

Харківський національний університет радіоелектроніки

Факультет Навчально-науковий центр заочної форми навчання

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 Лозовій Марії Сергіївні
(прізвище, ім'я, по батькові)

1. Тема роботи Інтелектуальна вебсистема моніторингу фітопатологій пасльонових культур методами комп'ютерного зору

затверджена наказом університету від 18 травня 2026 р. № 76Стз

2. Термін подання здобувачем роботи до екзаменаційної комісії 14 липня 2026 р.

3. Вихідні дані до роботи Зображення листків томату та картоплі з набору даних PlantVillage, параметри температури та вологості, методи комп'ютерного зору і глибокого навчання, архітектура MobileNetV2, програмні засоби TensorFlow, FastAPI, ASP.NET Core MVC та SQLite.

4. Перелік питань, що потрібно опрацювати в роботі _____

1) Аналіз предметної галузі та сучасних підходів

2) Проектування інтелектуальної системи моніторингу фітопатологій

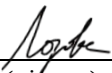
3) Реалізація модуля інтелектуальної діагностики та вебсистеми

4) Тестування та оцінка результатів роботи системи

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / термін виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	18.05.2026	виконано
2	Аналіз предметної галузі та постановка задачі дослідження	20.05.2026	виконано
3	Аналіз сучасних методів та існуючих програмних рішень	20.05.2026	виконано
4	Підготовка та дослідження набору даних	22.05.2026	виконано
5	Розроблення архітектури інтелектуальної системи	22.05.2026	виконано
6	Навчання базової моделі класифікації захворювань рослин	23.05.2026	виконано
7	Донавчання моделі та оцінювання результатів	28.05.2026	виконано
8	Реалізація механізму візуального пояснення рішень Grad-CAM	29.05.2026	виконано
9	Написання пояснювальної записки	26.06.2026	виконано
10	Нормоконтроль	03.07.2026	виконано
11	Перевірка на академічний плагіат	06.07.2026	виконано
12	Підготовка презентації та доповіді	10.07.2026	виконано
13	Рецензування	10.07.2026	виконано
14	Попередній захист	13.07.2026	виконано
15	Захист перед ЕК	14.07.2026	

Дата видачі завдання 18 травня 20 26 р.

Здобувач 
(підпис)

Керівник роботи 
(підпис)

ст. викл. Олена ГРИНЬОВА
(посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 128 с., 18 рис., 6 табл., 8 дод., 20 джерел.

ВЕБЗАСТОСУНОК, ГЛИБОКЕ НАВЧАННЯ, ДІАГНОСТИКА ЗАХВОРЮВАНЬ РОСЛИН, ЗГОРТКОВА НЕЙРОННА МЕРЕЖА, ІНТЕЛЕКТУАЛЬНА СИСТЕМА, КОМП'ЮТЕРНИЙ ЗІР, ПАСЛЬОНОВІ КУЛЬТУРИ, ФІТОПАТОЛОГІЇ, GRAD-CAM, MOBILENETV2, TRANSFER LEARNING.

Об'єкт дослідження – процес автоматизованої діагностики захворювань пасльонових культур за зображеннями листків рослин.

Предмет дослідження – методи комп'ютерного зору, глибокого навчання та побудови інтелектуальних вебсистем для виявлення фітопатологій.

Мета роботи – розроблення інтелектуальної системи мультимодального моніторингу фітопатологій та формування рекомендацій щодо захисту пасльонових культур на основі аналізу зображень листків і додаткових параметрів середовища.

Методи дослідження – методи комп'ютерного зору, згорткові нейронні мережі, transfer learning, fine-tuning, Grad-CAM, технології веброзробки ASP.NET Core MVC, FastAPI, TensorFlow та SQLite.

У кваліфікаційній роботі досліджуються методи комп'ютерного зору та глибокого навчання для автоматизованої діагностики захворювань пасльонових культур. Реалізується інтелектуальна вебсистема, що забезпечує завантаження зображень листків рослин, визначення їхнього стану, формування рекомендацій та збереження результатів перевірок. Для класифікації захворювань використовується архітектура MobileNetV2 із застосуванням підходів transfer learning і fine-tuning. Для підвищення інтерпретованості результатів використовується механізм Grad-CAM.

ABSTRACT

Bachelor's thesis contains: 128 pp., 18 fig., 6 tabl., 8 ann., 20 references.

COMPUTER VISION, CONVOLUTIONAL NEURAL NETWORK, DEEP LEARNING, GRAD-CAM, INTELLIGENT SYSTEM, MOBILENETV2, PHYTOPATHOLOGY, PLANT DISEASE DIAGNOSIS, SOLANACEOUS CROPS, TRANSFER LEARNING, WEB APPLICATION.

Object of research – the process of automated diagnosis of diseases of solanaceous crops using plant leaf images.

Subject of research – computer vision methods, deep learning approaches, and intelligent web-based systems for phytopathology detection.

Objective of the study – development of an intelligent multimodal phytopathology monitoring system and recommendation generation tool for the protection of solanaceous crops based on leaf image analysis and additional environmental parameters.

Research methods – computer vision methods, convolutional neural networks, transfer learning, fine-tuning, Grad-CAM, ASP.NET Core MVC, FastAPI, TensorFlow, and SQLite technologies.

The bachelor's thesis investigates computer vision and deep learning methods for automated diagnosis of diseases of solanaceous crops. An intelligent web-based system is implemented to provide plant leaf image uploading, plant condition identification, recommendation generation, and storage of diagnostic results. The MobileNetV2 architecture with transfer learning and fine-tuning approaches is used for disease classification. The Grad-CAM method is applied to improve the interpretability of classification results.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	8
Вступ	9
1 Аналіз предметної галузі та сучасних підходів	11
1.1 Особливості задачі автоматизованої діагностики захворювань рослин	11
1.2 Методи комп'ютерного зору для аналізу зображень рослин	13
1.3 Сучасні підходи глибокого навчання та інтерпретації результатів класифікації зображень рослин	15
1.4 Аналіз існуючих програмних рішень та наборів даних	18
1.5 Постановка задачі дослідження	20
2 Проєктування інтелектуальної системи моніторингу фітопатологій	23
2.1 Постановка вимог до системи	23
2.2 Архітектура та логіка функціонування системи	24
2.3 Проєктування бази даних системи	30
2.4 Проєктування модуля інтелектуальної діагностики	33
3 Реалізація модуля інтелектуальної діагностики та вебсистеми	38
3.1 Підготовка та аналіз набору даних	38
3.2 Реалізація та навчання моделі класифікації	41
3.3 Доновчання моделі та реалізація Grad-CAM	45
3.4 Реалізація сервісу інтелектуальної діагностики на FastAPI	49
3.5 Реалізація вебзастосунку ASP.NET Core MVC	52
4 Тестування та оцінка результатів роботи системи	56
4.1 Результати навчання базової моделі	56
4.2 Результати донавчання моделі	59
4.3 Аналіз роботи системи на тестових зображеннях	62
4.4 Перевірка функціонування вебсистеми	64
Висновки	68
Перелік джерел посилання	71

Додаток А Код навчання моделі класифікації.....	74
Додаток Б Код донавчання моделі класифікації	80
Додаток В Код реалізації модуля Grad-CAM	86
Додаток Г Код FastAPI-сервісу інтелектуальної діагностики	90
Додаток Д Код модуля рекомендацій	94
Додаток Е Код вебзастосунку ASP.NET Core MVC.....	102
Додаток Ж Код сторінок користувацького інтерфейсу	108
Додаток И Відомість кваліфікаційної роботи	128

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Accuracy – метрика точності класифікації;

API – Application Programming Interface – програмний інтерфейс взаємодії між компонентами системи;

ASP.NET Core MVC – фреймворк для розроблення вебзастосунків із використанням архітектурного шаблону Model-View-Controller;

Base64 – спосіб кодування бінарних даних у текстовий формат;

CNN – Convolutional Neural Network – згорткова нейронна мережа;

FastAPI – Python-фреймворк для створення вебсервісів та API;

Fine-tuning – донавчання частини шарів попередньо навченої нейронної мережі на новому наборі даних;

Grad-CAM – Gradient-weighted Class Activation Mapping – метод візуального пояснення результатів класифікації зображень;

Loss – значення функції втрат, що характеризує помилку моделі під час навчання;

MobileNetV2 – архітектура згорткової нейронної мережі, оптимізована для ефективної класифікації зображень;

SQLite – вбудована реляційна система керування базами даних;

Transfer learning – перенесення навчання, підхід до використання попередньо навченої моделі для розв’язання нової задачі;

XAI – Explainable Artificial Intelligence – пояснюваний штучний інтелект.

ВСТУП

У сучасних умовах розвитку агропромислового комплексу особливої актуальності набуває підвищення ефективності вирощування сільськогосподарських культур та забезпечення їх захисту від захворювань. Ураження рослин патогенами призводить до зниження врожайності, погіршення якості продукції та значних економічних втрат. Відповідно до сучасних досліджень, своєчасне виявлення ознак хвороб є одним із ключових чинників забезпечення стабільного агровиробництва.

Традиційне оцінювання стану рослин здійснюється шляхом візуального огляду агрономами та спеціалістами із захисту рослин. Проте такий підхід є трудомістким, потребує високої кваліфікації експерта та значною мірою залежить від людського фактора. Крім того, доступ до фахових консультацій часто є обмеженим для невеликих фермерських господарств і приватних землекористувачів, що ускладнює своєчасне виявлення фітопатологій та прийняття необхідних заходів.

У зв'язку з цим зростає інтерес до застосування технологій штучного інтелекту, зокрема методів комп'ютерного зору та глибокого навчання, для автоматизації процесів моніторингу стану рослин [1]. Результати сучасних досліджень свідчать, що згорткові нейронні мережі здатні ефективно розпізнавати захворювання за цифровими зображеннями листків, забезпечуючи високі показники точності класифікації [2], [3]. Це робить такі підходи перспективною основою для створення інтелектуальних засобів підтримки прийняття рішень у сільському господарстві.

Серед овочевих культур важливе місце займають представники родини пасльонових, які широко використовуються у виробництві та є вразливими до різноманітних грибкових, бактеріальних і вірусних захворювань. Необхідність регулярного контролю їхнього стану в поєднанні з потребою оперативного реагування на появу симптомів обумовлює доцільність використання автоматизованих засобів аналізу.

У наукових роботах останніх років показано, що використання методів глибокого навчання разом із відкритими наборами даних, зокрема PlantVillage, дозволяє досягати високих показників якості класифікації стану рослин. Це підтверджує актуальність розроблення програмних рішень, орієнтованих на практичне використання в умовах сучасного цифрового землеробства.

У цьому контексті актуальною є розробка інтелектуальної вебсистеми моніторингу фітопатологій пасльонових культур, яка дозволяє користувачу завантажувати зображення листків рослин, автоматично визначати їхній стан за допомогою моделі глибокого навчання, отримувати інформацію про виявлене захворювання, рекомендації щодо подальших дій та зберігати результати проведеної діагностики. Додатково у системі реалізовано експертний модуль рекомендацій, що формує поради щодо догляду за рослинами з урахуванням введених користувачем параметрів температури та вологості.

Важливим напрямом розвитку сучасних інтелектуальних систем є забезпечення інтерпретованості результатів роботи моделей машинного навчання. З цією метою використовуються методи Explainable Artificial Intelligence (XAI), які дозволяють візуалізувати причини прийняття рішення моделлю. Одним із найбільш поширених підходів є метод Grad-CAM, що забезпечує відображення ділянок зображення, які найбільше вплинули на результат класифікації. Це підвищує довіру користувача до отриманих результатів та спрощує їх аналіз.

Результати роботи можуть бути використані для створення доступних цифрових інструментів підтримки прийняття рішень у сільському господарстві, що сприятиме підвищенню ефективності вирощування пасльонових культур, своєчасному виявленню захворювань та зменшенню втрат урожаю.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА СУЧАСНИХ ПІДХОДІВ

1.1 Особливості задачі автоматизованої діагностики захворювань рослин

Автоматизована діагностика захворювань рослин є важливою складовою сучасного аграрного виробництва, оскільки своєчасне виявлення фітопатологій безпосередньо впливає на врожайність та якість сільськогосподарської продукції. За даними Продовольчої та сільськогосподарської організації ООН (FAO), втрати врожаю внаслідок впливу хвороб і шкідників можуть досягати 40% щорічно, що створює суттєві економічні ризики та негативно впливає на продовольчу безпеку [4]. У зв'язку з цим особливого значення набуває впровадження сучасних технологій моніторингу стану рослин.

Традиційні методи діагностики базуються на візуальному огляді рослин агрономами або спеціалістами із захисту рослин. Під час такого аналізу оцінюються зовнішні ознаки захворювань, зокрема зміна кольору листя, поява плям, деформацій, некрозів та інших симптомів. Проте ефективність подібного підходу значною мірою залежить від досвіду фахівця та умов проведення обстеження. Крім того, ручна діагностика є трудомісткою та не дозволяє оперативно обробляти великі обсяги інформації. З точки зору інформаційних технологій, задача діагностики захворювань рослин може бути сформульована як задача аналізу даних, у якій основним джерелом інформації виступають цифрові зображення листків. У більшості випадків така задача розглядається як задача класифікації, де необхідно визначити належність зображення до одного з наперед визначених класів, що відповідають певному захворюванню або здоровому стану рослини. У більш складних системах можуть також використовуватися підходи детекції та сегментації, які дозволяють локалізувати окремі уражені ділянки.

Реальні умови отримання зображень істотно ускладнюють процес автоматизованої діагностики. Зображення можуть відрізнятися за рівнем освітлення, ракурсом зйомки, якістю камери, фоном та іншими параметрами. Додаткову складність створює те, що прояви одного й того самого захворювання можуть змінюватися залежно від стадії його розвитку та фізіологічного стану рослини. Усе це впливає на якість класифікації та вимагає використання стійких до таких змін алгоритмів аналізу.

Окремою проблемою є подібність симптомів різних захворювань. Наприклад, пожовтіння або поява плям на листках можуть бути викликані як інфекційними захворюваннями, так і дефіцитом поживних речовин чи несприятливими умовами вирощування. У зв'язку з цим системи автоматизованої діагностики повинні враховувати широкий набір візуальних ознак для забезпечення достатньої точності розпізнавання.

Важливим фактором розвитку багатьох захворювань є умови навколишнього середовища. У фітопатологічних дослідженнях показано, що температура та вологість суттєво впливають на процес інфікування рослин і подальший розвиток хвороб [5]. Зокрема, підвищена вологість часто створює сприятливі умови для поширення грибкових інфекцій, тоді як температурний режим впливає на швидкість розвитку патогенів. Тому інформація про параметри середовища може використовуватися для формування додаткових рекомендацій щодо догляду за рослинами та профілактики захворювань.

Сучасні дослідження демонструють високу ефективність застосування методів глибокого навчання для розпізнавання захворювань рослин за зображеннями листя. Особливо широкого поширення набули згорткові нейронні мережі, які здатні автоматично виділяти інформативні ознаки зображень та виконувати класифікацію без необхідності ручного проєктування ознак. У наукових роботах повідомляється про досягнення високих показників точності класифікації на спеціалізованих наборах

даних, зокрема PlantVillage, що підтверджує ефективність використання методів комп'ютерного зору для автоматизації діагностики рослин [6].

Таким чином, автоматизована діагностика захворювань рослин є складною багатофакторною задачею, яка потребує використання сучасних методів комп'ютерного зору та машинного навчання. Застосування таких підходів створює передумови для розроблення інтелектуальних вебсистем моніторингу фітопатологій, здатних забезпечувати оперативне визначення стану рослин та підтримку прийняття рішень у сфері рослинництва.

1.2 Методи комп'ютерного зору для аналізу зображень рослин

Комп'ютерний зір є одним із провідних напрямів сучасного штучного інтелекту, що забезпечує автоматизований аналіз цифрових зображень та відеоданих. У сфері сільського господарства методи комп'ютерного зору широко використовуються для моніторингу стану рослин, виявлення захворювань, контролю якості продукції та підтримки прийняття рішень під час вирощування культур. У задачах діагностики захворювань рослин основним джерелом інформації виступають цифрові зображення листків, на яких можуть бути присутні характерні симптоми ураження: плями, зміна кольору, деформації, некрози або інші візуальні ознаки. Завдання системи комп'ютерного зору полягає у виділенні інформативних ознак зображення та подальшому визначенні стану рослини.

Методи комп'ютерного зору умовно можна поділити на класичні методи обробки зображень та сучасні підходи, засновані на глибокому навчанні. Класичні методи передбачають ручне проєктування ознак і використання алгоритмів цифрової обробки зображень. До них належать фільтрація шумів, аналіз кольорових характеристик, виділення контурів, сегментація та аналіз текстурних особливостей.

На етапі попередньої обробки зображень можуть застосовуватися методи нормалізації яскравості, корекції контрастності, зміни розміру

зображень та усунення шумів. Метою таких процедур є підвищення якості вхідних даних та зменшення впливу зовнішніх факторів, зокрема різних умов освітлення або особливостей використовуваної камери.

Одним із поширених підходів класичного комп'ютерного зору є сегментація зображень, яка полягає у виділенні окремих областей, що мають спільні характеристики. У задачах аналізу рослин сегментація може використовуватися для відокремлення листка від фону або локалізації потенційно уражених ділянок. Для цього застосовуються методи порогової обробки, кластеризації та аналізу кольорових просторів.

Проте класичні підходи мають низку обмежень. Ефективність їх роботи значною мірою залежить від якості ручного підбору ознак, а також від умов отримання зображень. У реальних умовах листки можуть відрізнятися формою, розміром, кольором, ракурсом зйомки та ступенем ураження, що ускладнює використання традиційних алгоритмів.

Суттєвий розвиток комп'ютерного зору відбувся завдяки появі методів глибокого навчання, зокрема згорткових нейронних мереж (Convolutional Neural Networks, CNN). На відміну від класичних методів, CNN здатні автоматично виділяти інформативні ознаки без необхідності їх ручного проєктування. Це дозволяє ефективно працювати зі складними наборами зображень та забезпечувати високу якість класифікації.

Принцип роботи згорткових нейронних мереж базується на використанні згорткових фільтрів, які послідовно аналізують локальні області зображення. На початкових рівнях мережі виділяються прості ознаки, такі як контури, градієнти та кольорові переходи. На більш глибоких рівнях формуються складніші представлення, які описують текстурні особливості, структуру об'єктів та характерні патерни захворювань. Такий ієрархічний підхід дозволяє враховувати як локальні, так і глобальні характеристики зображення [7].

Архітектура CNN зазвичай включає згорткові шари, шари підвибірки (pooling) та класифікаційні шари. Згорткові шари відповідають за виділення ознак, шари підвибірки зменшують розмірність даних і підвищують стійкість до шумів, а фінальні шари виконують класифікацію об'єкта до одного з визначених класів.

У сучасних системах аналізу зображень широко використовуються архітектури VGG, ResNet, EfficientNet та MobileNet. Особливий інтерес становить архітектура MobileNet, яка характеризується меншою кількістю параметрів і нижчими обчислювальними витратами порівняно з багатьма іншими моделями. Завдяки цьому вона придатна для використання у веборієнтованих системах та застосунках, де важливими є швидкість роботи та ефективне використання обчислювальних ресурсів.

Важливим етапом підготовки даних для навчання моделей комп'ютерного зору є попередня обробка та аугментація зображень. До таких операцій належать зміна масштабу, повороти, віддзеркалення, модифікація яскравості та контрастності. Аугментація дозволяє збільшити різноманітність навчальної вибірки, підвищити стійкість моделі до змін умов зйомки та зменшити ризик перенавчання.

Таким чином, сучасні методи комп'ютерного зору забезпечують ефективні засоби аналізу цифрових зображень рослин. Особливо перспективними для задач автоматизованої діагностики є згорткові нейронні мережі, які здатні автоматично виділяти ознаки захворювань та виконувати класифікацію з високою точністю. Саме такі підходи є основою сучасних інтелектуальних систем моніторингу фітопатологій рослин.

1.3 Сучасні підходи глибокого навчання та інтерпретації результатів класифікації зображень рослин

Подальший розвиток методів комп'ютерного зору безпосередньо пов'язаний із використанням технологій глибокого навчання. Завдяки

здатності автоматично виділяти інформативні ознаки із зображень згорткові нейронні мережі стали основою більшості сучасних систем аналізу візуальних даних, зокрема й систем діагностики захворювань рослин.

Незважаючи на високу ефективність глибоких нейронних мереж, їх навчання потребує значних обсягів розмічених даних та суттєвих обчислювальних ресурсів. У сфері фітопатології формування великих навчальних вибірок є складним завданням, оскільки потребує накопичення великої кількості зображень різних захворювань та участі фахівців для їх правильної класифікації. У зв'язку з цим широкого поширення набув підхід перенесення навчання (transfer learning).

Transfer learning передбачає використання знань, отриманих під час розв'язання однієї задачі, для підвищення ефективності розв'язання іншої задачі. У задачах комп'ютерного зору цей підхід реалізується шляхом використання попередньо навчених нейронних мереж, які були навчені на великих наборах зображень і вже містять узагальнені візуальні ознаки. Використання попередньо навчених моделей дає змогу переносити знання між спорідненими задачами та підвищувати якість класифікації навіть за обмеженої кількості даних [8].

Одним із найбільш поширених джерел попереднього навчання є набір даних ImageNet, що містить мільйони зображень різних категорій. У процесі навчання на такому наборі даних нейронна мережа формує універсальні представлення зображень, здатні описувати контури, текстури, форми та інші базові візуальні характеристики. Надалі ці знання можуть бути використані для розв'язання спеціалізованих прикладних задач, зокрема задач класифікації захворювань рослин.

Практичне застосування transfer learning зазвичай передбачає використання попередньо навченої моделі як базового екстрактора ознак із подальшою адаптацією до нової предметної галузі. Одним із поширених підходів є fine-tuning, який полягає у додатковому навчанні частини шарів нейронної мережі на новому наборі даних. Це дозволяє адаптувати модель

до особливостей конкретної задачі та водночас зберегти знання, отримані під час попереднього навчання.

Використання transfer learning і fine-tuning дозволяє скоротити час навчання моделей, зменшити потребу у великих обсягах розмічених даних та підвищити якість класифікації. Саме тому такі підходи активно використовуються у сучасних системах комп'ютерного зору для аграрної галузі [9].

Разом із підвищенням точності моделей машинного навчання зростає потреба в забезпеченні зрозумілості та інтерпретованості отриманих результатів. Більшість сучасних нейронних мереж мають складну внутрішню структуру, через що процес прийняття рішення часто залишається незрозумілим для кінцевого користувача. У зв'язку з цим останніми роками активно розвивається напрям Explainable Artificial Intelligence (XAI), спрямований на пояснення результатів роботи моделей штучного інтелекту.

У задачах аналізу зображень одним із найбільш поширених методів інтерпретації є Gradient-weighted Class Activation Mapping (Grad-CAM). Цей підхід дозволяє формувати теплові карти, які візуально відображають області зображення, що найбільше вплинули на результат класифікації. На отриманих картах більш теплі кольори відповідають ділянкам, на які модель звертала найбільшу увагу під час прийняття рішення, тоді як холодні кольори відповідають менш значущим областям.

Використання методів інтерпретації дозволяє оцінити коректність роботи моделі, підвищити довіру користувачів до отриманих результатів та забезпечити додатковий контроль якості автоматизованої діагностики. Особливо важливим це є для прикладних систем, що використовуються для підтримки прийняття рішень у сфері сільського господарства.

Таким чином, сучасні підходи глибокого навчання поєднують високу точність класифікації із можливістю адаптації попередньо навчених моделей до нових предметних областей. Додаткове використання методів

Explainable AI дозволяє підвищити інтерпретованість результатів роботи нейронних мереж, що є важливим фактором під час створення інтелектуальних систем моніторингу фітопатологій рослин.

1.4 Аналіз існуючих програмних рішень та наборів даних

Розвиток методів комп'ютерного зору та глибокого навчання сприяв появі значної кількості наукових і прикладних рішень, орієнтованих на автоматизовану діагностику захворювань рослин. Такі рішення зазвичай базуються на аналізі цифрових зображень листків, за якими модель визначає стан рослини або наявність певного захворювання. Ефективність таких систем значною мірою залежить від якості навчальних даних, обраної архітектури нейронної мережі та умов, у яких отримані зображення.

Одним із найбільш відомих відкритих наборів даних для задач діагностики захворювань рослин є PlantVillage. Він містить зображення здорових і уражених листків різних сільськогосподарських культур та широко використовується у дослідженнях з комп'ютерного зору. Згідно з відкритим репозиторієм PlantVillage, набір містить понад 54 тис. зображень листків рослин, що охоплюють різні культури та захворювання [10]. У бібліотеці TensorFlow Datasets цей набір також подано як колекцію зображень здорових і хворих листків, поділених на 38 категорій за видом рослини та захворюванням.

Перевагою PlantVillage є наявність великої кількості розмічених зображень, що дозволяє ефективно навчати та порівнювати моделі глибокого навчання. Саме цей датасет часто використовується як базовий набір для досліджень у сфері автоматизованої діагностики рослин, зокрема у роботах, присвячених застосуванню згорткових нейронних мереж. Водночас важливо враховувати, що значна частина зображень у PlantVillage отримана в контрольованих умовах, з відносно однорідним фоном і стабільним освітленням. Це спрощує процес класифікації, але може

створювати розрив між результатами на тестових даних і роботою системи в реальному середовищі. Для навчання та тестування моделі в межах даної роботи використано відкритий набір даних Plant Disease Classification Dataset Split into Train Test and Validation Sets, розміщений на платформі Kaggle [11]. Даний набір даних сформований на основі датасету PlantVillage та містить зображення рослин, попередньо розподілених на навчальну, валідаційну та тестову вибірки. Така структура спрощує процес побудови та оцінювання моделей машинного навчання, забезпечуючи коректне розділення даних на етапах навчання і тестування.

Окрім відкритих датасетів, існують також прикладні програмні рішення для діагностики стану рослин. Одним із відомих прикладів є мобільний застосунок Plantix, який дозволяє користувачу зробити або завантажити фото ураженої рослини та отримати діагноз і рекомендації щодо подальших дій. Такі рішення демонструють практичну затребуваність автоматизованої діагностики у сільському господарстві та підтверджують доцільність створення доступних інструментів для фермерів, агрономів і власників присадибних ділянок.

Типовими функціями подібних систем є завантаження зображення, автоматичне визначення проблеми, відображення короткого опису захворювання та надання рекомендацій. Деякі рішення також містять довідкову інформацію або додаткові матеріали щодо догляду за культурами. Водночас для навчальних і дослідницьких вебсистем важливими залишаються не лише точність класифікації, а й прозорість результатів, можливість збереження історії перевірок та зручність інтеграції моделі машинного навчання з користувацьким інтерфейсом.

Незважаючи на значний розвиток існуючих рішень, можна виділити низку обмежень. По-перше, якість роботи моделей суттєво залежить від того, наскільки навчальні дані відповідають реальним умовам використання. Якщо модель навчалася переважно на зображеннях із

контрольованим фоном, її точність може знижуватися під час аналізу фотографій, отриманих у складніших умовах освітлення, фону або ракурсу.

По-друге, не всі прикладні системи надають користувачу пояснення того, які саме ділянки зображення вплинули на рішення моделі. Для задач підтримки прийняття рішень це є важливим аспектом, оскільки користувач повинен мати змогу оцінити, чи справді модель звернула увагу на симптоми ураження, а не на випадкові особливості фону або якості зображення. У цьому контексті актуальним є використання методів Explainable AI, зокрема Grad-CAM.

По-третє, рекомендації у багатьох системах мають загальний характер і не завжди враховують додаткові умови, зазначені користувачем. Водночас температура та вологість є важливими факторами розвитку фітопатологій, тому їх урахування може бути корисним під час формування практичних порад щодо догляду за рослинами. У цьому випадку йдеться не про зміну результату нейромережевої класифікації, а про доповнення діагностичного висновку рекомендаціями експертного характеру.

Проведений аналіз показує, що відкриті набори даних, зокрема PlantVillage, створюють основу для навчання моделей комп'ютерного зору, а існуючі прикладні рішення підтверджують практичну цінність автоматизованої діагностики рослин. Водночас актуальним залишається створення веборієнтованих систем, які поєднують класифікацію захворювань за зображеннями, формування рекомендацій, збереження результатів перевірок та пояснення рішень моделі. Саме такі вимоги покладено в основу розроблення інтелектуальної вебсистеми моніторингу фітопатологій пасльонових культур.

1.5 Постановка задачі дослідження

Проведений аналіз предметної галузі, сучасних методів комп'ютерного зору, підходів глибокого навчання та існуючих програмних

рішень показав доцільність використання технологій штучного інтелекту для автоматизованої діагностики захворювань рослин. Водночас було встановлено, що для практичного застосування таких рішень важливими є не лише точність класифікації, а й зручність взаємодії з користувачем, можливість отримання рекомендацій та інтерпретація результатів роботи моделі.

Об'єктом дослідження є процес автоматизованої діагностики захворювань пасльонових культур за цифровими зображеннями листя.

Предметом дослідження є методи комп'ютерного зору, глибокого навчання та програмні засоби реалізації веборієнтованих систем моніторингу фітопатологій рослин.

Метою роботи є розроблення інтелектуальної вебсистеми моніторингу фітопатологій пасльонових культур, яка забезпечує автоматичне визначення стану рослини за зображенням листка, формування рекомендацій щодо подальших дій та надання користувачу засобів для аналізу результатів діагностики.

Для досягнення поставленої мети необхідно розв'язати такі задачі:

- проаналізувати особливості автоматизованої діагностики захворювань рослин та сучасні підходи до її реалізації;
- дослідити методи комп'ютерного зору та глибокого навчання, що застосовуються для класифікації зображень рослин;
- виконати аналіз існуючих програмних рішень і наборів даних для задач діагностики захворювань рослин;
- обґрунтувати вибір архітектури та технологій, необхідних для створення вебсистеми;
- підготувати та використати набір даних для навчання моделі класифікації захворювань рослин;
- реалізувати програмний модуль автоматичної діагностики на основі моделі глибокого навчання;

- реалізувати вебінтерфейс для завантаження зображень, перегляду результатів аналізу та збереження історії перевірок;
- розробити модуль формування рекомендацій на основі результатів діагностики та введених користувачем параметрів середовища;
- реалізувати механізм візуального пояснення результатів класифікації із використанням підходів Explainable Artificial Intelligence;
- провести тестування розробленої системи та оцінити коректність її роботи.

Результатом виконання роботи має стати інтелектуальна вебсистема моніторингу фітопатологій пасльонових культур, призначена для автоматизованого аналізу зображень листків рослин, отримання діагностичних висновків, рекомендацій щодо догляду та візуалізації причин прийняття рішення моделлю.

2 ПРОЄКТУВАННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ МОНІТОРИНГУ ФІТОПАТОЛОГІЙ

2.1 Постановка вимог до системи

На основі результатів аналізу предметної галузі, сучасних методів комп'ютерного зору та існуючих програмних рішень було сформовано основні вимоги до розроблюваної системи. Система повинна забезпечувати автоматизований аналіз зображень листя рослин, визначення їхнього стану та надання користувачу рекомендацій на основі результатів діагностики.

Основним призначенням системи є діагностика за цифровим зображенням листка із використанням методів глибокого навчання. Результатом роботи системи має бути прогнозований клас захворювання або підтвердження здорового стану рослини.

Відповідно до поставленої мети було сформовано функціональні вимоги до системи:

- завантаження користувачем зображення листка рослини для проведення аналізу;
- автоматична попередня обробка зображення перед передачею до моделі класифікації;
- визначення класу захворювання за допомогою моделі глибокого навчання;
- відображення результатів класифікації користувачу;
- формування рекомендацій щодо подальших дій на основі результатів діагностики;
- врахування введених користувачем параметрів температури та вологості під час формування рекомендацій;
- побудова теплової карти Grad-CAM для візуального пояснення результату класифікації;

- збереження результатів перевірок та можливість перегляду історії діагностики;
- надання користувачу доступу до раніше виконаних перевірок.

Окрім функціональних вимог, система повинна відповідати низці нефункціональних вимог.

Система має забезпечувати зручний та зрозумілий користувацький інтерфейс, доступний для користувачів без спеціальної підготовки у сфері машинного навчання або захисту рослин. Важливою вимогою є можливість роботи через веббраузер без необхідності встановлення додаткового програмного забезпечення.

Час обробки запиту повинен бути достатньо малим для забезпечення комфортної взаємодії користувача із системою. Також система повинна забезпечувати коректне зберігання даних виконаних перевірок.

З огляду на використання методів машинного навчання важливою вимогою є забезпечення інтерпретованості результатів роботи моделі. Для цього у системі передбачено використання методу Grad-CAM, який дозволяє візуально відображати ділянки зображення, що найбільше вплинули на прийняте рішення.

Таким чином, сформульовані вимоги визначають основні функціональні можливості та характеристики розроблюваної системи і є основою для подальшого проектування її архітектури та програмної реалізації.

2.2 Архітектура та логіка функціонування системи

Архітектура інтелектуальної системи моніторингу фітопатологій пасльонових культур була спроектована з урахуванням необхідності розділення функцій користувацького інтерфейсу, обробки даних, виконання алгоритмів машинного навчання та збереження результатів діагностики. Такий підхід забезпечує модульність програмного комплексу, спрощує його

супровід та створює можливість незалежного розвитку окремих компонентів системи.

У межах роботи система реалізована як розподілений програмний комплекс, що складається з двох незалежних серверів: вебзастосунку на платформі ASP.NET Core MVC та сервісу інтелектуальної діагностики на базі FastAPI і TensorFlow. Загальну архітектуру системи наведено на рисунку 2.1.

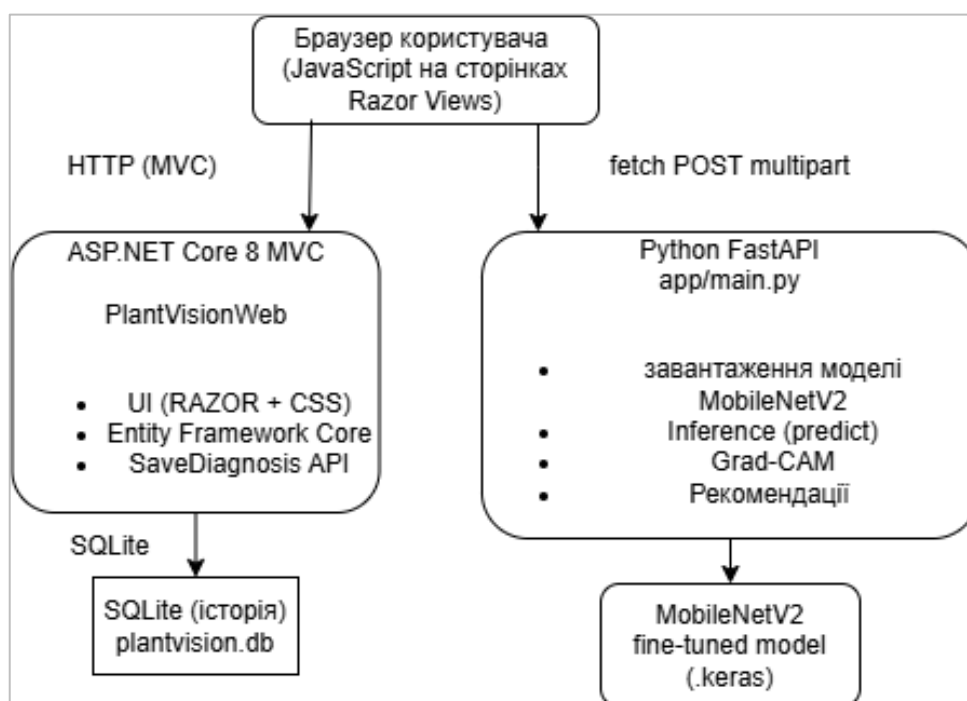


Рисунок 2.1 – Архітектура інтелектуальної системи моніторингу фітопатологій пасльонових культур

Ключовим архітектурним принципом системи є винесення всієї логіки машинного навчання в окремий Python-сервіс. Вебзастосунок не містить моделі класифікації та не виконує обчислень, пов'язаних із розпізнаванням захворювань. Уся логіка інтелектуального аналізу реалізована у REST API на базі FastAPI та TensorFlow. Такий підхід відповідає сучасному патерну побудови програмних систем типу «frontend/backend + ML microservice», за

якого користувацький інтерфейс і модулі машинного навчання функціонують незалежно один від одного.

Використання двох серверів обумовлене особливостями застосованих технологій. Навчання та використання нейронної мережі MobileNetV2 реалізовані за допомогою бібліотек TensorFlow та Keras, які забезпечують ефективні засоби створення й використання моделей глибокого навчання. Водночас платформа ASP.NET Core MVC надає зручні механізми побудови вебінтерфейсу, реалізації шаблону MVC, роботи з базою даних через Entity Framework Core та організації взаємодії з користувачем [12]. Розділення системи на два незалежні компоненти дозволяє оновлювати модель машинного навчання без внесення змін до вебінтерфейсу та незалежно розвивати користувацьку частину системи.

Основним засобом взаємодії користувача із системою є веббраузер. Через вебінтерфейс користувач обирає культуру для аналізу, завантажує фотографію листка рослини, за необхідності вводить дату спостереження, температуру та відносну вологість повітря, після чого запускає процес діагностики. Також браузер забезпечує відображення результатів аналізу, теплових карт Grad-CAM та історії попередніх перевірок.

Функції користувацького інтерфейсу реалізовано у вебзастосунку PlantVisionWeb на платформі ASP.NET Core MVC. У його структурі використовуються Razor Views для формування сторінок завантаження зображень, перегляду результатів діагностики та історії збережених перевірок. Серверна частина також забезпечує взаємодію з базою даних SQLite через Entity Framework Core та реалізує механізм збереження результатів аналізу.

Сервіс інтелектуальної діагностики реалізований на базі FastAPI та виконує прийом HTTP-запитів від клієнтської частини, попередню обробку зображень і взаємодію з моделлю машинного навчання [13]. Після отримання запиту сервіс перевіряє коректність файлу, виконує його

перетворення до необхідного формату та передає підготовлені дані на вхід моделі класифікації.

Центральним інтелектуальним компонентом системи є модель класифікації захворювань рослин, побудована на основі архітектури MobileNetV2. Для підвищення якості розпізнавання модель була навчена із використанням підходу transfer learning та додатково адаптована до предметної області методом fine-tuning. На вхід моделі подається зображення листка розміром 224×224 пікселів, а результатом роботи є визначення одного з класів захворювання або здорового стану рослини разом із рівнем впевненості класифікації.

Для підвищення інтерпретованості результатів до складу сервісу інтелектуальної діагностики входить модуль пояснення рішень на основі методу Grad-CAM. Після завершення класифікації формується теплова карта, яка відображає області зображення, що найбільше вплинули на прийняття рішення моделлю. Це дозволяє користувачу оцінити коректність роботи нейронної мережі та підвищує довіру до результатів аналізу.

Окремим компонентом системи є модуль формування рекомендацій. Його робота базується на результатах класифікації та додаткових параметрах середовища, введених користувачем. Значення температури та вологості не впливають на результат роботи нейронної мережі, але використовуються для уточнення рекомендацій щодо догляду за рослиною та профілактики подальшого розвитку захворювання.

Для збереження результатів діагностики використовується база даних SQLite. У базі зберігаються відомості про культуру, виявлене захворювання, стан рослини, рівень впевненості моделі, опис результату, сформовані рекомендації, дату спостереження, параметри середовища та завантажене зображення. Доступ до даних реалізовано через Entity Framework Core, що забезпечує зручну роботу з об'єктами предметної області [14].

Логіку функціонування системи наведено на рисунку 2.2.



Рисунок 2.2 – Схема процесу діагностики рослин у системі PlantVision

Процес роботи системи починається з вибору культури та завантаження користувачем фотографії листка рослини через вебінтерфейс. За потреби додатково вводяться параметри температури, вологості та дата спостереження. Після цього клієнтська частина виконує перевірку коректності введених даних та формує HTTP-запит до сервісу інтелектуальної діагностики.

У сервісі FastAPI здійснюється попередня обробка зображення: перевірка типу файлу, перетворення до кольорового формату RGB, зміна розміру до 224×224 пікселів та формування масиву даних для подальшої обробки нейронною мережею. Підготовлене зображення передається на вхід моделі MobileNetV2, яка визначає найбільш імовірний клас захворювання та рівень впевненості класифікації.

На основі отриманого результату модуль рекомендацій формує текстові поради щодо догляду за рослиною та профілактики розвитку захворювань. Одночасно виконується побудова теплової карти Grad-CAM, яка дозволяє візуалізувати області зображення, що найбільше вплинули на рішення моделі.

Після завершення аналізу сервіс FastAPI формує відповідь у форматі JSON, яка містить результати класифікації, опис захворювання, рівень впевненості моделі, рекомендації та теплову карту Grad-CAM. Отримані дані повертаються до браузера користувача та відображаються у вебінтерфейсі.

На клієнтській стороні додатково виконується перевірка відповідності культури, обраної користувачем, культурі, визначеній моделлю. У разі успішного проходження перевірки результат відображається користувачу та може бути збережений до бази даних. Під час збереження інформація передається до серверної частини ASP.NET Core MVC та записується до SQLite через Entity Framework Core. Надалі користувач має можливість переглядати історію виконаних перевірок та аналізувати отримані результати.

Таким чином, запропонована архітектура забезпечує ефективне розділення функцій між компонентами системи, а реалізована логіка функціонування дозволяє автоматизувати процес діагностики захворювань пасльонових культур, забезпечити інтерпретованість результатів роботи моделі та організувати зручне збереження інформації для подальшого використання.

2.3 Проєктування бази даних системи

Однією з важливих складових інтелектуальної системи моніторингу фітопатологій є підсистема збереження результатів діагностики. Наявність механізму накопичення та перегляду результатів дозволяє користувачу відстежувати стан рослин у часі, аналізувати історію виконаних перевірок та отримувати доступ до раніше сформованих діагностичних висновків. Для реалізації таких можливостей у системі передбачено використання локальної бази даних SQLite.

Вибір SQLite обумовлений її простотою використання, відсутністю необхідності встановлення окремого сервера баз даних та повною інтеграцією з платформою ASP.NET Core через технологію Entity Framework Core [15]. Такий підхід є доцільним для вебсистеми, орієнтованої на локальне використання та збереження відносно невеликих обсягів даних.

У розробленій системі база даних використовується для збереження результатів діагностики, отриманих після аналізу зображень листків рослин. Кожен запис містить інформацію про культуру, виявлене захворювання, рівень впевненості моделі, рекомендації щодо догляду та додаткові параметри середовища, введені користувачем.

У системі використовується одна основна таблиця `DiagnosisResults`, яка призначена для збереження результатів виконаних діагностичних перевірок. Кожний запис у таблиці відповідає окремому сеансу аналізу зображення рослини.

Первинним ключем таблиці є поле `Id` типу `int`, значення якого автоматично генерується під час створення нового запису. Це поле забезпечує унікальну ідентифікацію кожного результату діагностики.

Поле `Culture` містить назву культури, для якої було виконано аналіз. У поточній реалізації система підтримує дві культури — томат і картоплю. Поле `Disease` використовується для збереження назви визначеного захворювання або інформації про здоровий стан рослини.

Поле Status містить службову інформацію про стан рослини. Залежно від результату класифікації воно може набувати значень healthy або diseased. Використання окремого поля для стану рослини спрощує подальшу фільтрацію та аналіз результатів.

Поле ConfidencePercent призначене для збереження рівня впевненості моделі машинного навчання у відсотках. Значення цього поля формується на основі результатів роботи нейронної мережі та дозволяє користувачу оцінити надійність отриманого прогнозу.

Для надання додаткової інформації про результат класифікації використовуються поля Description та Recommendation. У полі Description зберігається текстовий опис виявленого стану рослини, а поле Recommendation містить рекомендації щодо подальших дій, профілактики або догляду. Рекомендації формуються експертним модулем системи на основі результатів класифікації та введених параметрів середовища.

Поле ObservationDate використовується для збереження дати спостереження, яку користувач може вказати під час проведення аналізу. Наявність цієї інформації дозволяє пов'язувати результати діагностики з конкретним періодом спостереження за рослиною.

Для врахування умов навколишнього середовища у структурі таблиці передбачено поля Temperature та Humidity. Вони зберігають значення температури та відносної вологості повітря, введені користувачем перед виконанням аналізу. Дані параметри використовуються підсистемою формування рекомендацій і можуть бути корисними під час подальшого аналізу історії спостережень.

Поле ImageData призначене для збереження завантаженого користувачем зображення листка рослини. У поточній реалізації зображення зберігається у вигляді рядка формату Base64. Такий підхід дозволяє уникнути необхідності створення окремого файлового сховища та забезпечує цілісність усіх даних діагностики в межах однієї бази даних.

Для автоматичної фіксації часу створення запису використовується поле CreatedAt типу DateTime. Значення цього поля встановлюється під час збереження результату та дозволяє впорядковувати історію перевірок за часом їх виконання.

Структуру основної таблиці бази даних наведено на рисунку 2.3.

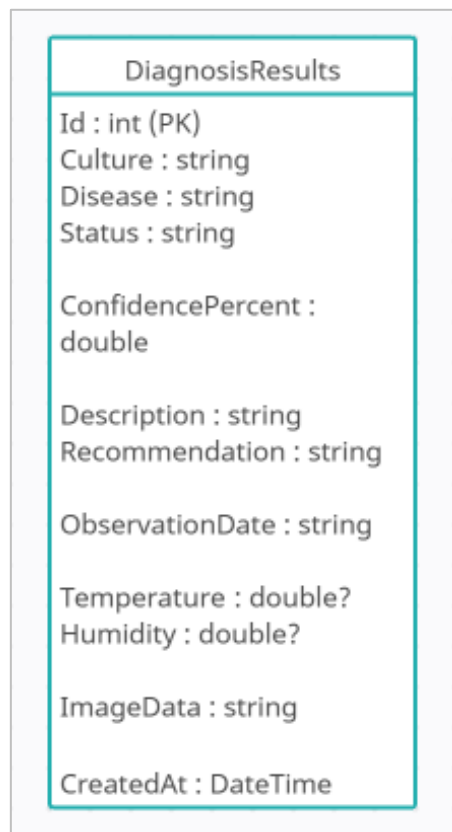


Рисунок 2.3 – Структура таблиці DiagnosisResults бази даних PlantVision

Взаємодія вебзастосунку з базою даних реалізована за допомогою технології Entity Framework Core. Для цього у проєкті створено контекст даних ApplicationDbContext, який забезпечує об'єктно-реляційне відображення між моделлю DiagnosisResult та таблицею DiagnosisResults у базі даних SQLite. Такий підхід дозволяє виконувати операції створення, збереження та отримання записів без необхідності безпосереднього використання SQL-запитів.

Під час збереження результату аналізу клієнтська частина формує структурований набір даних, який передається до методу SaveDiagnosis серверної частини ASP.NET Core MVC. Після отримання даних створюється новий об'єкт моделі DiagnosisResult, який зберігається у базі даних через механізми Entity Framework Core. Для перегляду історії перевірок система виконує отримання записів із таблиці DiagnosisResults та відображає їх у користувацькому інтерфейсі у порядку спадання дати створення.

Таким чином, спроектована база даних забезпечує надійне збереження результатів діагностики, підтримує механізм ведення історії перевірок та створює основу для подальшого розширення функціональних можливостей системи моніторингу фітопатологій пасльонових культур.

2.4 Проєктування модуля інтелектуальної діагностики

Модуль інтелектуальної діагностики є центральним функціональним компонентом розробленої системи та призначений для автоматизованого визначення стану рослин за цифровими зображеннями листків. Основною задачею модуля є класифікація зображення та визначення належності рослини до одного з наперед визначених класів захворювань або до класу здорового стану.

Під час проєктування модуля було враховано необхідність забезпечення достатньо високої точності класифікації, швидкої обробки зображень та можливості інтеграції моделі машинного навчання до веборієнтованої системи. З урахуванням цих вимог для реалізації інтелектуального аналізу було обрано архітектуру MobileNetV2, яка поєднує порівняно невеликі обчислювальні витрати та високу ефективність розпізнавання зображень.

Загальну структуру модуля інтелектуальної діагностики наведено на рисунку 2.4.

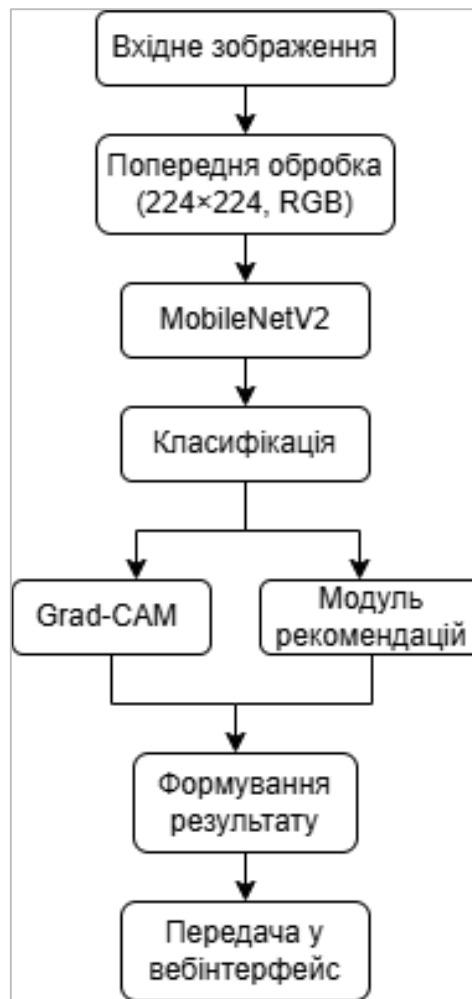


Рисунок 2.4 – Структура модуля інтелектуальної діагностики

Модуль інтелектуальної діагностики складається з декількох взаємопов'язаних компонентів: підсистеми попередньої обробки зображень, моделі класифікації на основі згорткової нейронної мережі, модуля формування рекомендацій та підсистеми візуального пояснення результатів класифікації.

Робота модуля починається з отримання цифрового зображення листка рослини від користувача. Перед передачею до нейронної мережі виконується попередня обробка даних. На цьому етапі здійснюється перевірка коректності зображення, його перетворення до кольорового простору RGB та зміна розміру до 224×224 пікселів. Після цього формується

масив даних, придатний для подальшої обробки моделлю машинного навчання.

Основою модуля є згортова нейронна мережа MobileNetV2. Вибір даної архітектури обумовлений її високою ефективністю під час роботи із зображеннями та відносно невеликою кількістю параметрів порівняно з багатьма іншими сучасними моделями. Це дозволяє забезпечити швидке виконання класифікації та зменшити вимоги до обчислювальних ресурсів сервера.

Для побудови моделі використано підхід transfer learning. Його застосування дозволяє використовувати знання, отримані під час попереднього навчання моделі на великому наборі зображень ImageNet, та адаптувати їх до задачі класифікації захворювань рослин. Такий підхід є особливо ефективним у випадках, коли обсяг спеціалізованих навчальних даних є обмеженим порівняно із загальними наборами зображень.

На першому етапі створення моделі використовується попередньо навчена архітектура MobileNetV2 без фінального класифікаційного шару. Вона виконує роль екстрактора ознак та забезпечує автоматичне виділення характерних візуальних особливостей зображень листків рослин. До архітектури додано власний класифікаційний блок, який виконує визначення належності зображення до одного з класів захворювань.

Для підвищення здатності моделі адаптуватися до особливостей предметної області застосовується додаткове донавчання окремих шарів нейронної мережі методом fine-tuning. Такий підхід дозволяє уточнити внутрішні представлення моделі відповідно до специфіки зображень рослин та покращити якість класифікації без необхідності повного перенавчання мережі.

Результатом роботи моделі є набір імовірностей для всіх підтримуваних класів. Для формування діагностичного висновку обирається клас із найбільшим значенням імовірності. Одночасно

розраховується рівень впевненості класифікації, який використовується під час формування результату для користувача.

Окрім визначення класу захворювання, модуль інтелектуальної діагностики забезпечує формування пояснення отриманого результату. Для цього використовується метод Grad-CAM, який належить до підходів Explainable Artificial Intelligence. Принцип його роботи полягає у використанні внутрішніх активацій згорткової нейронної мережі для визначення ділянок зображення, що найбільше вплинули на результат класифікації.

У результаті роботи Grad-CAM формується теплова карта, яка накладається на початкове зображення листка. Ділянки з більшою інтенсивністю кольору відповідають областям, які мали найбільший вплив на рішення моделі. Використання такого підходу дозволяє підвищити прозорість роботи системи та надати користувачу додаткові засоби оцінювання достовірності отриманого результату.

Важливою складовою модуля є підсистема формування рекомендацій. На відміну від нейронної мережі, яка виконує лише класифікацію зображень, даний компонент використовує набір експертних правил для генерації текстових порад щодо подальших дій користувача. Рекомендації формуються на основі визначеного класу захворювання та можуть додатково враховувати значення температури і вологості, введені користувачем.

У разі підвищеної вологості система доповнює рекомендації застереженнями щодо ризику розвитку грибкових захворювань, а за високих температур надає поради щодо режиму поливу та умов вирощування. Таким чином реалізується поєднання методів машинного навчання та експертного підходу до підтримки прийняття рішень.

Після завершення класифікації, побудови теплової карти та формування рекомендацій усі отримані результати об'єднуються у єдину структуру даних. До складу результату входять назва культури, визначене

захворювання, статус рослини, рівень впевненості класифікації, опис результату, рекомендації та візуалізація Grad-CAM. Сформована інформація передається до вебінтерфейсу та відображається користувачу у зручному для сприйняття вигляді.

Таким чином, спроектований модуль інтелектуальної діагностики поєднує сучасні методи глибокого навчання, механізми пояснення результатів роботи нейронних мереж та експертні правила формування рекомендацій. Це забезпечує не лише автоматизоване визначення захворювань рослин, а й підвищує практичну цінність системи для кінцевого користувача та створює основу для подальшого розвитку функціональних можливостей вебсистеми.

3 РЕАЛІЗАЦІЯ МОДУЛЯ ІНТЕЛЕКТУАЛЬНОЇ ДІАГНОСТИКИ ТА ВЕБСИСТЕМИ

3.1 Підготовка та аналіз набору даних

Якість роботи систем комп'ютерного зору значною мірою залежить від характеристик набору даних, який використовується для навчання моделі машинного навчання. Для задач автоматизованої діагностики захворювань рослин особливо важливим є використання достатньо великої кількості розмічених зображень, що охоплюють різні культури, типи захворювань та варіанти прояву симптомів.

Для реалізації інтелектуальної системи моніторингу фітопатологій пасльонових культур було використано відкритий набір даних Plant Village Dataset Plant Disease Classification Dataset Split into Train Test and Validation Sets, розміщений на платформі Kaggle. Даний набір сформовано на основі відомого датасету PlantVillage, який широко використовується у дослідженнях, пов'язаних із класифікацією захворювань рослин за цифровими зображеннями листків.

Однією з переваг обраного набору даних є наявність попередньо сформованих навчальної, валідаційної та тестової вибірок. Такий підхід дозволяє виконувати навчання, налаштування та оцінювання моделей машинного навчання без необхідності додаткового розділення даних. Крім того, використання окремої тестової вибірки забезпечує більш об'єктивну оцінку здатності моделі до узагальнення на нових даних.

У межах роботи використовувалися лише класи, що відповідають пасльоновим культурам, а саме томату та картоплі. Такий вибір обумовлений тематикою дослідження та спрямованістю системи на моніторинг фітопатологій саме цих культур.

Перелік класів, використаних для навчання моделі, наведено в таблиці 3.1.

Таблиця 3.1 – Класи захворювань та станів рослин, використані для навчання моделі

Клас	Опис класу	Train	Validation	Test
Potato___Early_blight	Альтернarioз картоплі	699	101	200
Potato___Late_blight	Фітофтороз картоплі	699	101	200
Potato___healthy	Здорова картопля	106	16	30
Tomato_Bacterial_spot	Бактеріальна плямистість томата	1488	213	426
Tomato_Early_blight	Альтернarioз томата	699	101	200
Tomato_Late_blight	Фітофтороз томата	1336	191	382
Tomato_Leaf_Mold	Листкова пліснява томата	666	96	190
Tomato_Septoria_leaf_spot	Септоріоз листя томата	1239	178	354
Tomato_Spider_mites_Two_spotted_spider_mite	Павутинний кліщ	1173	168	335
Tomato___Target_Spot	Цільова плямистість томата	982	141	281
Tomato___Tomato_mosaic_virus	Вірус тютюнової мозаїки	261	38	74
Tomato___Tomato_YellowLeaf___Curl_Virus	Вірус жовтого скручування листя томата	2246	321	642
Tomato_healthy	Здоровий томат	1113	160	318

Загалом модель підтримує тринадцять класів, серед яких присутні як здорові рослини, так і різні грибкові, бактеріальні та вірусні захворювання. Такий набір класів дозволяє охопити найбільш поширені проблеми, що виникають під час вирощування томатів і картоплі.

Фізична організація набору даних відповідає стандартному формату, який використовується засобами TensorFlow для задач класифікації

зображень. Кожен клас представлений окремою директорією, всередині якої розміщено відповідні зображення листків рослин.

Для навчання нейронної мережі використовувалася навчальна вибірка (train), яка містить основний обсяг зображень. Валідаційна вибірка (validation) застосовувалася для контролю процесу навчання та оцінювання якості моделі після кожної епохи. Тестова вибірка (test) використовувалася виключно для фінальної оцінки моделі після завершення навчання.

Перед початком навчання було виконано перевірку структури набору даних за допомогою допоміжного скрипта `check_dataset.py`. Скрипт проходить по навчальній, валідаційній та тестовій вибірках, перевіряє наявність директорій для вибраних класів і підраховує кількість графічних файлів із розширеннями `.jpg`, `.jpeg`, `.png` та `.bmp`. Для кожного класу виводиться кількість знайдених зображень і статус OK або MISSING, що дозволяє переконатися у коректності структури даних перед навчанням моделі.

Оскільки нейронна мережа MobileNetV2 працює з вхідними зображеннями фіксованого розміру, усі зображення під час завантаження автоматично приводилися до розміру 224×224 пікселів. Також здійснювалося перетворення до кольорового простору RGB, що забезпечує узгодженість вхідних даних незалежно від особливостей вихідних файлів.

На етапі підготовки даних було передбачено використання аугментації під час навчання моделі.

Використання аугментації є особливо важливим для задач аналізу зображень рослин, оскільки в реальних умовах фотографії можуть відрізнятися ракурсом зйомки, освітленням, положенням листка та якістю камери. Завдяки цьому модель навчається бути більш стійкою до подібних змін і демонструє кращі результати під час роботи з новими зображеннями.

Для завантаження набору даних у середовище TensorFlow використовувалися стандартні механізми `image_dataset_from_directory`, які

забезпечують автоматичне створення пакетів даних та присвоєння міток класів відповідно до структури директорій [16]. Такий підхід спрощує підготовку даних та забезпечує коректну інтеграцію набору даних із процесом навчання нейронної мережі.

Проведений аналіз набору даних показав, що він містить достатню кількість зображень для реалізації задачі багатокласової класифікації захворювань томатів і картоплі та забезпечує необхідну основу для навчання моделі MobileNetV2. Наявність попередньо сформованих навчальної, валідаційної та тестової вибірок, а також використання аугментації даних створюють передумови для побудови моделі, здатної ефективно розпізнавати захворювання рослин за цифровими зображеннями листків.

3.2 Реалізація та навчання моделі класифікації

Для автоматизованої діагностики захворювань пасльонових культур у роботі було реалізовано модель класифікації зображень на основі згорткової нейронної мережі MobileNetV2. Вибір даної архітектури обумовлений поєднанням високої точності розпізнавання з відносно невеликими обчислювальними витратами, що є важливим для подальшого використання моделі у веборієнтованій системі [17].

Реалізацію моделі виконано засобами бібліотек TensorFlow та Keras мовою програмування Python. Навчання проводилося із використанням підходу transfer learning, який передбачає використання попередньо навченої моделі як базового екстрактора ознак із подальшою адаптацією до нової предметної галузі.

Для завантаження навчальних даних використовувався механізм `image_dataset_from_directory`, що автоматично формує набори даних на основі структури каталогів. Навчальна, валідаційна та тестова вибірки завантажувалися окремо із відповідних директорій набору даних. Під час

завантаження всі зображення автоматично масштабувалися до розміру 224×224 пікселів, який відповідає вхідному формату архітектури MobileNetV2.

Після формування вибірок виконувалася оптимізація процесу читання даних за допомогою механізмів кешування та попереднього завантаження (prefetch). Це дозволяє зменшити час очікування під час навчання моделі та ефективніше використовувати обчислювальні ресурси.

Під час побудови моделі було використано попередньо навчену мережу MobileNetV2 із вагами набору даних ImageNet. Верхні класифікаційні шари базової моделі були видалені шляхом встановлення параметра `include_top=False`, що дозволило використовувати мережу лише для виділення ознак із зображень. На початковому етапі навчання всі шари базової моделі залишалися замороженими, а навчання підлягали лише нові класифікаційні шари.

Перед подачею зображень до базової мережі застосовувалися операції аугментації даних. До них належали горизонтальне віддзеркалення, випадковий поворот, масштабування та зміна контрастності. Використання аугментації дозволяє підвищити різноманітність навчальних даних і зменшити ризик перенавчання моделі.

Після шару MobileNetV2 використовувався шар глобального усереднення ознак `GlobalAveragePooling2D`, який перетворює багатовимірні карти ознак на компактне векторне представлення. Для додаткової боротьби з перенавчанням було застосовано шар `Dropout` із коефіцієнтом 0,25. Завершальний шар `Dense` із функцією активації `Softmax` виконує багатокласову класифікацію між тринадцятьма класами захворювань та станів рослин.

Фрагмент програмної реалізації побудови моделі класифікації наведено на рисунку 3.1.

Наведений фрагмент демонструє побудову моделі класифікації на основі архітектури MobileNetV2, використання аугментації даних,

заморожування попередньо навчених шарів та формування власного класифікаційного блоку для розпізнавання захворювань рослин.

```
def build_model(num_classes: int):
    data_augmentation = tf.keras.Sequential(
        [
            layers.RandomFlip("horizontal"),
            layers.RandomRotation(0.1),
            layers.RandomZoom(0.1),
            layers.RandomContrast(0.1),
        ],
        name="data_augmentation",
    )

    base_model = tf.keras.applications.MobileNetV2(
        input_shape=IMG_SIZE + (3,),
        include_top=False,
        weights="imagenet",
    )

    base_model.trainable = False

    inputs = tf.keras.Input(shape=IMG_SIZE + (3,))
    x = data_augmentation(inputs)
    x = tf.keras.applications.mobilenet_v2.preprocess_input(x)
    x = base_model(x, training=False)
    x = layers.GlobalAveragePooling2D()(x)
    x = layers.Dropout(0.25)(x)
    outputs = layers.Dense(num_classes, activation="softmax")(x)

    model = tf.keras.Model(inputs, outputs)

    model.compile(
        optimizer=tf.keras.optimizers.Adam(learning_rate=0.0005),
        loss="sparse_categorical_crossentropy",
        metrics=["accuracy"],
    )

    return model
```

Рисунок 3.1 – Фрагмент реалізації моделі класифікації на основі MobileNetV2

Для навчання моделі використовувався оптимізатор Adam із початковою швидкістю навчання 0,0005 [18]. Для оцінювання помилки класифікації використовувалася функція втрат Sparse Categorical Crossentropy, яка застосовується у задачах багатокласової класифікації, коли класи представлені числовими мітками [19]. Функція визначає ступінь відмінності між фактичним класом та ймовірностями, прогнозованими

моделлю. Мінімізація значення функції втрат під час навчання дозволяє підвищувати точність класифікації та покращувати узагальнювальну здатність моделі. Як основна метрика оцінювання використовувалася точність класифікації (accuracy).

Основні параметри навчання наведено в таблиці 3.2.

Таблиця 3.2 – Параметри навчання моделі

Параметр	Значення
Архітектура	MobileNetV2
Розмір зображення	224×224
Кількість класів	13
Розмір пакета (Batch Size)	32
Кількість епох	10
Оптимізатор	Adam
Швидкість навчання	0,0005
Функція втрат	Sparse Categorical Crossentropy
Метрика оцінювання	Accuracy
Seed	42

Для контролю процесу навчання використовувалися механізми Early Stopping та Model Checkpoint. Early Stopping автоматично припиняє навчання у випадку відсутності покращення значення функції втрат на валідаційній вибірці протягом трьох послідовних епох. Model Checkpoint забезпечує автоматичне збереження найкращої версії моделі за критерієм мінімального значення втрат на валідаційних даних.

Після завершення навчання модель була оцінена на тестовій вибірці, яка не використовувалася під час процесу навчання. Отримані значення точності класифікації та функції втрат були збережені для подальшого аналізу. Крім того, автоматично формувалися графіки зміни точності та функції втрат на навчальній і валідаційній вибірках, а також файл із

підсумковими метриками моделі. Повний код навчання моделі наведено у додатку А.

Таким чином, у результаті реалізації було створено базову модель класифікації захворювань рослин, яка може бути використана як основа для подальшого донавчання і підвищення якості розпізнавання.

3.3 Доновчання моделі та реалізація Grad-CAM

Після отримання базової моделі класифікації було виконано її подальшу адаптацію до предметної області шляхом донавчання (fine-tuning). Використання цього підходу дозволяє підвищити якість класифікації за рахунок часткового переналаштування параметрів попередньо навченої нейронної мережі під особливості конкретного набору даних.

На першому етапі навчання архітектура MobileNetV2 використовувалася як фіксований екстрактор ознак. Усі шари базової моделі були заморожені, а навчанням підлягав лише класифікаційний блок, доданий поверх попередньо навченої мережі. Такий підхід дозволив швидко адаптувати модель до задачі класифікації захворювань рослин без зміни параметрів базової архітектури.

Після завершення початкового навчання було виконано донавчання моделі. Для цього збережена базова модель завантажувалася повторно, після чого частина шарів MobileNetV2 переводилася у режим навчання. У реалізованій системі було розморожено останні 35 шарів базової мережі, тоді як інші шари залишалися зафіксованими. Такий підхід дозволяє зберегти загальні візуальні ознаки, отримані під час навчання на наборі даних ImageNet, і водночас адаптувати високорівневі представлення до специфіки захворювань рослин.

Фрагмент програмної реалізації процедури донавчання моделі MobileNetV2 наведено на рисунку 3.2.

```

model = tf.keras.models.load_model(BASE_MODEL_PATH)

base_model = find_mobilenet_layer(model)

print("Base model layer:", base_model.name)
print("Total layers in base model:", len(base_model.layers))

base_model.trainable = True

for layer in base_model.layers[:-UNFREEZE_LAST_LAYERS]:
    layer.trainable = False

for layer in base_model.layers[-UNFREEZE_LAST_LAYERS:]:
    layer.trainable = True

print(f"Unfrozen last layers: {UNFREEZE_LAST_LAYERS}")

model.compile(
    optimizer=tf.keras.optimizers.Adam(learning_rate=0.00001),
    loss="sparse_categorical_crossentropy",
    metrics=["accuracy"],
)

```

Рисунок 3.2 – Фрагмент реалізації процедури fine-tuning моделі MobileNetV2

У наведеному фрагменті базова модель завантажується з файлу, після чого визначається вкладений шар MobileNetV2. Для донавчання розморожуються лише останні шари базової мережі, а решта шарів залишаються зафіксованими. Після цього модель компілюється з меншою швидкістю навчання, що дозволяє обережно адаптувати ваги моделі до специфіки набору даних.

Після розморожування частини шарів модель була повторно скомпільована. Для донавчання використовувався оптимізатор Adam зі зменшеною швидкістю навчання 0,00001. Зменшення швидкості навчання є важливим етапом fine-tuning, оскільки дозволяє коригувати вагові коефіцієнти без суттєвого руйнування знань, отриманих під час попереднього навчання.

Для контролю процесу донавчання використовувалися ті самі механізми Early Stopping та Model Checkpoint, що й на етапі базового навчання. Найкраща версія моделі автоматично зберігалася на основі значення функції втрат на валідаційній вибірці. Після завершення процесу

донавчання модель оцінювалася на тестових даних, а результати зберігалися для подальшого аналізу ефективності. Програмну реалізацію процедури донавчання наведено в додатку Б.

Окрім підвищення точності класифікації, важливим завданням роботи було забезпечення інтерпретованості результатів роботи нейронної мережі. Для цього до складу системи було інтегровано метод Gradient-weighted Class Activation Mapping (Grad-CAM).

Grad-CAM належить до методів Explainable Artificial Intelligence та дозволяє візуалізувати області зображення, які найбільше вплинули на прийняте моделлю рішення. На відміну від звичайного результату класифікації, що містить лише назву класу та рівень впевненості, Grad-CAM надає додаткову інформацію про причини такого рішення [20].

Принцип роботи методу полягає у використанні градієнтів цільового класу відносно карт ознак останнього згорткового шару нейронної мережі. На основі отриманих градієнтів обчислюються вагові коефіцієнти важливості ознак, після чого формується карта активації, що відображає внесок різних ділянок зображення у результат класифікації.

У реалізованій системі генерація теплової карти виконується після завершення процесу класифікації. Для обраного класу захворювання визначаються активації останнього згорткового шару MobileNetV2 та відповідні градієнти. Далі формується двовимірний градієнт, який масштабується до розміру початкового зображення та накладається на фотографію листка рослини.

Зони з найбільшим впливом на результат класифікації відображаються теплими кольорами, тоді як менш значущі області позначаються холодними кольорами. Це дозволяє користувачу візуально оцінити, чи дійсно модель використовувала симптоми захворювання для прийняття рішення.

На рисунку 3.3 наведено фрагмент програмної реалізації формування теплової карти Grad-CAM.

```

last_conv_layer = base_model.get_layer("Conv_1")

inputs = tf.keras.Input(shape=IMG_SIZE + (3,))

x = data_augmentation(inputs, training=False)
x = tf.keras.applications.mobilenet_v2.preprocess_input(x)

feature_extractor = tf.keras.Model(
    inputs=base_model.input,
    outputs=[
        last_conv_layer.output,
        base_model.output,
    ],
)

conv_outputs, x = feature_extractor(x)

x = pooling_layer(x)
x = dropout_layer(x, training=False)
predictions = dense_layer(x)

grad_model = tf.keras.Model(
    inputs=inputs,
    outputs=[conv_outputs, predictions],
)

with tf.GradientTape() as tape:
    conv_outputs, predictions = grad_model(image_array)

    if pred_index is None:
        pred_index = tf.argmax(predictions[0])

    class_channel = predictions[:, pred_index]

gradients = tape.gradient(class_channel, conv_outputs)

pooled_gradients = tf.reduce_mean(gradients, axis=(0, 1, 2))
conv_outputs = conv_outputs[0]

heatmap = conv_outputs @ pooled_gradients[..., tf.newaxis]
heatmap = tf.squeeze(heatmap)
heatmap = tf.maximum(heatmap, 0)

```

Рисунок 3.3 – Фрагмент реалізації алгоритму Grad-CAM

У наведеному фрагменті використовується останній згортковий шар MobileNetV2 Conv_1, на основі якого формується карта ознак. За допомогою GradientTape обчислюються градієнти прогнозованого класу відносно карт ознак, після чого виконується усереднення градієнтів і побудова теплової карти.

Після побудови теплової карти отримане зображення кодується у формат Base64 та включається до JSON-відповіді сервісу FastAPI. Надалі теплова карта відображається у вебінтерфейсі разом із результатами діагностики, прогнозованим класом та сформованими рекомендаціями. Код реалізації механізму Grad-CAM наведено в додатку В.

Таким чином, у межах роботи було реалізовано механізм донавчання моделі MobileNetV2 та інтегровано метод Grad-CAM для пояснення результатів класифікації. Використання fine-tuning дозволяє адаптувати модель до особливостей предметної галузі, а застосування Explainable AI підвищує прозорість роботи системи та довіру користувачів до отриманих результатів.

3.4 Реалізація сервісу інтелектуальної діагностики на FastAPI

Сервіс інтелектуальної діагностики реалізовано як окремий Python-застосунок на базі FastAPI. Його основне призначення полягає в обробці зображення, запуску навченої моделі класифікації, формуванні рекомендацій та побудові Grad-CAM-візуалізації. Винесення цього функціоналу в окремий сервіс дозволяє ізолювати логіку машинного навчання від вебінтерфейсу ASP.NET Core MVC. Повний код FastAPI-сервісу інтелектуальної діагностики наведено у додатку Г.

Під час запуску сервісу виконується завантаження списку класів із файлу `class_names.json` та навченої моделі `plant_disease_mobilenetv2_finetuned.keras`. Такий підхід дозволяє не завантажувати модель повторно під час кожного запиту, що зменшує час обробки зображення.

У сервісі реалізовано два основні маршрути. Маршрут `/health` використовується для перевірки працездатності сервісу та повертає інформацію про стан API, факт завантаження моделі та кількість підтримуваних класів. Основним маршрутом є `/predict`, який приймає зображення листка рослини та необов'язкові параметри температури й вологості.

Перед виконанням класифікації сервіс перевіряє тип завантаженого файлу. Якщо файл не є зображенням, користувачу повертається повідомлення про помилку. Після успішної перевірки зображення

відкривається за допомогою бібліотеки PIL, перетворюється до формату RGB та масштабується до розміру 224×224 пікселів. Далі воно перетворюється у масив NumPy та передається на вхід моделі.

Фрагмент реалізації маршруту прогнозування який забезпечує приймання зображення, запуск моделі класифікації, формування рекомендацій та побудову Grad-CAM, наведено на рисунку 3.4.

```
@app.post("/predict")
async def predict(
    file: UploadFile = File(...),
    temperature: float | None = Form(None),
    humidity: float | None = Form(None),
):
    if not file.content_type or not file.content_type.startswith("image/"):
        raise HTTPException(
            status_code=400,
            detail="Файл має бути зображенням.",
        )

    image_array, original_image = preprocess_image(file.file)
    predictions = model.predict(image_array)

    predicted_index = int(np.argmax(predictions[0]))
    confidence = float(predictions[0][predicted_index])
    class_name = class_names[predicted_index]

    recommendation = get_recommendation(
        class_name,
        temperature=temperature,
        humidity=humidity,
    )
    heatmap = make_gradcam_heatmap(
        image_array,
        model,
        pred_index=predicted_index,
    )

    gradcam_image = create_gradcam_base64(
        original_image,
        heatmap,
    )
    return {
        "class_name": class_name,
        "culture": recommendation["culture"],
        "disease": recommendation["disease"],
        "status": recommendation["status"],
        "description": recommendation["description"],
        "confidence": confidence,
        "confidence_percent": round(confidence * 100, 2),
        "recommendation": recommendation["recommendation"],
        "gradcam_image": gradcam_image,
    }
```

Рисунок 3.4 – Фрагмент реалізації маршруту /predict у сервісі FastAPI

Після отримання прогнозу модель повертає масив імовірностей для всіх підтримуваних класів. Для визначення результату використовується індекс класу з найбільшою ймовірністю. На його основі визначається технічна назва класу, рівень впевненості моделі та відповідна інформація про культуру, захворювання, опис і статус рослини.

Окремо в сервісі викликається модуль формування рекомендацій. Він отримує назву визначеного класу, а також значення температури та вологості, якщо користувач їх ввів. На основі цих даних формується текстова рекомендація. Важливо, що параметри середовища не змінюють результат класифікації, а використовуються лише для доповнення рекомендацій експертними поясненнями. Код модуля рекомендацій наведено в додатку Д.

Після класифікації виконується побудова теплової карти Grad-CAM. Для цього сервіс викликає функцію формування карти активацій, після чого отримане зображення накладається на початкове фото та кодується у формат Base64. Цей формат дозволяє передати результат як JSON-відповідь без створення окремого файлу на сервері.

Після завершення всіх етапів обробки сервіс формує структуровану JSON-відповідь. До її складу входять технічна назва класу, культура, назва захворювання або здоровий стан, статус рослини, опис результату, рівень впевненості моделі у числовому та відсотковому форматах, рекомендація та Grad-CAM-зображення.

Таким чином, сервіс FastAPI виконує роль проміжного інтелектуального шару між вебінтерфейсом користувача та моделлю машинного навчання. Він забезпечує повний цикл обробки запиту: приймання зображення, попередню обробку, класифікацію, формування рекомендацій, побудову пояснення рішення моделі та повернення результату до клієнтської частини системи.

3.5 Реалізація вебзастосунку ASP.NET Core MVC

Вебчастина системи реалізована у вигляді застосунку ASP.NET Core MVC, що забезпечує введення даних користувачем, відображення результатів аналізу та збереження історії перевірок. Основні програмні компоненти наведено в додатку Е. У межах вебзастосунку реалізовано три основні сторінки: головну сторінку діагностики, сторінку детального результату та сторінку історії перевірок.

Користувацький інтерфейс побудовано за допомогою Razor Views, HTML, CSS та JavaScript. Код сторінок користувацького інтерфейсу наведено в додатку Ж. На головній сторінці користувач обирає культуру, завантажує зображення листка, вводить додаткові параметри та запускає аналіз. JavaScript виконує перевірку введених даних, формує об'єкт FormData і надсилає HTTP-запит до FastAPI-сервісу. Фрагмент реалізації цієї взаємодії наведено на рисунку 3.5.

```
const formData = new FormData();
formData.append("file", selectedFile);

if (temperature !== "") {
    formData.append("temperature", temperature);
}

if (humidity !== "") {
    formData.append("humidity", humidity);
}
try {
    const response = await fetch("http://127.0.0.1:8000/predict", {
        method: "POST",
        body: formData
    });

    if (!response.ok) {
        throw new Error("Не вдалося отримати результат аналізу.");
    }

    const result = await response.json();
```

Рисунок 3.5 – Фрагмент взаємодії вебінтерфейсу з FastAPI-сервісом

Після натискання кнопки аналізу формується об'єкт `FormData`, до якого додається файл зображення та, за наявності, значення температури і вологості. Далі за допомогою функції `fetch` виконується POST-запит до маршруту `/predict` FastAPI-сервісу. Отримана JSON-відповідь містить результат класифікації, рівень впевненості моделі, опис захворювання, рекомендації та Grad-CAM-зображення.

Важливою частиною реалізації є перевірка відповідності культури, обраної користувачем, культурі, визначеній моделлю. Якщо користувач обрав томат, а модель визначила картоплю, або навпаки, система виводить попередження про невідповідність. Це дозволяє зменшити кількість помилкових сценаріїв використання та звернути увагу користувача на можливу помилку під час вибору культури.

Після успішного отримання результату дані тимчасово зберігаються у сховищі браузера `localStorage`. Такий підхід використовується для передавання результату між головною сторінкою та сторінкою детального перегляду без негайного запису до бази даних. До `localStorage` заносяться завантажене зображення у форматі Base64, Grad-CAM-зображення, назва культури, дата спостереження, параметри температури і вологості, назва класу, діагноз, статус рослини, опис, рівень впевненості та рекомендації.

Сторінка детального результату призначена для повного відображення діагностичного висновку. На ній користувачу показується початкове зображення, теплова карта Grad-CAM, назва культури, введені параметри, стан рослини, назва захворювання, рівень впевненості моделі, опис результату та рекомендації. Якщо дані в `localStorage` відсутні, користувач автоматично перенаправляється на головну сторінку.

На сторінці результату також реалізовано можливість збереження діагностичної перевірки. Після натискання кнопки збереження браузер надсилає структуровані дані у форматі JSON до методу `SaveDiagnosis` контролера `HomeController`. Отримані дані записуються до бази даних SQLite через контекст `Entity Framework Core`.

Фрагмент реалізації збереження результату діагностики наведено на рисунку 3.6.

```
[HttpPost]
Ссылка: 0
public async Task<IActionResult> SaveDiagnosis([FromBody] DiagnosisResult result)
{
    if (result == null)
    {
        return BadRequest();
    }

    result.CreatedAt = DateTime.Now;

    _context.DiagnosisResults.Add(result);
    await _context.SaveChangesAsync();

    return Json(new { success = true, id = result.Id });
}
```

Рисунок 3.6 – Фрагмент реалізації методу SaveDiagnosis у контролері HomeController

У серверній частині вебзастосунку основну роль виконує контролер HomeController. Він містить методи для відображення головної сторінки, сторінки детального результату, сторінки історії та метод збереження результатів діагностики. Метод History отримує з бази даних усі збережені записи, упорядковує їх за датою створення у зворотному порядку та передає до відповідного представлення.

Для зберігання результатів використовується модель DiagnosisResult, яка описує структуру одного діагностичного запису. Вона містить поля для культури, діагнозу, статусу рослини, рівня впевненості, опису, рекомендацій, дати спостереження, температури, вологості, зображення та дати створення запису. Таким чином, кожна перевірка може бути збережена як окремий об'єкт у базі даних.

Сторінка історії перевірок реалізована як Razor View, що отримує список об'єктів `DiagnosisResult` від контролера. Якщо збережених результатів немає, користувачу відображається повідомлення про порожню історію. Якщо записи наявні, вони виводяться у вигляді карток із зображенням, назвою захворювання, культурою, станом рослини, рівнем впевненості та датою спостереження.

Таким чином, вебзастосунок ASP.NET Core MVC забезпечує користувацьку частину системи: взаємодію з FastAPI-сервісом, введення даних, відображення результатів діагностики, тимчасове збереження результату між сторінками та запис перевірок до бази даних. Реалізована структура дозволяє відокремити інтерфейс користувача від логіки машинного навчання та забезпечує зручну роботу з результатами моніторингу фітопатологій пасльонових культур.

4 ТЕСТУВАННЯ ТА ОЦІНКА РЕЗУЛЬТАТІВ РОБОТИ СИСТЕМИ

4.1 Результати навчання базової моделі

Після підготовки набору даних та реалізації архітектури MobileNetV2 було проведено навчання базової моделі класифікації захворювань пасльонових культур. Навчання виконувалося із використанням підходу transfer learning, за якого попередньо навчена на наборі даних ImageNet модель використовувалася як екстрактор ознак, а навчанню підлягали лише додані класифікаційні шари.

Під час навчання здійснювався моніторинг точності класифікації та значення функції втрат як на навчальній, так і на валідаційній вибірках. Отримані графіки дозволяють оцінити динаміку навчання моделі та її здатність до узагальнення даних.

На рисунку 4.1 наведено графік зміни точності моделі протягом десяти епох навчання.

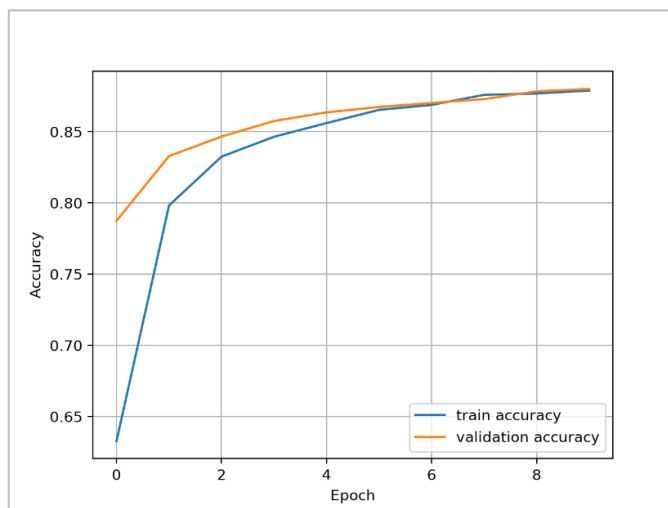


Рисунок 4.1 – Зміна точності базової моделі під час навчання

Як видно з графіка, вже після перших епох спостерігається суттєве зростання точності класифікації. Значення точності на навчальній вибірці

збільшилося приблизно з 63% до 88%, тоді як точність на валідаційній вибірці зросла з 79% до майже 88%. Наприкінці навчання значення навчальної та валідаційної точності практично збігаються, що свідчить про відсутність значного перенавчання моделі та її здатність коректно узагальнювати знання на нових даних.

На рисунку 4.2 наведено графік зміни функції втрат під час навчання.

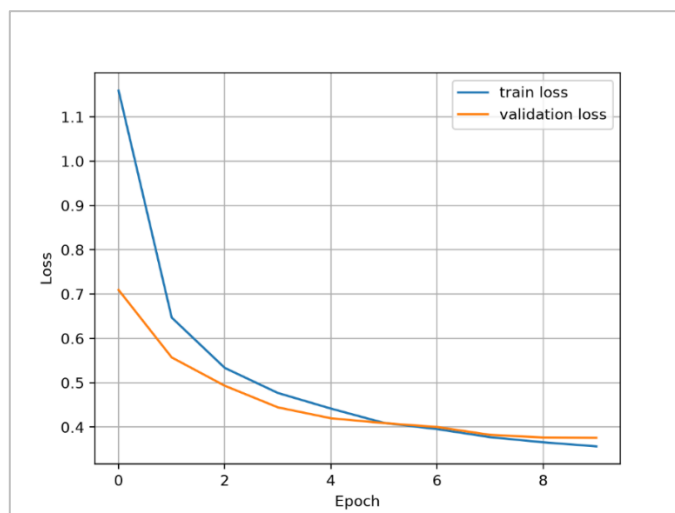


Рисунок 4.2 – Зміна функції втрат базової моделі під час навчання

Аналіз графіка показує стабільне зменшення значення функції втрат як на навчальній, так і на валідаційній вибірках. На початку навчання значення функції втрат для навчальної вибірки перевищувало 1,15, проте вже після декількох епох воно суттєво зменшилося та наприкінці навчання досягло приблизно 0,36. Аналогічна тенденція спостерігається і для валідаційної вибірки. Відсутність різкого розходження між кривими втрат свідчить про стабільний процес навчання та відсутність виражених ознак перенавчання.

Для об'єктивної оцінки якості моделі після завершення навчання було виконано тестування на незалежній тестовій вибірці, яка не використовувалася під час навчання або валідації. Результати тестування наведено в таблиці 4.1.

Таблиця 4.1 – Результати оцінювання базової моделі

Показник	Значення
Точність на тестовій вибірці (Accuracy)	88,49%
Функція втрат на тестовій вибірці (Loss)	0,356
Кількість епох навчання	10
Кількість класів	13
Розмір вхідного зображення	224×224
Розмір пакета (Batch Size)	32

За результатами тестування базова модель досягла точності класифікації 88,49% при значенні функції втрат 0,356. Отримані результати свідчать про достатньо високий рівень якості розпізнавання захворювань рослин та підтверджують ефективність використання архітектури MobileNetV2 для задачі автоматизованої діагностики фітопатологій.

Водночас аналіз отриманих результатів показує, що потенціал моделі не був повністю вичерпаний. Оскільки на початковому етапі всі шари базової мережі MobileNetV2 залишалися замороженими, модель використовувала лише попередньо сформовані ознаки без додаткової адаптації до специфіки зображень захворювань пасльонових культур. Для підвищення точності класифікації доцільним є застосування процедури донавчання (fine-tuning), яка дозволяє адаптувати частину параметрів базової мережі до особливостей предметної галузі.

Таким чином, результати навчання базової моделі підтвердили можливість ефективного використання підходу transfer learning для автоматизованої діагностики захворювань рослин та стали основою для подальшого донавчання моделі з метою покращення показників класифікації.

4.2 Результати донавчання моделі

Після завершення навчання базової моделі було виконано її додаткову адаптацію до предметної галузі методом *fine-tuning*. На відміну від початкового етапу навчання, де всі шари базової архітектури MobileNetV2 залишалися замороженими, під час донавчання було розморожено останні 35 шарів мережі. Такий підхід дозволяє моделі уточнити внутрішні представлення ознак відповідно до особливостей зображень захворювань томатів і картоплі, зберігаючи при цьому знання, отримані під час попереднього навчання на наборі даних ImageNet.

Для забезпечення стабільності процесу донавчання використовувався оптимізатор Adam зі зменшеною швидкістю навчання 0,00001. Зменшення швидкості навчання є важливим під час *fine-tuning*, оскільки дозволяє поступово коригувати ваги попередньо навченої моделі без втрати вже сформованих ознак.

Навчання виконувалося протягом десяти епох із використанням механізмів Early Stopping та Model Checkpoint. У процесі донавчання автоматично зберігалися значення точності класифікації та функції втрат для навчальної і валідаційної вибірок.

Динаміка зміни точності під час донавчання наведена на рисунку 4.3.

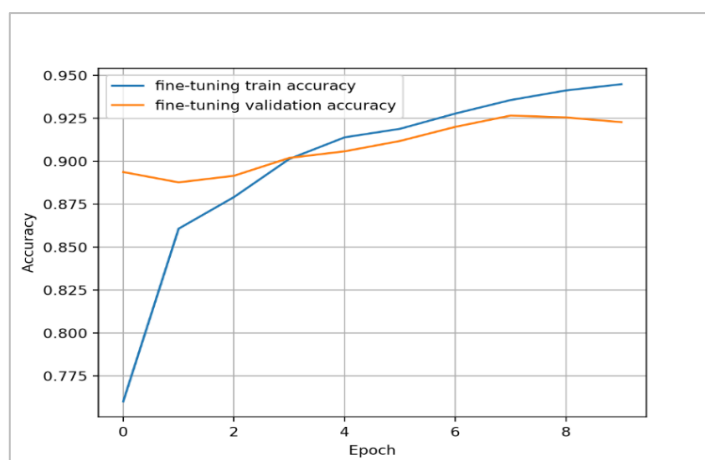


Рисунок 4.3 – Зміна точності під час донавчання моделі

З графіка видно, що вже на початкових етапах донавчання модель демонструє високі показники якості. Початкове значення точності на валідаційній вибірці становило близько 89%, що пояснюється використанням попередньо навченої базової моделі. У процесі подальшого навчання спостерігається поступове зростання як навчальної, так і валідаційної точності.

Наприкінці процесу донавчання точність на навчальній вибірці досягла приблизно 94,5%, тоді як точність на валідаційній вибірці становила близько 92,3%. Невелика різниця між цими показниками свідчить про добру узагальнювальну здатність моделі та відсутність суттєвого перенавчання.

Динаміка зміни функції втрат наведена на рисунку 4.4.

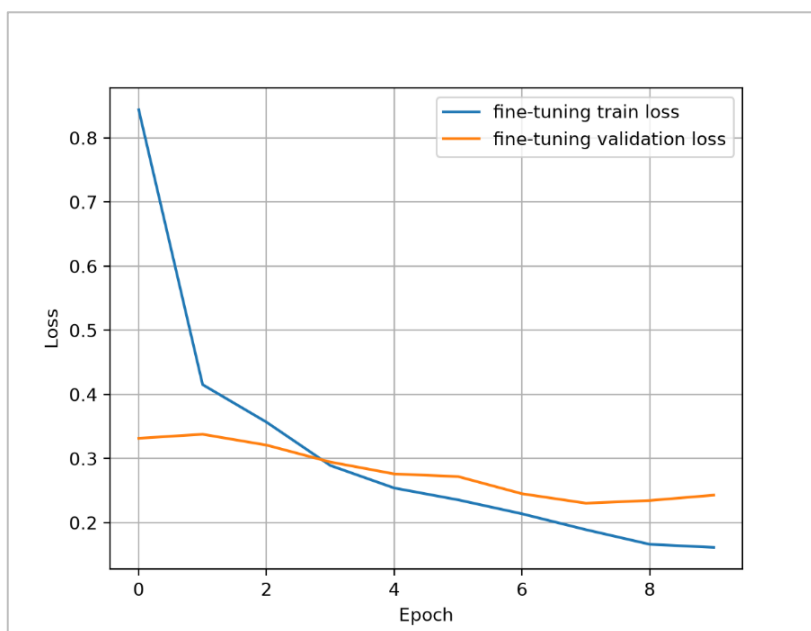


Рисунок 4.4 – Зміна функції втрат під час донавчання моделі

Аналіз графіка показує стабільне зменшення значення функції втрат як для навчальної, так і для валідаційної вибірок. Значення навчальної втрати зменшилося приблизно з 0,84 до 0,16, тоді як валідаційна втрата знизилася з 0,33 до 0,24.

Найменше значення втрат на валідаційній вибірці було досягнуто на сьомій епосі. Надалі спостерігається незначне збільшення валідаційної втрати при одночасному зростанні навчальної точності, що може свідчити про початок слабкого перенавчання. Проте масштаби цього ефекту є незначними та не мають істотного впливу на якість класифікації.

Після завершення процесу донавчання модель була протестована на незалежній тестовій вибірці, яка не використовувалася під час навчання або налаштування параметрів моделі.

Отримані результати наведено в таблиці 4.2.

Таблиця 4.2 – Результати донавченої моделі на тестовій вибірці

Показник	Значення
Точність класифікації (Accuracy)	92,24%
Функція втрат (Loss)	0,238
Кількість епох	10
Кількість розморожених шарів	35
Швидкість навчання	0,00001

Для оцінювання ефективності застосування fine-tuning було виконано порівняння результатів базової та донавченої моделей. Отримані результати наведено в таблиці 4.3.

Таблиця 4.3 – Порівняння результатів базової та донавченої моделей

Метрика	Базова модель	Донавчена модель
Accuracy	88,49%	92,24%
Loss	0,356	0,238
Покращення точності	—	+3,75%

Отримані результати демонструють позитивний вплив донавчання на якість класифікації. Після застосування fine-tuning точність моделі зросла з 88,49% до 92,24%, а значення функції втрат зменшилося з 0,356 до 0,238. Це свідчить про те, що додаткова адаптація останніх шарів MobileNetV2 дозволила моделі краще врахувати специфічні особливості зображень захворювань пасльонових культур.

Таким чином, використання підходу fine-tuning забезпечило підвищення точності класифікації та покращення узагальнювальної здатності моделі. Отримана донавчена модель була використана як основна модель інтелектуальної системи та надалі застосовувалася для аналізу тестових зображень і формування діагностичних висновків. Наступним етапом дослідження є оцінювання роботи моделі на окремих тестових прикладах та аналіз результатів, отриманих із використанням Grad-CAM методу.

4.3 Аналіз роботи системи на тестових зображеннях

Для оцінювання практичної придатності розробленої системи було проведено тестування на зображеннях листків томату та картоплі, які не використовувалися під час навчання моделі. Метою тестування була перевірка коректності класифікації захворювань, оцінка рівня впевненості моделі, а також аналіз роботи механізму пояснення рішень Grad-CAM.

Під час тестування користувач завантажував зображення листка через вебінтерфейс системи. Після виконання аналізу система формувала діагностичний висновок, відображала рівень впевненості класифікації, рекомендації щодо подальших дій та теплову карту Grad-CAM, яка демонструє області зображення, що найбільше вплинули на результат прогнозування. Результати тестування для окремих прикладів наведено в таблиці 4.4.

Таблиця 4.4 – Результати тестування системи на окремих зображеннях

Культура	Виявлений стан	Впевненість моделі, %
Томат	Здоровий стан	100,00
Томат	Альтернاریоз	73,45
Томат	Фітофтороз	86,68
Картопля	Альтернاریоз (рання плямистість)	100,00
Томат	Вірус тютюнової мозаїки	79,92

На рисунку 4.5 представлено приклад аналізу листка картоплі з ознаками ранньої плямистості. Модель класифікувала захворювання як альтернاریоз із рівнем впевненості 100%. Теплова карта демонструє концентрацію уваги моделі безпосередньо на уражених ділянках листової пластинки, що підтверджує коректність процесу прийняття рішення.

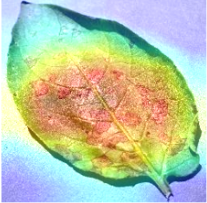

PlantVision
Детальний результат діагностики

Нова перевірка

Завантажене зображення



Пояснення рішення моделі



Теплова карта показує ділянки зображення, які найбільше вплинули на рішення моделі.

Культура: Картопля

Дата: 2026-06-10

Результат діагностики

● Рослина уражена захворюванням

Діагноз: Альтернاریоз

Рівень впевненості: 100%

Опис

Виявлено ознаки ранньої плямистості картоплі.

Рекомендації

Рекомендується видалити уражене листя, уникати надмірного зволоження ґрунту та забезпечити достатню циркуляцію повітря. За необхідності слід застосувати фунгіцидні засоби відповідно до інструкції виробника.

Зберегти результат

Повернутися на головну

Рисунок 4.5 – Результат діагностики альтернاریозу картоплі

Проведене тестування показало, що система здатна успішно розпізнавати як здорові рослини, так і різні типи захворювань пасльонових культур. У всіх наведених прикладах система сформувала коректний результат класифікації. Рівень впевненості моделі знаходився в межах від 73,45% до 100%, що свідчить про достатньо високий ступінь достовірності прогнозів.

Окремої уваги заслуговує використання механізму Grad-CAM. Візуальний аналіз теплових карт показав, що модель концентрує увагу саме на патологічно змінених ділянках листка, а не на фоні або сторонніх елементах зображення. Це підвищує інтерпретованість результатів роботи системи та дозволяє користувачу отримати додаткове підтвердження правильності діагностичного висновку.

Таким чином, результати тестування на реальних зображеннях підтвердили працездатність розробленої системи та її здатність виконувати автоматизовану діагностику захворювань пасльонових культур із високим рівнем точності та наочним поясненням отриманих результатів.

4.4 Перевірка функціонування вебсистеми

Після завершення реалізації програмних компонентів було проведено комплексну перевірку функціонування вебсистеми. Метою тестування було підтвердження коректної роботи користувацького інтерфейсу, механізмів взаємодії із сервісом інтелектуальної діагностики, відображення результатів аналізу та збереження інформації до бази даних.

На першому етапі було перевірено роботу головної сторінки системи. Користувач має можливість обрати культуру, завантажити зображення листка рослини, вказати дату спостереження та, за необхідності, додаткові параметри середовища, зокрема температуру й відносну вологість повітря. Завантаження файлу реалізовано із використанням механізму drag-and-drop або стандартного вибору файлу через файлову систему. Перед запуском

аналізу виконується перевірка коректності введених даних засобами JavaScript. Приклад головної сторінки із завантаженням зображення наведено на рисунку 4.6.

Рисунок 4.6 – Головна сторінка вебсистеми перед запуском аналізу

Після натискання кнопки «Аналізувати» клієнтська частина формує HTTP-запит до сервісу FastAPI, який виконує класифікацію зображення, побудову теплової карти Grad-CAM та формування рекомендацій. Отримані результати передаються назад до вебзастосунку та відображаються на окремій сторінці детального результату.

На сторінці результату користувач може переглянути вихідне зображення, теплову карту Grad-CAM, визначене захворювання або здоровий стан рослини, рівень впевненості моделі, опис виявленого стану

та рекомендації щодо подальшого догляду. Крім того, відображаються додаткові параметри, введені під час аналізу. Приклад сторінки результату наведено на рисунку 4.7.

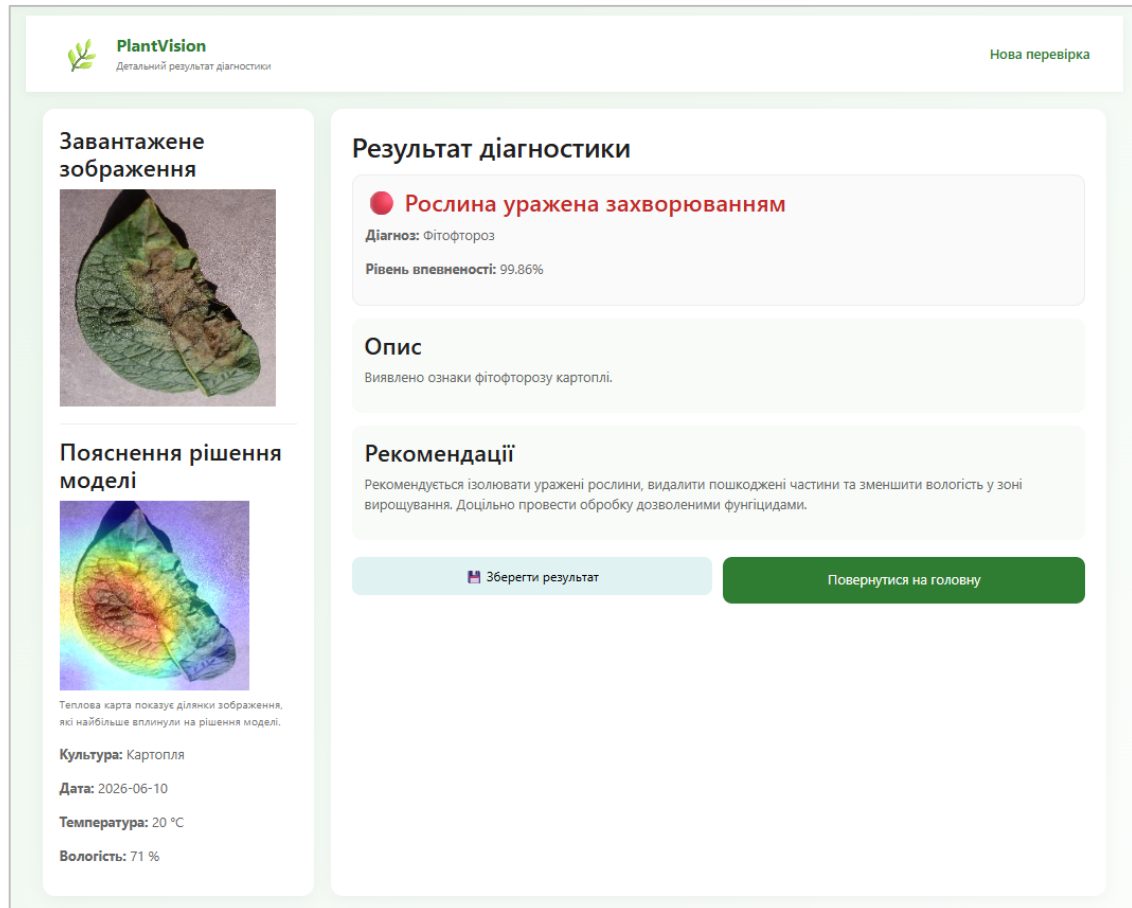


Рисунок 4.7 – Сторінка детального результату діагностики

Окремо було перевірено механізм збереження результатів аналізу. Після натискання кнопки «Зберегти результат» інформація передається до серверної частини ASP.NET Core MVC та записується до бази даних SQLite. Для кожної перевірки зберігаються дані про культуру, діагноз, статус рослини, рівень впевненості моделі, рекомендації, дату спостереження та завантажене зображення.

Для перегляду раніше виконаних перевірок у системі реалізовано сторінку історії. Інформація отримується з бази даних та відображається у вигляді окремих карток, що містять мініатюру зображення, назву

захворювання, культуру, стан рослини, рівень впевненості та дату проведення аналізу. Приклад сторінки історії наведено на рисунку 4.8.

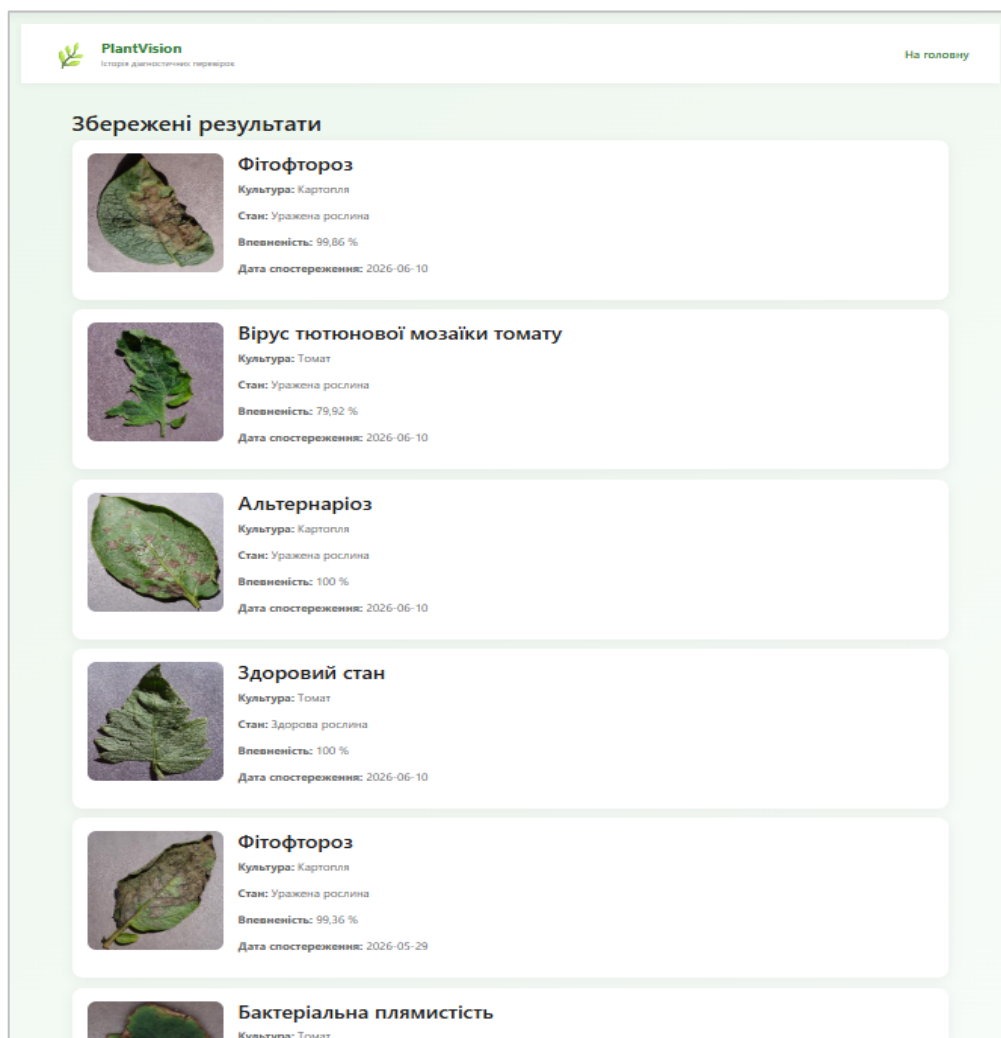


Рисунок 4.8 – Сторінка історії збережених результатів

Проведене тестування підтвердило коректну роботу всіх основних функцій вебсистеми. Користувач може виконувати повний цикл взаємодії із системою: завантажувати зображення рослин, отримувати результати інтелектуальної діагностики з поясненням рішення моделі, зберігати результати аналізу та переглядати історію раніше виконаних перевірок. Це свідчить про працездатність розробленого програмного забезпечення та його готовність до практичного використання.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було розроблено інтелектуальну вебсистему моніторингу фітопатологій пасльонових культур, призначену для автоматизованого аналізу зображень листків рослин, визначення їхнього стану, формування рекомендацій щодо догляду та візуального пояснення результатів класифікації.

У межах роботи було проаналізовано предметну галузь автоматизованої діагностики захворювань рослин, розглянуто сучасні методи комп'ютерного зору, глибокого навчання, transfer learning та підходи до інтерпретації результатів роботи моделей машинного навчання. Також було досліджено існуючі програмні рішення та відкриті набори даних, що застосовуються для задач класифікації захворювань рослин.

Поставлене завдання виконано у повному обсязі. Було спроектовано архітектуру системи, визначено функціональні й нефункціональні вимоги, розроблено структуру бази даних, реалізовано модуль інтелектуальної діагностики, вебінтерфейс користувача, механізм збереження результатів перевірок та модуль пояснення рішень моделі на основі Grad-CAM.

Для класифікації зображень було використано архітектуру MobileNetV2 із застосуванням підходу transfer learning. Базова модель досягла точності 88,49% на тестовій вибірці при значенні функції втрат 0,356. Після виконання процедури fine-tuning із розморожуванням останніх 35 шарів MobileNetV2 точність моделі зросла до 92,24%, а значення функції втрат зменшилося до 0,238. Отже, донавчання дозволило підвищити точність класифікації приблизно на 3,75 відсоткових пункти, що підтверджує доцільність використання fine-tuning для адаптації попередньо навченої моделі до задачі діагностики захворювань томату та картоплі.

Якісні результати роботи системи підтвердили її здатність коректно розпізнавати здоровий стан рослин і різні типи захворювань пасльонових культур. Під час тестування на окремих зображеннях система формувала

діагностичний висновок, відображала рівень впевненості моделі, надавала рекомендації та будувала теплову карту Grad-CAM. Використання методу Grad-CAM підвищує інтерпретованість роботи моделі, оскільки дозволяє користувачу побачити, які саме ділянки зображення найбільше вплинули на результат класифікації.

Розроблена система має переваги порівняно з типовими прикладними рішеннями для діагностики рослин, оскільки поєднує декілька функціональних можливостей: класифікацію захворювань за зображенням, візуальне пояснення рішення моделі, формування рекомендацій з урахуванням введених параметрів температури та вологості, а також збереження історії перевірок. Водночас система має певні обмеження, зокрема орієнтацію лише на дві культури – томат і картоплю, а також використання навчальних даних, значна частина яких отримана в контрольованих умовах.

Виконана робота відповідає напряму освітньої програми «Штучний інтелект» та може бути використана як приклад практичного застосування методів комп'ютерного зору, глибокого навчання й Explainable AI у задачах аграрної галузі. Матеріали роботи можуть бути корисними в навчальному процесі під час вивчення дисциплін, пов'язаних із машинним навчанням, комп'ютерним зором, розробленням інтелектуальних систем та веборієнтованих програмних рішень.

Практичне значення роботи полягає у створенні доступного програмного інструменту, який може використовуватися для попереднього аналізу стану рослин, підтримки прийняття рішень та ведення історії спостережень. Система не замінює фахову агрономічну експертизу, однак може бути корисною як допоміжний засіб для раннього виявлення можливих ознак захворювань.

Перспективним напрямом розвитку системи є інтеграція додаткових джерел даних для підвищення інформативності та практичної цінності результатів діагностики. Зокрема, доцільним є впровадження календаря

виращування культур із врахуванням фаз розвитку рослин, оптимальних температурних режимів, вимог до вологості та освітлення. Додаткові можливості можуть передбачати використання географічного розташування користувача для врахування кліматичних особливостей регіону та поточних погодних умов під час формування рекомендацій. У подальшому система може бути розширена модулями оцінювання стадії розвитку рослини за зображенням, рекомендаціями щодо живлення та профілактики захворювань, а також засобами моніторингу стану рослин упродовж усього вегетаційного періоду. Це створює передумови для трансформації розробленої системи у комплексну інтелектуальну платформу підтримки прийняття рішень у сфері рослинництва.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Лозова М. С., Гриньова О. Є. Вебдіагностика захворювань сільськогосподарських культур засобами комп'ютерного зору. *Радіотехніка та молодь у XXI столітті* : матеріали 30-го Міжнар. молодіж. форуму, 22-24 квітня 2026 р. Харків : ХНУРЕ, 2026. Т. 6. С. 69–71.
2. 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: 11.04.2026).
3. Enhancing plant disease detection: incorporating advanced CNN architectures for better accuracy and interpretability / A. González-Briones et al. *International journal of computational intelligence systems*. 2025. Vol. 18, no. 1. URL: <https://doi.org/10.1007/s44196-025-00835-2> (date of access: 11.04.2026).
4. About FAO's work on plant production and protection. FAO. URL: <https://www.fao.org/plant-production-protection/about/en/> (date of access: 18.04.2026).
5. Agrios G. N. Environmental factors that cause plant diseases. *Plant pathology*. 1969. P. 552–576. URL: <https://doi.org/10.1016/b978-0-12-044550-9.50018-6> (date of access: 18.04.2026).
6. Enhancing plant disease detection through deep learning: a Depthwise CNN with squeeze and excitation integration and residual skip connections / A. Y. Ashurov et al. *Frontiers in plant science*. 2025. Vol. 15. URL: <https://doi.org/10.3389/fpls.2024.1505857> (date of access: 18.04.2026).
7. Bengio Y., Courville A., Goodfellow I. Deep learning. MIT Press, 2016. 800 p.
8. Pan S. J., Yang Q. A survey on transfer learning. *IEEE transactions on knowledge and data engineering*. 2010. Vol. 22, no. 10. P. 1345–1359. URL: <https://doi.org/10.1109/tkde.2009.191> (date of access: 25.04.2026).

9. An effective ensemble convolutional learning model with fine-tuning for medicinal plant leaf identification / M. A. Hajam et al. *Information*. 2023. T. 14, № 11. C. 618. URL: <https://doi.org/10.3390/info14110618> (date of access: 31.05.2026).
10. GitHub - spMohanty/PlantVillage-Dataset: dataset of diseased plant leaf images and corresponding labels. *GitHub*. URL: <https://github.com/spMohanty/PlantVillage-Dataset> (date of access: 27.05.2026).
11. Mubashir C.A. Plant village dataset - plant disease classification dataset split into train test and validation sets. *Kaggle*. URL: <https://www.kaggle.com/datasets/adilmubashirchaudhry/plant-village-dataset/data> (date of access: 27.05.2026).
12. Overview of ASP.NET core MVC. *Microsoft Learn: Build with answers in reach*. URL: <https://learn.microsoft.com/en-us/aspnet/core/mvc/overview?view=aspnetcore-10.0> (date of access: 27.05.2026).
13. FastAPI Documentation. *FastAPI*. URL: <https://fastapi.tiangolo.com/> (date of access: 27.05.2026).
14. Overview of entity framework core - EF core. *Microsoft Learn: Build with answers in reach*. URL: <https://learn.microsoft.com/en-us/ef/core/> (date of access: 28.05.2026).
15. SQLite documentation. *SQLite Home Page*. URL: <https://sqlite.org/docs.html> (date of access: 28.05.2026).
16. Tf.keras.preprocessing.image_dataset_from_directory | tensorflow v2.16.1. *TensorFlow*. URL: https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image_dataset_from_directory (date of access: 30.05.2026).
17. Plant leaf disease detection using MobileNetV2 / M. Bala N et al. *ITM web of conferences*. 2025. Vol. 79. P. 01021. URL: <https://doi.org/10.1051/itmconf/20257901021> (date of access: 30.05.2026).

18. Adam: a method for stochastic optimization. *arXiv.org*.
URL: <https://doi.org/10.48550/arXiv.1412.6980> (date of access: 30.05.2026).
19. Tf.keras.losses.SparseCategoricalCrossentropy | tensorflow
v2.16.1. *TensorFlow*.
URL: https://www.tensorflow.org/api_docs/python/tf/keras/losses/SparseCategoricalCrossentropy (date of access: 30.05.2026).
20. 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: 31.05.2026).