

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук
(повна назва)
Кафедра _____ Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти _____ перший (бакалаврський)

_____ Інтелектуальна система визначення типу ураження листя рослин
_____ за зображенням
(тема)

Виконав:
здобувач _____ четвертого _____ року навчання,
групи _____ ІТШ-22-4

_____ Анастасія ГРИГОРОВА
(власне ім'я, прізвище)

Спеціальність 122 Комп'ютерні науки

(код і повна назва спеціальності)

Тип програми _____ освітньо-професійна

Освітня програма _____ Штучний інтелект

(повна назва освітньої програми)

Керівник _____ ас. Максим ПОЛІТ
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ШІ

(підпис)

_____ Лариса ЧАЛА
(власне ім'я, прізвище)

2026 р.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____

Кафедра _____ Штучного інтелекту _____

Рівень вищої освіти _____ перший (бакалаврський) _____

Спеціальність _____ 122 Комп'ютерні науки _____
(код і повна назва)

Тип програми _____ освітньо-професійна _____

Освітня програма _____ Штучний інтелект _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

« _____ » _____ 20 ____ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ Григоровій Анастасії Андріївні _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Інтелектуальна система визначення типу ураження листя рослин за зображенням _____

затверджена наказом університету від 5 червня 2026 р. № 488Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 8 липня 2026 р.

3. Вихідні дані до роботи _____ Наукові публікації з тематики комп'ютерного зору та класифікації зображень, документація бібліотек Python (TensorFlow, Keras, NumPy, Matplotlib, scikit-learn, OpenCV, Streamlit), набір даних Plant Leaf Disease MasterDataset (19 Crops) для навчання та тестування моделей класифікації уражень листя рослин _____

4. Перелік питань, що потрібно опрацювати в роботі _____

1) Аналіз предметної галузі та постановка задачі дослідження _____

2) Методи та моделі інтелектуального аналізу зображень _____

3) Розробка інтелектуальної системи діагностики ураження листя _____

4) Програмна реалізація та експериментальне дослідження _____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / термін виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	08.06.2026	виконано
2	Аналіз предметної галузі та постановка задачі	08.06.2026–10.06.2026	виконано
3	Теоретичне дослідження методів аналізу зображень	10.06.2026–12.06.2026	виконано
4	Вибір архітектур моделей, підбір та підготовка даних	13.06.2026–15.06.2026	виконано
5	Розробка та навчання моделей класифікації	15.06.2026–19.06.2026	виконано
6	Проведення експериментів та оцінювання результатів	20.06.2026–22.06.2026	виконано
7	Розробка застосунку на Streamlit та інтеграція компонентів	22.06.2026–24.06.2026	виконано
8	Оформлення пояснювальної записки	25.06.2026–27.06.2026	виконано
9	Нормоконтроль	28.06.2026–30.06.2026	виконано
10	Рецензування	01.07.2026	виконано
11	Попередній захист	04.07.2026	виконано
12	Захист перед ЕК	08.07.2026	

Дата видачі завдання 8 червня 2026 р.

Здобувач _____

(підпис)

Керівник роботи _____

(підпис)

ас. Максим ПОЛІТ

(посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 86 с., 51 рис., 1 дод., 33 джерела.

ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ, КЛАСИФІКАЦІЯ ЗОБРАЖЕНЬ, КОМП'ЮТЕРНИЙ ЗІР, УРАЖЕННЯ ЛИСТЯ РОСЛИН, EFFICIENTNET-B0, GRAD-CAM, MOBILE NETV2, RESNET50, STREAMLIT.

Об'єкт дослідження – процес аналізу зображень листя рослин для визначення їхнього стану.

Предмет дослідження – методи автоматичної класифікації зображень листя рослин за типом ураження з використанням згорткових нейронних мереж.

Мета роботи – розробка інтелектуальної системи визначення типу ураження листя рослин за зображенням із використанням моделей глибокого навчання.

Методи дослідження – згорткові нейронні мережі (MobileNetV2, ResNet50, EfficientNet-B0), transfer learning, методи оцінювання якості класифікації (accuracy, macro F1-міра, матриця помилок), метод Grad-CAM для візуалізації процесу прийняття рішення.

У роботі реалізовано інтелектуальну систему, що здійснює класифікацію зображень листя рослин за типом ураження. Проведено навчання та порівняння трьох архітектур нейронних мереж на одному наборі даних з оцінюванням якості моделей. Для підвищення інтерпретованості результатів застосовано метод Grad-CAM. Для демонстрації роботи системи підготовлено застосунок на Streamlit, який реалізує завантаження зображення, класифікацію, відображення ймовірності прогнозу, візуалізацію Grad-CAM та виведення текстової рекомендації.

ABSTRACT

Bachelor's thesis contains: 86 pp., 51 fig., 1 ann., 33 references.

CONVOLUTIONAL NEURAL NETWORKS, EFFICIENTNET-B0, GRAD-CAM, IMAGE CLASSIFICATION, MOBILENETV2, PLANT LEAF DISEASE, RESNET50, STREAMLIT, COMPUTER VISION.

Object of research – the process of analyzing plant leaf images to determine their condition.

Subject of research – methods for automatic classification of plant leaf images by type of disease using convolutional neural networks.

The purpose of the work is to develop an intelligent system for identifying the type of plant leaf disease from an image using deep learning models.

Research methods – convolutional neural networks (MobileNetV2, ResNet50, EfficientNet-B0), transfer learning, classification quality evaluation methods (accuracy, macro F1-score, confusion matrix), Grad-CAM method for visualizing the decision-making process.

The work implements an intelligent system for classifying plant leaf images by disease type. Three neural network architectures were trained and compared on the same dataset with performance evaluation. To improve interpretability, the Grad-CAM method was applied. A Streamlit-based application was developed to demonstrate the system, providing image upload, classification, prediction probability visualization, Grad-CAM visualization, and textual recommendations.

ЗМІСТ

Вступ.....	8
1 Аналіз предметної галузі та постановка задачі дослідження	9
1.1 Особливості ураження листя рослин та їх візуальні ознаки	9
1.2 Аналіз існуючих підходів та рішень	13
1.3 Постановка задачі дослідження.....	28
2 Методи та моделі інтелектуального аналізу зображень	30
2.1 Основи обробки зображень.....	30
2.2 Класичні методи аналізу зображень	32
2.3 Згорткові нейронні мережі.....	34
2.4 Transfer learning	37
2.5 Огляд сучасних архітектур CNN	40
2.6 Метрики оцінювання моделей	45
2.7 Методи пояснювального штучного інтелекту	48
3 Розробка інтелектуальної системи діагностики ураження листя.....	50
3.1 Архітектура системи.....	50
3.2 Підготовка та обробка даних	51
3.3 Вибір та конфігурація моделей.....	54
3.4 Процес навчання моделей	55
3.5 Реалізація пояснень рішень моделі (Grad-CAM)	56
3.6 Формування рекомендацій на основі результату класифікації.....	57
4 Програмна реалізація та експериментальне дослідження	59
4.1 Засоби та середовище розробки	59
4.2 Параметри експерименту та налаштування моделей	60
4.3 Результати навчання моделей.....	61
4.4 Порівняльний аналіз моделей.....	63
4.5 Аналіз помилок моделей	64
4.6 Візуалізація рішень моделі.....	72
4.7 Демонстрація роботи системи	74

Висновки	80
Перелік джерел посилання	82
Додаток А Відомість кваліфікаційної роботи	86

ВСТУП

Рослини відіграють важливу роль у житті людини, виконуючи як екологічні, так і господарські функції. Вони є джерелом харчових продуктів, сировини для промисловості, а також виконують важливі екологічні функції, пов'язані з підтриманням балансу природних екосистем. Здоровий стан рослин безпосередньо впливає на їх ріст, розвиток та здатність виконувати свої функції.

Одним із ключових факторів, що впливають на стан рослин, є різноманітні захворювання та пошкодження. Вони можуть бути спричинені грибковими, бактеріальними або вірусними інфекціями, а також впливом несприятливих умов навколишнього середовища. Подібні ураження часто проявляються у вигляді змін кольору листя, появи плям, деформацій або відмирання тканин. Листя при цьому виступає одним із найбільш інформативних елементів рослини, оскільки саме на його поверхні найчастіше проявляються характерні ознаки порушення нормального стану рослини.

Своєчасне виявлення таких змін є важливим для запобігання подальшому поширенню захворювань та погіршенню стану рослин. Проте візуальна діагностика може бути складною через різноманітність проявів уражень та подібність симптомів різних захворювань. У зв'язку з цим виникає потреба у застосуванні сучасних технологій, які дозволяють автоматизувати процес аналізу стану рослин на основі зображень та підвищити точність їх оцінювання.

Сучасний розвиток технологій комп'ютерного зору та методів штучного інтелекту відкриває нові можливості для автоматичного аналізу зображень рослин. Використання таких підходів дозволяє створювати системи, здатні аналізувати фотографії листя та виявляти характерні ознаки уражень, що значно спрощує процес первинної діагностики стану рослин.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Особливості ураження листя рослин та їх візуальні ознаки

Листя є одним із найбільш інформативних елементів рослини, за станом якого можна оцінити її загальне здоров'я. Саме на поверхні листя найчастіше проявляються перші ознаки порушення нормального розвитку рослини. Зміни кольору, поява плям, деформацій або пошкоджених ділянок можуть сигналізувати про наявність захворювань, шкідників або несприятливих умов середовища. У фітопатології симптоми захворювань рослин зазвичай оцінюються шляхом візуального аналізу рослинних тканин, зокрема листя, оскільки саме на ньому формуються характерні зовнішні прояви інфекційних уражень. Візуальні симптоми можуть проявлятися у вигляді локальних плям, пожовтіння, водянистих уражень або інших змін структури тканин листка (рисунок 1.1). Такі прояви є важливим джерелом інформації для оцінювання стану рослин і широко використовуються під час дослідження захворювань у рослинництві.



Рисунок 1.1 – Приклади візуальних симптомів ураження листя рослин [1]

Причини ураження листя рослин можуть бути різноманітними та мати як інфекційний, так і неінфекційний характер. До інфекційних причин належать захворювання, викликані різними типами патогенних організмів, зокрема грибами, бактеріями та вірусами [2]. Такі патогени проникають у тканини рослини, порушують нормальне функціонування клітин і призводять до змін у структурі та фізіологічному стані листя. У результаті цього на поверхні листка можуть з'являтися характерні симптоми, пов'язані з ураженням тканин, розвитком некротичних ділянок або порушенням процесів фотосинтезу. Більшість інфекційних захворювань рослин супроводжується появою специфічних візуальних проявів, що дозволяє використовувати стан листя як один із основних індикаторів наявності проблеми.

Разом із тим ураження листя може виникати і під впливом неінфекційних факторів, які не пов'язані з діяльністю патогенних організмів [3]. До таких факторів належать порушення умов вирощування рослин, зокрема нестача або надлишок поживних речовин у ґрунті, недостатній або надмірний полив, несприятливі температурні умови, а також недостатній або надлишковий рівень освітлення. Подібні чинники можуть призводити до порушення обміну речовин у рослині, що проявляється у вигляді змін кольору листя, уповільнення росту або поступового відмирання окремих ділянок тканин. Крім того, пошкодження листя можуть бути спричинені механічними впливами, наприклад ушкодженнями, що виникають унаслідок дії шкідників, вітру або інших зовнішніх факторів.

Незалежно від природи виникнення таких уражень, на поверхні листя зазвичай формуються зовнішні ознаки. Візуальні ознаки ураження листя можуть проявлятися у різних формах і є важливими індикаторами стану рослини.

До найбільш поширених симптомів належать зміни кольору тканин листка, поява локальних плям, пожовтіння, некроз окремих ділянок, а також

деформація або скручування листя. Наприклад, одним із типових симптомів є хлороз, який проявляється у вигляді пожовтіння тканин листка внаслідок порушення процесів фотосинтезу. Іншим поширеним проявом є некроз – відмирання тканин, що супроводжується утворенням темних або бурих ділянок на поверхні листка. Такі ураження можуть формувати чітко окреслені плями або поширюватися на значну частину листкової пластинки (рисунок 1.2).



Рисунок 1.2 – Хлороз та некротичні плями листкової поверхні

Крім того, для багатьох захворювань характерним є утворення локальних уражених ділянок (lesions) або плям, які можуть мати різну форму, розмір і колір. У деяких випадках навколо некротичних плям утворюється характерна жовта або світла облямівка, що є типовим симптомом бактеріальних або грибкових інфекцій [2]. Подібні прояви можуть поступово збільшуватися, зливаючись між собою та призводячи до

значного пошкодження листкової поверхні. У разі сильного ураження це може викликати передчасне відмирання листя або зниження здатності рослини до фотосинтезу.

Різноманітність таких візуальних симптомів пояснюється особливостями розвитку захворювань та реакцією рослинних тканин на вплив патогенів або несприятливих факторів середовища. Саме характерні зміни кольору, форми та структури поверхні листя використовуються у фітопатології як основні ознаки для попереднього визначення типу ураження рослин [1]. Завдяки цьому аналіз зображень листя може виступати ефективним інструментом для виявлення та дослідження захворювань рослин.

Важливо зазначити, що подібні візуальні ознаки можуть бути притаманні різним захворюванням або порушенням стану рослин. Наприклад, пожовтіння листя може виникати як унаслідок розвитку інфекційного захворювання, так і через нестачу поживних речовин або неправильний режим поливу [2]. Подібна ситуація спостерігається і з іншими симптомами, такими як поява плям, в'янення або деформація листкової пластинки, які можуть бути спричинені різними чинниками. У результаті схожі зовнішні прояви іноді мають різну природу виникнення, що ускладнює їх точну інтерпретацію під час візуального огляду. Через це визначення причини ураження лише на основі зовнішнього вигляду листя може бути складним і потребує відповідних знань, практичного досвіду або проведення додаткового аналізу.

Крім того, на сприйняття візуальних ознак можуть впливати різні зовнішні фактори. До них належать умови освітлення, кут зйомки, фон зображення, а також індивідуальні особливості рослини. Навіть одна й та сама проблема може виглядати по-різному на різних рослинах або на різних стадіях розвитку захворювання. Це створює додаткові труднощі під час аналізу зображень листя.

Таким чином, листя є важливим джерелом інформації про стан рослини, оскільки саме на ньому найчастіше проявляються характерні ознаки ураження. Разом із тим різноманітність візуальних проявів та складність їх інтерпретації ускладнюють процес діагностики лише на основі візуального огляду. Це зумовлює необхідність застосування сучасних підходів до аналізу зображень рослин, які дозволяють автоматизувати процес виявлення уражень та підвищити точність їх визначення.

1.2 Аналіз існуючих підходів та рішень

Традиційно визначення стану рослин здійснюється експертним шляхом, тобто за участю фахівця, який на основі зовнішнього вигляду листя встановлює наявність ураження та його тип. Такий підхід ґрунтується на візуальному аналізі характерних ознак, зокрема кольору листка, наявності плям, їх форми, розміру та розташування, а також додаткових проявів, таких як деформації або наліт. У ряді випадків враховуються також супутні фактори, зокрема умови вирощування, рівень вологості та освітлення.

Попри широке застосування, експертна діагностика має низку обмежень. По-перше, точність результату значною мірою залежить від досвіду та кваліфікації спеціаліста. По-друге, візуальні ознаки різних захворювань можуть бути подібними, що ускладнює однозначну ідентифікацію. Крім того, одна й та сама хвороба може проявлятися по-різному залежно від стадії розвитку або зовнішніх умов. Також варто враховувати, що такий підхід є трудомістким і не дозволяє швидко обробляти велику кількість рослин.

У зв'язку з цим виникає потреба у застосуванні більш автоматизованих і масштабованих підходів до діагностики стану рослин, що і зумовило активний розвиток методів комп'ютерного аналізу зображень.

У роботі [4] запропоновано метод автоматичної класифікації захворювань листя рослин на основі покращеної згорткової нейронної

мережі. Основна ідея полягає у підвищенні точності розпізнавання та зменшенні обчислювальної складності моделі шляхом модифікації класичної CNN-архітектури.

Архітектура моделі базується на поєднанні кількох підходів глибокого навчання. Зокрема, використано згорткові шари VGG16 для виділення ознак різного рівня абстракції, модуль Inception для багатомасштабного аналізу ознак, а також модуль Squeeze-and-Excitation (SE), який дозволяє адаптивно зважувати важливість каналів ознак. Додатково повнозв'язні шари замінено на глобальний середній pooling (GAP), що дозволило суттєво зменшити кількість параметрів та прискорити навчання моделі.

На рисунку 1.3 наведено загальну архітектуру запропонованої моделі, яка включає послідовність згорткових шарів, Inception-блок, SE-модуль та шар глобального усереднення. Така структура забезпечує ефективне поєднання локальних і глобальних ознак, а також покращує здатність моделі до узагальнення.

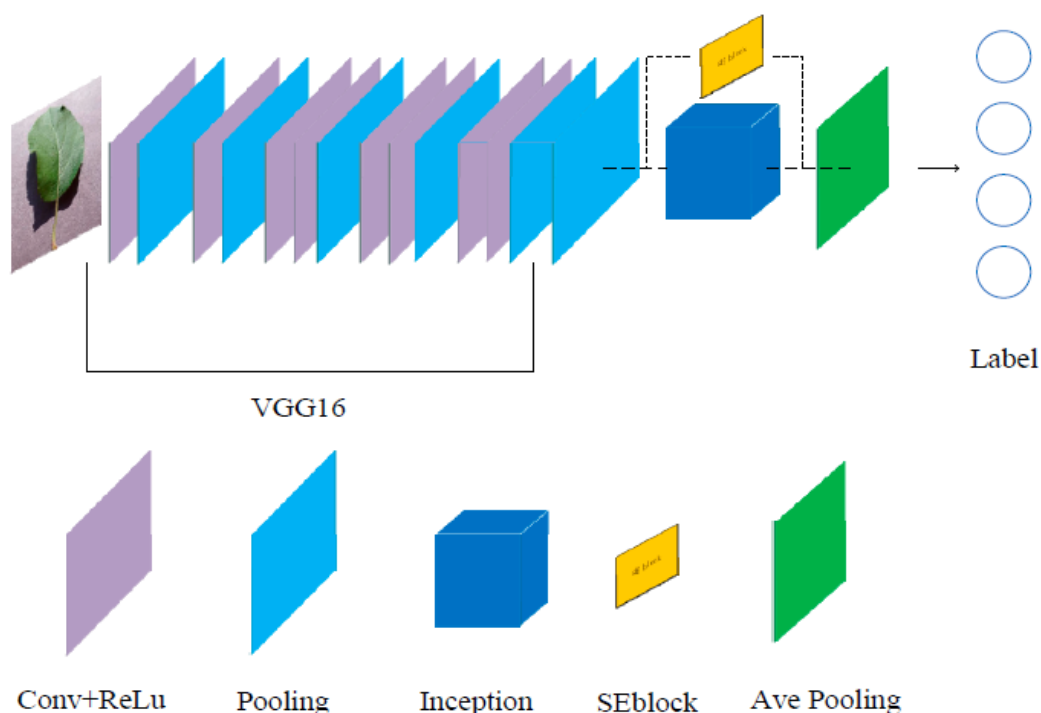


Рисунок 1.3 – Структура запропонованої згорткової нейронної мережі [4]

Для навчання використано набір даних із зображеннями 10 типів захворювань листя, який після аугментації містить понад 6000 зображень. Це дозволило підвищити збалансованість вибірки та якість навчання моделі.

Результати експериментів показали, що запропонований підхід досягає точності 91.7% на тестовому наборі даних, що перевищує показники класичних моделей, таких як AlexNet, VGG16, ResNet та Inception. Крім того, модель демонструє менший розмір та швидший час навчання завдяки використанню глобального pooling замість повнозв'язних шарів.

Подальші дослідження були спрямовані на підвищення ефективності моделей шляхом оптимізації їх архітектури. Зокрема, у роботі [5] використано архітектуру EfficientNet, яка забезпечує кращий баланс між точністю та обчислювальними витратами. Основна ідея полягає у відмові від ручного виділення ознак і застосуванні end-to-end моделі, здатної автоматично навчатися на сирих зображеннях.

Для навчання використано датасет PlantVillage, що містить понад 55 тис. зображень (та понад 61 тис. після аугментації). Усі моделі навчалися із використанням підходу transfer learning із донавчанням усіх шарів.

Результати експериментів показали, що моделі EfficientNet-B4 та EfficientNet-B5 досягли найкращих показників серед розглянутих підходів: точність до 99.97% та precision до 99.39%, що значно перевищує результати інших сучасних моделей глибокого навчання.

Окрім цього, актуальним напрямом є зменшення обчислювальної складності моделей. У роботі [6] запропоновано легковагу глибоку модель для виявлення захворювань рослин на основі модифікованої архітектури MobileNetV3Large. Основна ідея полягає у створенні ефективної та обчислювально економної системи, придатної для використання на мобільних пристроях у реальних польових умовах.

Архітектура моделі базується на попередньо натренованому MobileNetV3Large (рисунок 1.4) з доданою кастомною класифікаційною частиною, глобальним pooling та fine-tuning окремих шарів. Для підвищення

узагальнювальної здатності використано аугментацію даних та гібридний датасет, що поєднує реальні польові зображення та відкриті набори даних.

Результати експериментів показали, що запропонована модель досягла точності 99.37%, перевищивши базову MobileNetV3Large, EfficientNetB1 та класичну CNN-модель, що свідчить про високу ефективність підходу.

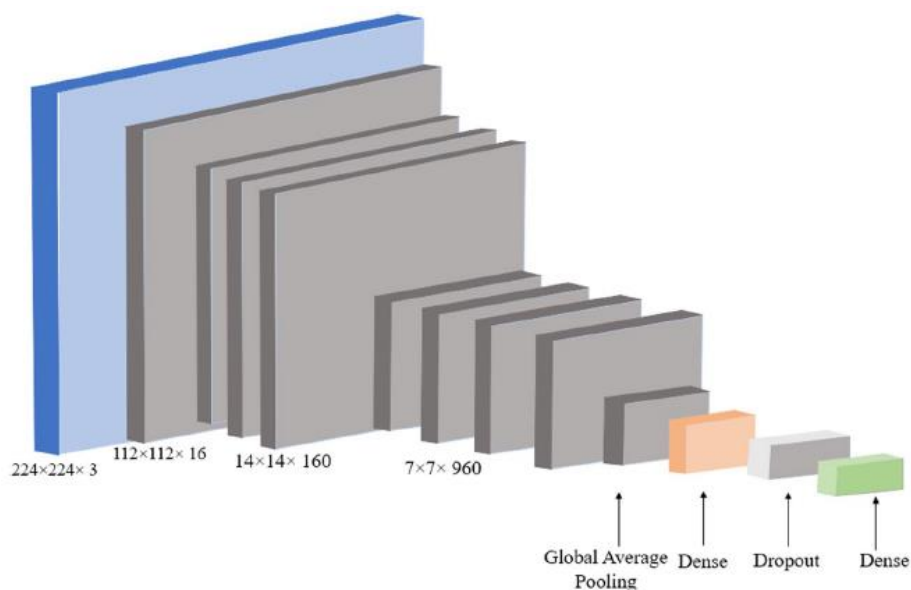


Рисунок 1.4 – Архітектура MobileNetV3Large [6]

Для підвищення якості класифікації також застосовуються механізми фокусування уваги моделі. У роботі [7] запропоновано інтерпретовану модель для виявлення захворювань листя рослин на основі архітектури CBAM-VGG16. Основна ідея полягає у поєднанні високої точності класифікації з можливістю пояснення рішень моделі за допомогою механізмів уваги та методів Explainable AI.

Архітектура моделі базується на VGG16, у яку інтегровано модулі уваги CBAM після кожного згорткового блоку. Це дозволяє моделі фокусуватись на важливих просторових та каналових ознаках. Додатково застосовано методи інтерпретації, зокрема Grad-CAM, Grad-CAM++ та LRP, що забезпечують візуалізацію важливих ділянок зображення.

На рисунку 1.5 наведено загальну схему роботи системи, яка включає етапи попередньої обробки даних, класифікації та пояснення результатів.

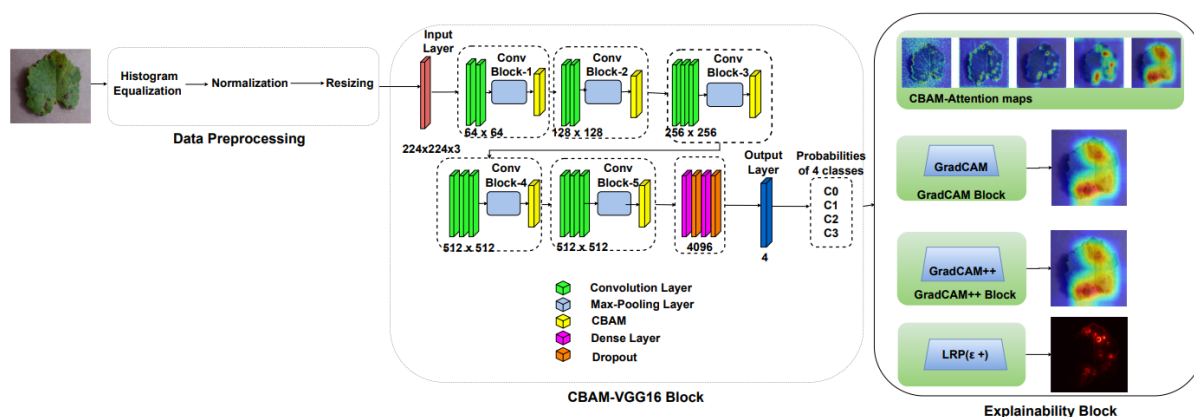


Рисунок 1.5 – Загальна схема системи класифікації захворювань рослин із використанням CBAM-VGG16 [7]

Результати експериментів показали високу ефективність моделі: точність до 98.87% на різних датасетах, а також хорошу узагальнювальну здатність. Крім того, використання attention-механізмів та XAI-підходів дозволило зробити модель прозорою та придатною для практичного застосування.

Подібний підхід розвивається і в інших дослідженнях. Зокрема, у роботі [8] запропоновано гібридну архітектуру CNN-SEEIB, що поєднує згорткові нейронні мережі з механізмом уваги Squeeze-and-Excitation. Основна ідея полягає в інтеграції SE-блоків у ідентифікаційні (residual) блоки для підсилення значущих ознак та покращення класифікації захворювань рослин.

Архітектура моделі включає послідовність згорткових та identity-блоків, доповнених механізмами нормалізації, pooling та глобального усереднення. Використання SE-механізму дозволяє моделі адаптивно переважувати канали ознак, фокусуючись на найбільш інформативних характеристиках зображення.

На рисунку 1.6 наведено загальну структуру моделі та принцип інтеграції SE-блоків у нейронну мережу.

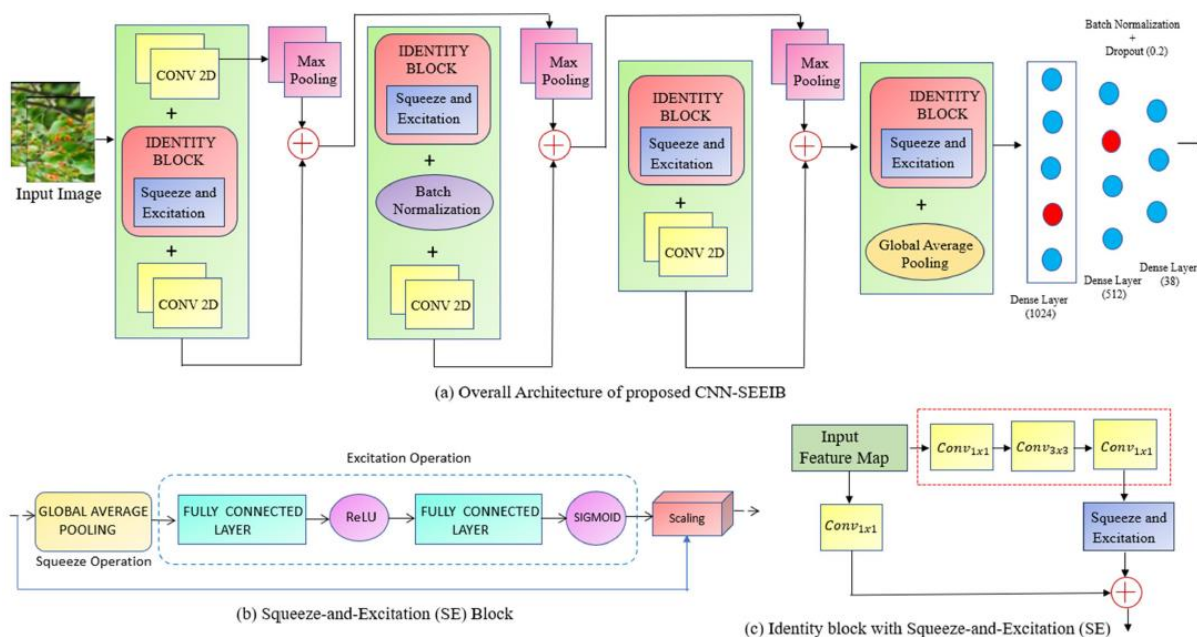


Рисунок 1.6 – Архітектура запропонованої моделі CNN-SEEIB для класифікації захворювань рослин [8]

Результати експериментів показали високу ефективність підходу: точність класифікації становить 99.79%, а також досягнуто високих значень precision, recall та F1-міри. Крім того, модель демонструє хорошу обчислювальну ефективність і придатність для використання в умовах обмежених ресурсів, зокрема на edge-пристроях.

Поряд із задачами класифікації, деякі дослідники змінюють саму постановку задачі. У роботі [9] запропоновано підхід до виявлення захворювань рослин на основі детекції об'єктів із використанням моделі YOLOv4. Основна ідея полягає у переході від простої класифікації зображення до локалізації уражених ділянок листя, що дозволяє точніше ідентифікувати симптоми хвороб.

Архітектура моделі базується на YOLOv4, яка забезпечує одночасне виявлення та класифікацію об'єктів у реальному часі. Такий підхід дозволяє

не лише визначити тип захворювання, але й виділити конкретні області ураження на зображенні. Для навчання використано анотовані зображення листя з позначеними хворобами, що забезпечує формування просторово чутливих ознак.

На рисунку 1.7 наведено загальний процес роботи моделі, включаючи етапи підготовки даних, навчання та детекції об'єктів.

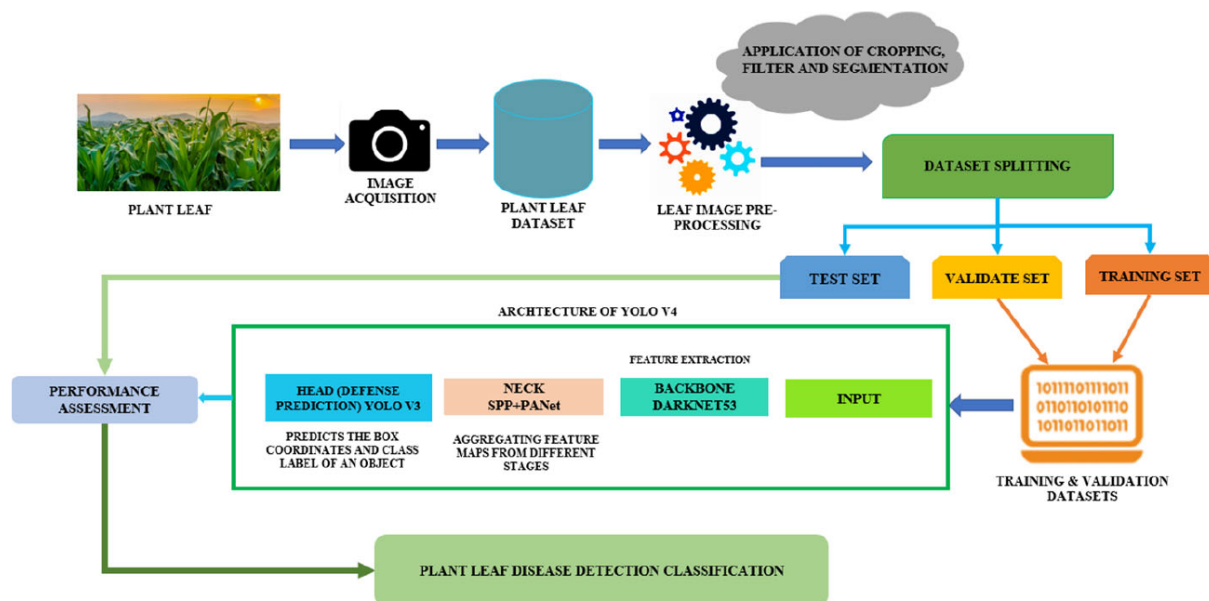


Рисунок 1.7 – Загальна схема реалізації підходу YOLOv4 для виявлення та ідентифікації хвороб листя рослин [9]

Результати експериментів показали, що запропонований підхід забезпечує високу точність детекції та дозволяє ефективно працювати в умовах реального часу. Крім того, використання YOLOv4 значно підвищує практичну цінність системи завдяки можливості локалізації уражених ділянок, а не лише їх класифікації.

Даний підхід отримав подальший розвиток. Зокрема, у роботі [10] поєднано модель YOLOv8 із алгоритмом оптимізації, що дозволяє підвищити точність виявлення захворювань.

У роботі запропоновано модель YOLOv8-POA, у якій базова архітектура YOLOv8 доповнюється метаевристичним алгоритмом оптимізації (Parrot Optimization Algorithm). Основна ідея полягає в оптимізації ваг нейронної мережі з використанням поведінкової моделі папуг (foraging, communication тощо), що дозволяє ефективніше знаходити оптимальні параметри моделі та підвищувати якість детекції.

Архітектура YOLOv8 складається з трьох основних компонентів: backbone (виділення ознак), neck (агрегація ознак різних рівнів) та head (детекція об'єктів), що забезпечує швидку та точну локалізацію уражених ділянок. У запропонованому підході після початкового навчання YOLOv8 застосовується POA, який генерує множину кандидатних ваг та ітеративно покращує їх на основі метрики mAP.

На рисунку 1.8 наведено загальний процес роботи моделі, включаючи етапи підготовки даних, навчання YOLOv8, оптимізації ваг за допомогою POA та оцінювання результатів.

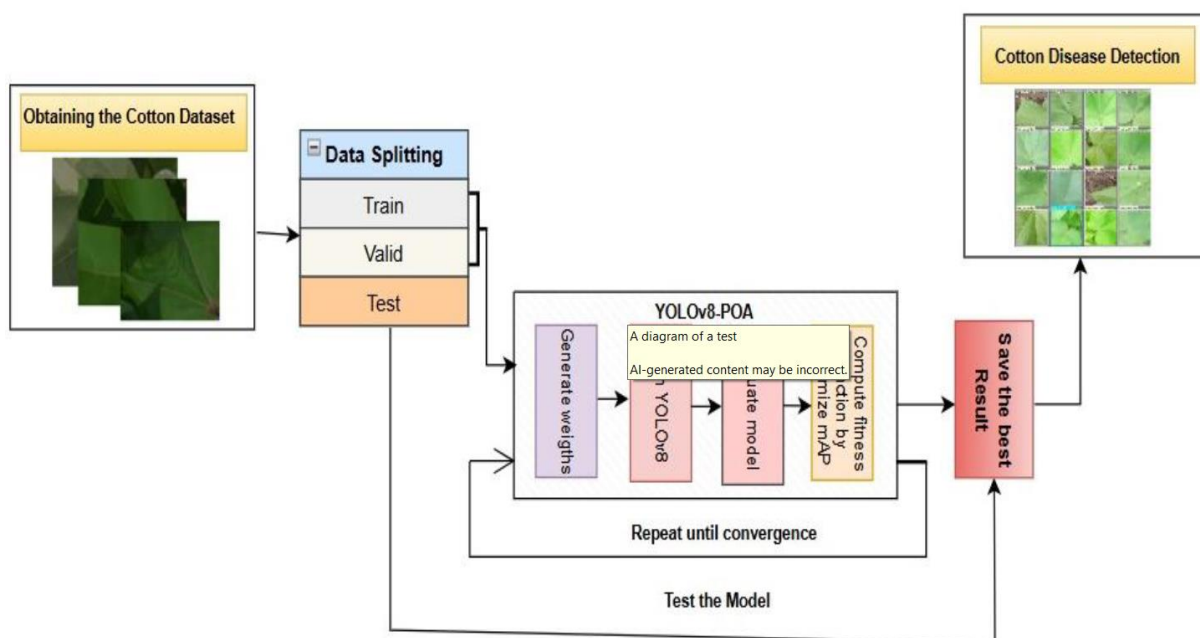


Рисунок 1.8 – Методологія запропонованої моделі YOLOv8-POA для виявлення захворювань бавовни [10]

Експериментальні результати показали, що запропонована модель досягає високих показників: значення $mAP@50$ становить 97.1% для навчальної вибірки, 97.0% для валідаційної та 96.3% для тестової, що підтверджує ефективність підходу порівняно з базовою моделлю YOLOv8 та іншими методами оптимізації. Крім того, модель демонструє здатність до роботи в реальному часі, що є важливим для практичного застосування в задачах моніторингу стану рослин.

В умовах обмеженої кількості навчальних даних застосовуються альтернативні підходи до навчання моделей. У роботі [11] запропоновано підхід *few-shot learning* (FSL) для класифікації захворювань рослин, який базується на вдосконаленому методі *Deep Brownian Distance Covariance* (DeepBDC). Основна ідея полягає у формуванні узагальнених представлень ознак навіть за наявності обмеженої кількості прикладів кожного класу.

Архітектура моделі включає *backbone* ResNet-12, доповнений модулем уваги CBAM, що дозволяє фокусуватися на найбільш інформативних просторових і каналових ознаках. Для зменшення перенавчання використано Dropout. Модуль DeepBDC вдосконалено за рахунок L2-нормалізації та температурного масштабування, що забезпечує стабільніші представлення ознак. Для обчислення подібності між зразками застосовуються різні метрики: косинусна подібність для 1-shot задач та гаусівське ядро для 5-shot і 10-shot налаштувань.

На рисунку 1.9 наведено загальну схему підходу, яка включає обробку *query* та *support* зображень, формування прототипів класів та подальшу класифікацію за допомогою функцій подібності.

Експериментальні результати показали, що на датасеті PlantVillage модель досягає точності 46.53% (1-shot), 65.86% (5-shot) та 69.67% (10-shot), а на датасеті CSMТ – 40.19%, 50.19% та 53.67% відповідно. Отримані результати підтверджують ефективність запропонованого підходу для задач класифікації захворювань рослин в умовах обмежених даних.

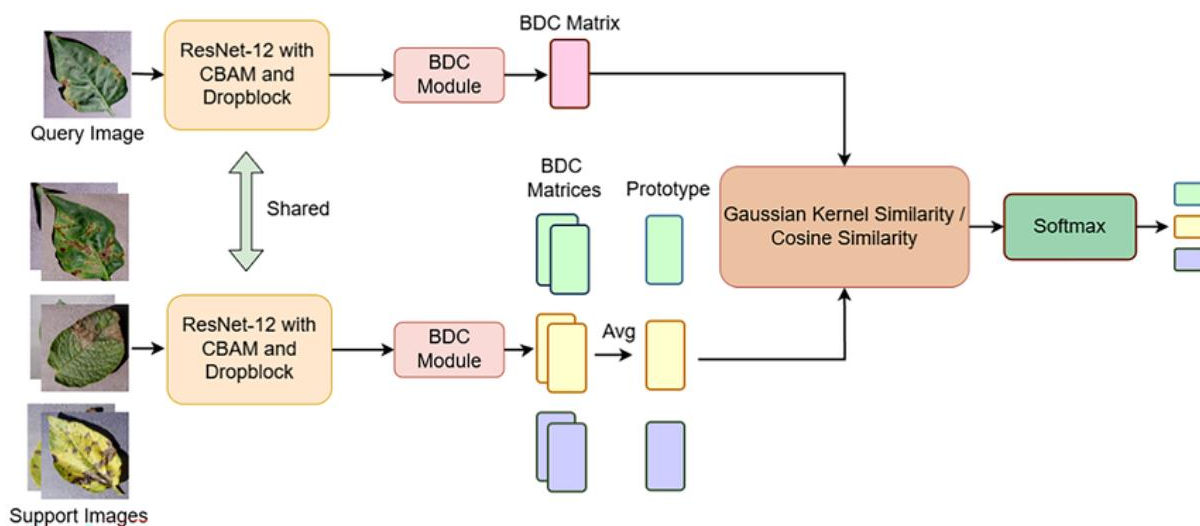


Рисунок 1.9 – Архітектура моделі DeepBDC для few-shot класифікації захворювань рослин [11]

Окрім традиційних згорткових мереж, досліджуються і принципово нові архітектури. У роботі [12] запропоновано архітектуру U-PKAN, яка поєднує encoder–decoder структуру з механізмами KAN. Основна ідея полягає у використанні нелінійного функціонального представлення, де замість класичних активацій застосовуються навчані функції, що дозволяє точніше апроксимувати складні залежності та підвищити інтерпретованість моделі. Архітектура включає два спеціалізовані модулі: P-KAN (для виділення та перетворення ознак у вигляді послідовностей із урахуванням просторових залежностей) та D-KAN (для відновлення просторової роздільної здатності та багаторівневої агрегації ознак).

На рисунку 1.10 наведено загальну архітектуру моделі, яка реалізує encoder-decoder підхід із використанням skip-з'єднань, KAN-блоків та операцій апсемплінгу.

Результати експериментів показали, що запропонована модель забезпечує стабільно високі показники сегментації: значення IoU знаходяться в межах 81.5–85.3%, а Dice – 89.8–92.2% для зображень різних культур та захворювань. Це підтверджує ефективність моделі при роботі з реальними даними та її здатність до узагальнення в різних умовах.

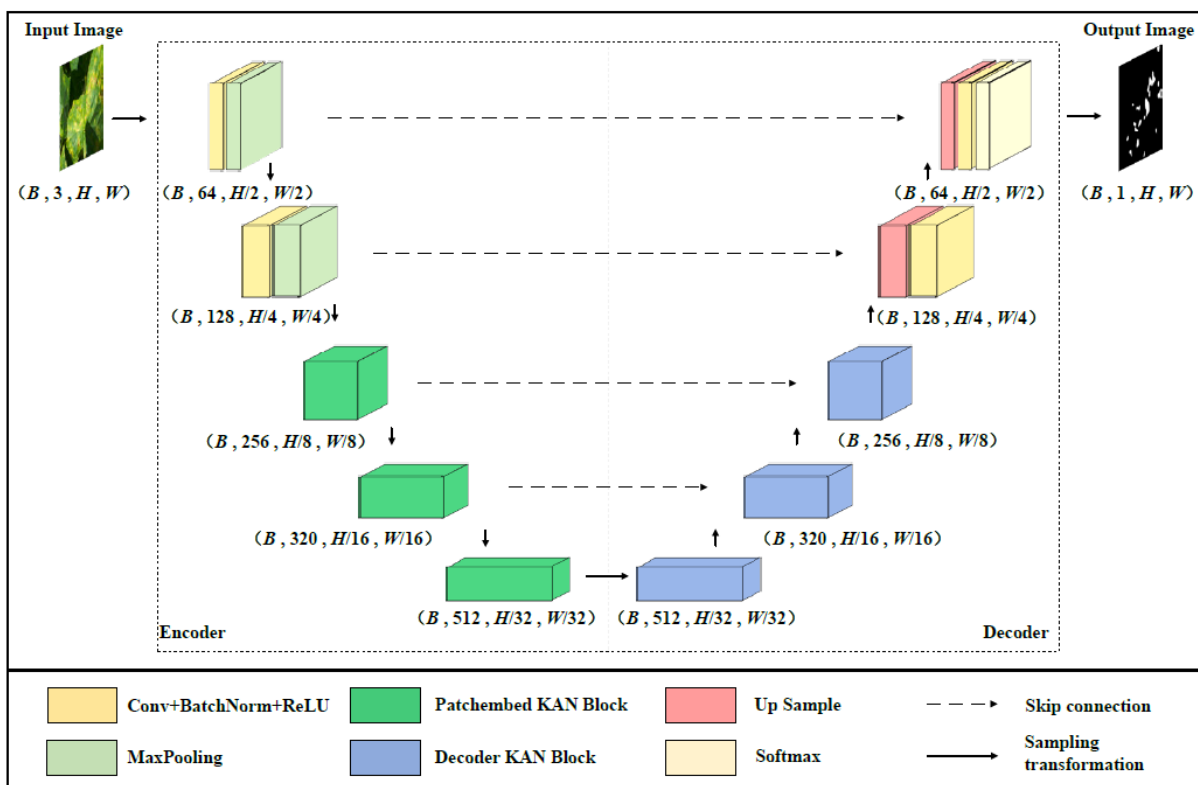


Рисунок 1.10 – Структура мережі U-PKAN [12]

Окрім згорткових підходів, у сучасних дослідженнях активно застосовуються архітектури на основі трансформерів. Зокрема, у роботі [13] запропоновано підхід до класифікації захворювань рослин із використанням моделі Vision Transformer (ViT), що дозволяє ефективно враховувати глобальні залежності в зображеннях.

Архітектура моделі базується на Vision Transformer, у якому зображення розбивається на патчі, що перетворюються у послідовність векторів ознак із додаванням позиційного кодування. Далі ці представлення обробляються через кілька шарів TransformerEncoder, які включають багатоголову self-attention та повнозв'язні шари, що дозволяє моделі ефективно враховувати як локальні, так і глобальні залежності в зображенні.

На рисунку 1.11 наведено загальну схему моделі, яка включає етапи розподілу даних на навчальну, валідаційну та тестову вибірки, навчання моделі та подальшу класифікацію зображень.

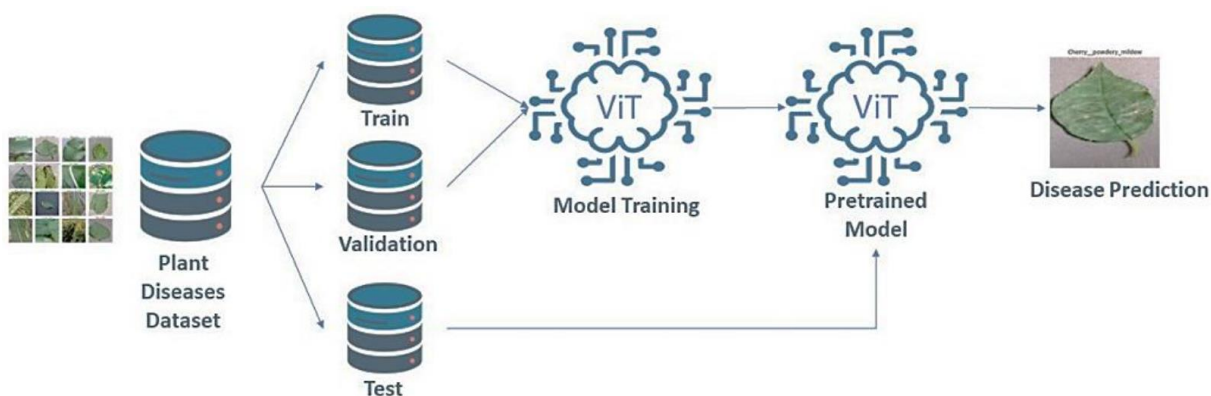


Рисунок 1.11 – Запропонована система на основі Vision Transformer для виявлення захворювань рослин [13]

Результати експериментів показали, що модель забезпечує високу якість класифікації: точність на тестовій вибірці становить 89.3%, а узагальнена оцінка якості (precision, recall, F1-score) знаходиться на рівні близько 90%, що свідчить про стабільну ефективність моделі для різних класів захворювань.

Одним із відомих програмних рішень для ідентифікації рослин є мобільний застосунок PlantNet. Даний сервіс призначений для визначення виду рослини за фотографією та побудований на використанні методів комп'ютерного зору. Користувач може зробити знімок рослини безпосередньо через камеру мобільного пристрою або завантажити зображення з галереї, після чого система виконує аналіз зображення та пропонує можливі варіанти виду рослини.

Інтерфейс застосунку є досить простим і орієнтованим на швидке виконання основної функції – ідентифікації рослини. На головному екрані користувачеві пропонується зробити фотографію або обрати зображення з галереї. Після обробки система відображає результати визначення рослини разом із додатковою інформацією, такою як наукова назва, родина та інші довідкові дані (рисунок 1.12).

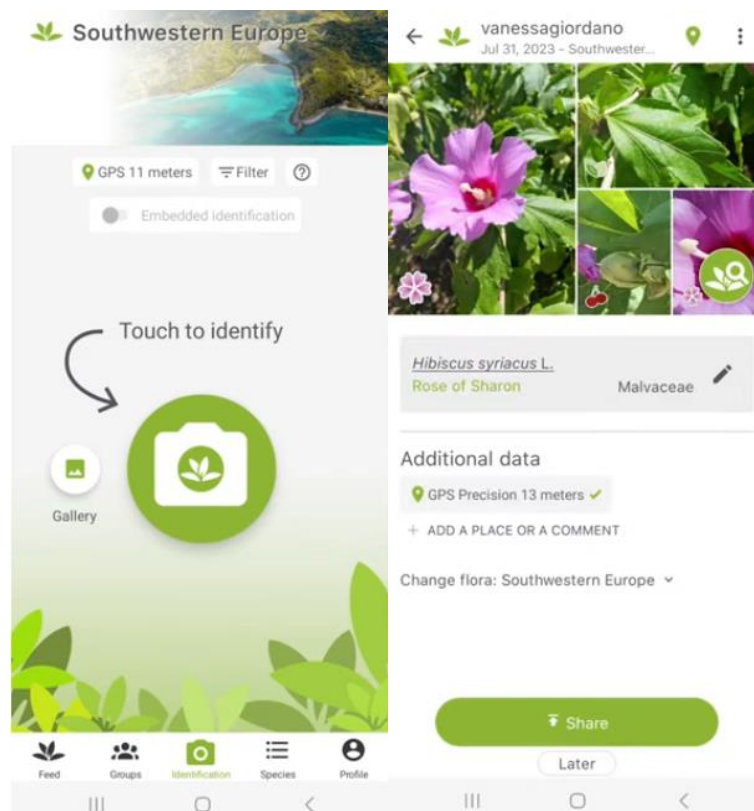


Рисунок 1.12 – Інтерфейс мобільного застосунку PlantNet: головний екран для фотографування рослини та приклад результату її ідентифікації [14]

Разом із тим функціональність застосунку обмежується переважно визначенням виду рослини. Система не орієнтована на детальний аналіз стану листя або діагностику захворювань, а також не надає інтерпретації причин можливих проблем із рослиною. Це обмежує можливості використання подібних рішень для автоматизованої діагностики уражень листя.

Ще одним прикладом мобільного застосунку для аналізу стану листя є Leaf Doctor. Даний застосунок призначений для оцінювання рівня пошкодження листя рослин за фотографією. Користувач робить знімок листка, після чого система виконує аналіз зображення та визначає приблизну площу уражених ділянок.

Принцип роботи застосунку полягає у виділенні пошкоджених областей на поверхні листя. Для цього користувач може налаштувати

параметр порогового значення, що дозволяє програмі відокремити уражені ділянки від здорової тканини листка. Після обробки зображення система підсвічує визначені області пошкодження та обчислює співвідношення між здоровою та ураженою частиною листка (рисунок 1.13).

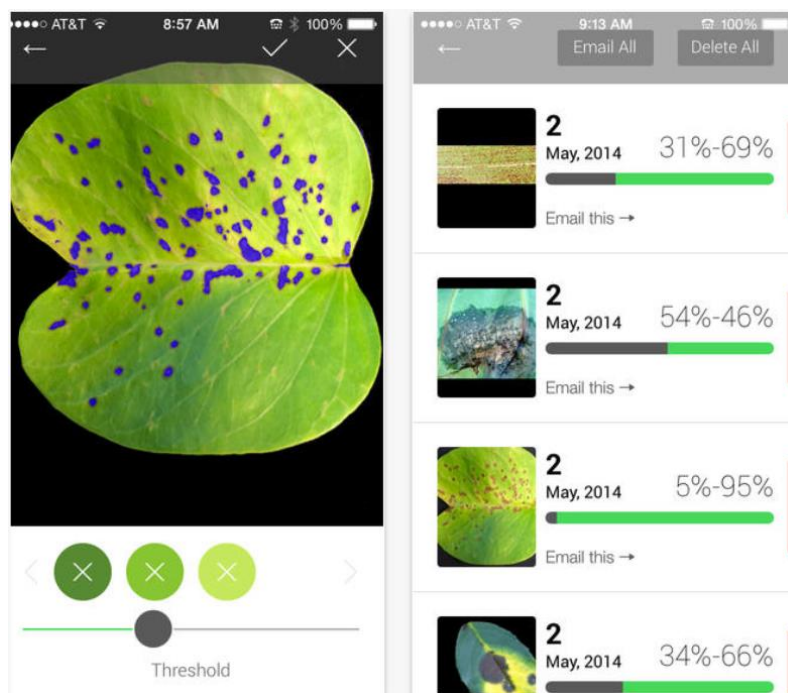


Рисунок 1.13 – Інтерфейс мобільного застосунку Leaf Doctor та приклад визначення площі ураження листя [15]

Інтерфейс програми є досить простим і орієнтований на виконання однієї основної функції – визначення відсотка пошкодженої поверхні листя. Користувач може переглядати результати аналізу для різних зображень та зберігати їх для подальшого використання.

Разом із тим функціональність застосунку є обмеженою. Система дозволяє лише оцінити площу пошкодження листя, але не визначає тип захворювання або причину появи уражень. Крім того, застосунок не надає рекомендацій щодо подальшого догляду за рослиною. У зв'язку з цим подібні рішення можуть використовуватися лише як допоміжний інструмент для попередньої оцінки стану листя.

Ще одним прикладом мобільного застосунку для визначення проблем рослин є Plant Disease Identifier. Даний сервіс дозволяє користувачу отримати попередню оцінку стану рослини на основі фотографії листа. Для цього необхідно зробити знімок рослини або завантажити зображення з галереї, після чого система виконує аналіз зображення та пропонує можливий варіант захворювання або проблеми з рослиною.

Інтерфейс застосунку орієнтований на просте використання та надання базових рекомендацій щодо догляду за рослинами. Після визначення проблеми користувач може отримати короткий план дій для покращення стану рослини, а також переглядати історію спостережень у вигляді так званого щоденника рослин (рисунк 1.14). Такий підхід дозволяє відстежувати зміни стану рослини протягом певного часу.

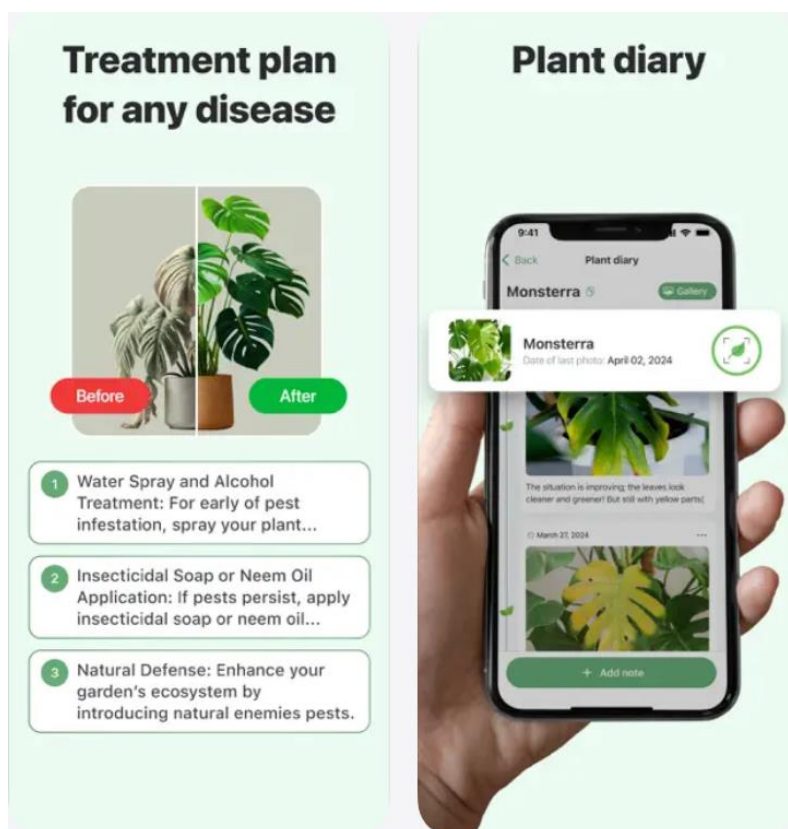


Рисунок 1.14 – Приклад інтерфейсу мобільного застосунку Plant Disease Identifier та рекомендацій щодо лікування рослин [16]

Разом із тим можливості застосування є обмеженими. Результат аналізу подається користувачу у вигляді готового висновку без пояснення того, які саме ознаки зображення були використані для визначення проблеми. Крім того, система не передбачає порівняння різних моделей або підходів до аналізу зображень, що ускладнює оцінку достовірності отриманого результату. У зв'язку з цим подібні рішення можуть використовуватися лише як допоміжний інструмент для попередньої оцінки стану рослин.

Таким чином, розглянуті програмні рішення здебільшого орієнтовані або на визначення виду рослини, або на базову оцінку її стану, що обмежує можливості їх використання для детальної діагностики захворювань. Крім того, користувач отримує лише результат класифікації без пояснення процесу прийняття рішення. Це зумовлює актуальність розробки системи, яка дозволить здійснювати аналіз зображень листя з використанням сучасних моделей глибокого навчання та забезпечувати інтерпретацію отриманих результатів.

1.3 Постановка задачі дослідження

Аналіз стану рослин є важливим завданням у сільському господарстві та екологічних дослідженнях, оскільки своєчасне виявлення захворювань дозволяє запобігти поширенню уражень і зменшити втрати врожаю. У сучасних умовах значного розвитку технологій обробки зображень і методів штучного інтелекту з'являється можливість автоматизувати процес аналізу стану рослин за їх візуальними ознаками.

Одним із найбільш інформативних об'єктів для такого аналізу є листя рослин, на якому проявляються характерні симптоми різних захворювань. У зв'язку з цим актуальним є створення інтелектуальної системи, здатної виконувати аналіз зображень листя рослин та визначати наявність і тип ураження.

Метою роботи є розробка інтелектуальної системи визначення типу ураження листя рослин за зображенням на основі методів глибокого навчання.

Для досягнення поставленої мети необхідно виконати наступні задачі:

- проаналізувати предметну галузь дослідження та особливості прояву уражень листя рослин;
- дослідити існуючі підходи та програмні рішення для аналізу стану рослин;
- розглянути сучасні методи обробки зображень і моделі глибокого навчання, що використовуються для задач класифікації;
- обрати датасет зображень листя рослин та виконати його підготовку до навчання моделей;
- реалізувати декілька моделей глибокого навчання для класифікації уражень листя;
- провести порівняльний аналіз ефективності використаних моделей;
- реалізувати програмну систему для аналізу зображень листя рослин та оцінити результати її роботи.

2 МЕТОДИ ТА МОДЕЛІ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ЗОБРАЖЕНЬ

Сучасні методи аналізу зображень відіграють важливу роль у вирішенні задач автоматичного розпізнавання та класифікації об'єктів. Завдяки розвитку технологій комп'ютерного зору та машинного навчання з'явилася можливість ефективно аналізувати візуальну інформацію та виявляти характерні ознаки об'єктів на зображеннях. У задачах дослідження стану рослин особливу увагу приділяють аналізу зображень листя, оскільки саме вони містять найбільш інформативні візуальні ознаки ураження. Використання методів обробки зображень та моделей глибокого навчання дозволяє автоматизувати процес виявлення таких ознак і підвищити точність діагностики стану рослин.

2.1 Основи обробки зображень

Цифрові зображення є однією з основних форм представлення візуальної інформації у комп'ютерних системах. У задачах комп'ютерного зору вони виступають вихідними даними, на основі яких здійснюється подальший аналіз, виділення ознак та класифікація об'єктів. Для того щоб алгоритми машинного навчання могли ефективно працювати із візуальною інформацією, зображення повинні бути представлені у формалізованому вигляді, придатному для математичної обробки [17].

Цифрове зображення розглядається як дискретне двовимірне представлення сцени, сформоване з елементів, що називаються пікселями. Кожен піксель містить числове значення, яке характеризує інтенсивність світла або колір у відповідній точці зображення. У найпростішому випадку використовується градація сірого, де значення пікселя визначає рівень яскравості. У кольорових зображеннях застосовується багатоканальне

представлення, найпоширенішим з яких є модель RGB, у якій колір описується трьома компонентами – червоною, зеленою та синьою.

З математичної точки зору цифрове зображення може бути представлене у вигляді матриці або тензора, елементи якого відповідають значенням пікселів (рисунок 2.1). Така форма представлення дозволяє застосовувати різноманітні операції лінійної алгебри та чисельного аналізу для обробки зображень. Саме матрична природа зображень робить можливим використання алгоритмів машинного навчання та нейронних мереж, які працюють із числовими масивами даних.

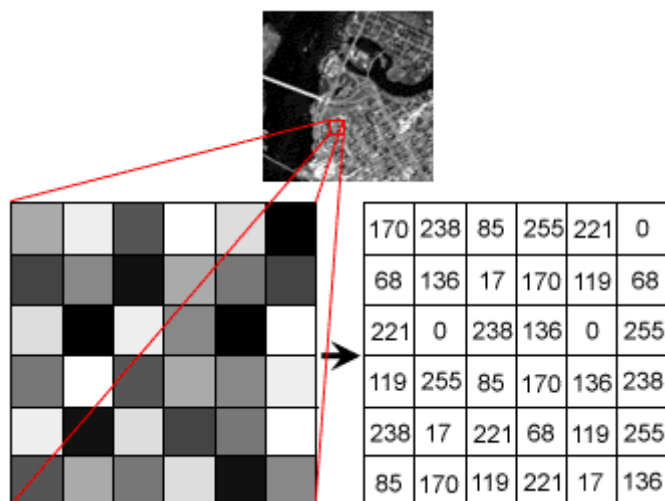


Рисунок 2.1 – Представлення цифрового зображення у вигляді матриці значень пікселів [18]

Базові операції обробки зображень спрямовані на покращення якості даних або підготовку їх до подальшого аналізу. До таких операцій належать зміна розміру зображення, нормалізація значень пікселів, фільтрація шумів та перетворення кольорового простору. Подібні перетворення дозволяють зменшити вплив випадкових спотворень, вирівняти характеристики зображень та зробити дані більш однорідними для алгоритмів аналізу.

Важливим аспектом є також просторові перетворення зображень. Вони включають масштабування, поворот або віддзеркалення, що дозволяє

адаптувати зображення до вимог моделі або забезпечити інваріантність алгоритмів до зміни положення об'єкта у кадрі. У задачах машинного навчання такі перетворення часто використовуються і як спосіб розширення навчальних даних, що дозволяє підвищити узагальнювальну здатність моделей.

Підготовка зображень до аналізу зазвичай включає декілька послідовних етапів. Спочатку здійснюється отримання або завантаження зображень із джерела даних. Далі виконуються перетворення, спрямовані на стандартизацію їх характеристик, зокрема розміру, формату та діапазону значень пікселів. Після цього зображення можуть бути представлені у вигляді числових масивів, що використовуються як вхідні дані для алгоритмів комп'ютерного зору.

У задачах дослідження стану рослин зображення листя містять значну кількість візуальної інформації, пов'язаної зі структурою поверхні, кольоровими змінами та текстурними особливостями [19]. Саме ці характеристики можуть свідчити про наявність різних типів ураження. Тому якість підготовки зображень та коректність їх представлення відіграють важливу роль для подальшого аналізу.

2.2 Класичні методи аналізу зображень

До появи глибоких нейронних мереж більшість задач комп'ютерного зору розв'язувалася за допомогою класичних методів аналізу зображень. Такі підходи базувалися на явному виділенні інформативних характеристик зображення та подальшому використанні цих характеристик у алгоритмах класифікації. Основна ідея полягала у перетворенні зображення на набір числових ознак, які описують форму, структуру або колір об'єкта [20].

Одним із важливих напрямів класичного аналізу зображень є виділення контурів. Контурні методи дозволяють визначати межі об'єктів на зображенні, де відбувається різка зміна інтенсивності пікселів. Для цього

застосовуються оператори градієнта, які оцінюють зміну яскравості у сусідніх пікселях (рисунк 2.2). Виявлені контури формують структурне представлення об'єкта та можуть використовуватися для подальшого аналізу його форми.

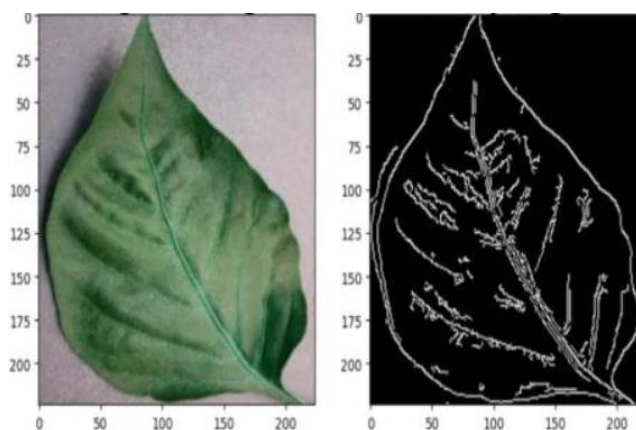


Рисунок 2.2 – Приклад виявлення контурів на зображенні за допомогою алгоритму Canny [21]

Іншим поширеним підходом є сегментація зображення. Її метою є поділ зображення на окремі області, що відповідають різним об'єктам або їх частинам. У процесі сегментації пікселі об'єднуються у групи відповідно до їхніх характеристик, наприклад інтенсивності, кольору або текстурних властивостей. Це дозволяє відокремити досліджуваний об'єкт від фону (рисунк 2.3) та зосередити подальший аналіз лише на релевантній частині зображення.



Рисунок 2.3 – Приклад сегментації зображення [22]

Після виділення області об'єкта зазвичай виконується обчислення описових характеристик, які використовуються як ознаки для класифікації. До таких характеристик можуть належати геометричні параметри форми, статистичні показники інтенсивності пікселів або текстурні властивості поверхні. Отриманий набір ознак формує вектор характеристик, який подається на вхід алгоритмам машинного навчання.

Класифікація об'єктів у класичних підходах зазвичай здійснюється за допомогою традиційних алгоритмів машинного навчання, таких як метод опорних векторів, k-найближчих сусідів або дерева рішень. Ефективність таких моделей значною мірою залежить від того, наскільки вдало сформовано набір ознак, що описують зображення. Таким чином, ключову роль у класичних системах комп'ютерного зору відіграє саме етап ручного конструювання ознак.

У задачах аналізу зображень листя рослин класичні методи дозволяють досліджувати такі характеристики, як зміни кольору, поява плям або порушення структури поверхні. Наприклад, аналіз кольорових характеристик може використовуватися для виявлення пожовтіння листя, а текстурні ознаки – для опису неоднорідностей поверхні, що виникають унаслідок розвитку захворювань.

2.3 Згорткові нейронні мережі

Попри те, що класичні методи обробки зображень відіграли важливу роль у розвитку комп'ютерного зору, вони мають певні обмеження. Ефективність таких підходів значною мірою залежить від правильності вибору ознак і може суттєво знижуватися у випадку складних або варіативних зображень. Саме ці обмеження стали однією з причин активного розвитку глибоких нейронних мереж, здатних автоматично навчатися виділяти інформативні ознаки без необхідності їх ручного конструювання.

Найбільш поширеним типом моделей для роботи із зображеннями є згорткові нейронні мережі.

Згорткова нейронна мережа являє собою спеціалізовану архітектуру штучної нейронної мережі, призначену для обробки даних із просторовою структурою. Основна ідея таких мереж полягає у використанні операції згортки, яка дозволяє аналізувати локальні області зображення та виявляти характерні шаблони. Завдяки цьому модель може поступово формувати ієрархію ознак – від простих структур, таких як краї та текстурні елементи, до більш складних візуальних характеристик об'єктів [23].

У процесі роботи згорткової мережі зображення подається на вхід у вигляді багатовимірного масиву чисел, що відповідають значенням пікселів. На початкових етапах мережа виконує локальний аналіз невеликих фрагментів зображення за допомогою спеціальних фільтрів. Кожен фільтр реагує на певний тип візуального патерна, наприклад на напрямок градієнта або характерну текстуру. У результаті застосування таких фільтрів формуються карти ознак, які відображають наявність відповідних структур у різних частинах зображення.

Згорткові шари зазвичай поєднуються з операціями підвибірки, що дозволяють поступово зменшувати просторову розмірність даних. Це сприяє зменшенню кількості параметрів моделі та робить її менш чутливою до незначних змін положення об'єкта на зображенні. У міру проходження даних через мережу формується узагальнене представлення зображення, яке містить інформацію про найбільш значущі ознаки.

На завершальному етапі отримані карти ознак перетворюються у компактний числовий вектор, який використовується для виконання задачі класифікації. Саме на цьому етапі модель визначає, до якого класу належить зображення, спираючись на виявлені візуальні характеристики. Таким чином, згорткова нейронна мережа поєднує процес виділення ознак і їх подальший аналіз у межах єдиної навчальної моделі.

Загальну послідовність перетворення зображення у процесі роботи згорткової нейронної мережі наведено на рисунку 2.4.

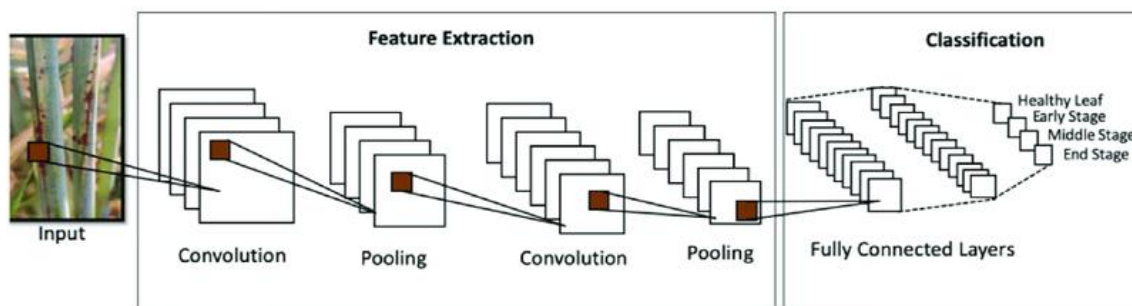


Рисунок 2.4 – Узагальнена схема обробки зображення у згортковій нейронній мережі [24]

Однією з ключових переваг згорткових нейронних мереж є здатність автоматично навчатися ефективних представлень даних. Під час навчання мережа самостійно підбирає параметри фільтрів таким чином, щоб вони максимально добре відображали характерні особливості зображень. Це дозволяє уникнути необхідності ручного конструювання ознак і значно підвищує точність моделей у задачах комп'ютерного зору.

У задачах аналізу зображень листя рослин згорткові нейронні мережі здатні ефективно виявляти складні візуальні патерни, пов'язані з різними типами ураження. Зокрема, вони можуть враховувати комбінацію кольорових змін, текстурних характеристик та просторового розташування плям або пошкоджень [17]. Завдяки цьому такі моделі демонструють високу ефективність у задачах класифікації зображень рослин, що зумовило їх широке використання у сучасних системах автоматичного моніторингу стану рослин.

Подальший розвиток методів глибокого навчання привів до появи підходів, які дозволяють використовувати вже навчені моделі для нових задач.

2.4 Transfer learning

Ефективність моделей глибокого навчання значною мірою залежить від обсягу навчальних даних. Навчання згорткових нейронних мереж з нуля зазвичай потребує великих наборів зображень, що містять достатню кількість прикладів для кожного класу. У багатьох практичних задачах отримати настільки великий датасет складно або потребує значних ресурсів. У таких випадках застосовується підхід transfer learning, який дозволяє використовувати знання, отримані моделлю під час розв'язання однієї задачі, для вирішення іншої. Перенесення навчання широко застосовується у задачах комп'ютерного зору, оскільки дозволяє ефективно використовувати попередньо навчені моделі та зменшувати вимоги до обсягу навчальних даних [25].

Основна ідея цього підходу полягає у використанні попередньо навченої моделі як основи для створення нової системи. Такі моделі зазвичай тренуються на великих наборах зображень загального призначення, наприклад ImageNet, що дозволяє їм навчитися виявляти універсальні візуальні ознаки. На ранніх рівнях мережі формуються базові характеристики зображення, зокрема контури, прості текстури та елементарні структури. Ці ознаки є спільними для великої кількості задач комп'ютерного зору, тому вони можуть бути ефективно використані і в інших предметних галузях.

Загальний принцип використання попередньо навченої моделі показано на рисунку 2.5. У цьому випадку мережа спочатку навчається на великому наборі зображень, після чого отримані вагові коефіцієнти переносяться до нової моделі. Частина параметрів мережі зберігається, тоді як завершальні шари можуть бути ініціалізовані повторно та адаптовані до нової задачі класифікації.

Залежно від постановки задачі можуть застосовуватися різні стратегії адаптації моделі. В одних випадках базова частина мережі використовується

як фіксований екстрактор ознак, а навчання відбувається лише для завершальних шарів, відповідальних за класифікацію. В інших випадках виконується додаткове донавчання всієї моделі або її окремих рівнів на новому наборі даних.

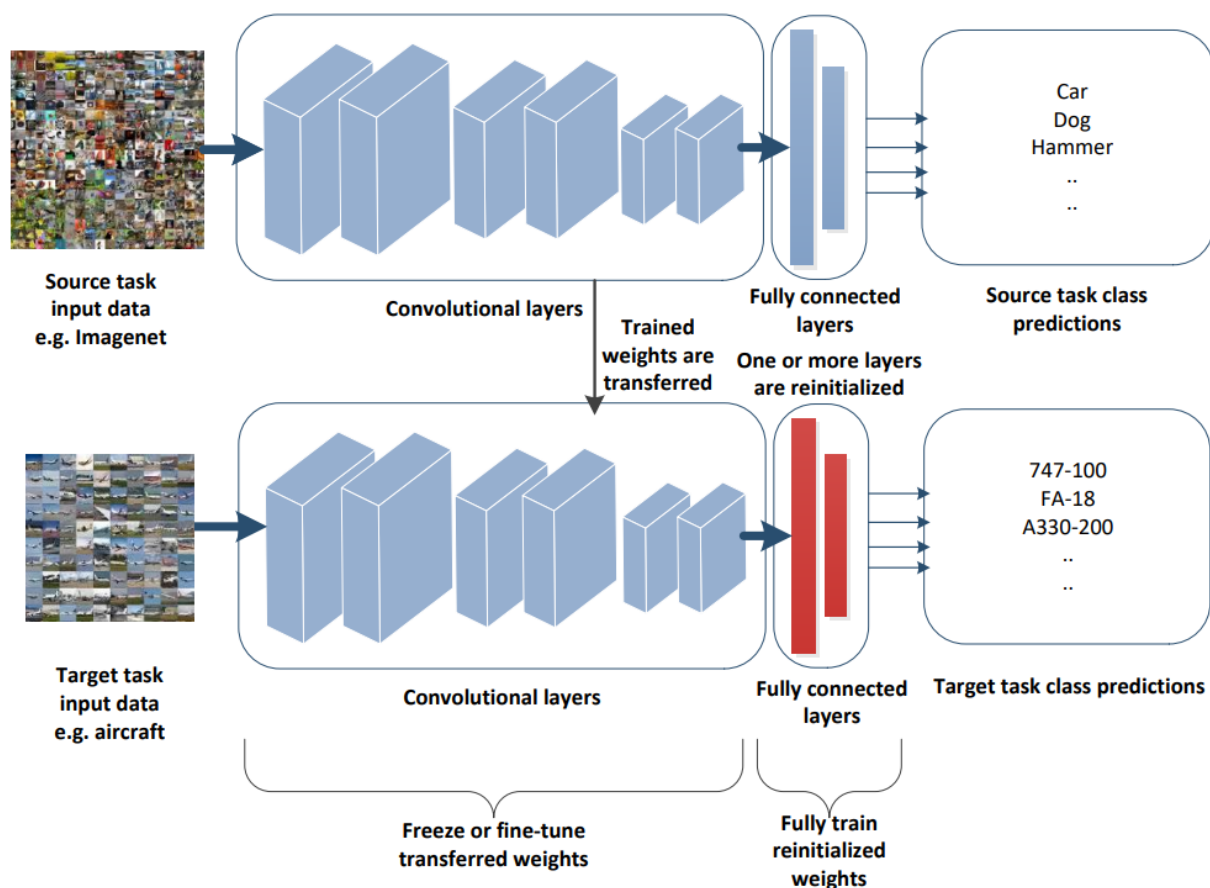


Рисунок 2.5 – Схема застосування transfer learning при використанні попередньо навченої згорткової нейронної мережі [25]

Основні стратегії використання transfer learning наведено на рисунку 2.6. У випадку екстракції ознак (feature extraction) базова частина попередньо навченої моделі залишається незмінною, а навчається лише новий класифікаційний шар. Такий підхід дозволяє значно скоротити час навчання. Інша стратегія – fine-tuning – передбачає донавчання частини або всіх шарів мережі, що дозволяє краще адаптувати модель до специфіки нового набору даних.

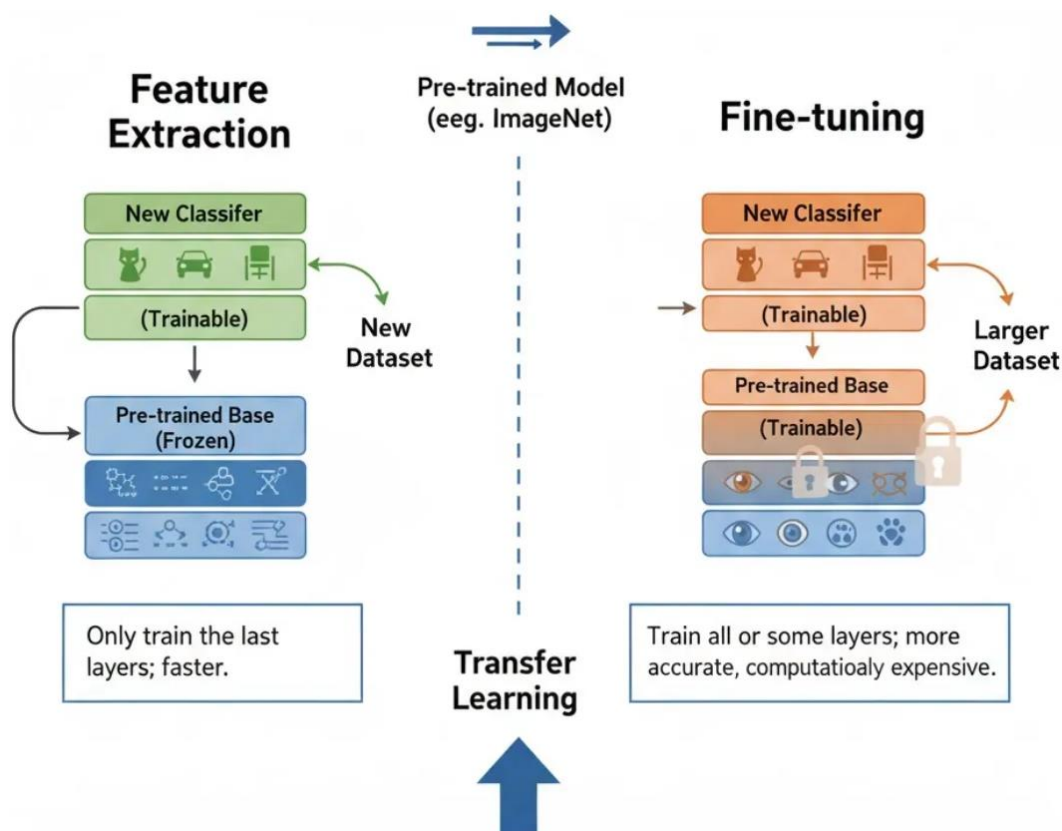


Рисунок 2.6 – Основні стратегії transfer learning: екстракція ознак та донавчання моделі [26]

Застосування transfer learning має кілька важливих переваг. По-перше, воно дозволяє суттєво скоротити обсяг необхідних навчальних даних, оскільки модель уже містить узагальнені знання про структуру зображень. По-друге, значно зменшується час навчання, адже більшість параметрів мережі вже налаштовані під час попереднього тренування. Крім того, використання попередньо навчених моделей часто дозволяє отримати більш стабільні результати у випадках, коли доступний датасет є відносно невеликим.

Використання transfer learning тісно пов'язане з вибором конкретної архітектури нейронної мережі, на основі якої будується модель. Сучасні дослідження у галузі комп'ютерного зору пропонують велику кількість архітектур згорткових мереж, що відрізняються структурою, складністю та ефективністю.

2.5 Огляд сучасних архітектур CNN

Після появи згорткових нейронних мереж активно розвиваються різні архітектури, спрямовані на підвищення точності моделей та ефективності обробки зображень. Розробка нових архітектур пов'язана з пошуком оптимальної структури мережі, яка дозволяє глибше аналізувати зображення, зберігати важливу просторову інформацію та водночас уникати проблем, що виникають під час навчання дуже глибоких моделей.

Розвиток архітектур згорткових нейронних мереж значною мірою пов'язаний із задачами великомасштабної класифікації зображень, зокрема змаганням ImageNet Large Scale Visual Recognition Challenge (ILSVRC). Саме результати цих досліджень стали основою для появи нових підходів до побудови глибоких моделей, що здатні ефективно працювати з великими наборами візуальних даних.

Однією з перших архітектур, що продемонструвала значний прорив у задачах класифікації зображень, стала мережа AlexNet [27]. У цій архітектурі використовуються кілька згорткових і повнозв'язних шарів, а також функція активації ReLU, що дозволило значно прискорити навчання моделі та підвищити точність розпізнавання зображень. Загальна структура архітектури AlexNet наведена на рисунку 2.7.

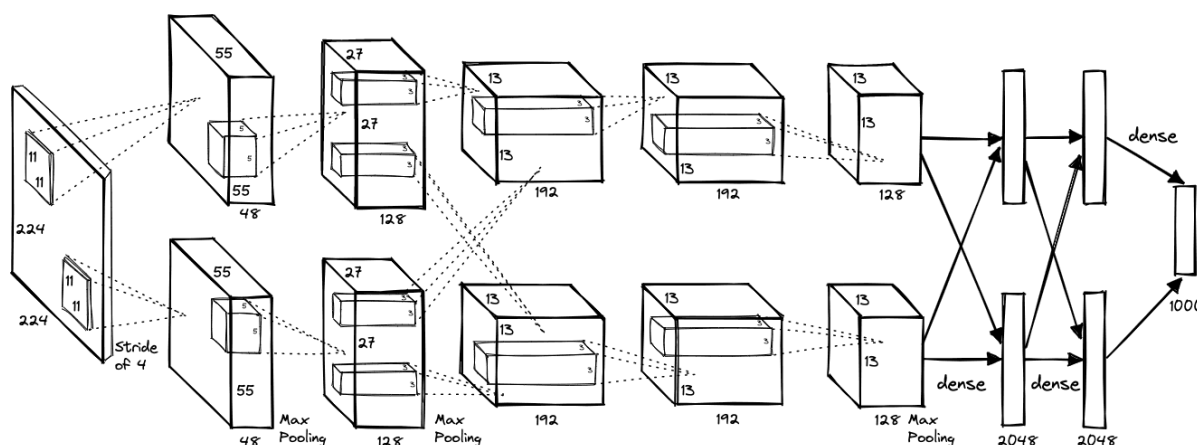


Рисунок 2.7 – Архітектура згорткової нейронної мережі AlexNet [27]

Подальші дослідження призвели до появи більш глибоких і структурно складніших моделей [28]. Одним із важливих етапів розвитку стали мережі сімейства VGG, які характеризуються використанням великої кількості послідовних згорткових шарів із невеликими ядрами розміру 3×3 . Такий підхід дозволив формувати більш детальні ієрархії ознак та забезпечив високу якість розпізнавання зображень. Водночас збільшення глибини мережі призвело до значного зростання кількості параметрів і підвищених вимог до обчислювальних ресурсів. Загальна структура архітектури VGG наведена на рисунку 2.8.

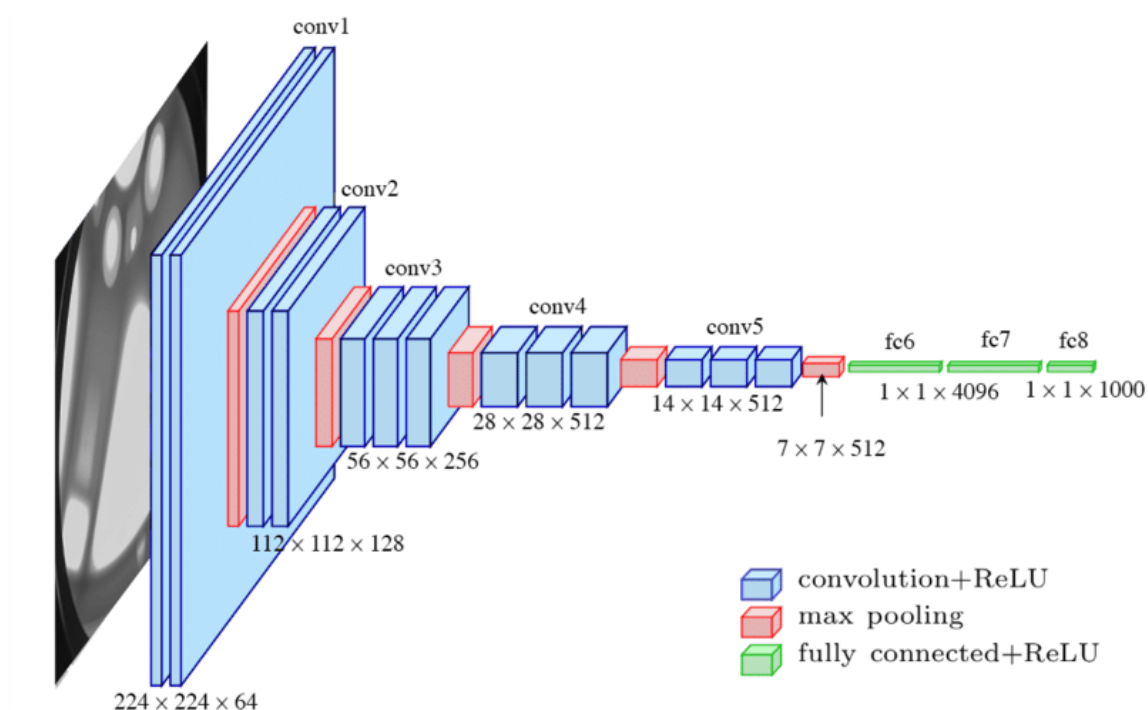


Рисунок 2.8 – Архітектура згорткової нейронної мережі VGG [28]

Подальший розвиток архітектур був спрямований на подолання проблем, пов'язаних із навчанням дуже глибоких моделей. Одним із ключових рішень стала поява мереж ResNet, у яких використано механізм залишкових з'єднань (residual connections). Завдяки такому підходу інформація може передаватися між шарами без втрати важливих характеристик, що дозволяє ефективно навчати значно глибші моделі. У

таких мережах використовуються спеціальні залишкові блоки, які забезпечують пряме передавання інформації між шарами. Приклад структури таких блоків у різних конфігураціях мережі ResNet наведено на рисунку 2.9.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				

Рисунок 2.9 – Структура блоків мережі ResNet50 для різної глибини архітектури [28]

Іншим напрямом розвитку стало створення архітектур, орієнтованих на підвищення обчислювальної ефективності. Наприклад, моделі сімейства MobileNet використовують спеціальний тип згорток – depthwise separable convolutions, що дозволяє суттєво зменшити кількість параметрів і обчислювальних операцій. Завдяки розділенню просторової та каналової обробки зображення такі мережі мають значно меншу складність порівняно з класичними CNN, що робить їх придатними для використання на мобільних пристроях або системах з обмеженими обчислювальними ресурсами. Приклад структури шарів у мережі MobileNet наведено на рисунку 2.10.

Подальшим розвитком ідеї використання розділених згорток стала архітектура Xception, у якій depthwise separable convolutions застосовуються більш системно та послідовно. У цій моделі стандартні згорткові операції

повністю замінено на розділені згортки, що дозволяє ефективніше моделювати просторові та каналові залежності у зображенні. Архітектура Xception складається з трьох основних частин: початкового блоку (entry flow), центральної частини (middle flow) та завершального блоку (exit flow). Така структура дозволяє формувати складніші представлення ознак при збереженні відносно високої обчислювальної ефективності. Загальна схема архітектури Xception наведена на рисунку 2.11.

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
5×	Conv dw / s1	$3 \times 3 \times 512$ dw
	Conv / s1	$1 \times 1 \times 512 \times 512$
	Conv dw / s2	$3 \times 3 \times 512$ dw
	Conv / s1	$1 \times 1 \times 512 \times 1024$
	Conv dw / s2	$3 \times 3 \times 1024$ dw
	Conv / s1	$1 \times 1 \times 1024 \times 1024$
	Avg Pool / s1	Pool 7×7
	FC / s1	1024×1000
	Softmax / s1	Classifier

Рисунок 2.10 – Приклад структури шарів архітектури MobileNetV1 [28]

Розвиток архітектур CNN також призвів до появи моделей, що поєднують високу точність із відносно невеликою кількістю параметрів. Одним із прикладів є EfficientNet, у якій застосовується підхід масштабування мережі, що одночасно враховує глибину, ширину та роздільну здатність вхідних зображень. Такий підхід дозволяє збалансовано збільшувати складність моделі та отримувати високу ефективність без суттєвого зростання обчислювальних витрат. Приклад структури базової архітектури EfficientNet наведено на рисунку 2.12.

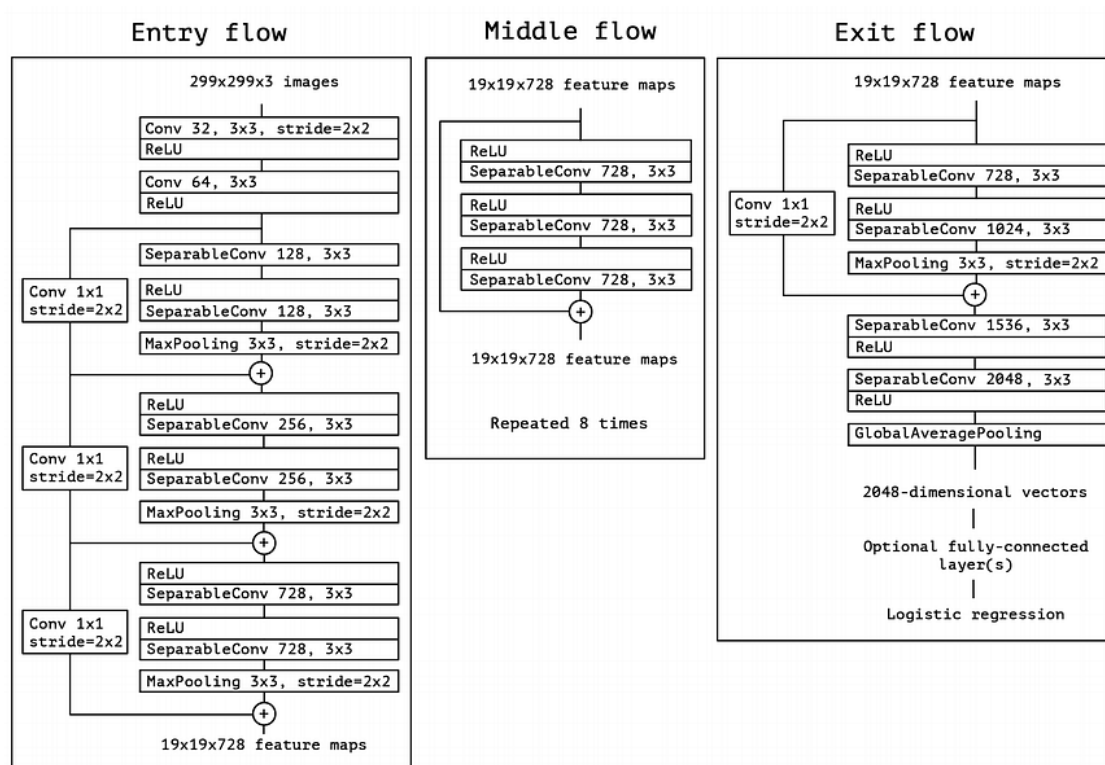


Рисунок 2.11 – Загальна структура архітектури Xception [28]

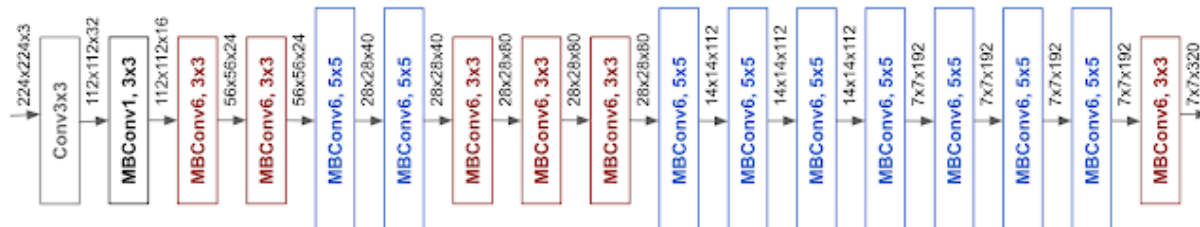


Рисунок 2.12 – Структура базової архітектури EfficientNet [29]

Таким чином, розвиток архітектур згорткових нейронних мереж відбувався у напрямі підвищення глибини моделей, покращення передачі інформації між шарами та зменшення обчислювальної складності. Починаючи з архітектур, орієнтованих на збільшення глибини мережі (VGG), дослідження поступово перейшли до використання спеціальних механізмів навчання глибоких моделей (ResNet), оптимізації обчислювальної ефективності (MobileNet, Xception) та збалансованого

масштабування параметрів мережі (EfficientNet). Така еволюція архітектур дозволила значно підвищити точність моделей комп'ютерного зору при збереженні прийнятної обчислювальної складності.

2.6 Метрики оцінювання моделей

Після побудови та навчання моделі машинного навчання необхідно оцінити її здатність правильно розпізнавати об'єкти на нових даних. Для цього використовуються спеціальні кількісні показники, які дозволяють визначити ефективність роботи моделі та порівнювати результати різних підходів [30]. У задачах класифікації зображень оцінювання ґрунтується на аналізі відповідності між передбаченими моделлю класами та фактичними мітками зображень.

Для детального аналізу результатів класифікації використовується матриця невідповідностей (confusion matrix). Вона відображає розподіл правильних і неправильних передбачень моделі та дозволяє визначити типи помилок, які виникають під час класифікації. Матриця невідповідностей містить чотири основні типи результатів: істинно позитивні (True Positive, TP), істинно негативні (True Negative, TN), хибно позитивні (False Positive, FP) та хибно негативні (False Negative, FN). Загальний вигляд такої матриці наведено на рисунку 2.13.

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Рисунок 2.13 – Матриця невідповідностей [30]

На основі елементів матриці невідповідностей обчислюються різні метрики, що характеризують якість роботи моделі. Однією з найпоширеніших є точність класифікації (accuracy), яка відображає частку правильно класифікованих прикладів серед усіх проаналізованих зображень. Формально цей показник визначається як відношення суми правильно передбачених позитивних та негативних результатів до загальної кількості прикладів:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}. \quad (2.1)$$

Ця метрика є інтуїтивно зрозумілою та часто використовується для загальної оцінки ефективності моделі. Однак у випадках, коли класи представлені нерівномірно, одного лише показника accuracy може бути недостатньо для повного аналізу результатів. Це особливо актуально для задач аналізу зображень рослин, де деякі типи уражень можуть бути представлені значно меншою кількістю прикладів порівняно з іншими класами, що призводить до дисбалансу даних і може спотворювати оцінку якості моделі.

Для більш детального оцінювання використовуються додаткові метрики. Однією з них є влучність (precision), яка визначає частку правильно визначених позитивних результатів серед усіх випадків, коли модель передбачила позитивний клас:

$$Precision = \frac{TP}{TP+FP}. \quad (2.2)$$

Цей показник дозволяє оцінити надійність передбачень моделі. Високе значення precision означає, що більшість передбачених позитивних результатів є правильними.

Іншою важливою характеристикою є повнота (recall). Вона відображає, яку частку фактичних позитивних прикладів модель змогла правильно виявити:

$$Recall = \frac{TP}{TP+FN}. \quad (2.3)$$

Високе значення recall означає, що модель здатна знаходити більшість об'єктів певного класу, однак при цьому може збільшуватися кількість хибно позитивних передбачень.

Для узагальнення показників precision та recall використовується F1-міра, яка враховує баланс між цими двома характеристиками. Вона визначається як гармонійне середнє значень precision та recall:

$$F1 = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)}. \quad (2.4)$$

F1-міра є корисною у випадках, коли важливо одночасно враховувати як здатність моделі знаходити об'єкти певного класу, так і точність таких передбачень.

У задачах класифікації зображень рослин використання декількох метрик є важливим, оскільки різні типи помилок можуть мати різні наслідки для практичного застосування системи. Аналіз метрик дозволяє оцінити не лише загальну точність моделі, а й її здатність правильно розпізнавати окремі типи ураження листя.

Окрім оцінювання точності передбачень, у сучасних дослідженнях значна увага приділяється також інтерпретації результатів роботи моделей. Це пов'язано з необхідністю зрозуміти, на основі яких ознак модель приймає рішення. Підходи, що дозволяють аналізувати та пояснювати поведінку моделей машинного навчання, розглядаються у наступному підрозділі.

2.7 Методи пояснювального штучного інтелекту

Зі зростанням складності моделей глибокого навчання постає проблема інтерпретації результатів їх роботи. Багато сучасних алгоритмів, зокрема згорткові нейронні мережі, демонструють високу точність у задачах аналізу зображень, проте механізм прийняття рішень у таких моделях часто залишається недостатньо зрозумілим для користувача. У таких випадках модель розглядається як «чорний ящик», оскільки складно визначити, які саме ознаки зображення вплинули на кінцевий результат класифікації.

Для розв'язання цієї проблеми розвивається напрям *explainable artificial intelligence* (XAI), спрямований на підвищення прозорості та інтерпретованості моделей машинного навчання. Основна мета таких підходів полягає у визначенні внеску окремих ознак у формування передбачення, що дозволяє краще зрозуміти поведінку моделі, виявити потенційні помилки та підвищити довіру до результатів її роботи.

У задачах аналізу зображень одним із найпоширеніших підходів до пояснення роботи моделей є побудова карт важливості (*saliency maps*). Такі методи дозволяють визначити області зображення, які найбільше вплинули на рішення нейронної мережі. У результаті формується додаткове зображення, на якому підсвічуються найбільш інформативні ділянки [31].

Серед сучасних методів пояснення для згорткових нейронних мереж широко застосовується підхід Grad-CAM (*Gradient-weighted Class Activation Mapping*). Він базується на використанні градієнтів цільового класу відносно ознак останніх згорткових шарів мережі та дозволяє формувати теплові карти, що відображають важливість різних областей зображення. Перевагою цього методу є можливість отримання інтерпретованих результатів без зміни архітектури моделі.

Приклад побудови карти важливості для різних класів наведено на рисунку 2.14.

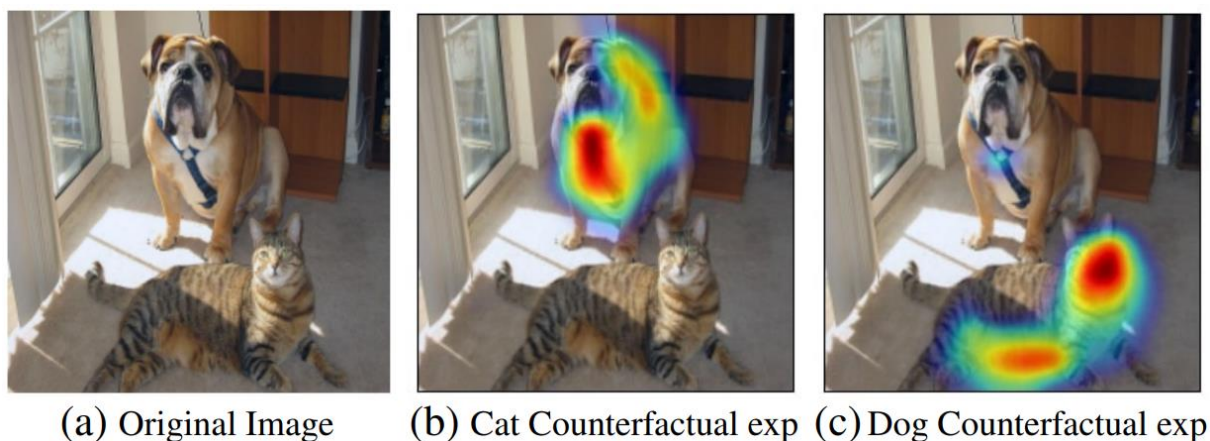


Рисунок 2.14 – Приклад візуалізації важливих областей зображення за допомогою методу Grad-CAM [32]

Як видно з рисунка, модель фокусується на різних ділянках зображення залежно від передбачуваного класу, що дозволяє оцінити обґрунтованість прийнятого рішення.

Також використовуються підходи, що аналізують внесок окремих пікселів або локальних областей зображення у кінцеве передбачення. На основі таких оцінок формуються теплові карти, які відображають ступінь впливу різних частин зображення на результат класифікації. Візуалізація цих карт дозволяє перевірити, чи дійсно модель орієнтується на релевантні ознаки об'єкта, а не на випадкові особливості фону або шуму.

Крім перевірки коректності роботи моделі, методи пояснювального штучного інтелекту можуть використовуватися для вдосконалення системи. Аналіз результатів пояснення дозволяє виявити випадки, коли модель помиляється або використовує неінформативні ознаки, що дає змогу скоригувати навчальні дані або уточнити процес навчання.

Таким чином, використання методів explainable AI доповнює процес оцінювання моделей машинного навчання та сприяє більш глибокому розумінню їх поведінки. Поєднання високої точності класифікації з можливістю інтерпретації результатів є важливою умовою створення надійних інтелектуальних систем аналізу зображень.

3 РОЗРОБКА ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ ДІАГНОСТИКИ УРАЖЕННЯ ЛИСТЯ

Розробка інтелектуальної системи визначення типу ураження листя рослин за зображенням вимагає комплексного підходу, що поєднує обробку візуальних даних, використання моделей глибокого навчання та формування результату у зручному для користувача вигляді. Така система повинна забезпечувати не лише автоматичне розпізнавання ознак ураження, але й надавати зрозумілу інтерпретацію отриманих результатів.

Побудова системи передбачає поєднання кількох взаємопов'язаних компонентів, зокрема підготовки даних, вибору та налаштування моделей, а також формування результату у зрозумілому для користувача вигляді. Важливою складовою є не лише досягнення високої точності класифікації, але й забезпечення можливості інтерпретації отриманих результатів та надання рекомендацій щодо подальших дій.

3.1 Архітектура системи

Інтелектуальна система визначення типу ураження листя рослин реалізована у вигляді послідовності взаємопов'язаних етапів обробки зображення, кожен з яких виконує окрему функцію в загальному процесі аналізу (рисунок 3.1). Така архітектура забезпечує перехід від вхідного зображення до інтерпретованого результату, що включає класифікацію стану листя та пояснення прийнятого рішення.

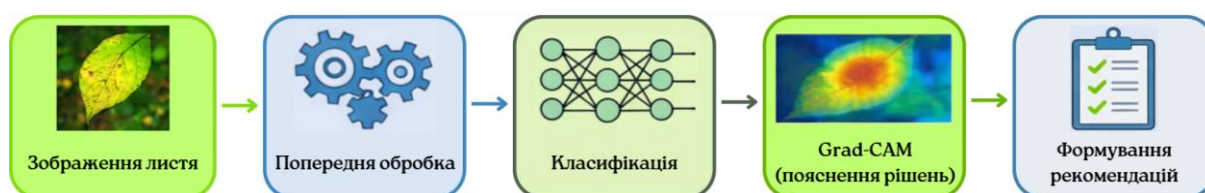


Рисунок 3.1 – Архітектура системи

На вхід системи подається зображення листя рослини, отримане за допомогою камери або завантажене користувачем. На першому етапі виконується попередня обробка, що передбачає приведення зображення до єдиного формату, зміну розміру та нормалізацію. Це дозволяє забезпечити узгодженість вхідних даних і коректну роботу моделей глибокого навчання.

Після попередньої обробки зображення передається до підсистеми класифікації, реалізованої на основі згорткових нейронних мереж. У системі використовується кілька моделей, що дозволяє виконати їх порівняльний аналіз і визначити найбільш ефективну для поставленої задачі. Результатом цього етапу є визначення класу, що відповідає типу ураження листя.

Наступним етапом є пояснення рішення моделі. Для цього застосовується метод Grad-CAM, який забезпечує візуалізацію областей зображення, що мали найбільший вплив на формування прогнозу. У результаті формується теплова карта, яка накладається на вихідне зображення та дозволяє оцінити коректність локалізації уражених ділянок.

Завершальним етапом є формування рекомендацій щодо догляду за рослиною. На основі визначеного типу ураження система надає короткі текстові поради, спрямовані на усунення або зменшення негативного впливу виявлених факторів, що підвищує практичну цінність розробленого рішення.

3.2 Підготовка та обробка даних

Підсистема аналізу зображень у розробленій системі базується на використанні попередньо підготовленого набору даних, що містить зображення листя рослин із різними типами уражень. Якість та структура даних мають визначальний вплив на ефективність навчання моделей, тому особлива увага приділяється їх організації та попередній обробці.

У роботі використано датасет «Plant Leaf Disease MasterDataset (19 Crops)» [33], що включає 86 класів, які відповідають різним типам уражень

листя, а також здоровому стану рослин. Загальна кількість зображень становить понад 140 тисяч, що забезпечує достатню репрезентативність вибірки для навчання моделей глибокого навчання.

Для коректної оцінки якості моделей дані було розподілено на три підмножини: навчальну (train), валідаційну (validation) та тестову (test). Навчальна вибірка використовується для оптимізації параметрів моделей, валідаційна – для контролю процесу навчання та вибору найкращої моделі, а тестова – для фінального оцінювання якості. Розподіл зображень між підмножинами наведено на рисунку 3.2.

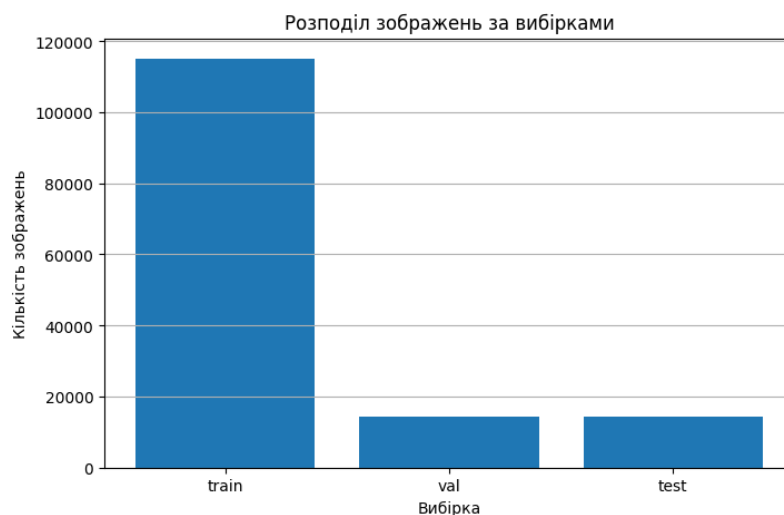


Рисунок 3.2 – Розподіл зображень за вибірками

Аналіз розподілу кількості зображень за класами показує наявність певного дисбалансу, оскільки окремі класи представлені більшою кількістю прикладів, ніж інші. Це є типовою особливістю реальних датасетів і може впливати на здатність моделі коректно розпізнавати рідкісні типи уражень. Відповідний розподіл наведено на рисунку 3.3.

Для кращого розуміння структури даних та характеру зображень на рисунку 3.4 наведено приклади зображень із датасету. Вони демонструють різноманітність форми листя, а також візуальні прояви різних типів уражень, зокрема плями, зміни кольору та некротичні ділянки.

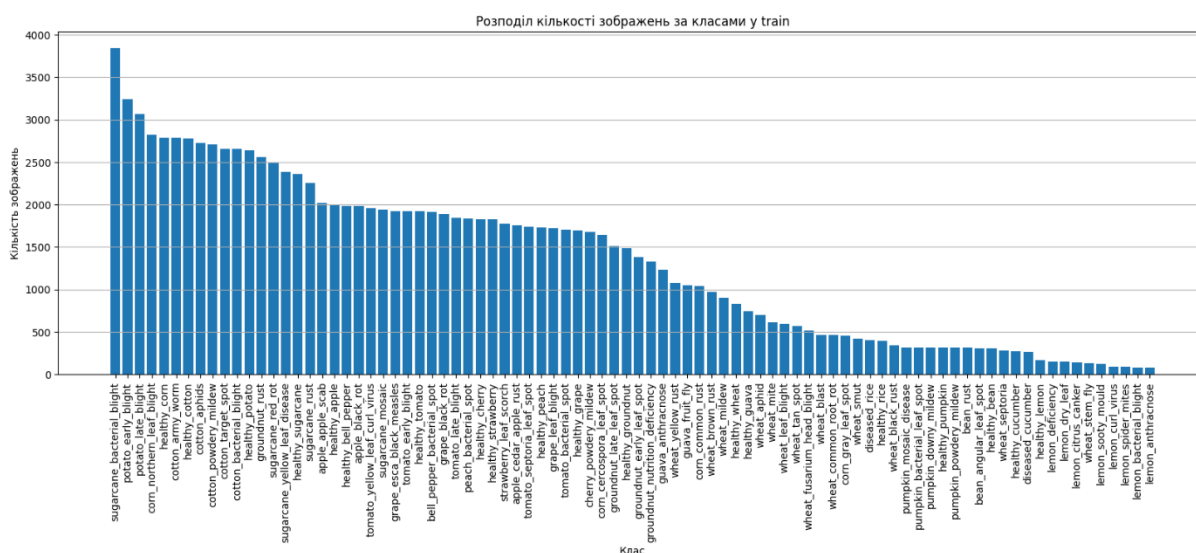


Рисунок 3.3 – Розподіл кількості зображень за класами

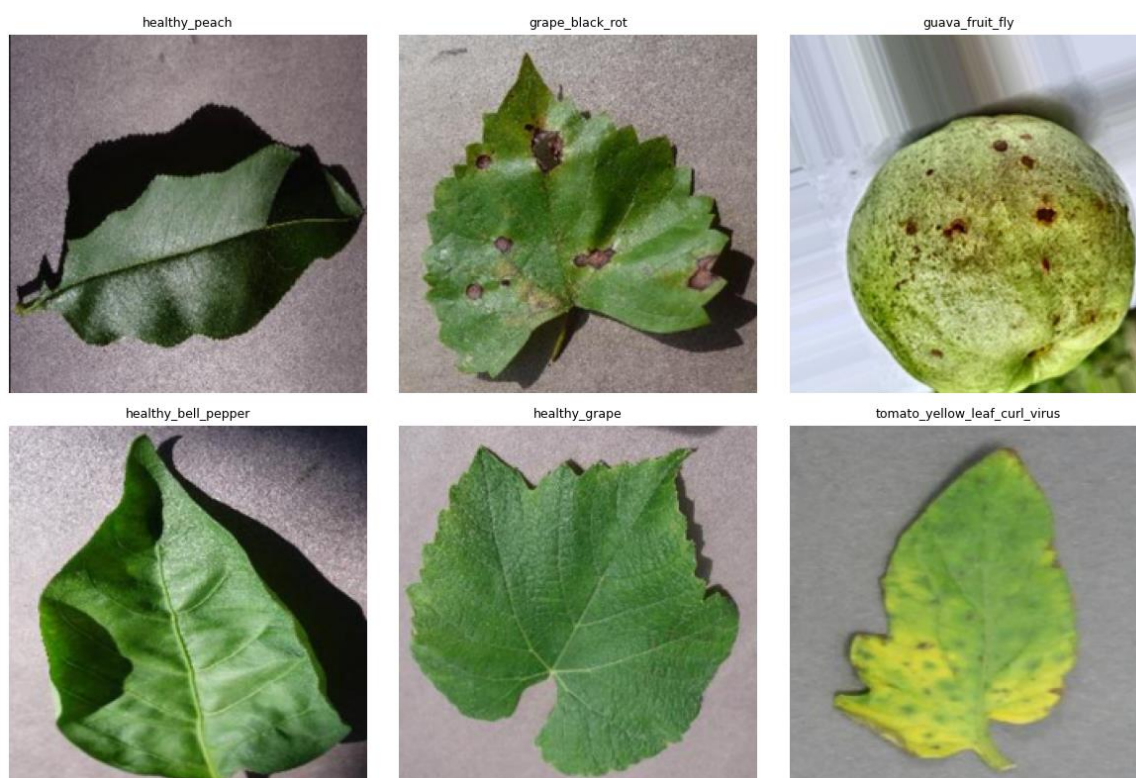


Рисунок 3.4 – Приклади зображень із датасету

Перед подачею на вхід моделей усі зображення проходять етап попередньої обробки. Зокрема, виконується зміна розміру до фіксованого значення 224×224 пікселів, що відповідає вимогам обраних архітектур

нейронних мереж, а також нормалізація значень пікселів. Такі перетворення забезпечують узгодженість вхідних даних і стабільність процесу навчання.

3.3 Вибір та конфігурація моделей

Для реалізації підсистеми класифікації було обрано три сучасні архітектури згорткових нейронних мереж: MobileNetV2, ResNet50 та EfficientNet-B0. Використання кількох моделей обумовлено необхідністю проведення порівняльного аналізу їх ефективності для задачі визначення типу ураження листа рослин.

Обрані архітектури відрізняються за кількістю параметрів, складністю та підходами до побудови мережі, що дозволяє оцінити їхню продуктивність у різних умовах. MobileNetV2 є легкою моделлю, оптимізованою для швидкої роботи та використання на пристроях з обмеженими ресурсами. ResNet50 належить до більш глибоких мереж і використовує залишкові з'єднання, що забезпечують стабільність навчання. EfficientNet-B0 представляє сучасний підхід до масштабування нейронних мереж і дозволяє досягати високої точності при відносно невеликій кількості параметрів.

Усі моделі були використані в рамках підходу transfer learning із застосуванням попередньо навчених ваг. Це дозволило суттєво скоротити час навчання та підвищити якість класифікації за рахунок використання вже сформованих низькорівневих ознак.

Оскільки початкові версії моделей орієнтовані на задачу класифікації зображень на великій кількості класів, їх фінальні класифікаційні шари були модифіковані відповідно до кількості класів у досліджуваному датасеті. Зокрема, вихідний шар кожної моделі було замінено на повнозв'язний шар із кількістю нейронів, що відповідає 86 класам.

Таким чином, обрані архітектури були адаптовані до поставленої задачі, що дозволило забезпечити коректне навчання моделей та можливість їх подальшого порівняння.

3.4 Процес навчання моделей

Навчання моделей класифікації виконувалося з використанням єдиного підходу для всіх обраних архітектур, що забезпечує коректність їх подальшого порівняння. Для кожної моделі використовувалися однакові параметри навчання та однакова організація вхідних даних.

Процес навчання здійснювався протягом фіксованої кількості епох, що дорівнює 5. На кожній епосі модель послідовно обробляла всі зображення навчальної вибірки, після чого виконувалося оцінювання якості на валідаційній підмножині. Це дозволяло відстежувати динаміку зміни функції втрат та точності класифікації.

Для оптимізації параметрів моделей використовувався оптимізатор Adam, який забезпечує стабільне та швидке збіження в задачах комп'ютерного зору. Як функцію втрат застосовано categorical cross-entropy, що є стандартним вибором для задач багатокласової класифікації.

У процесі навчання реалізовано механізм збереження найкращої моделі. Після кожної епохи перевірялося значення точності на валідаційній вибірці, і у випадку його покращення ваги моделі зберігалися. Це дозволило зафіксувати модель із найкращою узагальнюючою здатністю.

Фрагмент реалізації процесу збереження найкращої моделі наведено нижче:

```
if val_acc > best_val_acc:
    best_val_acc = val_acc
    torch.save(model.state_dict(), "best_model.pth")
```

Такий підхід забезпечує отримання оптимальної конфігурації моделі без необхідності ручного аналізу результатів кожної епохи.

У результаті навчання для кожної з моделей було отримано набір ваг, що відповідає найкращому значенню точності на валідаційній вибірці, які надалі використовуються для оцінювання якості на тестових даних та інтеграції в програмний застосунок.

3.5 Реалізація пояснень рішень моделі (Grad-CAM)

Для підвищення інтерпретованості результатів класифікації у системі реалізовано механізм пояснення рішень моделі на основі методу Grad-CAM. Його використання дозволяє візуально визначити області зображення, які мали найбільший вплив на формування прогнозу.

Інтеграція Grad-CAM здійснюється шляхом отримання активацій останнього згорткового шару нейронної мережі та відповідних градієнтів, що відображають внесок кожної ознаки у кінцевий результат. На основі цих даних формується карта важливості, яка після нормалізації накладається на вихідне зображення у вигляді теплової карти.

Для візуалізації використовується накладання теплової карти з використанням кольорової палітри, що дозволяє виділити найбільш значущі області.

Реалізація методу передбачає виконання зворотного поширення помилки для обраного класу та обчислення ваг для карт ознак. Узагальнений фрагмент коду, що демонструє ключовий етап обчислення карти важливості, наведено нижче:

```
weights = gradients.mean(dim=(2, 3), keepdim=True)
cam = (weights * activations).sum(dim=1, keepdim=True)
cam = torch.relu(cam)
```

Отримана карта масштабується до розміру вхідного зображення та поєднується з ним, що дозволяє візуально оцінити, на які саме ділянки листя орієнтується модель під час визначення типу ураження.

Застосування Grad-CAM дає змогу не лише підвищити довіру до результатів роботи системи, але й виявити можливі помилки моделі, пов'язані з фокусуванням на нерелевантних областях зображення. Це є важливим елементом аналізу якості моделі та обґрунтування отриманих результатів.

3.6 Формування рекомендацій на основі результату класифікації

Завершальним етапом роботи системи є формування рекомендацій щодо подальшого догляду за рослиною на основі результату класифікації. У даній реалізації використовується контрольований підхід, який не передбачає генерації тексту, а базується на попередньо визначених відповідностях між типом ураження та рекомендаціями.

Рекомендації реалізовано у вигляді словника, де кожному типу ураження відповідає короткий текстовий опис дій, спрямованих на усунення або зменшення негативного впливу виявлених факторів (лістинг 3.1). Формування таких рекомендацій здійснювалося шляхом узагальнення типових підходів до догляду за рослинами залежно від характеру ураження.

Лістинг 3.1 – Приклад структури словника рекомендацій

```
disease_recommendations = {
    "Healthy": "Ознак ураження не виявлено. Рекомендується продовжувати стандартний догляд за рослиною.",
    "Scab": "Рекомендується видалити уражене листя, застосувати фунгіцидну обробку та контролювати рівень вологості.",
    "Black Rot": "Рекомендується видалити уражені частини рослини та провести обробку відповідними захисними препаратами.",
    "Rust": "Доцільно контролювати поширення ураження та за потреби застосувати фунгіцидні засоби.",
    "Gray Leaf Spot": "Рекомендується зменшити вологість середовища та провести фунгіцидну обробку.",
    "Northern Leaf Blight": "Слід видалити сильно уражене листя та застосувати засоби захисту рослин.",
    "Powdery Mildew": "Рекомендується зменшити вологість, покращити циркуляцію повітря та застосувати відповідний препарат.",
}
```

Залежно від результату класифікації система визначає відповідну рекомендацію та повертає її користувачу разом із прогнозом. У випадку, якщо визначено здоровий стан листя, виводиться повідомлення про відсутність ознак ураження та рекомендації щодо підтримання належних умов догляду.

Додатково враховується рівень впевненості моделі. У разі, якщо значення ймовірності є низьким (менше 0.5), система рекомендує повторну перевірку або додатковий огляд рослини (лістинг 3.2).

Лістинг 3.2 – Функція, що враховує рівень впевненості моделі

```
def get_recommendation_by_disease(disease_name, confidence):  
    if confidence < 0.5:  
        return "Рекомендується додатково оглянути рослину або  
        зробити інше фото."  
  
    if disease_name in disease_recommendations:  
        return disease_recommendations[disease_name]
```

Таким чином, реалізований підхід забезпечує стабільність та передбачуваність результатів, а також дозволяє отримувати зрозумілі рекомендації без використання генеративних моделей.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ

4.1 Засоби та середовище розробки

Реалізація інтелектуальної системи здійснювалася із використанням мови програмування Python, що є однією з найбільш поширених у задачах машинного навчання та комп'ютерного зору завдяки наявності великої кількості спеціалізованих бібліотек.

Для побудови та навчання моделей глибокого навчання було використано бібліотеку PyTorch, яка забезпечує гнучкі засоби роботи з нейронними мережами та підтримує ефективне виконання обчислень. Для роботи із згортковими архітектурами застосовувалися готові моделі з бібліотеки torchvision, що дозволило використати підхід transfer learning.

Обробка зображень та підготовка вхідних даних здійснювалася за допомогою бібліотек torchvision.transforms та PIL, які забезпечують виконання таких операцій, як зміна розміру зображень, нормалізація та перетворення у тензорний формат.

Для реалізації підсистеми візуалізації пояснень було використано бібліотеки matplotlib та OpenCV, що дозволяють формувати теплові карти Grad-CAM та накладати їх на вихідні зображення.

Розрахунок метрик якості моделей, зокрема accuracy та macro F1-міри, здійснювався із використанням бібліотеки scikit-learn, що забезпечує зручні засоби оцінювання результатів класифікації.

Розробка та тестування моделей здійснювалися у середовищі Google Colab, що забезпечує доступ до обчислювальних ресурсів та спрощує процес експериментування з моделями глибокого навчання.

Для створення користувацького інтерфейсу було використано фреймворк Streamlit, який дозволяє швидко реалізувати веб-застосунок для

демонстрації роботи системи без необхідності розробки складної серверної частини.

Під час реалізації системи було забезпечено узгодженість етапів обробки даних, навчання моделей та формування результатів, що дозволило отримати стабільні результати класифікації на тестових вибірках.

4.2 Параметри експерименту та налаштування моделей

Навчання всіх моделей виконувалося за однакових умов, що забезпечує коректність подальшого порівняння їх результатів. Для кожної архітектури використовувалися однакові параметри навчання та однакова організація даних.

Кількість епох навчання встановлено рівною 5, що дозволяє досягти стабільного значення метрик якості без суттєвого збільшення часу обчислень. Розмір пакета (batch size) обрано рівним 32, що є компромісом між швидкістю навчання та використанням обчислювальних ресурсів.

Для оптимізації параметрів моделей застосовано оптимізатор Adam із фіксованим значенням швидкості навчання 0.001. Як функцію втрат використано cross-entropy, що відповідає задачі багатокласової класифікації.

Налаштування моделі та параметрів навчання реалізовано відповідно до фрагмента коду, наведеного у лістингу 4.1.

Лістинг 4.1 – Ініціалізація моделі та параметрів навчання

```
model = models.mobilenet_v2(weights=models.MobileNet_V2_Weights.DEFAULT)
model.classifier[1] = nn.Linear(model.last_channel, num_classes)
model = model.to(device)
criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.parameters(), lr=0.001)
num_epochs = 5
```

Усі моделі навчалися з використанням попередньо навчених ваг із подальшою адаптацією фінального шару до кількості класів у датасеті, що становить 86 класів.

Додатково було визначено основні характеристики моделі, зокрема кількість параметрів та її розмір. Відповідний фрагмент коду та результат наведено на рисунку 4.1.

```
model_summary = summary(model, input_size=(32, 3, 224, 224), verbose=0)

print("Input size:", (3, 224, 224))
print("Total params:", model_summary.total_params)
print("Trainable params:", model_summary.trainable_params)
print("Model size (MB):", round(model_summary.total_param_bytes / (1024**2), 2))
```

```
Input size: (3, 224, 224)
Total params: 2334038
Trainable params: 2334038
Model size (MB): 8.9
```

Рисунок 4.1 – Визначення характеристик моделі

4.3 Результати навчання моделей

У процесі навчання моделей було отримано значення функції втрат та точності як на навчальній, так і на валідаційній вибірках. Динаміка зміни цих показників дозволяє оцінити ефективність навчання та стабільність роботи моделей.

Для всіх розглянутих архітектур спостерігається зменшення значення функції втрат із кожною епохою, що свідчить про поступове покращення якості моделі. Водночас значення точності на валідаційній вибірці зростає, що підтверджує здатність моделей узагальнювати отримані знання.

Найкращі результати серед розглянутих моделей продемонструвала архітектура EfficientNet-B0. За результатами навчання значення точності на

валідаційній вибірці досягло 0.9531, що є найвищим показником серед усіх моделей.

Результати навчання моделей наведено на рисунках 4.2–4.4, де представлено залежності значень функції втрат та точності від кількості епох для кожної архітектури.

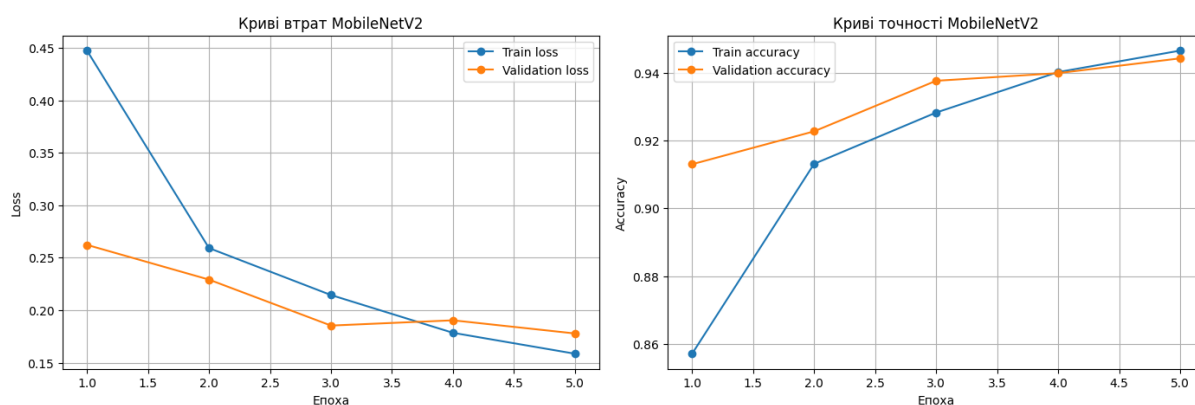


Рисунок 4.2 – Loss/Accuracy для MobileNetV2

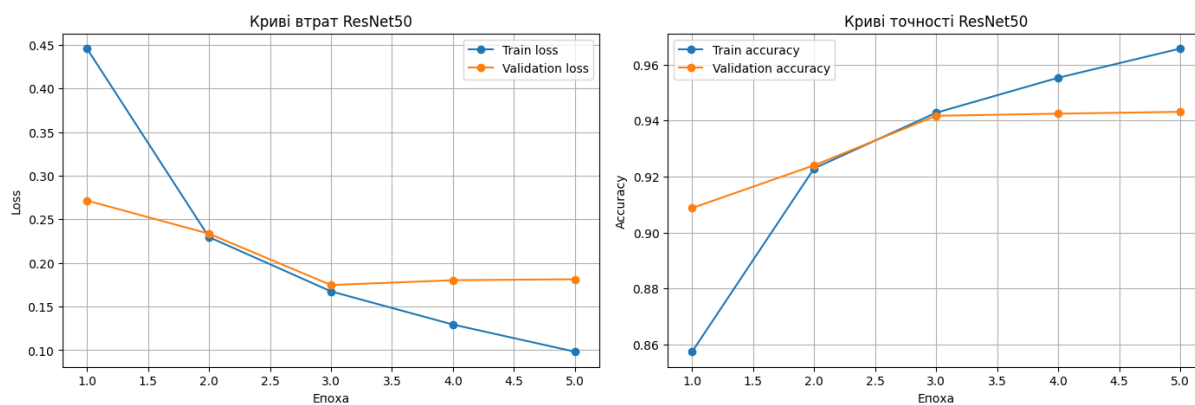


Рисунок 4.3 – Loss/Accuracy для ResNet50

Аналіз отриманих графіків показує, що всі моделі демонструють стабільну збіжність без різких коливань значень функції втрат. При цьому відсутні ознаки значного перенавчання, оскільки значення метрик на навчальній та валідаційній вибірках залишаються близькими.

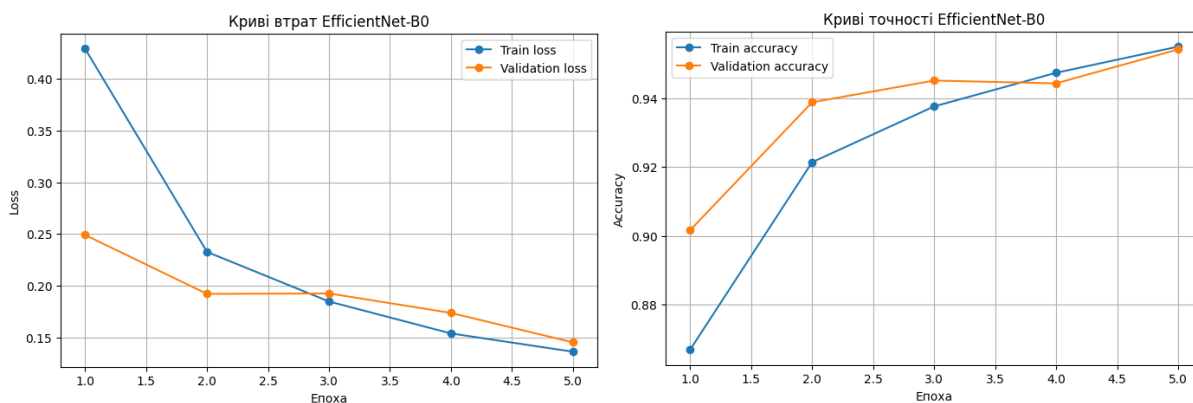


Рисунок 4.4 – Loss/Accuracy для EfficientNet-B0

Таким чином, результати навчання свідчать про ефективність використаних моделей для задачі класифікації типів ураження листя рослин та підтверджують доцільність їх подальшого порівняльного аналізу.

4.4 Порівняльний аналіз моделей

Після завершення навчання та тестування моделей було виконано їх порівняння за основними метриками якості. Для оцінювання використовувалися accuracy та macro F1-міра, що дозволяє врахувати не лише загальну частку правильних класифікацій, але й якість розпізнавання окремих класів за умов нерівномірного розподілу зображень у датасеті.

Результати порівняння моделей наведено на рисунку 4.5.

	Model	Val Accuracy	Test Accuracy	Test Macro F1
0	ResNet50	0.9431	0.944721	0.903531
1	EfficientNet-B0	0.9531	0.950325	0.903391
2	MobileNetV2	0.9442	0.946382	0.896415

Рисунок 4.5 – Порівняння моделей за основними метриками якості

Аналіз отриманих результатів показує, що найвище значення accuracy на валідаційній та тестовій вибірках продемонструвала модель EfficientNet-B0. Це свідчить про її високу загальну ефективність у задачі класифікації типів ураження листя рослин.

Разом із цим найбільше значення macro F1-міри було отримано для моделі ResNet50, хоча різниця між ResNet50 та EfficientNet-B0 є мінімальною. Такий результат вказує на те, що обидві моделі демонструють близьку якість розпізнавання окремих класів, проте EfficientNet-B0 має кращу загальну точність.

Модель MobileNetV2 показала дещо нижче значення macro F1-міри, однак її результати також залишаються високими. Це підтверджує доцільність використання цієї архітектури як легшої альтернативи для задач, де важливими є не лише точність, але й обчислювальна ефективність.

З урахуванням отриманих результатів найбільш ефективною моделлю для подальшого використання в системі було обрано EfficientNet-B0, оскільки вона забезпечує найкраще співвідношення між загальною точністю класифікації та якістю розпізнавання окремих класів.

4.5 Аналіз помилок моделей

Для більш глибокого дослідження якості роботи побудованих моделей було проведено аналіз помилок класифікації, який дозволяє виявити характерні труднощі розпізнавання та встановити причини виникнення некоректних передбачень. На відміну від узагальнених метрик якості, такий підхід дає змогу детально оцінити поведінку моделей у складних випадках та визначити закономірності помилок.

У межах реалізації класифікація здійснювалася за комбінованими класами, що враховують як тип рослини, так і тип ураження. Такий підхід обумовлений структурою використаного набору даних і дозволяє враховувати специфіку проявів захворювань для різних культур. Водночас

проведений аналіз показує, що моделі у більшості випадків орієнтуються саме на візуальні ознаки ураження, а не на належність до конкретного виду рослини. Це підтверджується характером помилок, у яких спостерігається плутанина між класами з подібними симптомами навіть за умови різних рослин, що свідчить про формування узагальнених ознак захворювань.

Для оцінювання результатів класифікації використано нормалізовані матриці помилок для кожної моделі (рисунки 4.6–4.8), а також діаграми розподілу класів з найбільшою кількістю помилок.

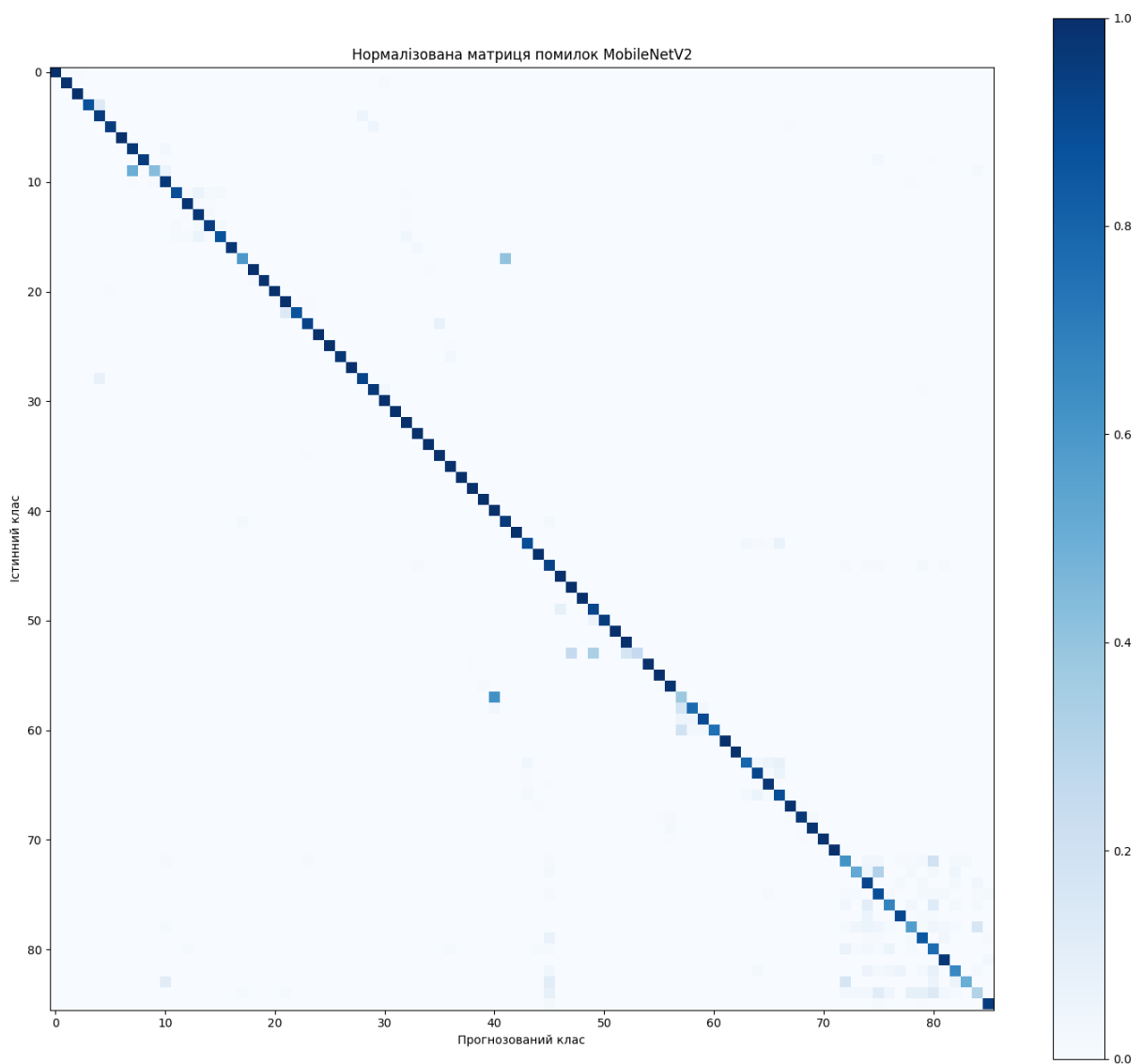


Рисунок 4.6 – Нормалізована матриця помилок моделі MobileNetV2

Нормалізована матриця помилок для моделі MobileNetV2 характеризується чітко вираженою головною діагоналлю, що свідчить про правильну класифікацію більшості зображень. Однак у матриці присутні окремі відхилення від діагоналі, які вказують на наявність плутанини між окремими класами, переважно тими, що мають схожі візуальні ознаки ураження.

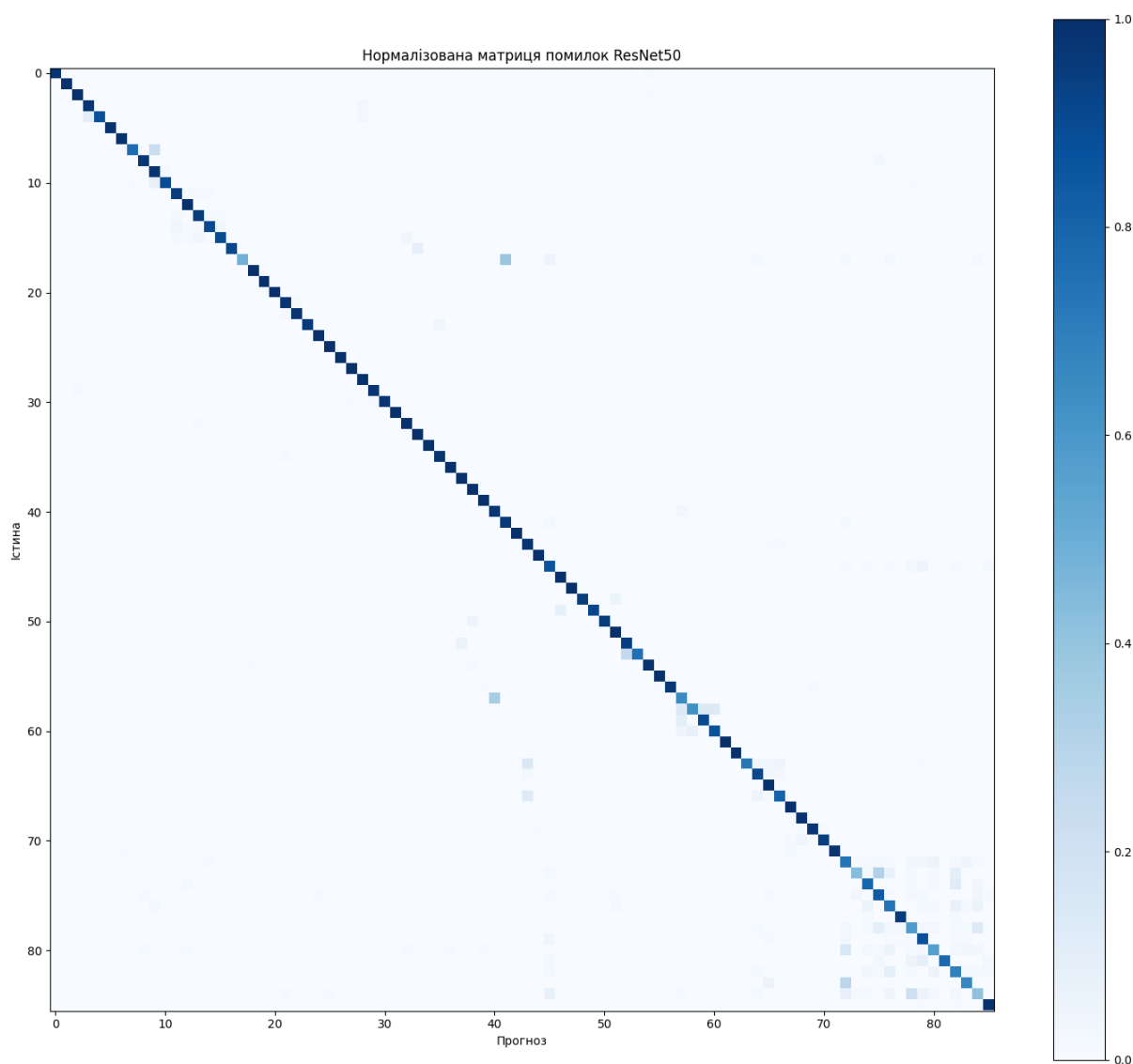


Рисунок 4.7 – Нормалізована матриця помилок моделі ResNet50

Матриця помилок для моделі ResNet50 також демонструє домінування правильних класифікацій, проте характер помилок є більш

структурованим: спостерігаються локальні групи відхилень, що свідчить про систематичну плутанину між певними класами. Така поведінка моделі вказує на те, що вона стабільно помиляється у випадках, де ознаки ураження є близькими за текстурою, формою або кольором.

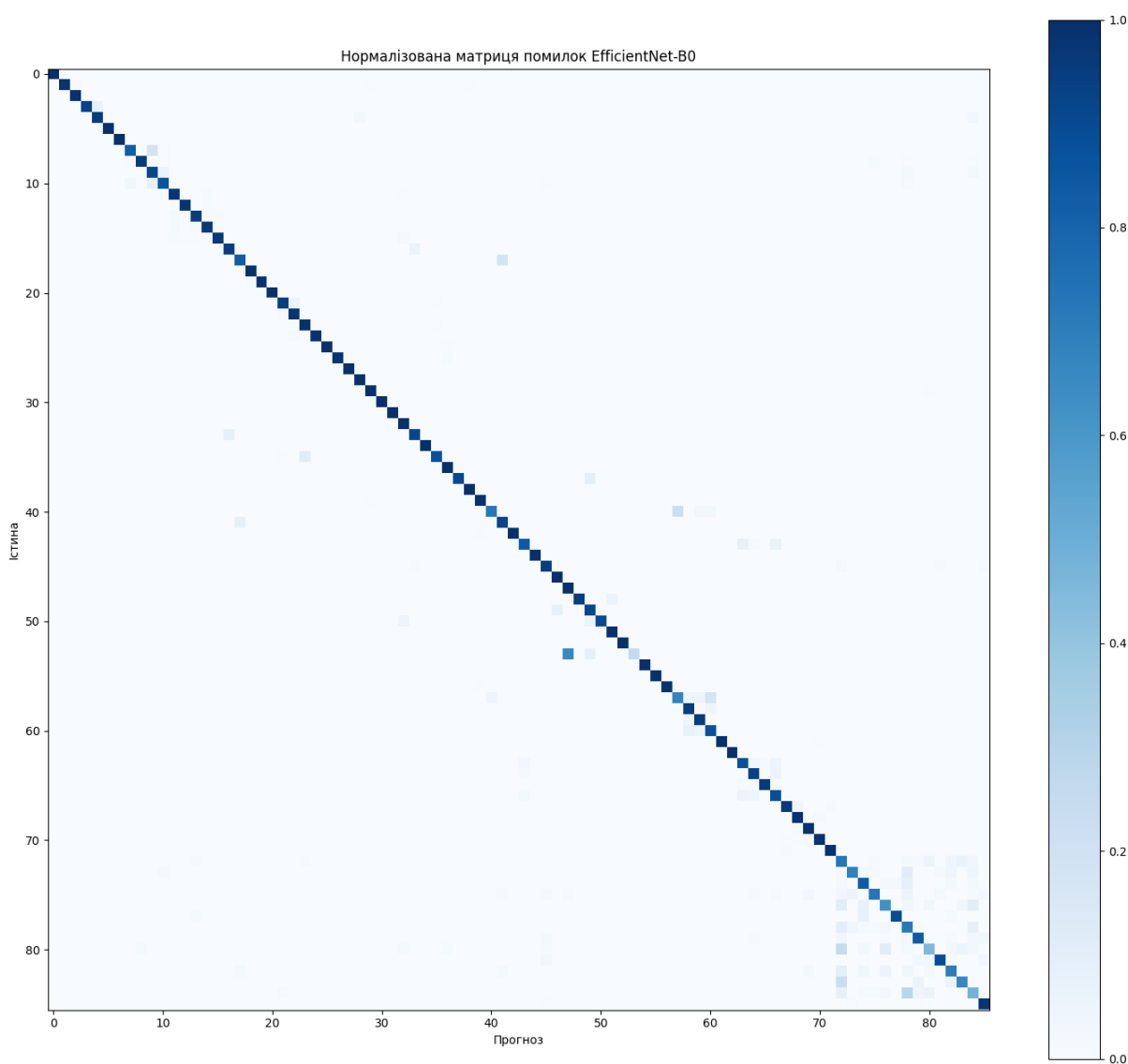


Рисунок 4.8 – Нормалізована матриця помилок моделі EfficientNet-B0

У свою чергу, матриця помилок для моделі EfficientNet-B0 характеризується найбільш чіткою діагоналлю та меншою кількістю випадкових помилок. Відхилення від діагоналі мають локалізований

характер, що свідчить про кращу здатність моделі до узагальнення та більш стабільну роботу на різних типах зображень.

Нормалізовані матриці помилок були використані для більш об'єктивного аналізу результатів класифікації в умовах нерівномірного розподілу зображень між класами. На відміну від звичайної матриці, що відображає абсолютну кількість передбачень, нормалізована матриця показує відносну частку правильних і помилкових класифікацій для кожного класу. У даній роботі нормалізація виконувалася по рядках матриці, що відповідають істинним класам, шляхом ділення кожного значення на загальну кількість зразків відповідного класу. Це дозволяє оцінити якість розпізнавання незалежно від кількості зображень у класі та уникнути переважного впливу більш представлених категорій. Застосування такого підходу дозволяє точніше визначити класи, для яких характерні найбільші труднощі класифікації.

Додатковий аналіз було проведено шляхом визначення класів, для яких спостерігається найбільша кількість помилок (рисунки 4.9–4.11).

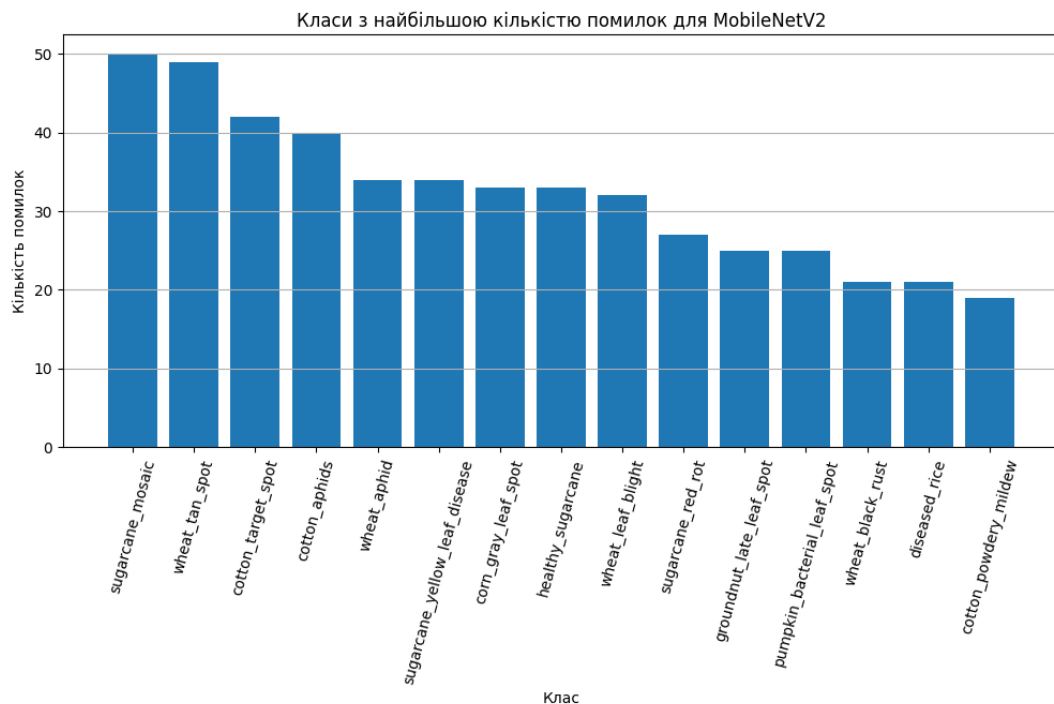


Рисунок 4.9 – Класи з найбільшою кількістю помилок для MobileNetV2

Для моделі MobileNetV2 найбільш проблемними виявилися класи, що відповідають різним видам плямистостей, мозаїчних уражень та захворювань листя. Це пояснюється тим, що подібні типи уражень мають схожі візуальні характеристики, зокрема форму та розподіл плям, що ускладнює їх розмежування навіть для людини.

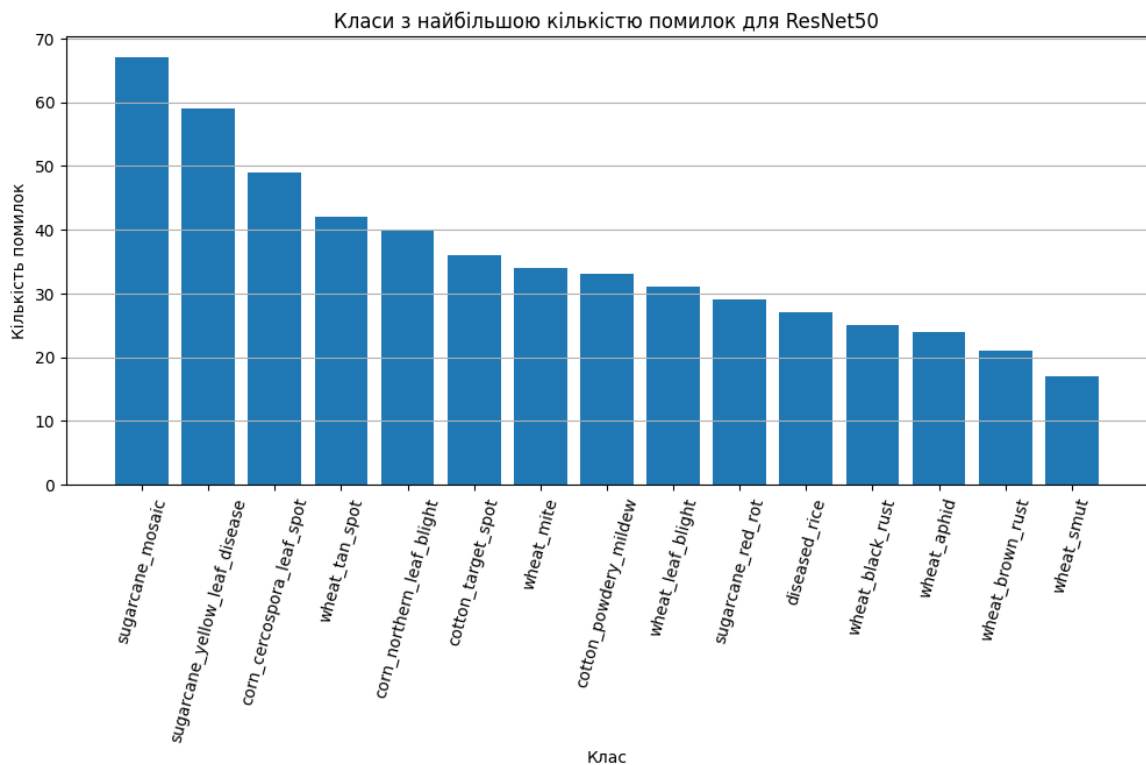


Рисунок 4.10 – Класи з найбільшою кількістю помилок для ResNet50

Аналогічна тенденція спостерігається і для моделі ResNet50, однак загальна кількість помилок для окремих класів є більшою. Це свідчить про підвищену чутливість моделі до складних випадків, хоча при цьому вона демонструє більш рівномірний розподіл якості між класами. Найбільш проблемними залишаються ті самі групи уражень, що підтверджує їх об'єктивну складність.

Для моделі EfficientNet-B0 спостерігається зменшення кількості помилок для окремих класів, проте загальна структура проблемних категорій зберігається. Це дозволяє зробити висновок, що основні труднощі

класифікації пов'язані не з особливостями конкретної архітектури, а з природою самих даних, зокрема високою схожістю ознак різних захворювань.

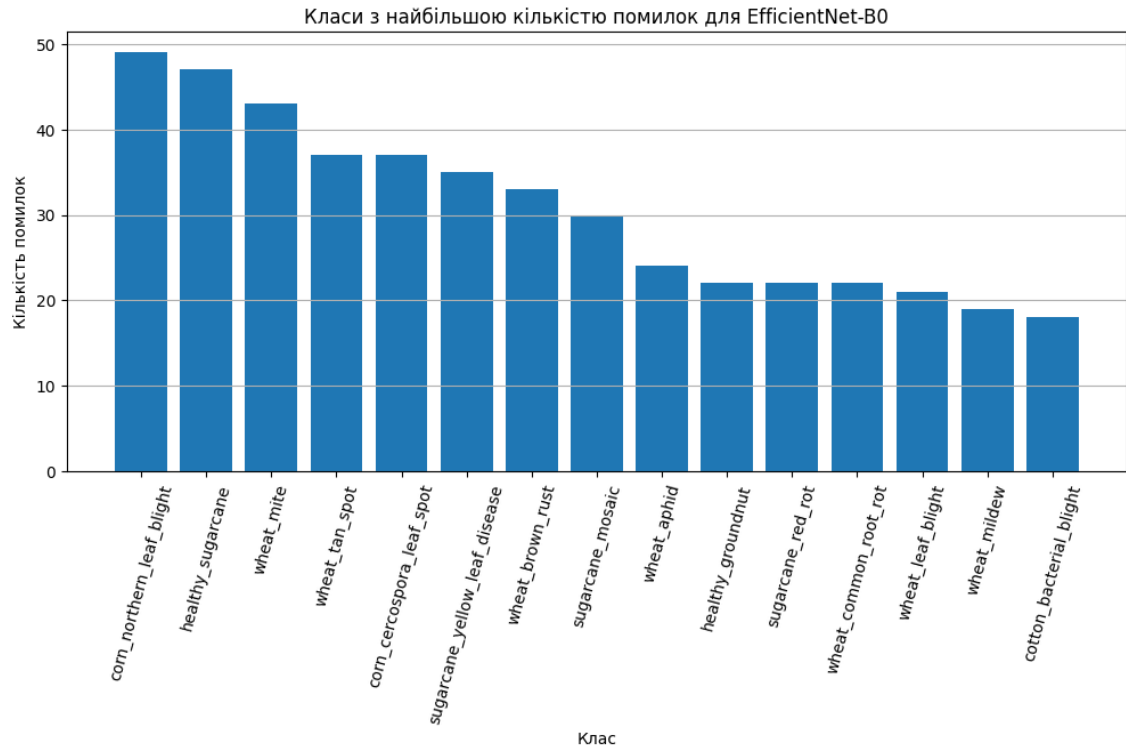


Рисунок 4.11 – Класи з найбільшою кількістю помилок для EfficientNet-B0

Для більш детального розуміння причин виникнення помилок було розглянуто окремі приклади некоректних класифікацій (рисунки 4.12–4.14).



Рисунок 4.12 – Приклади помилок класифікації MobileNetV2

Для моделі MobileNetV2 характерною є плутанина між різними типами плямистостей, а також випадки, коли слабо виражені ураження інтерпретуються як здоровий стан. Це свідчить про обмежену чутливість моделі до ранніх стадій розвитку захворювань.



Рисунок 4.13 – Приклади помилок класифікації ResNet50

Аналіз прикладів для моделі ResNet50 підтверджує аналогічні закономірності: модель схильна плутати захворювання з подібними візуальними проявами, а також іноді класифікує уражене листя як здорове у випадках слабо виражених симптомів.

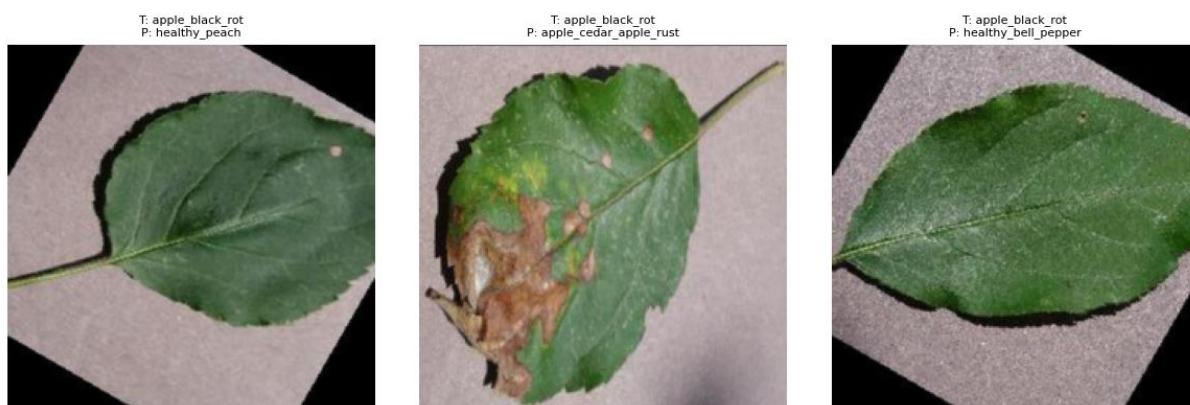


Рисунок 4.14 – Приклади помилок класифікації EfficientNet-B0

У випадку моделі EfficientNet-B0 спостерігається більш узагальнений підхід до аналізу зображення: модель орієнтується на загальний стан листя, що дозволяє зменшити кількість помилок, однак іноді призводить до ігнорування незначних дефектів. Крім того, фіксуються випадки класифікації різних типів уражень як одного класу через подібність їхніх візуальних характеристик.

Загалом результати аналізу показують, що основні помилки класифікації пов'язані з високою візуальною подібністю різних типів уражень, особливо у випадках плямистостей, грибкових захворювань та уражень листя. Додатковим фактором є наявність слабо виражених симптомів, що ускладнює їх виявлення. Важливо також відзначити, що всі розглянуті моделі демонструють подібні патерни помилок, що свідчить про складність задачі в цілому. Найкращі результати з точки зору узагальнення демонструє модель EfficientNet-B0, яка забезпечує більш стабільну класифікацію та меншу кількість випадкових помилок.

4.6 Візуалізація рішень моделі

Для підвищення інтерпретованості результатів класифікації у роботі використано метод Grad-CAM, який дозволяє визначити області зображення, що мали найбільший вплив на прийняття рішення моделлю. На відміну від числових метрик, такий підхід дає можливість візуально оцінити, чи дійсно модель орієнтується на релевантні ознаки ураження, а не на сторонні фактори, такі як фон або форма листя.

Суть методу полягає у побудові теплової карти на основі активацій згорткових шарів нейронної мережі. Отримана карта відображає інтенсивність впливу різних ділянок зображення на кінцевий результат класифікації: області з найбільшим впливом підсвічуються більш яскравими кольорами. Подальше накладання теплової карти на оригінальне

зображення дозволяє наочно проаналізувати, які саме фрагменти були використані моделлю для прийняття рішення.

На рисунку 4.15 наведено приклади результатів візуалізації Grad-CAM для правильно класифікованих зображень класу `apple_apple_scab`.

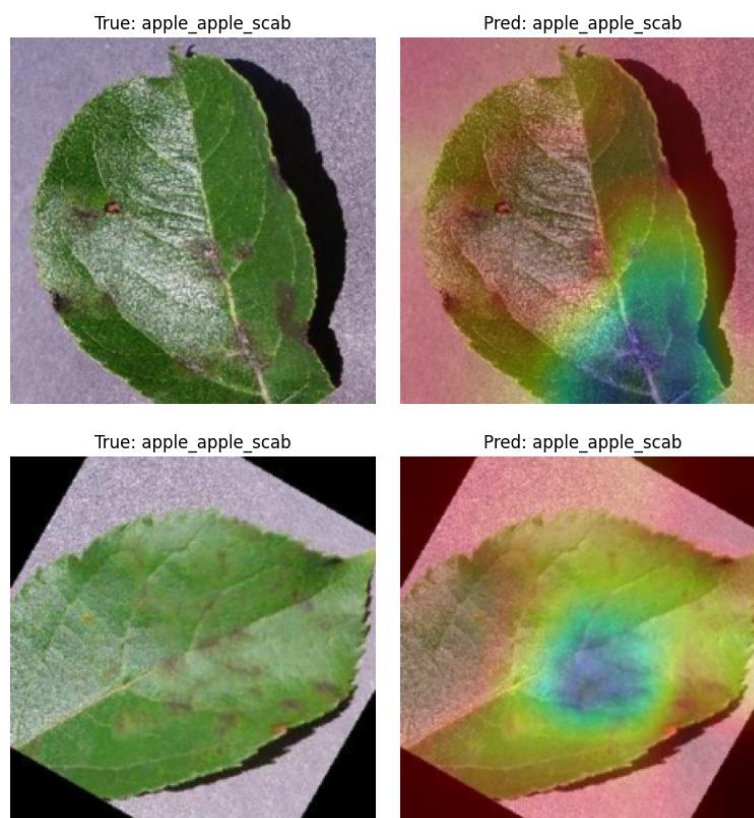


Рисунок 4.15 – Візуалізація Grad-CAM для прикладів класифікації ураження листя

Аналіз отриманих результатів показує, що модель концентрує увагу саме на уражених ділянках листя, які характеризуються наявністю темних плям та змін текстури поверхні. Зокрема, у першому прикладі чітко видно, що найбільш інтенсивні області теплової карти відповідають зоні локального ураження, тоді як здорові частини листя мають значно менший вплив на прийняття рішення. Це свідчить про те, що модель коректно ідентифікує ключові ознаки захворювання.

У другому прикладі також спостерігається виражена концентрація уваги у центральній частині ураження, що підтверджує здатність моделі локалізувати характерні патологічні зміни. При цьому інші ділянки листа, які не містять виражених симптомів, практично не впливають на результат класифікації.

Отримані результати підтверджують, що модель формує рішення на основі змістовних візуальних ознак, пов'язаних із ураженням, а не на основі другорядних характеристик зображення. Це свідчить про коректність навчання та підвищує довіру до результатів роботи системи. Таким чином, використання Grad-CAM дозволяє не лише пояснити поведінку моделі, але й додатково перевірити адекватність її роботи з точки зору предметної галузі.

4.7 Демонстрація роботи системи

Для перевірки працездатності розробленого підходу та забезпечення зручної взаємодії з користувачем було створено демонстраційний застосунок, що реалізує повний цикл обробки зображення — від завантаження до отримання результату класифікації та пояснення рішення моделі. Для реалізації інтерфейсу використано бібліотеку Streamlit, що дозволяє швидко створювати інтерактивні веб-застосунки без необхідності розробки складної клієнт-серверної архітектури.

Для демонстрації роботи системи підготовлено застосунок на Streamlit, який реалізує завантаження зображення, класифікацію, відображення ймовірності прогнозу, візуалізацію Grad-CAM та виведення текстової рекомендації.

Інтерфейс застосунку передбачає можливість завантаження зображення листа рослини у форматах JPG, JPEG або PNG. Після вибору файлу виконується автоматична обробка зображення, що включає його приведення до необхідного розміру та нормалізацію відповідно до вимог

навченої моделі. Далі зображення передається до моделі класифікації, яка визначає найбільш ймовірний клас ураження.

Результат аналізу відображається у вигляді текстової інформації, що містить визначений тип ураження українською мовою, а також значення ймовірності прогнозу. Крім того, реалізовано підсистему формування рекомендацій, яка на основі отриманого класу повертає короткі поради щодо подальших дій. У випадках, коли ймовірність прогнозу є нижчою за встановлений поріг, рекомендація формується у більш загальному вигляді, що дозволяє уникнути потенційно некоректних порад.

На рисунку 4.16 наведено приклад роботи системи для зображення з чітко вираженими ознаками ураження.

Інтелектуальна система визначення типу ураження листя рослин

Завантажте зображення для аналізу

Drag and drop file here
Limit 200MB per file • JPG, JPEG, PNG

Browse files

4.jpg 110.4KB



Завантажене зображення



Grad-CAM

Результат аналізу

Тип ураження: Чорна гниль

Ймовірність прогнозу: 95.16%

Рекомендація: Рекомендується видалити уражені частини рослини та провести обробку відповідними захисними препаратами.

Рисунок 4.16 – Приклад класифікації ураження листя з високою ймовірністю прогнозу

У даному випадку система коректно визначає тип ураження як «чорна гниль» із високим значенням ймовірності. Теплова карта Grad-CAM демонструє, що модель концентрує увагу саме на уражених ділянках листа, що підтверджує обґрунтованість отриманого результату.

На рисунку 4.17 представлено приклад класифікації для випадку з менш вираженими ознаками ураження.

Інтелектуальна система визначення типу ураження листя рослин

Завантажте зображення для аналізу

Drag and drop file here
Limit 200MB per file • JPG, JPEG, PNG

Browse files

2.jpg 76.2KB



Завантажене зображення



Grad-CAM

Результат аналізу

Тип ураження: Вірус мозаїки

Ймовірність прогнозу: 52.15%

Рекомендація: Потрібно ізолювати уражену рослину та контролювати можливих переносників вірусної інфекції.

Рисунок 4.17 – Приклад класифікації з помірною впевненістю моделі

Незважаючи на відносно нижчу ймовірність прогнозу, система визначає тип ураження як «вірус мозаїки» та надає відповідну рекомендацію. Такий приклад демонструє здатність моделі працювати з менш очевидними проявами захворювань.

На рисунку 4.18 наведено приклад обробки зображення здорового листя.

Інтелектуальна система визначення типу ураження листя рослин

Завантажте зображення для аналізу

Drag and drop file here
Limit 200MB per file • JPG, JPEG, PNG

Browse files

one.png 181.5KB

×



Завантажене зображення



Grad-CAM

Результат аналізу

Тип ураження: Здоровий стан

Ймовірність прогнозу: 97.86%

Рекомендація: Ознак ураження не виявлено. Рекомендується продовжувати стандартний догляд за рослиною.

Рисунок 4.18 – Приклад класифікації здорового стану листя

У даному випадку система правильно визначає відсутність ураження та формує рекомендацію щодо стандартного догляду. Теплова карта не містить чітко локалізованих зон уваги, що узгоджується з відсутністю патологічних змін.

На рисунку 4.19 представлено приклад класифікації для зображення кімнатної рослини.

Інтелектуальна система визначення типу ураження листя рослин

Завантажте зображення для аналізу

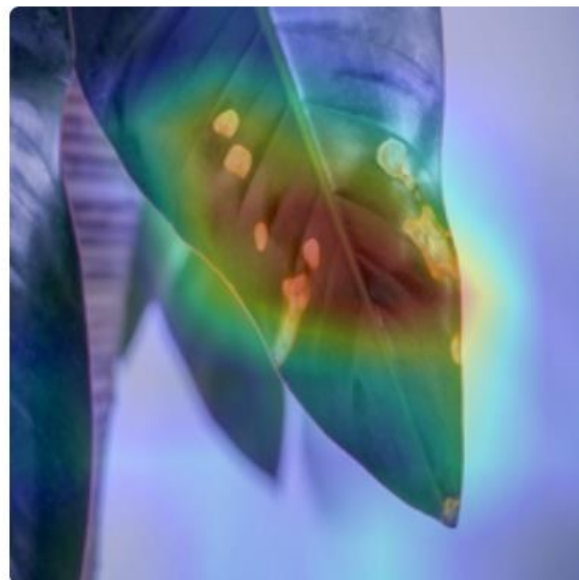

 Drag and drop file here
 Limit 200MB per file • JPG, JPEG, PNG
 Browse files

 111.jpeg 94.7KB

×



Завантажене зображення



Grad-CAM

Результат аналізу

Тип ураження: Сіра гниль

Ймовірність прогнозу: 57.06%

Рекомендація: Рекомендується видалити уражені частини та контролювати поширення інфекції.

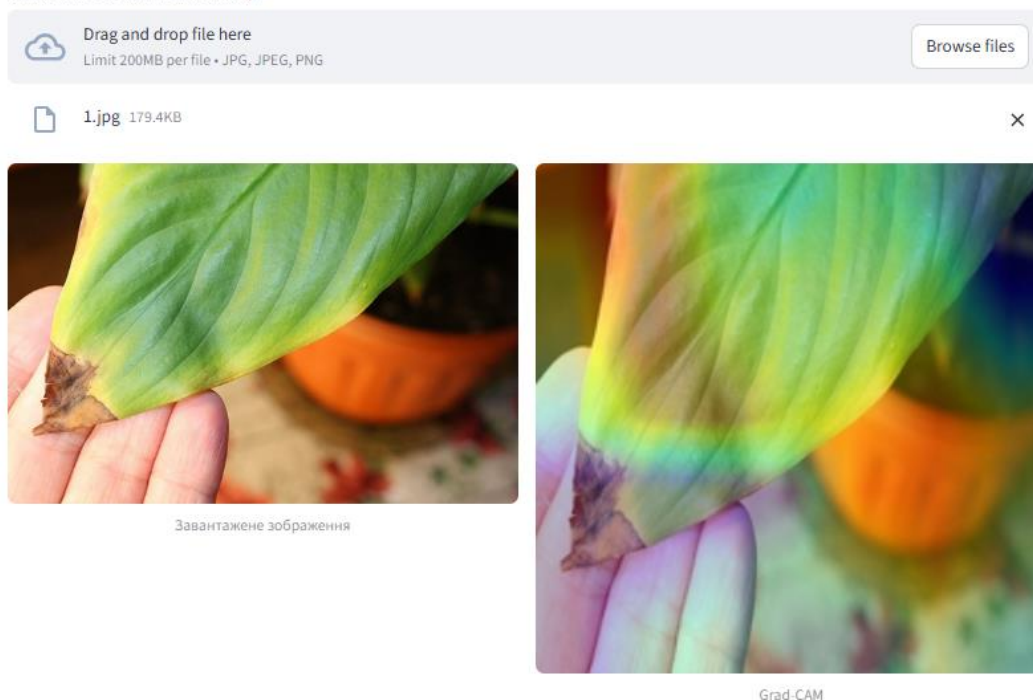
Рисунок 4.19 – Приклад роботи системи для зображення кімнатної рослини

Хоча модель навчалася переважно на зображеннях сільськогосподарських культур, вона демонструє здатність узагальнювати ознаки уражень та застосовувати їх до інших типів рослин. У даному випадку система визначає «сіру гниль» та коректно локалізує уражені ділянки.

На рисунку 4.20 наведено ще один приклад класифікації для кімнатної рослини з іншим типом пошкодження.

Інтелектуальна система визначення типу ураження листя рослин

Завантажте зображення для аналізу



Результат аналізу

Тип ураження: Некроз тканин (висихання)

Ймовірність прогнозу: 73.29%

Рекомендація: Рекомендується видалити уражені частини рослини та контролювати умови зволоження.

Рисунок 4.20 – Приклад класифікації некрозу тканин у кімнатної рослини

Модель визначає ураження як некроз тканин (висихання), що підтверджується локалізацією уваги у зоні пошкодження. Це свідчить про здатність системи орієнтуватися на загальні візуальні ознаки ураження незалежно від конкретного виду рослини.

Отримані результати демонструють, що розроблена система забезпечує коректну класифікацію як для типових випадків, так і для більш складних або нетипових зображень. Поєднання класифікації, візуалізації рішень та текстових рекомендацій дозволяє отримати не лише результат, але й пояснення його формування, що підвищує практичну цінність системи.

ВИСНОВКИ

У процесі виконання роботи було реалізовано повний цикл проєктування, розробки та дослідження інтелектуальної системи визначення типу ураження листя рослин за зображенням.

У ході аналізу предметної галузі розглянуто особливості задач комп'ютерного зору у сільському господарстві та обґрунтовано доцільність використання згорткових нейронних мереж для класифікації зображень рослин. Визначено основні підходи до розв'язання задачі та обрано архітектури моделей, що дозволяють забезпечити баланс між точністю та обчислювальною ефективністю.

Для реалізації системи використано набір даних Plant Leaf Disease MasterDataset (19 Crops), на основі якого проведено навчання та тестування моделей MobileNetV2, ResNet50 та EfficientNet-B0 із застосуванням підходу transfer learning. Оцінювання якості здійснювалося за метриками accuracy та macro F1-міри, а також за допомогою матриці помилок, що дозволило детально проаналізувати результати класифікації.

Порівняльний аналіз моделей показав відмінності у їхній здатності розпізнавати окремі класи уражень, що дозволило визначити найбільш ефективну архітектуру для розв'язання поставленої задачі. Додатковий аналіз було проведено із застосуванням методу Grad-CAM, який дозволив візуалізувати області зображення, що впливають на прийняття рішення моделлю, та оцінити коректність її поведінки.

Практична реалізація системи виконана у вигляді застосунку на Streamlit, що забезпечує можливість завантаження зображення, отримання результату класифікації, відображення ймовірностей прогнозу, візуалізацію Grad-CAM та виведення текстової рекомендації на основі визначеного типу ураження.

Отримані результати підтверджують ефективність застосування моделей глибокого навчання для автоматичного визначення уражень листя

рослин та демонструють можливість їх використання у прикладних задачах моніторингу стану рослин. Використання методів інтерпретації дозволяє підвищити довіру до результатів роботи системи та зробити її більш зрозумілою для користувача.

У подальшому можливий розвиток системи у напрямі розширення набору даних, підвищення стійкості моделей до різних умов зйомки, а також вдосконалення блоку рекомендацій шляхом розширення словника відповідностей або використання генеративних мовних моделей.

Окремим напрямом розвитку є створення повноцінного програмного продукту, зокрема веб- або мобільного застосунку з підтримкою збереження історії аналізу, можливістю пакетної обробки зображень та інтеграцією з хмарними сервісами для масштабування обчислень.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Mutka A. M., Bart R. S. Image-based phenotyping of plant disease symptoms. *Frontiers in plant science*. 2015. Vol. 5. URL: <https://doi.org/10.3389/fpls.2014.00734> (date of access: 08.06.2026).
2. Signs and symptoms of plant disease: is it fungal, viral or bacterial? URL: https://www.canr.msu.edu/news/signs_and_symptoms_of_plant_disease_is_it_fungal_viral_or_bacterial (date of access: 08.06.2026).
3. Diseases and disorders. *NC State Extension Publications | Browse Popular Publications*. URL: <https://content.ces.ncsu.edu/extension-gardener-handbook/5-diseases-and-disorders> (date of access: 08.06.2026).
4. Classification of Plant Leaf Diseases Based on Improved Convolutional Neural Network / Hang et al. *Sensors*. 2019. Vol. 19, no. 19. P. 4161. URL: <https://doi.org/10.3390/s19194161> (date of access: 09.06.2026).
5. Plant leaf disease classification using EfficientNet deep learning model / Ü. Atila et al. *Ecological Informatics*. 2021. Vol. 61. P. 101182. URL: <https://doi.org/10.1016/j.ecoinf.2020.101182> (date of access: 09.06.2026).
6. A customized MobileNetV3Large-based deep learning framework for plant disease detection / J. Rahaman et al. *Discover Artificial Intelligence*. 2026. URL: <https://doi.org/10.1007/s44163-025-00733-8> (date of access: 09.06.2026).
7. Singh B., Sharma R. P., Dey S. Interpretable plant leaf disease detection using attention-enhanced CNN. *arXiv*. 2025. URL: <https://doi.org/10.48550/arXiv.2512.17864> (date of access: 09.06.2026).
8. Enhanced plant disease classification with attention-based convolutional neural network using squeeze and excitation mechanism / S. Karthikeyan et al. *Frontiers in Artificial Intelligence*. 2025. Vol. 8. URL: <https://doi.org/10.3389/frai.2025.1640549> (date of access: 09.06.2026).
9. Aldakheel E. A., Zakariah M., Alabdall A. H. Detection and identification of plant leaf diseases using YOLOv4. *Frontiers in Plant Science*.

2024. Vol. 15. URL: <https://doi.org/10.3389/fpls.2024.1355941> (date of access: 09.06.2026).

10. Parrot Optimization Algorithm Integrated with YOLOv8 for Improving the Detection of Diseases in the Cotton Plant / A. N. Hasoon et al. *Journal of Studies in Science and Engineering*. 2025. Vol. 5, no. 2. P. 187–202. URL: <https://doi.org/10.53898/josse2025539> (date of access: 09.06.2026).

11. Ranjan R., Singh J. P., Titoriya A. K. Few-shot learning for plant disease detection using DeepBDC. *Soft Computing*. 2025. URL: <https://doi.org/10.1007/s00500-025-10884-6> (date of access: 10.06.2026).

12. Xi D., Zhang B., Wang Y.-J. U-PKAN: A dual-module Kolmogorov–arnold network for agricultural plant disease detection. *Agriculture*. 2025. Vol. 15, no. 24. P. 2599. URL: <https://doi.org/10.3390/agriculture15242599> (date of access: 10.06.2026).

13. Plant disease detection using vision transformers / M. Ali et al. *International Journal of Electrical and Computer Engineering (IJECE)*. 2025. Vol. 15, no. 2. P. 2334. URL: <https://doi.org/10.11591/ijece.v15i2.pp2334-2344> (date of access: 10.06.2026).

14. How? Why? Pl@ntNet. *Pl@ntNet*. URL: <https://plantnet.org/en/how-why/> (date of access: 10.06.2026).

15. Quantitative Plant | Leaf Doctor. *Quantitative Plant | Quantitative Plant*. URL: <https://www.quantitative-plant.org/software/leaf-doctor> (date of access: 10.06.2026).

16. Plant Disease Identifier. App – App Store. *App Store*. URL: <https://apps.apple.com/qa/app/plant-disease-identifier/id1550499292> (date of access: 10.06.2026).

17. Mohanty S. P., Hughes D. P., Salathé M. Using deep learning for image-based plant disease detection. *Frontiers in plant science*. 2016. Vol. 7. URL: <https://doi.org/10.3389/fpls.2016.01419> (date of access: 10.06.2026).

18. Tutorial 1: image filtering. *Stanford Artificial Intelligence Laboratory*.
URL: <https://ai.stanford.edu/~syyeong/cvweb/tutorial1.html> (date of access: 11.06.2026).
19. Kamilaris A., Prenafeta-Boldú F. X. Deep learning in agriculture: a survey. *Computers and electronics in agriculture*. 2018. Vol. 147. P. 70–90.
URL: <https://doi.org/10.1016/j.compag.2018.02.016> (date of access: 11.06.2026).
20. Nalawade R. R. Image processing techniques for plant disease detection. *Agriculture & Food e-Newsletter*. 2023. Vol. 5, no. 5. P. 90–92.
URL: <http://agrifoodmagazine.co.in/2023/volume-5-issue-5-may-2023/> (date of access: 11.06.2026).
21. Jha P., Dembla D., Dubey W. Implementation of machine learning classification algorithm based on ensemble learning for detection of vegetable crops disease. *International journal of advanced computer science and applications*. 2024. Vol. 15, no. 1.
URL: <https://doi.org/10.14569/ijacsa.2024.0150157> (date of access: 11.06.2026).
22. Image segmentation of overlapping leaves based on Chan–Vese model and Sobel operator / Z. Wang et al. *Information processing in agriculture*. 2018. Vol. 5, no. 1. P. 1–10. URL: <https://doi.org/10.1016/j.inpa.2017.09.005> (date of access: 12.06.2026).
23. Dutta S. Understanding the convolutional layer in convolutional neural networks (CNNs). *Medium*. 2024. URL: https://medium.com/@sanjay_dutta/understanding-the-convolutional-layer-in-convolutional-neural-networks-cnns-ef4065a0e3ca (date of access: 12.06.2026).
24. Wheat disease severity estimation: a deep learning approach / S. Nigam et al. *Internet of things and connected technologies*. Cham, 2022. P. 185–193.
URL: https://doi.org/10.1007/978-3-030-94507-7_18 (date of access: 12.06.2026).

25. Plested J., Phiri M., Gedeon T. Deep transfer learning for image classification: a survey. *Artificial intelligence review*. 2026. URL: <https://doi.org/10.1007/s10462-026-11491-z> (date of access: 12.06.2026).
26. Transfer learning: feature extraction vs fine-tuning. *Medium*. URL: <https://medium.com/@ayasc/transfer-learning-feature-extraction-vs-fine-tuning-3735eff6091e> (date of access: 12.06.2026).
27. AlexNet and imagenet: the birth of deep learning. *Pinecone*. URL: <https://www.pinecone.io/learn/series/image-search/imagenet/> (date of access: 12.06.2026).
28. Mostafid T. Overview of VGG16, ResNet50, Xception and MobileNet neural networks. *Medium*. URL: <https://medium.com/@t.mostafid/overview-of-vgg16-xception-mobilenet-and-resnet50-neural-networks-c678e0c0ee85> (date of access: 12.06.2026).
29. Kim D.-K. EfficientNet: smarter, not just bigger, neural networks. *Medium*. URL: <https://medium.com/@kdk199604/efficientnet-smarter-not-just-bigger-neural-networks-94db3e2f8699> (date of access: 12.06.2026).
30. Kumar S. Evaluation metrics for classification model. *AnalyticsVidhya*. URL: <https://www.analyticsvidhya.com/blog/2021/07/metrics-to-evaluate-your-classification-model-to-take-the-right-decisions/> (date of access: 12.06.2026).
31. What is saliency mapping in Explainable AI? *Milvus*. URL: <https://milvus.io/ai-quick-reference/what-is-saliency-mapping-in-explainable-ai> (date of access: 13.06.2026).
32. Grad-CAM: visual explanations from deep networks via gradient-based localization / R. R. Selvaraju et al. *International journal of computer vision*. 2019. Vol. 128, no. 2. P. 336–359. URL: <https://doi.org/10.1007/s11263-019-01228-7> (date of access: 13.06.2026).
33. Plant leaf disease masterdataset (19 crops). *Kaggle: The World's AI Proving Ground*. URL: <https://www.kaggle.com/datasets/adiithape1/plant-disease-detection-dataset-master-version> (date of access: 13.06.2026).