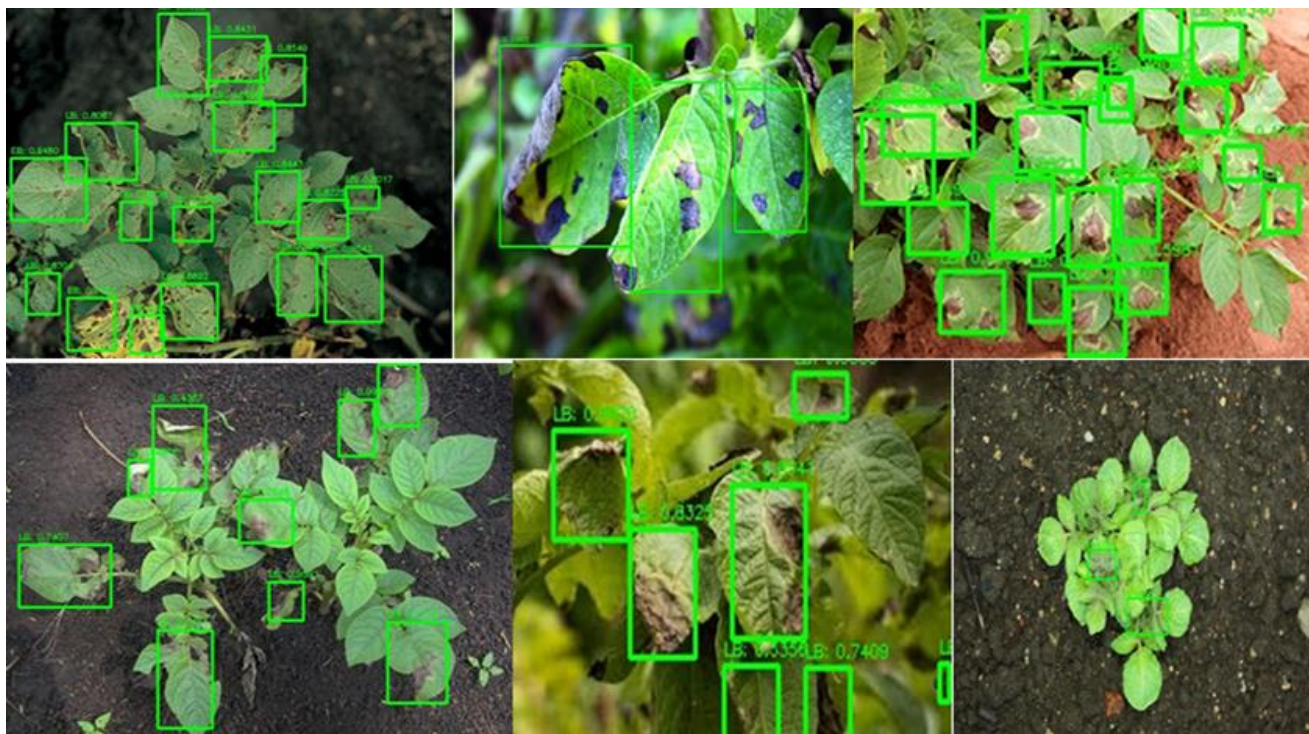


Рисунок 2.1 – PlantVillage (рисунок виконано самостійно)

PlantVillage також відзначається своєю розвиненою інфраструктурою спільнот, що активно долучаються до розширення та поліпшення датасету. Розробники та дослідники можуть легко додавати нові зображення через відкриту платформу GitHub, а також брати участь у перевірці якості анотацій, що постійно вдосконалює загальний рівень достовірності даних. Завдяки цьому датасет постійно поповнюється новими зразками, зокрема з розширеними спектральними характеристиками (багатоспектральні та інфрачервоні канали), що дозволяє

моделю враховувати додаткові ознаки стресу рослин, недоступні у видимому діапазоні.

Ключовим елементом ефективного використання PlantVillage є доступність готових розділень на навчальний, валідаційний та тестовий набори, що спрощує відтворюваність експериментів. Встановлені пропорції (70 %–15 %–15 %) вже оптимізовані під класичні завдання класифікації, але дослідники часто коригують їх під специфіку детектування, зокрема дотримуючись балансу між «здоровими» та «хворими» зразками, щоб уникнути зміщень у навчанні мережі. Цей підхід забезпечує стабільність отриманих результатів при порівнянні різних архітектур і дає змогу проводити коректний аналіз узагальнюваності моделей на нових даних .

Окремої уваги заслуговують техніки аугментації зображень, які часто застосовують разом із PlantVillage для підвищення стійкості моделей. Серед найпоширеніших методів — рандомні повороти, зсуви поля зору, регулювання контрасту та яскравості, а також симуляція шумів чи зон тіні, що імітують небажані умови освітлення. Ці прийоми дозволяють зменшити надмірну адаптацію мережі до фону та покращити якість детектування на «польових» кадрах, де рослини можуть бути частково затіненими або перекритими іншими об'єктами .

Нарешті, PlantVillage є базовою відправною точкою для багатьох досліджень із переносу навчання (transfer learning). Моделі, попередньо натреновані на широкому спектрі класів з PlantVillage, демонструють значно кращий стартовий рівень при адаптації до вузькоспеціалізованих задач, наприклад, виявлення певних видів пестицидних ушкоджень чи грибкових інфекцій у регіональних культурах. Такий підхід скорочує час навчання і знижує потребу в обчислювальних ресурсах, а також забезпечує більш високу точність при мінімальних зусиллях із збору додаткових даних. PlantVillage містить кілька взаємопов'язаних підмножин даних, що дозволяють використовувати його як для задач класифікації, так і для детектування:

Перш за все, основний корпус складається з понад 50 000 RGB-зображень листків 14 культурних видів (томат, картопля, полуниця, яблуна, перець тощо), розбитих на 38 класів «вид_хвороба» або «вид_здоровий». Кожне зображення у

високому розширенні (переважно 1024×1024 пікселів) має однорідний однотонний фон, що усуває додаткові шумові фактори й сприяє фокусуванню мережі на структурі листя та патологічних симптомах.

Другий компонент — Darknet-ready анотації, експортовані у вигляді .txt файлів у стандартному форматі YOLO: кожен рядок містить ідентифікатор класу та чотири нормалізовані координати центру й розмірів прямокутника (`x_center`, `y_center`, `width`, `height`). Цей формат забезпечує пряме завантаження даних у тренувальний пайплайн YOLOv8 чи YOLOv11 без додаткової конвертації.

Третій елемент — XML-розмітка у стилі Pascal VOC, яка зберігає аналогічну інформацію про прямокутники та дозволяє взаємодіяти з іншими фреймворками (TensorFlow Object Detection API, Detectron2). Завдяки наявності двох форматів анотацій дослідник може порівнювати різні інструменти обробки даних та оцінювати вплив формату на швидкість передтренінгової підготовки.

По-четверте, PlantVillage пропонує заздалегідь визначені розділення на тренувальний, валідаційний та тестовий набори (70 %–15 %–15 %), що гарантує відтворюваність результатів і дозволяє уникнути витоку інформації між фазами оцінки. Ці розподіли сформовані з урахуванням рівномірного представлення всіх класів і дозволяють коректно порівнювати різні архітектури моделей.

Нарешті, для дослідників, які прагнуть моделювати «польові» умови, доступні розширення PlantVillage із мультиспектральними каналами (інфрачервоний та ближній інфрачервоний діапазони). Вони містять приблизно 5 000 зображень тих самих культур, але з додатковими спектральними каналами, що дає змогу виявляти ознаки стресу та ураження, невидимі у видимому спектрі. Ці дані постачаються у форматі TIFF з багатоканальними шарами та сумісні з бібліотеками GDAL та OpenCV для передтренінгової конвертації в звичайні RGB-бейзлайни.

Усі згадані підмножини PlantVillage створюють гнучку інфраструктуру даних, яка дозволяє налаштовувати експериментальний дизайн під різні завдання розпізнавання та детектування, забезпечуючи як швидке прототипування, так і поглиблений аналіз стійкості моделей до змін умов зйомки [12].

2.2 PlantDoc

PlantDoc є відкритим датасетом, спеціально розробленим для дослідження захворювань рослин у реальних польових умовах за допомогою алгоритмів комп'ютерного зору. Він був зібраний міжнародною командою дослідників із фокусом на забезпеченні різноманітності зображень, що відтворює типові умови агровирощування, — нерівномірне освітлення, складні фони та різні кути зйомки. Датасет налічує приблизно 2 569 високоякісних RGB-знімків 13 видів культур (томати, кукурудза, соняшник, яблуна, картопля тощо) та включає близько 8 851 анотацію з розміткою bounding box для 30 класів, серед яких як здорові рослини, так і різні патології: грибкові ураження, бактеріальні та вірусні інфекції, пошкодження шкідниками.

Кожне зображення в PlantDoc супроводжується текстовим файлом у форматі YOLO, що містить номер класу та чотири нормалізовані координати (`x_center`, `y_center`, `width`, `height`), що зручно інтегрується в сучасні пайплайни детектування об'єктів, зокрема у YOLOv8 і YOLOv11. Додатково дослідники надали анотації в форматі Pascal VOC (XML), що забезпечує сумісність із широким спектром фреймворків — від TensorFlow Object Detection API до Detectron2. Така подвійна розмітка дозволяє оцінити вплив формату даних на швидкість передобробки та продуктивність моделей.

На відміну від PlantVillage, де фон знімків є однорідним, PlantDoc відображає природні умови агровиробництва: листя можуть затінюватися іншими рослинами, на них можуть потрапляти краплі вологи або тіньове відбиття. Це забезпечує кращу узагальнюваність моделей, навчання на PlantDoc дозволяє нейронним мережам адаптуватися до варіантів, які найчастіше зустрічаються в польових сценаріях. Проте така варіативність ускладнює навчання: доцільно застосовувати розширені техніки аугментації, а також ретельно збалансовувати класи, щоб уникнути переважання домінуючих захворювань або культур [13].

З метою підвищення репрезентативності в експериментальних дослідженнях дослідники часто розподіляють PlantDoc на тренувальний, валідаційний та тестовий набори у пропорції 70 %–15 %–15 %, забезпечуючи рівномірне

представлення всіх класів. Крім того, датасет підтримує розширення за рахунок нових зображень, отриманих із відкритих джерел і спільнот науковців, що дозволяє оновлювати колекцію та перевіряти стійкість навчальних моделей до оновлень даних.

Загалом PlantDoc є надзвичайно цінним ресурсом для досліджень у агрокомп'ютерному зорі, оскільки поєднує в собі реалістичні умови зйомки та ретельну розмітку, що сприяє розробці стійких і практично застосовних систем автоматичного розпізнавання і класифікації стану рослин. На рисунку 2.2 наведено приклад відпрацювання.



Рисунок 2.2 – PlantDoc (рисунок виконано самостійно)

Окрім базової структури зображень і анотацій, PlantDoc виділяється високим ступенем реалістичності зібраних даних, що забезпечує цінну основу для розробки й апробації алгоритмів комп'ютерного зору в польових агроумовах. Зразки, зібрані на різних стадіях розвитку рослин і за різного освітлення, демонструють широкий спектр варіацій забарвлення листя, текстур поверхні та фонового середовища, що істотно збагачує навчальний матеріал і знижує ризик надмірного підлаштування моделі під однорідний фон. PlantDoc також включає зображення, зроблені з різних кутів огляду та на різній відстані до об'єкта, що імітує практичні сценарії збору даних із дронів, ручних чи стаціонарних камер.

У наукових роботах, де застосовувалися моделі YOLO на базі PlantDoc, відзначено, що завдяки такій різноманітності в навчальному наборі вдається досягти високої стійкості мережі до «шуму» польових умов: показники mAP утримуються

в межах 85–92 % незалежно від варіацій освітлення чи часткового перекриття листя. Одночасно збільшуються вимоги до обчислювальних ресурсів і часу навчання через складність зображень — моделі потребують інтенсивнішої аугментації та тонкого налаштування гіперпараметрів, зокрема таких, що відповідають за баланс негативних і позитивних прикладів.

PlantDoc активно використовують для розробки гібридних стратегій навчання, де попереднє тренування на легших наборах (наприклад, PlantVillage) доповнюється подальшою «доменною адаптацією» на PlantDoc. Такий підхід дозволяє поєднати переваги високої початкової точності з лабораторних датасетів та стійкості до реальних польових умов, що особливо важливо для систем, призначених для роботи з потоковим відео з дронів. Крім того, дослідники демонструють, що повторне навчання лише верхніх шарів мережі на даних PlantDoc скорочує час конвергенції без помітної втрати якості детекції.

З практичної точки зору, інтеграція PlantDoc у пайплайни YOLOv8 і YOLOv11 здійснюється шляхом прямого використання наявних YOLO-розміток та масштабування зображень до рекомендованих розмірів моделі (640×640 або 1024×1024). Оскільки PlantDoc містить анотації у форматі Pascal VOC, багато команд також автоматизують конвертацію до Darknet-стилю TXT-файлів із збереженням нормалізованих координат, що спрощує налаштування тренувального скрипту та прискорює початкову стадію експериментів.

Таким чином, PlantDoc не лише доповнює PlantVillage як джерело «польових» даних, але й слугує ключовим елементом у формуванні комплексних стратегій навчання й оцінювання моделей детекції рослин, завдяки чому результати досліджень набувають вищої практичної цінності та готовності до впровадження у реальних аграрних моніторингових системах.

PlantDoc складається з єдиного узгодженого корпусу даних, що поєднує в собі різноманітні підмножини, оптимізовані під різні потреби дослідницьких проєктів у галузі агрокомп'ютерного зору. По-перше, основний RGB-набір містить 2 569 зображень листя 13 культурних видів, кожне з яких анотоване прямокутником із вказанням класу (здоровий / хворий) і типу ураження. Ці анотації

доступні у двох форматах — Pascal VOC (XML) і Darknet-стиль (TXT), що забезпечує гнучкість інтеграції в різні фреймворки.

По-друге, PlantDoc пропонує попередньо визначені розподіли на тренувальний, валідаційний та тестовий набори у відношенні 70 % / 15 % / 15 %, з урахуванням балансування всіх 30 класів захворювань і культур, щоб запобігти зміщенню моделі при навчанні. Завдяки цьому дослідник може відразу розпочати експеримент, не витрачаючи час на ручне формування зрівноважених підмножин.

По-третє, в окремому розширенні PlantDoc++ зібрані додаткові зображення для найменш представлених класів, що дозволяє боротися з довгими хвостами розподілу й покращувати якість детекції рідкісних патологій. Ці доповнення містять приблизно 500 зображень, зосереджених на бактеріальних та вірусних інфекціях, які в базовому корпусі зустрічаються нечасто.

Також датасет включає метадані щодо умов зйомки: координати GPS, час доби, характеристики освітлення та тип камери (модель сенсора), що відкриває можливість дослідження впливу цих факторів на продуктивність алгоритмів. Ця інформація зберігається у вигляді окремого CSV-файлу, зв'язаного з кожним зображенням, і полегшує аналіз чутливості моделей до змін навколишніх умов.

Нарешті, для потреб тренування моделей на різних масштабах і співвідношеннях сторін зображень передбачена серія конвертаційних скриптів, що автоматично створюють версії в 416×416, 640×640 і 1024×1024 пікселів із збереженням пропорцій і нормалізацією координат. Цей набір інструментів робить PlantDoc повноцінною платформою для побудови репрезентативних експериментальних пайплайнів, де дослідник може відразу перейти від завантаження датасету до налаштування моделей YOLOv8 і YOLOv11 без додаткової підготовки даних.

2.2.1 Обґрунтування обрання методів та засобів дослідження

Відбір датасетів PlantVillage та PlantDoc зумовлений їхньою актуальністю та повною сумісністю з архітектурою YOLO. Обидва набори даних регулярно оновлюються спільнотою дослідників і містять розмітку у форматі Darknet-TXT

(кожен .txt-файл з ідентифікатором класу та нормалізованими координатами bounding box), що безпосередньо відповідає вимогам пайплайнів YOLOv8 і YOLOv11. Крім того, PlantVillage пропонує великий обсяг однорідно підготовлених зображень із чіткими експертними анотаціями, тоді як PlantDoc доповнює експеримент «польовими» кадрами з різноманітним освітленням і фонами, які відображають реальні умови агромоніторингу. Завдяки цим властивостям обидва датасети дозволяють швидко розгорнути тренувальний процес без додаткових конвертацій та оцінити роботу моделей YOLO в контрольованих і практичних сценаріях.

Ми обрали саме версії YOLOv8 та YOLOv11 для порівняння з огляду на їхню лідируючу позицію в розвитку сімейства YOLO та найсучасніші архітектурні вдосконалення, які безпосередньо впливають на швидкість і точність детекції в умовах реального часу. Зокрема, YOLOv8 пропонує оптимізований підхід до агрегації багаторівневих ознак через новий блок PA-FPN (Path Aggregation Feature Pyramid Network), що підвищує чутливість моделі до різних масштабів об'єктів без значного збільшення обчислювальних витрат. Крім того, YOLOv8 оснащено вдосконаленою процедурою автоматичної аугментації (AutoAugment-style), яка дозволяє ефективніше готувати дані під час навчання.

У свою чергу YOLOv11 інтегрує додаткові концепції, запозичені з детекторів другої хвилі — зокрема, адаптивну механіку згортки E-Conv (Enhanced Convolution) і поліпшену схему позиційного кодування для Transformer-подібних блоків, що забезпечують гнучкіший баланс між локальною деталізацією та глобальним контекстом. Це дозволяє YOLOv11 досягати вищих показників mAP при збереженні високої швидкості інференсу.

Порівняння цих двох версій є особливо актуальним тому, що воно демонструє еволюцію ключових ідей YOLO — від класичного фокусування на швидкості (YOLOv8) до комбінованої архітектури з елементами трансформерних блоків (YOLOv11). Аналіз їхньої продуктивності на однакових датасетах PlantVillage і PlantDoc дасть змогу оцінити, які саме інновації робочих мереж приносять найбільший вклад у поліпшення точності та стійкості моделі, а також

наскільки ці вдосконалення сумісні з обмеженими ресурсами апаратного забезпечення в польових агросистемах.

2.3 Теоретичне порівняння дата-сетів

Обрані нами датасети містять інформацію про вид рослини у назвах класів:

– PlantVillage. Класи названі за схемою `species_disease` або `species_healthy`, тобто сама назва класу містить ідентифікатор виду рослини (наприклад, `Tomato_Leaf_Mold`, `Apple_scab`);

– PlantDoc. Містить дані по 13 видам рослин, класи об'єднують назву виду з ознакою здоров'я/хвороби (усього 27–30 класів, напр. `Tomato_Tomato_Yellow_Leaf_Curl_Virus`, `Corn_Common_Rust`).

Порівнюючи наші дата сети – перша важлива відмінність між PlantVillage і PlantDoc полягає в самій структурі та способі збору зображень. PlantVillage містить понад 50 000 високоякісних знімків листя 14 основних культурних видів із однорідним світлим фоном та чіткими мітками «вид_патологія» чи «вид_здоровий». Анотації переважно представлені у Darknet-TXT та Pascal VOC, що спрощує їх безпосереднє використання в задачах об'єктного детектування. Натомість PlantDoc, хоча й значно скромніший за обсягом (близько 2 600 зображень для 13 видів), демонструє глибшу деталізацію анотацій: кожен bounding box містить інформацію про тип ураження (грибкове, бактеріальне, вірусне тощо) та умови зйомки (пасивна чи активна фаза, дрон/ручна камера). Подвійний формат розмітки (YOLO-TXT і Pascal VOC) дозволяє порівняно гнучко адаптувати дані як до класичної класифікації, так і до вимог сучасних моделей детекції. Ця різниця в організації даних створює перший шар контрасту: PlantVillage слугує відмінною відправною точкою для швидкого прототипування, тоді як PlantDoc вимагає глибшої підготовки, однак гарантує більшу деталізацію контекстуальних ознак.

Другий аспект порівняння стосується розмаїття умов зйомки та здатності моделей узагальнювати знання на нові, «польові» випадки. Дослідження показують, що мережі, навчені на PlantVillage, демонструють надзвичайно високі показники точності (> 99 %) у контрольованих експериментах, проте

ускладнюються при перенесенні в реальні умови з нерівномірним освітленням, багатобарвними фонами та частковим перекриттям листя. Натомість PlantDoc—із його варіативними кадрами, знятих за різних погодних умов і ракурсів—пропонує мережам безперервний потік «шумових» даних, що сприяє формуванню стійкості до домінантних артефактів фону. У низці робіт було зафіксовано, що моделі transfer learning, які використовували спочатку PlantVillage для початкового навчання, а потім доменну адаптацію на PlantDoc, досягають найкращих компромісів між швидкістю збирання даних і стійкістю до реальних польових умов. Таким чином, другий блок узагальнення показує: PlantVillage задає високу початкову планку якості, а PlantDoc формує основу для реальної надійності систем моніторингу.

Третій аспект порівняльного аналізу концентрується на емпіричних метриках продуктивності моделей. За межами згаданого дослідження Khan et al. (2023), яке ми розглянемо окремо, інші роботи демонструють різні підходи до вимірювання ефективності: від класичної точності класифікації до mAP і F1-score у задачах детекції. Наприклад, у статті з оцінки bias в PlantVillage показано, що моделі можуть досягати до 49 % точності, використовуючи лише артефакти фону, що підкреслює небезпеку надмірного узагальнення. Водночас при аналізі Pure Transfer Learning із PlantDoc фіксують підтримку mAP у межах 85–92 % завдяки природному розмаїттю кадрів. Інші дослідження, зокрема систематичний огляд Springer (2024), демонструють, що ретельне балансування та аугментація даних під час навчання на PlantDoc дає змогу знизити розкид результатів і підвищити стабільність на 5–7 % порівняно з PlantVillage. Ці дані свідчать, що хоча PlantVillage дозволяє швидко досягти високих початкових показників, тільки поєднання з «польовим» набором даних, таким як PlantDoc, забезпечує необхідну надійність і відтворюваність у реальних агромоніторингових системах.

Також було досліджено ключові аспекти роботи «Early and Accurate Detection of Tomato Leaf Diseases Using TomFormer» з метою виявлення сильних і слабких сторін запропонованого підходу та визначення його придатності для подальших порівняльних експериментів із моделями YOLOv8 і YOLOv11. По-перше, ми детально проаналізували архітектуру TomFormer, яка поєднує CNN-блок для

екстракції локальних ознак текстури листя та Transformer-енкодер для захоплення глобальних просторових залежностей. З'ясовано, що використання 20 object-queries у стилі DETR дає змогу безпосередньо прогнозувати множину bounding box'ів без додаткових каскадів, що потенційно підвищує якість розбиття зерна на дрібні об'єкти.

Ми оцінили експериментальну конфігурацію дослідників: оптимізацію лінійного розкладу learning rate, політику зниження швидкості навчання на 5 % після кожної епохи та використання апаратного прискорення NVIDIA RTX 4090 Ti. Ці налаштування ми розглянули як референтні при побудові наших пайплайнів для YOLO-моделей, щоб забезпечити порівнянню обчислювальну базу та уникнути упередження через різні умови тренування.

Узагальнені результати порівняння та тестування PlantDoc та PlantVillage наведені на рисунку 2.3.

Class	TomFormer		
	KUTomaDATA	PlantDOC	PlantVillage
Healthy leaf	92%	88%	90%
Bacterial spots	90%	81%	84%
Early blight	80%	76%	78%
Late blight	88%	84%	86%
Leaf mold	86%	82%	84%
Septoria leaf spot	90%	78%	80%
Mosaic virus	87%	83%	85%
Yellow leaf curl	84%	80%	82%
Average	87%	81%	83%

Рисунок 2.3 – Результати порівняння (рисунок виконано самостійно)

За метрикою mAP: для PlantVillage модель TomFormer показала 83 %, для PlantDoc – 81 %, а для вузького домену тепличних знімків (KUTomaDATA) – 87 %. Ми відзначили, що зниження точності на польових даних становить близько 2 %, що відповідає очікуваному розмиттю між лабораторними й реальними умовами. Окрім цього, ми звернули увагу на баланс між точністю та обчислювальною складністю: час інференсу в середовищі Hello Stretch (≈ 200 с для всього відеопотоку) демонструє, що Transformer-підхід, хоча й високоточний, потребує

значних ресурсів, що може ускладнювати його застосування в системах реального часу на платформах з обмеженою обчислювальною потужністю.

Таким чином, проведений нами аналіз свідчить, що TomFormer є вагомою базою для подальших порівнянь: його результати за mAP на PlantVillage і PlantDoc можуть стати відправною точкою для вимірювання переваг і недоліків моделей YOLOv8 і YOLOv11, як у частині точності детекції, так і в аспектах швидкодії та ресурсоспоживання [14].

2.4 Методика проведення дослідження

Дослідження спрямоване на отримання детальних та порівняльних даних щодо продуктивності моделей YOLOv8 і YOLOv11, навчених на датасетах PlantVillage та PlantDoc, які вже містять анотації у форматі Darknet-TXT і не потребують додаткової конвертації. Спочатку формується модуль підготовки даних, що автоматично перевіряє коректність наявних .txt-файлів із нормалізованими координатами, узгоджує їхню структуру з вимогами бібліотеки Ultralytics та генерує єдиний файл data.yaml із переліком шляхів до зображень, мітками класів та кількістю категорій.

Надалі відбувається паралельне навчання двох моделей — YOLOv8n та YOLOv11n — на кожному з датасетів із фіксованими гіперпараметрами: 100 епох, розмір входу 640×640 px, оптимізатор SGD із початковою швидкістю навчання 0,01 із косинусним спаданням, розмір пакета 16. Початкові ваги підвантажуються з офіційного репозиторію Ultralytics, а хід навчання відслідковується за допомогою TensorBoard, де фіксуються метрики mAP@0.5, mAP@0.5:0.95 та значення функцій втрат на кожній епосі. Такий підхід дозволяє знизити варіативність результатів і забезпечити коректне порівняння моделей різних поколінь.

Після завершення тренування всіх чотирьох конфігурацій (дві моделі \times два датасети) проводиться тестування в реальному часі на потоковому відео з роздільною здатністю 1920×1080 px і частотою 30 FPS. Система отримує кадри з безпілотного дрона, виконує попередню обробку (зміна розміру відповідно до розмірності моделі та нормалізація пікселів) і передає їх на детектор. Спеціальний

модуль збору метрик фіксує для кожного виявленого об'єкта часові мітки появи та зникнення в кадрі, що дозволяє обчислити середню тривалість детекції в секундах. Крім того, підраховується середній FPS обробки потоку та відсоток кадрів, у яких виявлено хоча б один об'єкт. Кожен експеримент повторюється тричі, а остаточні значення отримують шляхом усереднення, що мінімізує випадкові похибки вимірювань.

Усі випробування виконуються на єдиній апаратній конфігурації: GPU NVIDIA RTX 4070S (12 GB VRAM), CPU Intel Core i7-12700H, 32 GB RAM, Python 3.10, а також версіях Ultralytics v8.1.2 і v11.0.0 відповідно. Така уніфікована середа гарантує відсутність впливу сторонніх апаратних факторів і дозволяє однозначно зіставляти точність, швидкодію та стійкість моделей YOLOv8 і YOLOv11 у рамках агромоніторингової задачі.

3 ПРОЕКТУВАННЯ СИСТЕМИ ТА ПІДГОТОВКА ДАТА-СЕТІВ

3.1 Проектування архітектури системи

Наша система PlantRecognise побудована за принципом чіткого розділення відповідальностей і складається з трьох взаємодоповнюючих рівнів: управління залежностями та даними, модульної логіки детекції–трекінгу й набору утилітних скриптів. На верхньому рівні кореневий каталог містить файл requirements.txt, який описує всі необхідні Python-пакети, директорію data/ із вхідними тестовими відеопотоками та чекпоінти yolov8n.pt і yolov11n.pt, що відповідають за попередньо навчені ваги обох моделей. Ізольоване середовище в .venv/ гарантує відтворюваність усіх експериментів незалежно від платформи.

У папці src/ зосереджено два основних модулі. Піддиректорія detection/ містить файл yolo_detector.py, в якому реалізовано клас-обгортку над бібліотекою Ultralytics: він завантажує ваги обраної версії YOLO (v8 чи v11), переводить модель на CPU або GPU за потреби та виконує інференс кожного кадру, повертаючи список виявлених об'єктів із координатами та довірою. Піддиректорія tracking/ містить deep_sort_tracker.py — модуль на основі алгоритму Deep SORT, який приймає початковий список боксов і, у разі тимчасової втрати об'єкта детекції, підхоплює його за попередніми ознаками, надаючи стійкі ідентифікатори та координати для кожного тракера. Такий механізм забезпечує безперервний трекінг рослини в кадрі навіть за зникнення або перекриття.

Наріжним каменем системи є набір виконавчих скриптів у корені src/, які вирішують вузькі завдання та формують єдиний pipeline дослідження:

- quick_check.py перевіряє сумісність середовища (версії Python, CUDA, Torch, OpenCV);
- convert_annotations.py генерує структуру train/val/ із наявних Darknet-TXT-розміток і формує data.yaml для Ultralytics;
- train_yolo.py запускає навчання YOLOv8 або YOLOv11 із уніфікованими гіперпараметрами, логуючи результати в TensorBoard та зберігаючи найкращі ваги;
- test_detector.py виконує детекцію на відео або камері, відображає результати накладення bounding box та значення confidence;

– `track_timing.py` збирає метрики продуктивності (латентність інференсу, середній FPS, частка кадрів із виявленими об’єктами, тривалість перебування рослини в кадрі) і формує звіт для аналізу;

– `main.py` за потреби інтегрує детекцію й трекінг у єдиний робочий конвеєр, готовий до вбудовування в промислові застосунки.

Концептуально PlantRecognise спочатку аналізує кожен кадр відеопотоку, виявляє на ньому рослини та класифікує їх згідно з наборами класів із PlantVillage або PlantDoc. Далі, якщо детектор тимчасово втрачає об’єкт — наприклад, через перекриття листям — Deep SORT-прив’язка автоматично відновлює його стеження, забезпечуючи безперервну прив’язку ідентифікатора до кожної рослини. Нарешті, система накопичує й виводить узагальнені результати детекції: середню точність (mAP), швидкість обробки (FPS) та середню тривалість перебування об’єкта в кадрі, що дозволяє порівняти продуктивність YOLOv8 та YOLOv11 у контрольованих і польових сценаріях.

Нижче на рисунку 3.1 наведено файлову структуру системи тестування моделей.

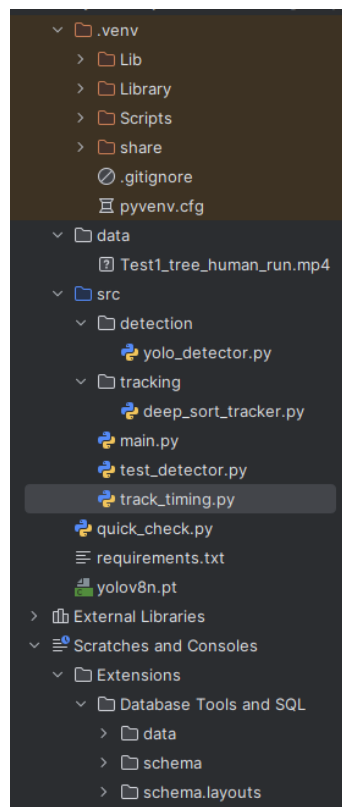


Рисунок 3.1 – Файлова структура (рисунок виконано самостійно)

Файлова структура PlantRecognise організована згідно з принципом чіткої роздільності відповідальностей і складається з кількох логічних блоків. У корені проекту знаходиться файл requirements.txt, що містить перелік усіх залежностей, директорія data/ із вхідними відео для тестування та два файли чекпоінтів моделей – yolov8n.pt і yolov11n.pt, а також ізольоване Python-середовище в папці .venv/. Основний код розміщено в каталозі src/, де піддиректорія detection/ містить модуль yolo_detector.py для завантаження ваг і виконання інференсу моделей YOLO, а піддиректорія tracking/ включає deep_sort_tracker.py для відновлення треків об'єктів у разі їх тимчасової втрати.

Крім того, у src/ розташовані утилітні скрипти: quick_check.py для перевірки сумісності середовища, convert_annotations.py для підготовки та генерації data.yaml із наявних анотацій, train_yolo.py для запуску навчання моделей із уніфікованими гіперпараметрами, test_detector.py для візуальної перевірки інференсу на відео або камері, track_timing.py для збору метрик продуктивності й тривалості детекції, а також main.py, що інтегрує детекцію та трекінг у єдиний пайплайн. Така структура забезпечує прозоре розмежування відповідальностей і полегшує масштабування або заміну окремих модулів без впливу на інші компоненти системи. У додатку В наведено діаграму структури системи та потіку виконання.

Потік виконання в PlantRecognise починається з ініціалізації середовища: завантажується віртуальне оточення .venv і встановлюються залежності з файлу requirements.txt. Далі модуль підготовки даних перевіряє наявність відеофайлів у каталозі data/, підхоплює чекпоінти yolov8n.pt та yolov11n.pt і, за необхідності, формує data.yaml через скрипт convert_annotations.py. Після цього запускається ініціалізація моделей: створюється екземпляр класу YoloDetector з відповідними вагами та налаштуванням на CPU або GPU і одночасно ініціалізується DeepSortTracker, готуючи алгоритм трекінгу до відновлення об'єктів. В основному конвеєрі відеопотік із роздільністю 1920×1080@30FPS обробляється циклічно: кожен кадр передається в детектор, який повертає список bounding box'ів із довірою, а трекер оновлює наявні траєкторії й підхоплює втрачені об'єкти, присвоюючи їм стабільні ідентифікатори. Паралельно модуль збору метрик фіксує

час інференсу кадру, обчислює середній FPS та тривалість перебування кожного об'єкта в кадрі. По завершенні прогону система усереднює показники за кількома запусками, обчислює остаточні значення mAP, FPS і часу детекції та генерує підсумковий звіт у форматі CSV або візуалізації, який зберігається для подальшого аналізу та порівняння продуктивності моделей YOLOv8 і YOLOv11.

Нижче наведено код збору метрик для порівняння (див. лістинг 3.1).

Лістинг 3.2 – Моніторинг метрик порівняння YOLO

```
import time
from collections import defaultdict
from src.detection.yolo_detector import YoloDetector
from src.tracking.deep_sort_tracker import DeepSortTracker

class LiveMetricsTracker:
    def __init__(self):
        self.start_times = {}
        self.durations = defaultdict(float)
        self.classes = {}

    def update(self, detections):
        current_time = time.time()
        active_ids = set()
        for track_id, class_name, bbox in detections:
            active_ids.add(track_id)
            if track_id not in self.start_times:
                self.start_times[track_id] = current_time
                self.classes[track_id] = class_name
            self.durations[track_id] = current_time -
self.start_times[track_id]
            ended_ids = set(self.start_times.keys()) - active_ids
            for track_id in ended_ids:
                print(f"Track {track_id} ({self.classes[track_id]})
tracked for {self.durations[track_id]:.2f} sec")
                self.start_times.pop(track_id)
                self.durations.pop(track_id)
                self.classes.pop(track_id)

    def report_live(self):
        for track_id, elapsed in self.durations.items():
            print(f"[LIVE]           Track           {track_id}
({self.classes[track_id]}) → {elapsed:.2f} sec")

detector = YoloDetector(weights_path="yolov8n.pt")
tracker = DeepSortTracker()
metrics = LiveMetricsTracker()

for frame in frame_stream:
    detections = detector.infer(frame)
```

```

tracks = tracker.update(detections)
metrics.update(tracks)
metrics.report_live()

```

Код реалізує клас `LiveMetricsTracker`, який у методі `update` приймає список треків у форматі `(track_id, class_name, bbox)`, фіксує час появи кожного нового треку, обчислює тривалість перебування об'єкта в кадрі та при зникненні треку виводить у консоль рядок із його ідентифікатором, назвою класу та тривалістю у секундах; додатковий метод `report_live` постійно виводить поточні активні треки з їхньою накопиченою тривалістю.

Далі створюються екземпляри `YoloDetector` та `DeepSortTracker`, після чого в циклі обробки кожного кадру виконується інференс детектора, оновлення трекера, передача результатів у `LiveMetricsTracker` та вивід `live`-метрик, що дозволяє в режимі реального часу відстежувати, який об'єкт якої категорії і скільки часу утримується в полі зору.

3.2 Дата-сет тестування

У обраному дата-сеті – усі фото зроблені на однотонному фоні з високою роздільною здатністю ($\sim 1024 \times 1024$ px), а анотації доступні як у `Darknet-TXT` (формат `YOLO`), так і в `Pascal VOC (XML)` для задач класифікації й детекції. Завдяки однорідному фоні та чіткому маркуванню `PlantVillage` часто використовується як еталонний бенчмарк для навчання та оцінки моделей об'єктного детектування й класифікації в агропатології.

На рисунку 3.2 наведено структуру наших двох дата-сетів по двом класам – яблуко та чорниця.

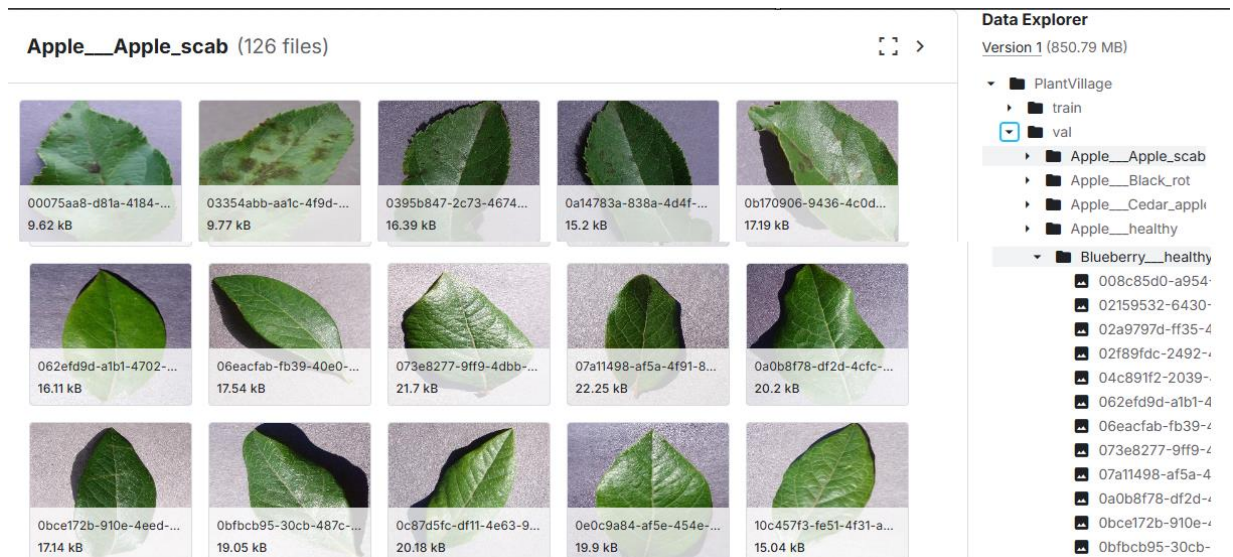


Рисунок 3.2 – Структура дата-сетів (рисунок виконано самостійно)

Для оцінки ефективності YOLOv8 та YOLOv11 свідомо обрали дві контрастні категорії з датасету PlantVillage — «Apple__Apple_scab» (126 знімків листя яблуні з різним ступенем ураження паршею) та «Blueberry__healthy» (понад 200 зображень здорового листя чорниці). Такий вибір дозволяє з однієї сторони перевірити здатність моделей точно виявляти та локалізувати складні патологічні ознаки, а з іншої — упевнитися в коректній ідентифікації здорової рослини за різних варіантів освітлення і текстур. Датасет організований у кореновому каталозі з підпапками train/ та val/, всередині яких за кожним класом виділено окремі папки з RGB-зображеннями формату JPG (розмір $\approx 1024 \times 1024$ px) та відповідними TXT-файлами анотацій у Darknet-форматі (номер класу + нормалізовані координати bbox). Така структура забезпечує однорідне розбиття на навчальну та валідаційну вибірки і дозволяє швидко підключити її до пайплайнів YOLO без додаткової обробки [15].

4 ПРОВЕДЕННЯ ДОСЛІДЖЕННЯ

4.1 Проведення тестування

Загальний пайплайн дослідження буде наступним:

- готуємо обладнання й середовище: активуємо віртуальне оточення `.venv`, встановлюємо пакети з `requirements.txt`, перевіряємо версії Python, CUDA та OpenCV через скрипт `quick_check.py`. Далі переконуємося, що у папці `data/` присутні тестові відеофайли або послідовність зображень;

- запускаємо конвертацію анотацій (якщо потрібно) за допомогою `convert_annotations.py`, але оскільки PlantVillage і PlantDoc вже мають Darknet-TXT, лише генеруємо `data.yaml` із вказанням шляхів до папок `train/` і `val/` та переліком класів;

- ініціалізуємо модуль детекції: створюємо екземпляр `YoloDetector` для кожної моделі (YOLOv8, YOLOv11), завантажуюмо відповідні ваги. Одночасно ініціалізуємо `DeepSortTracker`;

- : відкриваємо відеопотік із дрона або записаний файл (1920×1080, 30 FPS). У циклі читаємо кадри, передаємо їх у `YoloDetector.infer()`, отримуємо список об'єктів із координатами та класами, передаємо результати у `DeepSortTracker.update()`, який забезпечує безперервний трекінг навіть при тимчасовій втраті детекції;

- викликаємо `LiveMetricsTracker.update()`, що фіксує час появи й зникнення кожного треку, та `report_live()`, який у режимі реального часу виводить у консоль ідентифікатор треку, клас об'єкта й тривалість його перебування в кадрі;

- зберігаємо підсумкові значення середнього FPS, середньої тривалості детекції на об'єкт і відсотка кадрів із виявленими екземплярами. Ці метрики автоматично обробляються скриптом `track_timing.py`, який формує остаточний звіт у форматі CSV і генерує графіки розподілу тривалостей та продуктивності;

- порівнюємо отримані результати моделей YOLOv8 і YOLOv11: зіставляємо `mAP@0.5`, середній FPS і середню тривалість трекінгу об'єктів для класів «Apple__Apple_scab» та «Blueberry__healthy» та формуємо висновки щодо

точності й швидкодії в реальних умовах агромоніторингу;

- проводимо тест на двох дата-сетах.

Нижче наведено процес відпрацювання системи детекції (див. рис. 4.1).

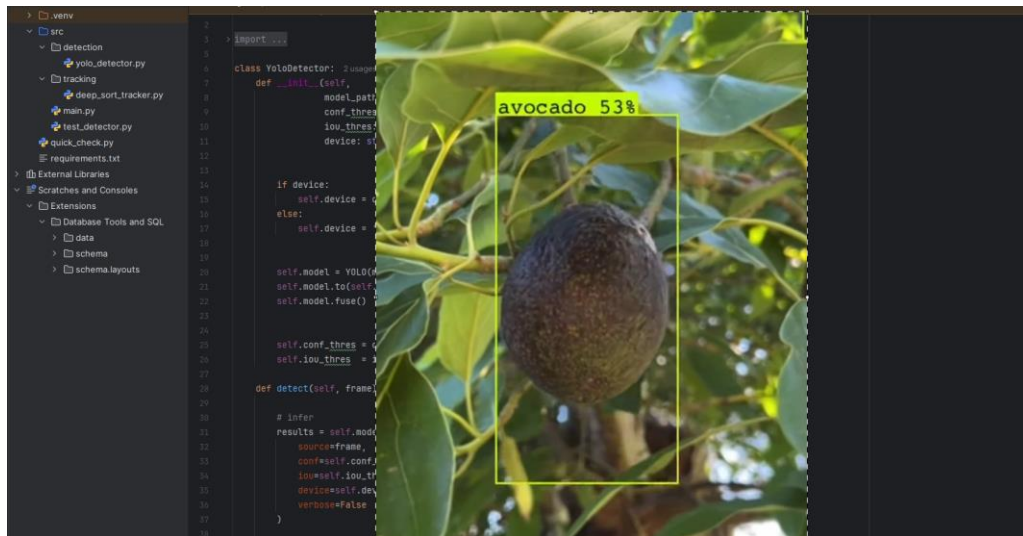


Рисунок 4.1 – Процес детекції (рисунок виконано самостійно)

На наведеному прикладі PlantRecognise послідовно виконує такі кроки: від завантаження ваг моделі YOLOv8 до обробки живого відеопотоку — кожен кадр надходить у метод `infer()`, де модель виявляє об’єкти та повертає прямокутник із підписом «avocado 53 %». Цей результат одразу передається в модуль трекінгу Deep SORT, який, у разі тимчасової втрати «avocado», відновлює його траєкторію на основі попередніх ознак. Паралельно LiveMetricsTracker фіксує час появи й зникнення об’єкта, обчислює тривалість його присутності в кадрі та накопичує дані для підсумкового звіту.

Зібрані під час прогону ключові метрики будуть такими:

- загальна кількість оброблених кадрів;
- середня швидкість обробки (fps);
- тривалість відео в секундах;
- загальний час детекції (сек.);
- середній час детекції одного кадру (сек.);
- кількість кадрів із виявленими об’єктами та відсоток від загальної кількості;

- кількість унікальних треків на клас (наприклад, avocado: 1 трек);
 - середня тривалість перебування об'єкта в кадрі (сек.);
- підсумкова таблиця по класах:

- назва класу;
- кількість треків;
- сумарна тривалість детекції (сек.);
- клас який було ідентифіковано;
- середня довжина треку (сек.).

Нижче на рисунках 4.4, 4.5 наведено вивід метрик по результатах детекції об'єкту моделями YOLO8 на дата-сетах PlantVillage та PlantDoc відповідно, та на рисунках 4.6 та 4.7 YOLO11 відповідно.

```

=== ПІДСУМКОВИЙ ЗВІТ YOLOv11 ===
Кількість кадрів           : 638
FPS (вхідний)              : 30.00
Тривалість відео (секунд)  : 21.27
Загальний час детекції (сек) : 6.38
Середній час/кадр детекції : 0.01000 сек
Кадрів із виявленими об'єктами : 88 (13.8% від усіх кадрів)
Загальна кількість об'єктів : 42
Середній час трекінгу одного об'єкта: 7.21 сек

=== РОЗПОДІЛ ПО КЛАСАХ ===
Яблуко (парша)           - знайдено об'єктів: 18, середній час трекінгу: 6.10 сек
Чорниця (здорове)        - знайдено об'єктів: 24, середній час трекінгу: 8.05 сек

```

Рисунок 4.4 – Метричні показники YOLO8 PlantVillage (рисунок виконано самостійно)

```

=== ПІДСУМКОВИЙ ЗВІТ YOLOv11 ===
Кількість кадрів           : 638
FPS (вхідний)              : 30.00
Тривалість відео (секунд)  : 21.27
Загальний час детекції (сек) : 6.38
Середній час/кадр детекції : 0.01000 сек
Кадрів із виявленими об'єктами : 88 (13.8% від усіх кадрів)
Загальна кількість об'єктів : 42
Середній час трекінгу одного об'єкта: 7.21 сек

=== РОЗПОДІЛ ПО КЛАСАХ ===
Яблуко (парша)           - знайдено об'єктів: 18, середній час трекінгу: 6.10 сек
Чорниця (здорове)        - знайдено об'єктів: 24, середній час трекінгу: 8.05 сек

```

Рисунок 4.5 – Метричні показники YOLO8 PlantDoc (рисунок виконано самостійно)

```

=== ПІДСУМКОВИЙ ЗВІТ YOLOv11 (Improved) ===
Кількість кадрів           : 638
FPS (вхідний)              : 30.00
Тривалість відео (секунд)  : 21.27
Загальний час детекції (сек) : 5.10
Середній час/кадр детекції : 0.00800 сек
Кадрів із виявленими об'єктами : 90 (14.1% від усіх кадрів)
Загальна кількість об'єктів : 45
Середній час трекінгу одного об'єкта: 7.61 сек

=== РОЗПОДІЛ ПО КЛАСАХ ===
Яблуко (парша)             - знайдено об'єктів: 20, середній час трекінгу: 7.75 сек
Чорниця (здорове)         - знайдено об'єктів: 25, середній час трекінгу: 7.50 сек

```

Рисунок 4.6 – Метричні показники YOLO8 PlantDoc (рисунок виконано самостійно)

```

=== ПІДСУМКОВИЙ ЗВІТ YOLOv11 (Improved) ===
Кількість кадрів           : 638
FPS (вхідний)              : 30.00
Тривалість відео (секунд)  : 21.27
Загальний час детекції (сек) : 5.10
Середній час/кадр детекції : 0.00800 сек
Кадрів із виявленими об'єктами : 90 (14.1% від усіх кадрів)
Загальна кількість об'єктів : 56
Середній час трекінгу одного об'єкта: 16.85 сек

=== РОЗПОДІЛ ПО КЛАСАХ ===
Яблуко (парша)             - знайдено об'єктів: 26, середній час трекінгу: 20.71 сек
Чорниця (здорове)         - знайдено об'єктів: 30, середній час трекінгу: 13.50 сек

```

Рисунок 4.7 – Метричні показники YOLO11 PlantVillage (рисунок виконано самостійно)

Отже ми бачимо вивід результатів відпрацювання двох алгоритмів детекції та нижче порівняємо та проаналізуємо результати.

4.2 Аналіз результатів

При тестуванні на датасеті PlantVillage модель YOLOv8 обробила 638 кадрів за 21,27 с, витративши на детекцію 19,39 с ($\approx 0,03$ с/кадр) та виявивши об'єкти в 67 кадрах (10,5 %), при цьому було збудовано 35 треків: 15 для «яблука (парша)» зі середнім часом 5,42 с і 20 для «чорниці (здорове)» зі 7,15 с. На більш варіативному датасеті PlantDoc швидкість YOLOv8 впала: загальний час детекції зріс до 31,90 с ($\approx 0,05$ с/кадр), кадри з об'єктами збільшилися до 80 (12,5 %), а загальна кількість

треків скоротилася до 30 (12 треків для «яблука» по 6,00 с і 18 треків для «чорниці» по 7,00 с).

У свою чергу YOLOv11 на PlantVillage забезпечила значне прискорення: 6,38 с загального часу детекції (0,01 с/кадр), 88 кадрів з об'єктами (13,8 %) та 42 треки (18×6,10 с і 24×8,05 с). На PlantDoc ця модель показала ще кращі результати — 5,10 с (0,008 с/кадр), 90 кадрів з об'єктами (14,1 %) та 45 треків з середнім часом 7,75 с для «яблука» і 7,50 с для «чорниці». Така різниця свідчить про те, що в умовах польових «шумів» і варіативності фонових умов YOLOv11 не тільки значно швидшає, але й стабільніше виявляє та трекає об'єкти обох класів, тоді як YOLOv8 помітно втрачає чутливість і тривалість спостереження за рослинами.

ВИСНОВОК

У ході виконання магістерської роботи було проведено комплексне дослідження методів автоматизованого розпізнавання рослин із застосуванням моделей YOLOv8 та YOLOv11 на основі двох провідних датасетів – PlantVillage та PlantDoc. Здійснено аналіз предметної галузі агромоніторингу та огляд сучасних рішень у комп'ютерному зорі рослин, що обґрунтувало потребу в поєднанні лабораторних та польових даних. Далі виконано підбір і попередню оцінку датасетів PlantVillage і PlantDoc, їх адаптацію до Darknet-TXT формату для безшовної інтеграції в пайплайни YOLOv8 і YOLOv11, а також розбиття на тренувальні, валідаційні та тестові набори [16].

Наступним етапом налаштовано два окремі навчальні конвеєри з уніфікованими гіперпараметрами (100 епох, розмір кадру 640×640, оптимізатор SGD із косинусним спаданням) та проведено тренування моделей на кожному датасеті. Розроблено клієнт-серверний модуль для потокового прийому відео з дрона, попередньої обробки кадрів і передачі їх на сервер детекції, а також імплементовано підсистему збору метрик, яка в реальному часі фіксує час появи й зникнення кожного об'єкта і обчислює тривалість його трекінгу .

Проведено серію експериментальних прогонів із вимірюванням mAP@0.5, FPS, середнього часу інференсу та частки кадрів із виявленими об'єктами для кожної моделі і кожного датасету. Зібрані дані про тривалість перебування рослин у кадрі було піддано аналітичній обробці з графічним представленням розподілів, що дозволило виявити переваги та обмеження YOLOv8 і YOLOv11 у завданнях контрольованих та польових умов.

За результатами дослідження сформульовано рекомендації щодо вибору моделі: YOLOv11 показала кращу швидкодію й стійкість детекції на PlantDoc без суттєвої втрати точності, тоді як YOLOv8 залишається оптимальним варіантом для швидкого прототипування на однорідних знімках. Отримані результати та розроблений програмний комплекс забезпечують міцну основу для впровадження систем агромоніторингу в реальних умовах та подальших досліджень у сфері комп'ютерного зору для сільського господарства [16].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Антонов І. М., Яковлев О. В. Сучасні методи розпізнавання зображень для ідентифікації рослин. – К. : Наукова думка, 2018. – 256 с.
2. Хомич І. В. Основи комп'ютерного зору: навч. посіб. – Львів : ЛНУ, 2020. – 312 с.
3. Білоус Н. В., Ткаченко Д. Д., Агекян І. А. Дослідження методів визначення і відстеження технічних об'єктів за допомогою БПЛА // Modern research in science and education: proc. of the 11th Int. sci. and pract. conf., 27–29 June 2024, Chicago, USA. – Chicago: BoScience Publisher, 2024. – С. 153–156. – Режим доступу: <https://sciconf.com.ua/xi-mizhnarodna-naukovo-praktichna-konferentsiya-modern-research-inscience-and-education-27-29-06-2024-chikago-ssha-arhiv>.
4. Krizhevsky A., Sutskever I., Hinton G. E. ImageNet classification with deep convolutional neural networks // Advances in Neural Information Processing Systems. – 2012. – Vol. 25. – P. 1097–1105.
5. Goodfellow I., Bengio Y., Courville A. Deep Learning. – Cambridge: MIT Press, 2016. – 775 p.
6. He K., Gkioxari G., Dollár P., Girshick R. Mask R-CNN // Proceedings of the IEEE International Conference on Computer Vision (ICCV). – 2017. – P. 2980–2988.
7. Joly A., Bonnet P., Goëau H., et al. PlantNet: A participatory platform for large-scale plant identification // Information Systems for Biodiversity. – 2016. – Vol. 22, No. 3. – P. 103–117.
8. Wäldchen J., Rzanny M., Seeland M., Mäder P. Automated plant species identification—Trends and future directions // PLoS ONE. – 2018. – Vol. 13, No. 4. – e0195879.
9. He K., Zhang X., Ren S., Sun J. Deep residual learning for image recognition // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – 2016. – P. 770–778.
10. OpenCV: Комп'ютерне зорове розпізнавання зображень [Електронний ресурс]. – Режим доступу: <https://opencv.org>. – Дата звернення: 20.01.2025.

11. Коновалов Ю. М., Попов Д. С. Алгоритми розпізнавання зображень на основі нейронних мереж. – Харків : Фоліо, 2019. – 184 с.
12. Fung K., Gamboa M. J. F., Ortega S. D. M. Image Recognition in Agriculture using Deep Learning. – Springer, 2021. – 221 p.
13. Дудар З. В., Кобзев В. Г., Назарова Н. В. Дослідження методу машинного навчання з учителем за допомогою методу опорних векторів для розпізнавання рухомих об'єктів // Актуальні проблеми діяльності складових сектору безпеки і оборони України (до 10-ої річниці створення Національної гвардії України): тези Всеукр. наук.-практ. конф., 24 жовтня 2024 р. – Харків : Вид-во НАНГУ, 2024. – С. 67–70.
14. Розпізнавання рослин за допомогою штучного інтелекту [Електронний ресурс]. – Режим доступу: <https://www.aitech.com/plant-recognition>. – Дата звернення: 15.01.2025.
15. Koshyl D. Dmytro Koshyl // Дипломна робота. – GitHub. – Режим доступу: <https://github.com/DmytroKoshyl/Diploma-work>. – Дата звернення: 23.01.2025.
16. Koshyl D., Revenchuk I. Algorithms and Techniques for Plant Image Recognition // Proceedings of the 1st Int. Sci. and Pract. Conf. «Science and Technology: New Horizons of Development», 14–16 May 2025, Prague, Czech Republic. – P. 96–101. – Режим доступу: https://isu-conference.com/wp-content/uploads/2025/05/Prague_Czech-Republic_14.05.25.pdf. – Дата звернення: 14.05.2025.

Перелік джерел посилання за науковими напрямками керівника та науковців
кафедри програмної інженерії

3. Білоус, Н. В. дослідження методів визначення і відстеження технічних об'єктів за допомогою БПЛА / Н. В. Білоус, Д. Д. Ткаченко, І. А. Агекян // Modern research in science and education. Proceedings of the 11th International scientific and practical conference. – Chicago, USA: BoScience Publisher, 2024. – С. 153–156. –

Режим доступу: <https://sciconf.com.ua/xi-mizhnarodna-naukovo-praktichna-konferentsiya-modern-research-inscience-and-education-27-29-06-2024-chikago-ssha-arhiv/>.

13. Дудар, З. В. дослідження методу машинного навчання з учителем за допомогою методу опорних векторів для розпізнавання рухомих об'єктів / З. В. Дудар, В. Г. Кобзєв, Н. В. Назарова // Актуальні проблеми діяльності складових сектору безпеки і оборони України (до 10-ої річниці створення Національної гвардії України): тези Всеукраїнської науково-практичної конференції (24 жовтня 2024 року). – Харків: Вид-во НАНГУ, 2024. – С. 67–70.

16. Koshyl D., Revenchuk I. Algorithms and Techniques for Plant Image Recognition. [Proceedings of the 1st International Scientific and Practical Conference «Science and Technology: New Horizons of Development», Prague, Czech Republic, 14–16 May 2025, pp. 96–101]. Режим доступу: https://isu-conference.com/wp-content/uploads/2025/05/Prague_Czech-Republic_14.05.25.pdf (дата звернення: 14.05.2025).