Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)
Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

ДОСЛІДЖЕННЯ ТА РЕАЛІЗАЦІЯ МЕТОДУ РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ ДЛЯ ДІАГНОСТИКИ ЗАХВОРЮВАННЯ РОСЛИН
(тема)

Виконав:
здобувач 2 року навчання,
групи ІНФм-24-2
Свістельник Д.О.
(прізвище, ініціали)

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

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Науковий керівник доц. Кобилін О. А.
(посада, прізвище, ініціали)

Допускається до захисту

Завідувач кафедри інформатики _____ Кобилін О. А.
(підпис) (прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту

Кафедра Інформатики

Рівень вищої освіти другий (магістерський)

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

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Свістельнику Даніілу Олеговичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та реалізація методу розпізнавання зображень для діагностики захворювання рослин

затверджена наказом університету від 14 листопада 2025 року № 1045Ст
2. Термін подання здобувачем роботи до екзаменаційної комісії 6 грудня 2025 р.
3. Вихідні дані до роботи методи класифікації зображень, літературні джерела щодо застосування методів класифікації, програмні засоби для реалізації вибраних методів класифікації та десктоп-застосунку, зображення хворих рослин для тренування та тестування моделей.

4. Перелік питань, що потрібно опрацювати в роботі
1. Аналіз сучасних методів класифікації об'єктів на зображеннях.
2. Аналіз літературних джерел щодо апробації методів класифікації об'єктів на зображеннях.
3. Формування алгоритму для вибраного методу класифікації.
4. Розробка програмного застосунку, що надасть змогу класифікувати захворювання рослин на зображеннях

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) актуальність проблеми класифікації об'єктів на зображеннях, об'єкт та мета дослідження, постановка задачі, етапи реалізації поставленої задачі

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	29.09.2025	
2	Аналіз завдання, підбір літератури	30.09.25-07.10.25	
3	Аналіз літератури з досліджуваної проблеми	08.10.25-14.10.25	
4	Особливості методів класифікації зображень	15.10.25-20.10.25	
5	Дослідження методів класифікації зображень	21.10.25-27.10.25	
6	Програмна реалізація	28.10.25-05.11.25	
7	Обґрунтування отриманих результатів	06.11.25-11.11.25	
8	Оформлення пояснювальної записки	12.11.25-14.11.25	
9	Перевірка на нормоконтроль	03.12.25-04.12.25	
10	Перевірка на плагіат	05.12.25	
11	Рецензування	06.12.25-10.12.25	
12	Підготовка презентації та доповіді	01.12.25-12.12.25	
13	Занесення роботи в електронний архів	15.12.25	
14	Попередній захист кваліфікаційної роботи	15.12.25	

Дата видачі завдання 29 вересня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ доц. Кобилін О. А.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи: 77 с., 15 рис., 48 джерел.

АУГМЕНТАЦІЯ ДАНИХ, БІБЛІОТЕКА KERAS, БІБЛІОТЕКА TENSORFLOW, ДІАГНОСТИКА ЗАХВОРЮВАНЬ РОСЛИН, ЗГОРТКОВА НЕЙРОННА МЕРЕЖА, КЛАСИФІКАЦІЯ ЗОБРАЖЕНЬ, КОМП'ЮТЕРНИЙ ЗІР, МАШИННЕ НАВЧАННЯ, BATCH NORMALIZATION, CNN, DATASET PLANTVILLAGE, DEEP LEARNING, DROPOUT.

Об'єктом дослідження є цифрові зображення листя рослин з різними типами захворювань та здорових рослин.

Предмет дослідження – методи глибокого навчання для автоматичної діагностики та класифікації захворювань рослин за зображеннями.

Метою дослідження є розробка ефективного методу розпізнавання зображень для автоматичної діагностики захворювань рослин шляхом створення програмного забезпечення на основі згорткових нейронних мереж з високою точністю класифікації.

Використано методи глибокого навчання, зокрема згорткову нейронну мережу на основі MobileNetV2.

Наукова новизна роботи полягає у розробці оптимізованої архітектури згорткової нейронної мережі для класифікації захворювань рослин із застосуванням комплексу сучасних технік регуляризації для забезпечення високої здатності моделі до генералізації та запобігання перенавчанню.

Дослідження базується на фундаментальних роботах у галузі глибокого навчання та їх адаптації до задач аграрного сектору.

Розроблене програмне забезпечення рекомендується для впровадження у практику аграрних підприємств, фермерських господарств та консультаційних центрів як інструмент швидкої діагностики захворювань рослин.

ABSTRACT

Explanatory note to the qualification work: 77 pages, 15 figures, 48 sources.

DATA ENHANCEMENT, KERAS LIBRARY, TENSORFLOW LIBRARY, PLANT DISEASE DIAGNOSTICS, CONVOLUTIONAL NEURAL NETWORK, IMAGE CLASSIFICATION, COMPUTER VISION, MACHINE LEARNING, BATCH NORMALIZATION, CNN, DATASET PLANTVILLAGE, DEEP LEARNING, DROPOUT.

The object of the research is digital images of plant leaves with various types of diseases and healthy plants.

The subject of the research is deep learning methods for automatic diagnosis and classification of plant diseases based on images.

The goal of the research is to develop an effective image recognition method for the automatic diagnosis of plant diseases by creating software based on convolutional neural networks with high classification accuracy.

Deep learning methods were used, in particular a convolutional neural network based on MobileNetV2.

The scientific novelty of the work lies in the development of an optimized convolutional neural network architecture for the classification of plant diseases using a set of modern regularization techniques to ensure high model generalization ability and prevent overfitting.

The research is based on fundamental work in the field of deep learning and its adaptation to the tasks of the agricultural sector.

The developed software is recommended for implementation in the practice of agricultural enterprises, farms, and consulting centers as a tool for rapid diagnosis of plant diseases.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ПЛР – полімеразна ланцюгова реакція

CNN – Convolutional Neural Networks (згорткові нейронні мережі)

GAP – Global Average Pooling (глобальне усереднююче згортання)

Grad-CAM – Gradient-weighted Class Activation Mapping (створення карти активації класу з вагою градієнта)

HOG – Histogram of Oriented Gradients (гістограма орієнтованих градієнтів)

IoT – Internet of Things (інтернет речей)

ORB – Oriented FAST and Rotated BRIEF

ReLU – Rectified Linear Unit (випрямлений лінійний блок)

ROC-крива – Receiver Operating Characteristic (характеристична крива приймача)

SGD – Stochastic Gradient Descent (стохастичний градієнтний спуск)

SIFT – Scale-Invariant Feature Transform (масштабно-інваріантне перетворення ознак)

SURF – Speeded-Up Robust Features (прискорені робастні ознаки)

SVM – Support Vector Machine (метод опорних векторів)

ВСТУП

Сільське господарство є однією з найважливіших галузей економіки, що забезпечує продовольчу безпеку населення та відіграє ключову роль у глобальній економічній стабільності. Захворювання рослин становлять серйозну загрозу для врожайності та якості сільськогосподарської продукції, призводячи до значних економічних втрат у всьому світі. За оцінками Продовольчої та сільськогосподарської організації ООН, щорічні втрати врожаю від шкідників та хвороб становлять близько 20 – 40% світового потенціалу виробництва продовольства, що в грошовому еквіваленті перевищує сотні мільярдів доларів щорічно [1].

Традиційні методи діагностики захворювань рослин базуються на візуальному огляді досвідченими агрономами та лабораторних аналізах. Такий підхід є трудомістким процесом і потребує високої кваліфікації спеціалістів, значних часових та фінансових ресурсів. Суб’єктивність людської оцінки може призводити до помилок у діагностиці, особливо на ранніх стадіях розвитку захворювань, коли симптоми ще недостатньо виражені. При цьому саме раннє виявлення захворювань є критично важливим для своєчасного прийняття рішень щодо захисту рослин та мінімізації втрат врожаю.

Сучасний стан досліджень у галузі автоматизованої діагностики захворювань рослин характеризується активним впровадженням технологій штучного інтелекту, комп’ютерного зору та машинного навчання. Останні досягнення в галузі глибокого навчання, зокрема використання згорткових нейронних мереж, демонструють високу точність розпізнавання патологій рослин за цифровими зображеннями їх листя, стебел та плодів. Різні архітектури нейронних мереж, такі як ResNet, VGG, MobileNet та EfficientNet, активно досліджуються науковцями для вирішення задач класифікації фітопатологій. Впровадження автоматизованих систем діагностики на основі аналізу зображень відкриває нові можливості для точного та швидкого виявлення захворювань у

польових умовах, що особливо актуально в контексті цифровізації сільського господарства та розвитку концепції розумного фермерства.

Інтеграція методів комп’ютерного зору та машинного навчання у процес моніторингу стану посівів відкриває перспективи для створення масштабованих рішень, які можуть застосовуватися як на великих агропромислових підприємствах, так і в умовах невеликих фермерських господарств. Своєчасна та точна діагностика захворювань на основі аналізу зображень дозволяє оптимізувати використання засобів захисту рослин, знизити екологічне навантаження від надмірного застосування пестицидів та забезпечити раціональне управління ресурсами, що є важливим кроком на шляху до сталого розвитку аграрного сектору.

Актуальність роботи полягає у необхідності розробки ефективного та доступного методу автоматизованої діагностики захворювань рослин, який би дозволив підвищити точність класифікації захворювань, пришвидшити процес діагностики та зменшити залежність від висококваліфікованих спеціалістів. Створення такого програмного забезпечення має практичне значення для сільськогосподарських підприємств різного масштабу та сприятиме підвищенню ефективності виробництва, збереженню врожаю та зменшенню економічних втрат.

1 ОГЛЯД МЕТОДІВ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ

1.1 Основні захворювання сільськогосподарських культур

Захворювання рослин можна класифікувати за різними критеріями, однак найпоширенішою є класифікація за етіологією – причиною виникнення хвороби. За цим критерієм розрізняють інфекційні та неінфекційні захворювання.

Інфекційні захворювання викликаються патогенними організмами, які можуть передаватися від однієї рослини до іншої.

Грибкові захворювання є найчисленнішою групою хвороб рослин [2]. Гриби-патогени можуть уражувати всі частини рослини – від коренів до плодів. Серед найпоширеніших грибкових захворювань можна виділити наступні:

– борошниста роса (проявляється у вигляді білого порошкоподібного нальоту на листках, стеблах та плодах);

– іржа (характеризується появою рудувато-коричневих пустул на листках);

– фітофтороз (швидко поширюється та може знищити весь урожай за кілька днів);

– плямистість листя (проявляється появою некротичних плям різної форми та кольору).

Бактеріальні захворювання спричинюються бактеріями, які проникають у рослину через природні отвори або механічні пошкодження [3]. Типовими проявами бактеріозів є бактеріальна плямистість або в’янення, бактеріальний рак та мокра гниль.

Вірусні захворювання часто є найнебезпечнішими, оскільки не піддаються лікуванню і можуть швидко поширюватися через комах-переносників. Віруси викликають глибокі незворотні зміни в рослинах, порушують процеси фотосинтезу, вуглецевого та азотного обміну. Характерними симптомами вірусних інфекцій можна вважати наступні:

– мозаїчність листя (специфічне плямисте забарвлення);

– жовтяниці (пожовтіння рослин та надмірне кущіння);

– деформація листків та плодів;

– затримка росту та розвитку.

Неінфекційні захворювання виникають внаслідок несприятливих умов навколишнього середовища і не передаються від рослини до рослини. Основними причинами неінфекційних хвороб є такі:

– температурний стрес (морози, спека);

– порушення водного режиму (посуха, надлишок вологи);

– дефіцит або надлишок поживних елементів;

– забруднення ґрунту токсичними речовинами;

– механічні пошкодження.

Симптоми захворювань рослин можуть проявлятися на різних органах та у різних формах [4]. На листках найчастіше спостерігаються плямистість, хлороз (пожовтіння), некроз (відмирання тканин), деформація, в’янення. Стебла можуть уражуватися гнилями, раком, некрозами судинної системи. Плоди і насіння схильні до гнилей, плямистості, деформацій, що значно знижує їхню товарну якість та схожість.

Сучасне інтенсивне землеробство з великими площами монокультур створює особливо сприятливі умови для розвитку та поширення хвороб. Тому раннє виявлення перших ознак захворювання і своєчасне втручання є критично важливими для запобігання масовим втратам врожаю та забезпечення продовольчої безпеки.

1.2 Традиційні методи діагностики захворювань рослин

Діагностика захворювань рослин є першим і найважливішим етапом у системі захисту сільськогосподарських культур. Точна та своєчасна ідентифікація патогена дозволяє обрати найбільш ефективні методи боротьби,

мінімізувати використання хімічних засобів для захисту та запобігти поширенню хвороби на великі площі. Протягом тривалого часу фітопатологія використовувала переважно традиційні методи діагностики, які базувалися на фундаментальних принципах ботаніки, мікробіології та мікології.

Візуальна діагностика є найдавнішим і досі найпоширенішим методом виявлення захворювань рослин. Цей метод базується на розпізнаванні характерних симптомів хвороби за зовнішніми ознаками ураженої рослини [5]. Досвідчений фітопатолог може визначити захворювання за специфічними проявами: кольором, формою та розташуванням плям на листках, характером некрозів, деформацією органів, наявністю спороношення патогена.

Переваги візуальної діагностики полягають у її простоті, доступності та можливості швидкої оцінки стану рослин безпосередньо в середовищі їх вирощування. Однак цей метод має суттєві недоліки [6]. По-перше, він вимагає високої кваліфікації спеціаліста та багаторічного досвіду роботи. По-друге, візуальна діагностика є суб’єктивною і може призводити до помилок, особливо на ранніх стадіях захворювання, коли симптоми ще слабко виражені. По-третє, різні захворювання можуть мати схожі зовнішні прояви, що ускладнює точну ідентифікацію.

Мікроскопічна діагностика передбачає вивчення морфологічних особливостей патогена під мікроскопом. Для грибкових захворювань аналізують форму та розміри спор, тип міцелію тощо. Для бактеріальних інфекцій досліджують форму клітин, їхню рухливість, наявність джгутиків. Мікроскопія може бути світловою або електронною, залежно від об’єкта дослідження та необхідної роздільної здатності.

Цей метод забезпечує більш точну ідентифікацію патогена порівняно з візуальним оглядом, дозволяє виявити збудника на ранніх стадіях інфікування. Однак мікроскопічна діагностика потребує спеціального обладнання, підготовки зразків, високої кваліфікації персоналу та займає значний час. Крім того, деякі патогени не мають характерних морфологічних ознак, що ускладнює їхню ідентифікацію цим методом [7].

Мікологічний аналіз полягає у виділенні патогена з ураженої тканини рослини та його культивуванні на спеціальних поживних середовищах. Після отримання чистої культури проводять детальне вивчення морфологічних, культуральних та фізіологічних властивостей гриба. Аналізують характер росту колоній, їхній колір, текстуру, швидкість розвитку тощо.

Мікологічний метод дає можливість отримати чисту культуру патогена для подальших досліджень, точно ідентифікувати вид збудника, вивчити його біологічні особливості. Проте цей підхід є дуже трудомістким та займає багато часу – отримання чистої культури може займати від кількох днів до кількох тижнів [8]. Деякі патогени взагалі не культивуються на штучних середовищах, що робить цей метод неприйнятним для їхньої діагностики.

Біологічна діагностика з використанням рослин-індикаторів застосовується переважно для виявлення вірусних інфекцій. Метод базується на штучному зараженні рослин-індикаторів, які реагують на присутність певного вірусу специфічними симптомами. Різні віруси викликають характерні реакції на певних рослинах-індикаторах, що дозволяє їх ідентифікувати.

Цей метод є відносно простим і не вимагає складного обладнання, однак він надзвичайно тривалий у часі (результати можна отримати через кілька тижнів), потребує підтримання колекції рослин-індикаторів та контрольованих умов вирощування. Крім того, метод має обмежену специфічність і не завжди дозволяє точно ідентифікувати конкретний штам вірусу.

Серологічні методи базуються на реакції антиген-антитіло і широко використовуються для діагностики вірусних та бактеріальних захворювань. Найпоширенішим серологічним методом є імуноферментний аналіз, який дозволяє швидко та з високою чутливістю виявити специфічні білки патогена в рослинному матеріалі. Метод передбачає використання антитіл, мічених ферментом, які специфічно зв’язуються з антигенами патогена.

Серологічні методи характеризуються високою чутливістю та специфічністю, можливістю аналізу великої кількості зразків одночасно, виявленням патогена на ранніх стадіях інфікування, ще до появи симптомів.

Однак вони потребують наявності специфічних антитіл, які не завжди доступні, спеціального обладнання та реактивів, кваліфікованого персоналу.

Молекулярно-генетичні методи представляють собою найсучасніший напрям традиційної діагностики. Полімеразна ланцюгова реакція дозволяє виявити присутність патогена за специфічними фрагментами його ДНК або РНК. Метод базується на багаторазовому копіюванні певної ділянки нуклеїнової кислоти, що робить можливим виявлення навіть мінімальних кількостей патогена.

ПЛР-діагностика характеризується надзвичайно високою чутливістю з можливістю виявлення одиничних клітин патогена, ідентифікацією на рівні виду і навіть штаму, швидкістю отримання результатів та можливістю виявлення патогенів, які не культивуються [9].

До недоліків методу належать висока вартість обладнання та реактивів, необхідність спеціально обладнаної лабораторії, кваліфікованого персоналу, можливість хибно-позитивних результатів через контамінацію, неможливість визначення життєздатності патогена.

Незважаючи на різноманітність і надійність багатьох традиційних методів, усі вони мають спільні недоліки, які обмежують їхнє застосування в умовах сучасного інтенсивного землеробства:

– трудомісткість і часозатратність – більшість методів вимагають значних витрат часу (від кількох днів до тижнів) на проведення аналізу, що неприйнятно за необхідності швидкого прийняття рішень про захисні заходи;

– необхідність кваліфікованого персоналу – точна діагностика вимагає спеціалістів з глибокими знаннями у фітопатології, мікології, вірусології, що обмежує доступність методів;

– висока вартість – лабораторні методи потребують дорогого обладнання, реактивів, витратних матеріалів, що робить їх економічно недоступними для багатьох господарств;

– обмежена масштабованість – традиційні методи не дозволяють швидко обробляти велику кількість зразків для моніторингу великих площ;

– суб’єктивність – багато методів залежать від досвіду та кваліфікації конкретного спеціаліста, що може призводити до розбіжностей в інтерпретації результатів.

Ці обмеження традиційних підходів зумовили необхідність розробки нових, автоматизованих методів діагностики захворювань рослин, які б поєднували швидкість, точність, доступність та можливість масштабування для використання в умовах виробництва.

1.3 Сучасні підходи до автоматизованої діагностики

Обмеження традиційних методів діагностики захворювань рослин, поєднані з зростаючими вимогами до продуктивності сільського господарства та необхідністю швидкого реагування на загрози, стимулювали розвиток автоматизованих систем діагностики на основі сучасних інформаційних технологій [10]. Особливо значний прогрес у цій галузі було досягнуто завдяки стрімкому розвитку комп’ютерного зору, машинного навчання та глибокого навчання протягом останніх двох десятиліть.

Перші спроби автоматизації діагностики захворювань рослин датуються кінцем XX століття і базувалися на класичних методах обробки зображень та аналізу окремих візуальних ознак [11]. Ці підходи передбачали ручне виділення характеристик зображення (колірних показників, текстури, форми плям) та їхню подальшу класифікацію за допомогою простих алгоритмів машинного навчання.

Типовий процес обробки в ранніх автоматизованих системах включав кілька етапів: попередня обробка зображення (корекція освітлення, фільтрація шумів), сегментація (виділення ураженої ділянки від здорового фону), вилучення ознак (обчислення колірних гістограм, текстурних параметрів, геометричних характеристик) та класифікація за допомогою традиційних алгоритмів машинного навчання.

Ці ранні системи мали значні обмеження: вони були чутливі до умов зйомки (освітлення, фон, кут), вимагали ручного проєктування ознак, що потребувало експертних знань, погано узагальнювалися на нові умови та види рослин, і мали обмежену точність, особливо при роботі з складними або схожими захворюваннями.

Справжній прорив в автоматизованій діагностиці захворювань рослин стався з появою методів глибокого навчання, зокрема згорткових нейронних мереж [12]. Принципова відмінність CNN від попередніх підходів полягає у здатності автоматично навчатися безпосередньо з вихідних зображень, без необхідності ручного проєктування ознак.

Згорткові нейронні мережі складаються з декількох типів шарів, кожен з яких виконує специфічну функцію. Згорткові шари застосовують набір навчуваних фільтрів до вхідного зображення, автоматично виявляючи локальні паттерни (краї, текстури, форми). Шари підвибірки зменшують розмірність даних, забезпечуючи інваріантність до невеликих зсувів та деформацій. Повнозв’язні шари на завершальному етапі виконують фінальну класифікацію на основі вивчених ознак [13].

Ключовою перевагою CNN є їхня здатність автоматично навчатися ієрархічним представленням: нижні шари виявляють прості ознаки (краї, кольорові переходи), середні шари комбінують їх у більш складні структури (текстури, частини об’єктів), а верхні шари формують високорівневі семантичні представлення, що відповідають цілим об’єктам або концепціям.

1.3.1 Архітектури CNN

Розвиток глибокого навчання призвів до створення численних архітектур CNN, кожна з яких має свої особливості та переваги для різних застосувань.

Архітектура VGG (Visual Geometry Group) характеризується простотою та регулярністю структури, використовуючи послідовні блоки малих згорткових

фільтрів розміром 3×3. Незважаючи на значну кількість параметрів, що робить мережу важкою для навчання та порівняння продуктивності, VGG демонструє стабільні результати і часто використовується як базова архітектура для порівняння [14 – 16].

Архітектура ResNet (Residual Network) вирішила проблему деградації точності при збільшенні глибини мережі за допомогою залишкових з'єднань, які дозволяють градієнтам проходити через мережу без затухання. Це дало можливість навчати надзвичайно глибокі мережі (до 152 шарів і більше), досягаючи високої точності на складних задачах розпізнавання.

Архітектура MobileNet розроблена спеціально для мобільних та вбудованих пристроїв з обмеженими обчислювальними ресурсами [17, 18]. Архітектура використовує глибинно-роздільні згортки, які драматично зменшують кількість параметрів та обчислювальну складність при збереженні прийнятної точності. Це робить MobileNet ідеальним вибором для створення мобільних застосунків діагностики захворювань рослин.

Архітектура EfficientNet представляє сучасний підхід до проєктування архітектури, що базується на принципі компонованого масштабування. Замість довільного збільшення глибини, ширини або роздільної здатності вхідних зображень, EfficientNet балансує всі три виміри за допомогою спеціального коефіцієнта, досягаючи оптимального співвідношення точності та ефективності. Різні варіанти EfficientNet (від B0 до B7) дозволяють обирати баланс між точністю та швидкістю відповідно до конкретних вимог застосування.

Архітектура DenseNet використовує щільні з'єднання між шарами, де кожен шар отримує вхідні дані від усіх попередніх шарів. Така архітектура сприяє повторному використанню ознак, зменшує проблему зникаючого градієнта і забезпечує ефективне використання параметрів, досягаючи високої точності при відносно невеликій кількості параметрів.

1.3.2 Передавальне навчання (Transfer Learning)

Одним з найважливіших практичних досягнень у застосуванні глибокого навчання до діагностики захворювань рослин стало використання підходу передавального навчання (transfer learning). Суть методу полягає в тому, що модель спочатку навчається на великому загальному наборі даних (таких як ImageNet, що містить мільйони зображень з тисячами категорій), а потім адаптується, тобто довчається, для специфічної задачі діагностики захворювань [19].

Цей підхід має кілька важливих переваг. По-перше, він дозволяє досягати високої точності навіть при обмеженій кількості спеціалізованих даних про захворювання рослин, оскільки модель вже навчилася виділяти базові візуальні ознаки на етапі попереднього навчання. По-друге, значно скорочується час навчання, адже не потрібно навчати всю мережу з нуля. По-третє, знижуються вимоги до обчислювальних ресурсів, що робить технологію доступнішою.

На практиці передавальне навчання (transfer learning) може застосовуватися різними способами: або заморожуються нижні шари мережі і довчаються лише верхні класифікаційні шари, або виконується тонке налаштування усієї мережі або її частини з невеликою швидкістю навчання для адаптації до нової задачі.

1.3.3 Типові проблеми методів

Однією з проблем використання глибоких нейронних мереж в критичних застосуваннях, таких як діагностика захворювань, є складність розуміння того, як саме модель приймає рішення. Це викликає питання довіри до автоматизованих систем діагностики.

Для вирішення цієї проблеми розроблено механізми уваги, які дозволяють моделі явно фокусуватися на найбільш інформативних ділянках зображення.

Наприклад, при діагностиці захворювання механізм уваги може виділяти саме ураженні ділянки листка, ігноруючи несуттєві деталі фону. Це не лише покращує точність моделі, але й робить її рішення більш інтерпретованими для людини-експерта.

Додатково застосовуються методи візуалізації, такі як Grad-CAM, які генерують теплові карти, що показують, які ділянки зображення найбільше вплинули на рішення моделі. Це дозволяє фахівцям перевірити, чи справді модель спирається на релевантні ознаки захворювання.

1.3.4 Навчальні дані

Успіх застосування глибокого навчання значною мірою залежить від наявності якісних навчальних даних. У галузі діагностики захворювань рослин створено кілька великих публічних датасетів, що стали стандартними мірилами.

PlantVillage є одним з найвідоміших датасетів, що містить понад 54 тисячі зображень здорових та хворих листків, розділених на 38 категорій, які охоплюють 14 видів рослин та різноманітні захворювання. Датасет створений у контрольованих лабораторних умовах, що забезпечує високу якість зображень, але може обмежувати здатність моделей узагальнювати на реальні польові умови.

Набір даних PlantDoc є більш складним, що містить зображення, зняті в польових умовах з різноманітними фонами, освітленням та кутами зйомки. Це робить його більш реалістичним для тренування моделей, призначених для практичного використання в сільському господарстві.

Набір даних Plant Leaves Dataset, доступний через колекцію TensorFlow Datasets, містить високоякісні зображення листків рослин у різних станах здоров’я, організовані за видами та типами захворювань. Датасет оптимізований для використання з сучасними фреймворками глибокого навчання.

1.3.5 Практичні застосування

Сучасні автоматизовані системи діагностики реалізуються у різних формах. Мобільні застосунки, такі як Plantix, Agrobase та Pestoz, дозволяють фермерам безпосередньо в полі фотографувати уражені рослини та отримувати миттєву діагностику з рекомендаціями щодо лікування. Ці системи використовують легкі моделі, оптимізовані для роботи на смартфонах.

Веб-платформи надають можливість більш детального аналізу з використанням потужніших моделей, збереження історії діагнозів, статистичного аналізу поширення захворювань. Інтеграція з IoT дозволяє створювати системи безперервного моніторингу, де камери, встановлені в полі або теплиці, автоматично аналізують стан рослин та сповіщають про виявлені захворювання.

Системи з використанням дронів та супутниковий моніторинг використовують методи глибокого навчання для виявлення захворювань на великих площах, аналізуючи зображення для раннього виявлення стресу рослин, який може бути непомітним для звичайних камер.

Сучасні системи на основі глибокого навчання демонструють вражаючі результати: точність діагностики 95 – 99% на багатьох датасетах, швидкість аналізу кілька мілісекунд на одне зображення при використанні GPU, можливість одночасного виявлення десятків різних захворювань, робота в режимі реального часу на мобільних пристроях.

Однак залишаються й значні виклики. Проблема узагальнення – моделі, навчені на одних умовах, можуть погано працювати в інших (інше освітлення, фон, вид камери). Необхідність великих обсягів розмічених даних для навчання робить створення систем для рідкісних захворювань або нових культур складним завданням. Складність діагностики на ранніх стадіях, коли симптоми слабко виражені, залишається проблемою навіть для найсучасніших моделей. Потреба

в обчислювальних ресурсах для навчання складних моделей може бути бар’єром для дослідників з обмеженим бюджетом.

Незважаючи на це, автоматизовані системи діагностики на основі глибокого навчання представляють надзвичайно перспективний напрямок, який має зробити революцію у практиці захисту рослин, зробивши точну та швидку діагностику доступною для широкого кола користувачів.

Проведений аналіз показує, що традиційні методи діагностики захворювань рослин, незважаючи на високу точність ідентифікації патогенів, мають критичні обмеження для практичного застосування в умовах сучасного сільського господарства. Висока трудомісткість, значні витрати часу, необхідність спеціалізованого обладнання та кваліфікованого персоналу, а також обмежена масштабованість роблять їх недостатньо ефективними для швидкого прийняття рішень та моніторингу великих сільськогосподарських площ.

Автоматизовані системи діагностики на основі згорткових нейронних мереж демонструють революційний прогрес у вирішенні цих проблем. Сучасні архітектури CNN, такі як ResNet, MobileNet та EfficientNet, у поєднанні з методом Transfer Learning забезпечують точність розпізнавання захворювань на рівні 95 – 99% при можливості обробки зображень у режимі реального часу. Доступність великих публічних датасетів та різноманітність практичних реалізацій у вигляді мобільних та веб-застосунків підтверджує готовність технології до широкого впровадження. Це обґрунтовує доцільність розробки нових методів автоматичної діагностики як перспективного напрямку підвищення ефективності захисту рослин у сучасному землеробстві.

1.4 Класичні методи комп’ютерного зору

Комп’ютерний зір як наукова дисципліна має багатолітню історію, що передує ері глибокого навчання. Класичні методи комп’ютерного зору, розроблені протягом 1960-2000-х років, заклали фундаментальні принципи

обробки та аналізу зображень, багато з яких залишаються актуальними і сьогодні. Розуміння цих традиційних підходів є важливим для повного усвідомлення переваг та обмежень сучасних методів глибокого навчання.

Класичний конвеєр обробки зображень для задач розпізнавання та класифікації зазвичай складається з декількох послідовних етапів, кожен з яких вирішує специфічну проблему.

Попередня обробка є першим і критично важливим етапом, що забезпечує покращення якості вхідних даних та їх стандартизацію. Основні операції попередньої обробки включають корекцію освітлення для нормалізації яскравості зображення та усунення тіней чи бліків; фільтрацію шумів за допомогою різноманітних фільтрів, таких як медіанний, гаусівський чи білатеральний; зміну розміру зображення до стандартних розмірів; корекцію кольору для забезпечення консистентності кольорових характеристик; перетворення колірних просторів RGB, HSV або LAB залежно від специфіки задачі.

Сегментація зображення – процес поділу зображення на окремі області або об’єкти, що мають певні спільні характеристики. У контексті діагностики захворювань рослин сегментація дозволяє виділити листок від фону або ідентифікувати уражені ділянки на здоровій тканині. Існує кілька основних підходів до сегментації.

Порогова сегментація є найпростішим методом, що базується на виборі порогового значення інтенсивності пікселів. Пікселі, яскравість яких перевищує поріг, відносяться до одного класу, решта – до іншого. Метод Отсу автоматично визначає оптимальне порогове значення, що мінімізує внутрішньокласову дисперсію. Незважаючи на простоту, метод добре працює лише для зображень з чітким бімодальним розподілом яскравості.

Сегментація на основі країв виявляє межі між різними регіонами зображення, де відбуваються різкі зміни інтенсивності. Детектори країв, такі як оператори Собеля, Превітта чи Канні, обчислюють градієнти інтенсивності та виявляють точки з максимальними змінами. Детектор Канні, зокрема,

використовує багатоетапний алгоритм з придушенням немаксимумів та подвійною пороговою обробкою для виявлення справжніх країв та відсіювання шуму.

Сегментація на основі регіонів групує пікселі зі схожими характеристиками у зв'язні області. Метод нарощування регіонів починає з початкових точок та ітеративно додає сусідні пікселі, що задовольняють критерію подібності. Метод водозбору розглядає зображення як топографічну поверхню, де інтенсивність відповідає висоті, та знаходить межі між регіонами як вододіли між басейнами.

Кластеризація використовує алгоритми машинного навчання без вчителя для групування пікселів у кластери на основі їхніх кольорових або текстурних характеристик. Метод *k*-середніх є найпопулярнішим підходом, що ітеративно розподіляє пікселі між *k* кластерами так, щоб мінімізувати внутрішньо-кластерну дисперсію.

Після сегментації наступним критичним етапом є вилучення ознак – процес перетворення вихідного зображення або його частин у компактний набір чисельних характеристик, що описують суттєві властивості об'єкта. Якість вилучених ознак безпосередньо впливає на успішність подальшої класифікації.

Колірні ознаки є одними з найпростіших, але часто дуже інформативними для діагностики захворювань рослин, оскільки багато патологій змінюють колір уражених тканин. Колірні гістограми представляють розподіл інтенсивності кольорових каналів та є інваріантними до масштабу та обертання об'єкта. Статистичні моменти, такі як середнє, дисперсія, асиметрія та ексцес кожного кольорового каналу надають компактне представлення кольорової інформації. Домінантні кольори, отримані шляхом кластеризації кольорового простору, ефективно описують основні кольорові характеристики.

Текстурні ознаки описують просторове розташування інтенсивностей пікселів та є особливо корисними для розрізнення типів поверхонь. Матриця співвустрічності рівнів сірого (GLCM) аналізує просторові взаємозв'язки між пікселями на різних відстанях та в різних напрямках. З GLCM обчислюються такі

характеристики як контрастність, однорідність, ентропія, кореляція, енергія. Локальні бінарні шаблони (LBP) кодують локальну текстуру навколо кожного пікселя, порівнюючи його значення з сусідніми, що забезпечує інваріантність до монотонних змін освітлення.

Геометричні ознаки описують форму об’єктів та їхніх компонентів. Площа та периметр є базовими характеристиками форми. Компактність, тобто співвідношення площі до квадрату периметру, вказує на «округлість» форми. Ексцентриситет описує ступінь подовженості об’єкта. Моменти Ху є набором з семи величин, інваріантних до масштабу, обертання та відображення, що дозволяють описувати складні форми.

Дескриптори ключових точок представляють більш складний підхід до вилучення ознак, що базується на виявленні характерних особливостей зображення.

SIFT (Scale-Invariant Feature Transform) виявляє та описує локальні ознаки зображення, інваріантні до масштабу, обертання та частково до змін освітлення та точки зору. Алгоритм складається з декількох етапів: виявлення екстремумів у просторі масштабів, локалізація ключових точок, визначення орієнтації, формування дескрипторів. SIFT дескриптори широко використовуються для зіставлення зображень, розпізнавання об’єктів та панорамної зшивки.

SURF (Speeded-Up Robust Features) є прискореною версією SIFT, що використовує інтегральні зображення та апроксимації фільтрів для більш швидкого обчислення, зберігаючи при цьому подібну якість дескрипторів.

ORB (Oriented FAST and Rotated BRIEF) поєднує детектор кутів FAST та бінарний дескриптор BRIEF з додаванням інваріантності до обертання. ORB є вільною альтернативою SIFT та SURF та характеризується високою швидкістю обчислення.

HOG (Histogram of Oriented Gradients) описує розподіл напрямків градієнтів у локальних областях зображення. Спочатку розроблений для виявлення пішоходів, HOG знайшов широке застосування в різних задачах розпізнавання об’єктів, включаючи аналіз зображень рослин.

Після вилучення ознак виконується власне класифікація – віднесення об’єкта до одного з заздалегідь визначених класів. Класичні методи машинного навчання пропонують різноманітні підходи до цієї задачі.

Метод опорних векторів (SVM – Support Vector Machine) знаходить оптимальну гіперплощину, що максимізує відстань між класами в багатовимірному просторі ознак. SVM особливо ефективний у високовимірних просторах та при роботі з невеликими наборами даних. Використання ядрових функцій дозволяє вирішувати нелінійні задачі класифікації без явного перетворення даних у простір вищої розмірності.

Метод *k*-найближчих сусідів класифікує об’єкт на основі класів k найближчих до нього навчальних прикладів у просторі ознак. Метод є непараметричним, не вимагає етапу навчання моделі, проте має високу обчислювальну складність на етапі класифікації та чутливий до вибору метрики відстані та значення *k*.

Дерева рішень будують ієрархічну структуру правил класифікації, послідовно розділяючи простір ознак на основі порогових значень окремих ознак. Алгоритми ID3, C4.5 та CART використовують різні критерії для вибору оптимальних розбиттів: інформаційний приріст, коефіцієнт приросту та індекс Джині. Дерева рішень є інтерпретованими, але схильні до перенавчання.

Випадковий ліс (Random Forest) є ансамблевим методом, що об’єднує множину дерев рішень, кожне з яких навчається на випадковій підвибірці даних та підмножині ознак. Фінальна класифікація визначається голосуванням усіх дерев. Випадковий ліс зменшує проблему перенавчання окремих дерев та зазвичай демонструє вищу точність.

Наївний байєсівський класифікатор базується на теоремі Байєса з припущенням незалежності ознак. Незважаючи на нереалістичність цього припущення для більшості реальних задач, метод часто показує добрі результати, особливо при обмеженій кількості навчальних даних.

Логістична регресія, всупереч назві, є методом класифікації, що моделює ймовірність належності до класу за допомогою логістичної функції. Метод є простим, інтерпретованим та ефективним для лінійно роздільних класів.

1.5 Постановка задачі дослідження

Проведений аналіз показав, що класичні методи комп'ютерного зору для діагностики захворювань рослин мають суттєві обмеження при практичному застосуванні, а саме необхідність ручного проєктування ознак, обмежену здатність до узагальнення, складність роботи з багатокласовими задачами та високу залежність від якості попередньої обробки зображень. Необхідність ручного проєктування ознак вимагає глибокого експертного знання предметної області та методів обробки зображень, що робить процес трудомістким та займає багато часу. Обмежена здатність до узагальнення проявляється в тому, що ознаки, ефективні для одного набору даних чи умов зйомки, можуть погано працювати при зміні освітлення, фону, кута зйомки чи пори року. Крім того, складність роботи з багатокласовими задачами зростає експоненційно зі збільшенням кількості класів захворювань, а втрата інформації на етапі вилучення ознак є неминучою, оскільки компактне представлення не може повністю зберегти всю інформацію вихідного зображення. Відсутність ієрархічного представлення означає, що класичні методи зазвичай працюють з плоским набором ознак, не враховуючи природну ієрархічну структуру візуальної інформації, а залежність від якості сегментації критично впливає на фінальну точність класифікації.

Основна проблема полягає у необхідності швидкої, точної та економічно ефективної системи автоматизованого розпізнавання захворювань рослин, яка могла б адаптуватися до різних умов зйомки та автоматично навчатися представленням даних безпосередньо з вихідних зображень без необхідності ручного проєктування ознак.

Об’єктом дослідження є зображення рослин з різними захворюваннями та симптомами ураження.

Метою дослідження є розробка та реалізація методу автоматизованого розпізнавання зображень для діагностики захворювань рослин на основі застосування згорткових нейронних мереж для підвищення точності діагностики та забезпечення ефективного моніторингу стану рослин.

Для досягнення мети необхідно вирішити наступні завдання:

– підготувати набір даних для класифікації захворювань рослин на основі набору зображень з розділенням на навчальну, валідаційну та тестову вибірки;

– реалізувати метод розпізнавання захворювань за допомогою згорткових нейронних мереж для класифікації стану рослин;

– провести експериментальне дослідження методу з оцінкою метрик якості та швидкості роботи;

– розробити програмне забезпечення для практичного застосування методу діагностики захворювань рослин;

– сформулювати практичні рекомендації щодо застосування розробленої системи для моніторингу стану рослин у різних сценаріях використання.

Очікуваним результатом є функціонуючий прототип системи розпізнавання захворювань рослин та розроблене програмне забезпечення з рекомендаціями щодо практичного застосування.

2 ВИБІР МЕТОДУ РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ ДЛЯ КЛАСИФІКАЦІЇ РОСЛИН

2.1 Нейронні мережі та глибоке навчання

Глибоке навчання є підмножиною машинного навчання, що базується на штучних нейронних мережах з багатьма шарами обробки. Протягом останнього десятиліття глибоке навчання радикально змінило ландшафт комп'ютерного зору, досягаючи рівня точності, що іноді перевершує людські можливості у специфічних задачах розпізнавання образів.

Ідея штучних нейронних мереж не є новою – вона виникла ще у 1940-х роках з роботи Маккалока та Піттса, які запропонували математичну модель нейрона. Перший персептрон, створений Розенблаттом у 1958 році, був здатний навчатися лінійній класифікації. Однак обмеження простих персептронів, продемонстровані Мінським та Пейпертом у 1969 році, призвели до першої «зими штучного інтелекту».

Важливим проривом стала розробка алгоритму зворотного поширення помилки у 1980-х роках, що дозволило ефективно навчати багатошарові мережі. Однак через обмеження обчислювальних потужностей та недостатню кількість даних нейронні мережі залишалися здебільшого теоретичним інструментом.

Справжня революція глибокого навчання почалася у 2012 році, коли згорткова нейронна мережа AlexNet, розроблена Алексом Крижевським та його колегами, вражаюче перемогла у змаганні ImageNet з класифікації зображень, знизивши показник помилок майже вдвічі порівняно з класичними методами. Цей успіх став можливим завдяки збігу кількох факторів: доступності великих наборів даних (ImageNet містив мільйони розмічених зображень), можливості використання графічних процесорів для прискорення обчислень, розробці нових методів регуляризації та функцій активації.

2.1.1 Базові принципи нейронних мереж

Штучна нейронна мережа складається з взаємопов'язаних вузлів – нейронів, організованих у шари. Найпростіша архітектура включає вхідний шар, один або більше прихованих шарів та вихідний шар.

Штучний нейрон отримує множину вхідних сигналів, кожен з яких множиться на відповідну вагу. Зважена сума входів додається до зміщення, а результат пропускається через функцію активації, що визначає вихідний сигнал нейрона. Математично це можна записати як:

$$y = f(\Sigma(w_i * x_i) + b, \tag{2.1}$$

де x_i – вхідні сигнали;

w_i – ваги;

b – зміщення;

f – функція активації.

Функції активації вносять нелінійність у мережу, що є критично важливим для моделювання складних залежностей. Сигмоїдна функція та гіперболічний тангенс були популярні на ранніх етапах, але мають проблему зникаючого градієнта при глибоких мережах. ReLU (Rectified Linear Unit), що обчислюється як:

$$f(x) = max(0, x), \tag{2.2}$$

стала стандартом завдяки простоті обчислення, відсутності насичення для додатних значень та здатності сприяти розрідженості активацій. Варіації ReLU, такі як Leaky ReLU, Parametric ReLU, ELU, вирішують проблему «мертвих» нейронів, коли градієнт дорівнює нулю для від'ємних входів.

Прямий прохід – це процес обчислення виходу мережі для заданого входу, де сигнал послідовно проходить через усі шари від входу до виходу. Зворотне

поширення помилки – це алгоритм навчання, що обчислює градієнти функції втрат відносно ваг мережі, використовуючи правило ланцюгового диференціювання для ефективного розповсюдження помилки від виходу до входу.

Оптимізація ваг виконується ітеративно за допомогою градієнтних методів. Стохастичний градієнтний спуск (SGD) оновлює ваги після обробки кожного прикладу або міні-батчу прикладів. Сучасні оптимізатори, такі як Adam, RMSprop або AdaGrad використовують адаптивні швидкості навчання для різних параметрів, що прискорює збіжність та покращує якість навчання.

2.1.2 Проблеми навчання глибоких мереж

Навчання глибоких нейронних мереж пов’язане з низкою специфічних проблем, для вирішення яких розроблено різноманітні техніки.

Перенавчання виникає, коли модель занадто добре підлаштовується під навчальні дані, запам’ятовуючи їх замість вивчення загальних закономірностей, що призводить до поганої генералізації на нових даних. Для боротьби з перенавчанням використовуються різні методи регуляризації.

L1 та L2 регуляризації додають до функції втрат штраф за великі значення ваг, стимулюючи мережу використовувати більш прості моделі. L1 регуляризація сприяє розрідженості ваг. L2 регуляризація більш широко використовується та відома як «weight decay», або «зниження ваги».

Dropout – це техніка, запропонована Хінтоном у 2012 році, що під час навчання випадково «вимикає» певну частку нейронів, зазвичай 20 – 50%, змушуючи мережу вчитися більш стійким представленням, які не залежать від присутності конкретних нейронів. Під час тестування використовуються всі нейрони, але їхні виходи масштабуються відповідно до ймовірності dropout.

Збільшення даних штучно розширює навчальний набір шляхом застосування різноманітних перетворень до вхідних зображень: обертання,

зсуви, масштабування, відображення, зміни яскравості та контрасту, додавання шуму. Це допомагає моделі навчитися інваріантності до цих перетворень та покращує генералізацію.

Рання зупинка припиняє навчання, коли продуктивність на підтверджувальному наборі даних перестає покращуватися, запобігаючи подальшому перенавчанню на тренувальних даних.

Батч-нормалізація, запропонована Іоффе та Сегеді у 2015 році, нормалізує входи кожного шару до стандартного розподілу (нульове середнє, одинична дисперсія) в межах міні-батчу. Це стабілізує процес навчання, дозволяє використовувати вищі швидкості навчання, зменшує чутливість до ініціалізації ваг та має регуляризуючий ефект.

Зникаючий та вибухаючий градієнт є проблемами глибоких мереж, де градієнти можуть ставати надто малими або надто великими при поширенні через багато шарів. Зникаючий градієнт особливо проблематичний для сигмоїдних функцій активації, оскільки їхні похідні малі для великих за модулем значень. Використання ReLU активації, батч-нормалізації та залишкових з’єднань допомагає пом’якшити ці проблеми.

2.1.3 Згорткові операції

Згорткові нейронні мережі є спеціалізованим типом нейронних мереж, розробленим спеціально для обробки даних з решітчастою топологією, зокрема зображень. Ключовою операцією є згортка – застосування навчального фільтра до вхідних даних.

Згорткова операція математично визначається як обчислення скалярного добутку фільтра з локальною областю входу у кожній позиції. Фільтр «ковзає» по входу з певним кроком, обчислюючи відгук у кожній позиції. Результатом є карта ознак, що представляє присутність певної характеристики (краю, текстури тощо) в різних частинах зображення.

Ключові переваги згорткових операцій включають локальну зв’язність, коли кожен нейрон з’єднаний лише з невеликою локальною областю попереднього шару, спільне використання параметрів (один і той самий фільтр застосовується до всіх позицій, що драматично зменшує кількість параметрів порівняно з повнозв’язними шарами), трансляційну інваріантність (здатність виявляти ознаки незалежно від їхньої позиції в зображенні).

Згортковий шар зазвичай застосовує не один, а множину фільтрів, кожен з яких виявляє різні типи ознак. Глибина вихідного тензора дорівнює кількості фільтрів. У перших шарах мережі фільтри зазвичай виявляють прості ознаки (краї різних орієнтацій, кольорові переходи), у глибших шарах – більш складні та абстрактні паттерни.

Паддінг – додавання нулів або інших значень по краях входу, використовується для контролю просторових розмірів виходу та збереження інформації з країв зображення. Valid padding зменшує просторові розміри, same padding зберігає їх.

Підвибірка зменшує просторові розміри карт ознак, забезпечуючи трансляційну інваріантність та зменшуючи кількість параметрів і обчислень у наступних шарах. Max pooling вибирає максимальне значення в локальній області, average pooling обчислює середнє. Max pooling є більш поширеним, оскільки зберігає найсильніші активації та забезпечує стійкість до невеликих зсувів.

2.1.4 Архітектура типової CNN

Типова згорткова нейронна мережа складається з чергування згорткових шарів, шарів активації та шарів підвибірки, за якими слідують один або кілька повнозв’язних шарів для фінальної класифікації.

Згорткові блоки зазвичай включають послідовність: згортковий шар, батч-нормалізація, активація (ReLU), можливо ще згортковий шар, підвибірка.

Такі блоки повторюються кілька разів, з поступовим збільшенням кількості фільтрів та зменшенням просторових розмірів.

Повнозв’язні шари у кінці мережі виконують фінальну класифікацію на основі ознак, вилучених згортковими шарами [20]. Вихідний шар має кількість нейронів, що дорівнює кількості класів, з softmax активацією для отримання розподілу ймовірностей по класах [21].

Global Average Pooling (GAP) є альтернативою повнозв’язним шарам, що обчислює середнє значення кожної карти ознак перед фінальним класифікаційним шаром. Це значно зменшує кількість параметрів та зменшує ризик перенавчання.

Навчання нейронної мережі вимагає визначення функції втрат, що кількісно оцінює розбіжність між передбаченнями моделі та справжніми мітками [22, 23].

Крос-ентропія є стандартною функцією втрат для задач класифікації. Для бінарної класифікації використовується бінарна крос-ентропія, для багатокласової – категоріальна крос-ентропія. Функція штрафує модель пропорційно до «впевненості» неправильних передбачень.

Фокальна втрата, запропонована для задач розпізнавання об’єктів, модифікує крос-ентропію, зменшуючи вагу простих, легко класифікованих прикладів та зосереджуючи навчання на складних випадках. Це особливо корисно при сильному дисбалансі класів [24, 25].

2.1.5 Метрики оцінювання

Для оцінки якості моделі класифікації використовуються різноманітні метрики.

Точність – частка правильно класифікованих прикладів – є найпростішою метрикою, але може бути оманливою при дисбалансі класів.

Precision вимірює частку справді позитивних серед усіх передбачених як позитивні. Recall (повнота, чутливість) вимірює частку справді позитивних, що були правильно ідентифіковані. F1-score є гармонічним середнім precision та recall, забезпечуючи баланс між ними.

Матриця плутанини показує детальний розподіл правильних та неправильних класифікацій для кожного класу, дозволяючи виявити, які класи найчастіше плутаються між собою.

ROC-крива (Receiver Operating Characteristic) та площа під нею, оцінюють якість класифікатора для різних порогових значень, що особливо корисно при дисбалансі класів або коли вартість різних типів помилок відрізняється [26 – 28].

Глибоке навчання, особливо згорткові нейронні мережі, продемонструвало вражаючі результати в комп'ютерному зорі та продовжує активно розвиватися, пропонуючи нові архітектури та методи навчання для вирішення все більш складних задач.

2.2 Архітектури згорткових нейронних мереж (CNN)

Розвиток згорткових нейронних мереж призвів до створення численних архітектур, кожна з яких внесла унікальні ідеї та покращення. Розуміння еволюції цих архітектур та їхніх ключових особливостей є важливим для обґрунтованого вибору моделі для конкретної задачі діагностики захворювань рослин.

2.2.1 Архітектура LeNet

LeNet-5, розроблена Яном ЛеКуном у 1998 році, стала першою успішною згортковою нейронною мережею для практичного застосування. Створена для

розпізнавання рукописних цифр у системах автоматичного читання чеків, LeNet продемонструвала ключові принципи CNN:

– використання згорткових шарів для автоматичного вилучення ознак;
– підвибірка для зменшення розмірності;
– чергування згорткових та підвибіркових шарів.

Архітектура LeNet-5 включала 7 шарів (не рахуючи вхідний): два блоки згортка-підвибірка, за якими слідували три повнозв'язні шари [29, 30]. Мережа мала приблизно 60000 параметрів. Вхідне зображення розміром 32×32 пікселів послідовно обробляється згортковими шарами з фільтрами 5×5, після чого застосовується підвибірка 2×2. Хоча LeNet була успішною для своєї задачі, обмеження обчислювальних ресурсів та відсутність великих наборів даних стримували її застосування до більш складних задач комп'ютерного зору.

2.2.2 Архітектура AlexNet

Архітектура AlexNet, представлена Алексом Крижевським, Ільєю Суцкевером та Джеффрі Хінтоном у 2012 році, стала переломним моментом у розвитку глибокого навчання для комп'ютерного зору. Мережа перемогла у змаганні ImageNet Large Scale Visual Recognition Challenge 2012, знизивши показник помилок з 26% до 15,3%.

Архітектура AlexNet складалася з 8 шарів: 5 згорткових та 3 повнозв'язних, з загальною кількістю 60 мільйонів параметрів. Математично, згортковий шар обчислює вихід як:

$$y_{\mathrm{ij}}{}^{\mathrm{k}} = f(\Sigma_c * \Sigma_m * \Sigma_n * \ w_{\mathrm{mn}}{}^{\mathrm{kc}} * \ x_{(\mathrm{i+m})(\mathrm{j+n})}{}^{\mathrm{c}} \ + \ b^{\mathrm{k}}), \qquad (2.3)$$

де $y_{ij}{}^{k}$ – вихід k-го фільтра в позиції;

x^{c} – вхідна карта ознак каналу c;

$w_{mn}{}^{kc}$ – ваги фільтра;

b^k – зміщення;

f – функція активації;

$m*n$ – розмір фільтра.

Ключові інновації включали використання ReLU активації (формула (2.2)) замість tanh або сигмоїди, що значно прискорило навчання; застосування dropout з коефіцієнтом p=0,5 у повнозв'язних шарах для боротьби з перенавчанням; data augmentation для збільшення різноманітності навчальних даних; використання двох GPU для паралелізації обчислень.

2.2.3 Архітектура VGGNet

Архітектура VGGNet, розроблена дослідницькою групою Visual Geometry Group Оксфордського університету у 2014 році, систематично дослідила вплив глибини мережі на точність розпізнавання. Карен Сімоньян та Ендрю Зіссерман запропонували дві основні версії: VGG16 (16 шарів з навчуваними параметрами) та VGG19 (19 шарів).

Ключовою особливістю VGG була простота та регулярність архітектури. Мережа використовувала виключно малі згорткові фільтри розміром 3×3 з кроком s=1. Рецептивне поле R послідовності згорткових шарів обчислюється як:

$$R = 1 + \Sigma_{i=1}^{L} (k_i - 1) \cdot \prod_{j=1}^{i-1} s_j, \quad (2.4)$$

де L – кількість шарів;

k_i – розмір фільтра i-го шару;

s_j – крок j-го шару.

Для VGG із L=2 шарами 3×3 із s=1:

$$R = 1 + (3-1) \cdot 1 + (3-1) \cdot 1 = 5,$$

що еквівалентно одному шару 5×5, але з меншою кількістю параметрів: $18C^2$ проти $25C^2$, де C – кількість каналів.

Архітектура організована у блоки згорткових шарів, після кожного з яких йде max pooling з вікном 2×2 та кроком 2, що зменшує просторові розміри за формулою:

$$H_{\text{out}} = \lfloor (H_{\text{in}} - k)/s \rfloor + 1 = \lfloor (H_{\text{in}} - 2)/2 \rfloor + 1 = H_{\text{in}}/2. \quad (2.5)$$

Кількість фільтрів подвоюється після кожного pooling: 64, 128, 256, 512. VGG16 містить близько 138 мільйонів параметрів[31, 32]. Незважаючи на це, VGG залишається популярною завдяки простоті та ефективності як feature extractor для Transfer Learning.

2.2.4 Архітектура GoogLeNet

Архітектура GoogLeNet, розроблена командою Google у 2014 році, виграла ILSVRC 2014, запровадивши концепцію Inception-модулів [33]. Мережа досягла точності, порівнянної з VGG, але з лише 7 мільйонами параметрів.

Inception-модуль застосовує згортки різних розмірів паралельно до того самого входу:

$$Output = Concat([Conv_{1x1}(X), Conv_{3x3}(X), Conv_{5x5}(X), Pool(X)]),$$

де *Concat* – операція конкатенації по осі каналів.

Для зменшення обчислювальної складності використовуються 1×1 згортки перед дорогими операціями.

Кількість операцій для 3×3 згортки без вузького місця:

$$3 \times 3 \times C_{\text{in}} \times C_{\text{out}} \times H \times W.$$

Кількість операцій для 3×3 згортки з вузьким місцем:

$$(1\times1\times C_{in}\times C_{re} + 3\times3\times C_{re}\times C_{out})\times H\times W,$$

де $C_{re} << C_{in}$, що дає значну економію при $C_{re} = C_{in}/4$.

Архітектура GoogLeNet складається з 22 шарів та містить дев'ять Inceptionмодулів [34]. Замість великих повнозв'язних шарів використовується global average pooling, який для кожної карти ознак розміром H×W обчислює:

$$GAP = \left(\frac{1}{H\times W}\right) * \Sigma_{i=1}^{H} \Sigma_{j=1}^{W} x_{ij}.$$

Це зменшує кількість параметрів з $H\times W\times C\times K$ до $C\times K$, де K – кількість класів.

2.2.5 Архітектура ResNet

Архітектура ResNet (Residual Network), представлена Кайміном Хе та його колегами з Microsoft Research у 2015 році, виграла ILSVRC 2015 з помилкою 3,57%, що краще за середню людську продуктивність, яка сягає приблизно 5%.

Головною інновацією архітектури ResNet є залишкові зв'язки, які реалізують відображення:

$$H(x) = F(x) + x, \tag{2.6}$$

де $F(x)$ – залишкова функція, яку навчає мережа;

x – вхід блоку;

$H(x)$ – вихід блоку.

Замість прямого навчання відображення

$$H(x): x \rightarrow y,$$

мережа навчає залишок

$$F(x) = H(x) - x.$$

Якщо оптимальне відображення близьке до ідентичності, легше навчити F(x) → 0, ніж навчати безпосередньо H(x) → x.

Градієнт для residual блоку обчислюється як:

$$\partial L/\partial x = \partial L/\partial H \cdot \partial H/\partial x = \partial L/\partial H \cdot (\partial F/\partial x + I), \quad (2.7)$$

де I – одинична матриця.

Доданок I забезпечує пряме проходження градієнта, запобігаючи його зникненню.

ResNet існує у декількох варіаціях: ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-152 [35, 36]. Глибші версії використовують bottleneck блоки з трьома шарами (1×1, 3×3, 1×1), що зменшує кількість параметрів: Параметри regular block:

$$2 \times (3 \times 3 \times C \times C) = 18C^2.$$

Параметри bottleneck:

$$1 \times 1 \times C \times \left(\frac{C}{4}\right) + 3 \times 3 \times \left(\frac{C}{4}\right) \times \left(\frac{C}{4}\right) + 1 \times 1 \times \left(\frac{C}{4}\right) \times C =$$
$$= C^2/2 + 9C^2/16 + C^2/4 \approx 1.69C^2.$$

2.2.6 Архітектура DenseNet

Архітектура DenseNet (Densely Connected Network), запропонована Хуаном Лю у 2017 році, реалізує максимальну зв'язність між шарами. В архітектурі DenseNet кожен шар отримує вхідні дані від усіх попередніх шарів [37]:

$$x_l = H_l([x_0, x_1, \ldots, x_{l-1}]), \tag{2.8}$$

де $[x_0, x_1, \ldots, x_{l-1}]$ – конкатенація карт ознак усіх попередніх шарів;

H_l – композитна функція (BatchNorm, ReLU, Conv).

У dense-блоці з L шарами існує $L(L+1)/2$ з'єднань. Кількість вхідних каналів для l-го шару:

$$C_l^{\text{in}} = C_0 + k \times (l - 1), \tag{2.9}$$

де C_0 – кількість вхідних каналів у блок;

k – growth rate (приріст каналів на шар, зазвичай k = 12 – 32).

2.2.7 Архітектура MobileNet

Архітектура MobileNet, розроблена командою Google у 2017 році, оптимізована для мобільних пристроїв через використання глибинно роздільної згортки (depthwise separable convolutions).

Стандартна згортка обчислює:

$$Y = \Sigma\_\{c = 1\}\^\{C_{\text{in}}\} Kc * Xc, \tag{2.10}$$

де Kc – фільтр для каналу c;

* – операція згортки.

Кількість операцій:

$$Ops_{\text{stanar}} = k \times k \times C_{\text{in}} \times C_{\text{out}} \times H \times W.$$

Глибинно роздільна згортка розбивається на два етапи.

Етап 1. Depthwise convolution (окремо для кожного каналу):

$$Y^{\text{cDW}} = Kc * Xc,$$

для $c = 1$,

$$C_{\text{in}}.Ops^{\text{DW}} = k \times k \times C_{\text{in}} \times H \times W.$$

Етап 2. Pointwise convolution (1×1 згортка для комбінування):

$$Y^{\text{PW}} = \Sigma\_\{c = 1\}\hat{}\{C_{\text{in}}\}\, K^{\text{c1x1}} \cdot Y^{\text{cDW}}.$$

$$Ops^{\text{PW}} = C_{\text{in}} \times C_{\text{out}} \times H \times W.$$

Загальна кількість операцій:

$$Ops_{\text{separaβle}} = k \times k \times C_{\text{in}} \times H \times W + C_{\text{in}} \times C_{\text{out}} \times H \times W.$$

Відношення обчислювальних витрат:

$$Ops_{\text{separaβle}}/Ops_{\text{stanar}} = 1/C_{\text{out}} + 1/k^2.$$

Для k=3, C_{out}=128:

$$1/128 + 1/9 \approx 0{,}119,$$

тобто економія близько 88%.

Архітектура MobileNet використовує два параметри: width multiplier $\alpha \in (0,1]$ та resolution multiplier $\rho \in (0,1]$. Кількість каналів та роздільна здатність масштабуються:

$$C' = \alpha \times C, H' = \rho \times H, W' = \rho \times W.$$

Обчислювальна складність:

$$Ops = \alpha^2 \rho^2 \times Ops_{\text{base}}.$$

2.2.8 Архітектура EfficientNet

Архітектура EfficientNet, запропонована командою Google Brain у 2019 році, представляє систематичний підхід до масштабування архітектури.

Compound scaling method використовує єдиний коефіцієнт φ для масштабування трьох вимірів: глибина, ширина та роздільна здатність (depth, width та resolution):

$$d = \alpha^{\varphi},$$

$$w = \beta^{\varphi},$$

$$r = \gamma^{\varphi},$$

з обмеженням $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$, що забезпечує приблизно подвоєння обчислювальних витрат при $\varphi = \varphi + 1$.

Для базової моделі (φ=0) через grid search знайдено α=1,2, β=1,1, γ=1,15. Обчислювальна складність:

$$FLOPS \approx (\alpha \cdot \beta^2 \cdot \gamma^2)^{\varphi} \times FLOPS_{\beta ase} \approx 2^{\varphi} \times FLOPS_{\beta ase}.$$

Кількість параметрів:

$$Params \approx (\alpha \cdot \beta^2)^{\varphi} \times Params_{\beta ase}.$$

Це призводить до сімейства моделей EfficientNet-B0 до B7, де кожна наступна модель масштабується збалансовано. EfficientNet-B0 має 5,3M параметрів та 0,39B FLOPS, тоді як B7 має 66M параметрів та 37B FLOPS.

2.3 Вибір архітектури для діагностики захворювань рослин

Вибір архітектури залежить від обмежень та вимог задачі. Для оцінки було враховано наступні критерії:

– обсяг даних (більші моделі, такі як ResNet-101 та EfficientNet-B7, вимагають більше даних);

– обчислювальні ресурси (FLOPS та кількість параметрів визначають вимоги до пам’яті та швидкості);

– точність (складніші моделі зазвичай досягають вищої точності);

– deployment (мобільні застосунки потребують легких моделей, наприклад, MobileNet);

– передавальне навчання (попередньо навчені моделі на ImageNet ефективні при обмежених даних).

Для діагностики захворювань рослин рекомендується: EfficientNet B0 – B4 для оптимального балансу, MobileNetV2 для мобільних застосунків, ResNet-50 для стандартних задач з передавальним навчанням (transfer learning), DenseNet-121 для високої точності при обмеженій пам’яті.

Проведений комплексний огляд методів розпізнавання зображень виявив принципові відмінності між класичними підходами комп’ютерного зору та сучасними методами глибокого навчання. Класичні методи, що базуються на послідовному конвеєрі обробки з ручним проєктуванням ознак та традиційними алгоритмами класифікації, досягають точності 70 – 92% на контрольованих датасетах, однак їхня продуктивність критично погіршується в реальних умовах – падіння на 25 – 40% при зміні освітлення або фону. Це обмежує їхню практичну застосовність для польової діагностики захворювань рослин.

Згорткові нейронні мережі демонструють революційний прогрес завдяки здатності автоматично навчатися ієрархічним представленням безпосередньо з вихідних зображень. Еволюція архітектур від AlexNet через VGG, GoogLeNet, ResNet до сучасних EfficientNet та MobileNet показує постійне вдосконалення балансу між точністю, обчислювальною ефективністю та стійкістю до варіацій умов [38 – 42]. Порівняльний аналіз показав, що сучасні CNN з Transfer Learning досягають 96 – 99% точності на лабораторних датасетах та 88 – 95% у польових умовах при падінні точності лише на 5 – 12% при зміні умов зйомки.

На основі проведеного аналізу встановлено, що для задач діагностики захворювань рослин оптимальним вибором є архітектури сімейства EfficientNet (B0 – B3) або ResNet-50 з Transfer Learning на ImageNet та інтенсивною data augmentation, які забезпечують найкращий баланс точності та обчислювальної ефективності. Для мобільних застосунків рекомендується MobileNetV2 з обчислювальною складністю 0,3 GFLOPS та часом інференсу 25 – 50 мс на CPU. Ці висновки визначають архітектурні рішення для проєктування та реалізації програмного забезпечення в наступних розділах роботи.

3 ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Вибір технологій та інструментів розробки

Для реалізації системи діагностики захворювань рослин була обрана мова програмування Python версії 3.10. Python є оптимальним рішенням для задач машинного навчання та комп'ютерного зору.

По-перше, Python володіє найбільш розвиненою екосистемою бібліотек для машинного навчання. Такі фреймворки як TensorFlow, PyTorch, Keras забезпечують повний цикл розробки моделей від прототипування до продакшн-розгортання. По-друге, синтаксична простота мови дозволяє зосередитися на алгоритмічній частині замість боротьби з технічними деталями компіляції та управління пам'яттю.

По-третє, велика спільнота розробників забезпечує широкий спектр успішних практик та документації.

Для побудови нейронної мережі було обрано бібліотеку TensorFlow 2.*x* з високорівневим API Keras. TensorFlow є однією з найпопулярніших платформ для глибокого навчання, розроблена компанією Google Brain. Keras, інтегрований у TensorFlow як tf.keras, надає зручний інтерфейс для швидкого прототипування моделей.

Основні переваги TensorFlow/Keras включають підтримку різних апаратних прискорювачів (GPU через CUDA та DirectML, TPU), можливість розгортання моделей на різних платформах, ефективні інструменти для візуалізації процесу навчання, а також вбудовані засоби для аугментації даних та попередньої обробки зображень.

Keras API дозволяє будувати моделі послідовно, додаючи шари один за одним, що робить код читабельним та легким для модифікації. Бібліотека також надає готові реалізації популярних архітектур згорткових мереж, що значно прискорює розробку.

Як інтегроване середовище розробки було обрано PyCharm від компанії JetBrains. PyCharm забезпечує потужні засоби для розробки Python-додатків: вбудований відладчик, інтеграцію з системами контролю версій, можливість роботи з віртуальними середовищами та інструменти для профілювання продуктивності.

Для управління залежностями проєкту використовувався менеджер пакетів pip та віртуальне середовище venv, що дозволяє ізолювати бібліотеки проєкту від системних пакетів Python. Для прискорення обчислень на етапі навчання моделі використовувався графічний процесор NVIDIA GeForce RTX 3050 Ti Laptop з підтримкою DirectML.

Додатково використовувалися бібліотеки NumPy для роботи з багатовимірними масивами, Matplotlib для побудови графіків процесу навчання, та Pillow для попередньої обробки зображень. Для зберігання моделі використовувався формат HDF5 (.h5), що забезпечує ефективне зберігання великих наборів даних та метаданих моделі.

3.2 Архітектура програмного забезпечення

3.2.1 Загальна структура системи

Розроблене програмне забезпечення має модульну архітектуру, що складається з трьох основних компонентів: модуля підготовки та завантаження даних, модуля навчання нейронної мережі та модуля інференсу для класифікації нових зображень (рис. 3.1). Така архітектура забезпечує гнучкість, можливість незалежного тестування компонентів та простоту модифікації окремих частин системи.

Модуль підготовки даних відповідає за завантаження зображень з файлової системи, їх нормалізацію, аугментацію та формування батчів для навчання. Модуль навчання реалізує побудову архітектури згорткової нейронної мережі, її компіляцію з вибраними оптимізатором та функцією втрат, а також

безпосередньо процес навчання з валідацією. Модуль інференсу забезпечує завантаження навченої моделі та класифікацію нових зображень з поверненням ймовірностей для кожного класу захворювань.

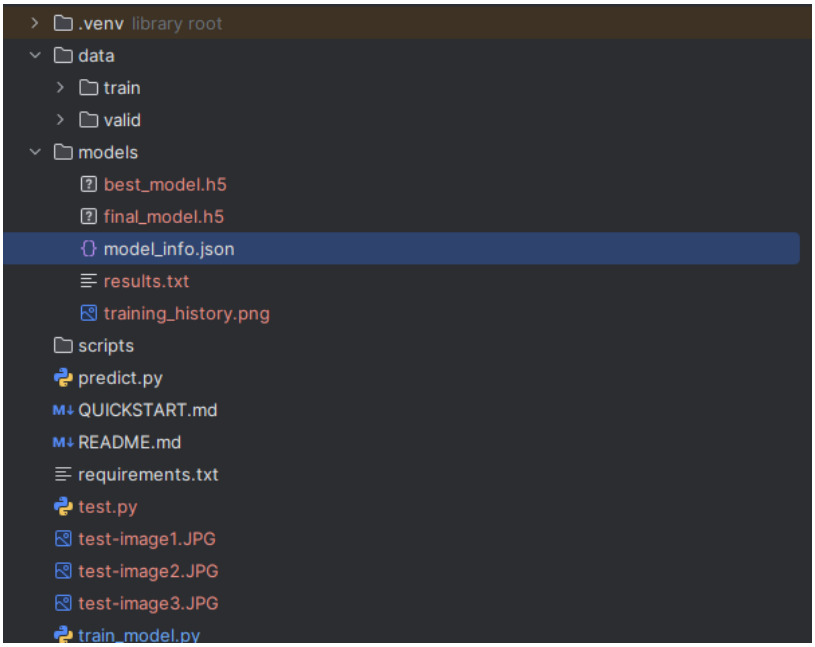

Рисунок 3.1 – Структура проєкту

3.2.2 Опис модулів програми

Програмне забезпечення організовано у вигляді двох основних скриптів: train_model.py та predict.py. Скрипт train_model.py містить функції для конфігурації GPU, підготовки генераторів даних, побудови моделі, налаштування callback-функцій та збереження результатів навчання.

Конфігураційні параметри, такі як розмір зображення, розмір батчу, кількість епох, швидкість навчання, винесені на початок скрипту для зручності експериментування. Це дозволяє швидко змінювати гіперпараметри без необхідності модифікації основного коду. Результати навчання зберігаються у директорії models у форматі HDF5 разом з метаданими у JSON-форматі та графіками у форматі PNG.

Потік обробки даних починається з організації датасету у структуру директорій, де кожна піддиректорія відповідає окремому класу захворювання. ImageDataGenerator з бібліотеки Keras автоматично сканує цю структуру та створює відповідність між зображеннями та їх мітками. Зображення завантажуються батчами, що дозволяє ефективно використовувати пам’ять при роботі з великими датасетами.

При завантаженні кожне зображення масштабується до розміру 224×224 пікселі та нормалізується діленням значень пікселів на 255, щоб перевести їх у діапазон [0, 1]. Генератор даних також виконує випадкову аугментацію: повороти на кут до 20 градусів, зміщення по горизонталі та вертикалі до 20%, масштабування до 20%, горизонтальне відзеркалення. Це штучно збільшує різноманітність тренувальних даних та покращує здатність моделі до узагальнення.

3.3 Реалізація модуля підготовки даних

3.3.1 Структура датасету

Датасет PlantVillage організований у ієрархічну структуру директорій: кореневі директорії train та valid містять піддиректорії для кожного з 10 класів захворювань різних рослин. Загальна кількість зображень у тренувальному наборі становить 18345 зображень, у валідаційному – 4585 зображень, що забезпечує співвідношення приблизно 80/20.

Розподіл зображень по класах є відносно збалансованим: кожен клас містить від 1702 до 1961 зображень у тренувальному наборі. Найменшу кількість зображень має клас бактеріальної плями – 1702, найбільшу – клас вірусу скручування листків – 1961. Такий збалансований розподіл важливий для уникнення зміщення моделі у бік більш представлених класів.

Для пришвидшення етапу навчання моделі було вирішено взяти лише захворювання одного типу рослин – томатів.

У датасеті наявні як здорові, так і хворі листя рослин. Зображення здорових листків характеризуються рівномірним зеленим забарвленням, відсутністю плям, некротичних ділянок та деформацій (рис. 3.2).

Рисунок 3.2 – Приклад зображення здорового листя томату

Зображення хворих листків, уражених бактеріальними захворюваннями, демонструють характерні темно-коричневі або чорні плями з жовтим ореолом, некротичні ділянки неправильної форми, водянисті ділянки (рис. 3.3). Алгоритм має виявити ці патологічні ознаки шляхом аналізу текстурних особливостей та кольорових характеристик RGB-каналів.

Рисунок 3.3 – Листя томату уражене бактерією

Листки з вірусними захворюваннями відрізняються мозаїчним малюнком, чергуванням світлих і темних ділянок, деформацією листкової пластини та зміною пігментації (рис. 3.4). Для вірусних уражень характерні системні зміни: порушення симетрії та морфологічної структури листя. Очікується, що розроблена модель ідентифікує ці аномалії за геометричними та колірними ознаками для точної діагностики вірусної інфекції та їх відмінності від бактеріальних уражень.

Рисунок 3.4 – Ураження листя вірусом

3.3.2 Аугментація зображень

Аугментація даних є критично важливою технікою для підвищення якості навчання моделі, особливо при обмеженій кількості тренувальних зображень. У даному проєкті застосовувалися наступні види аугментації: геометричні перетворення, зміна яскравості та контрасту, горизонтальне відзеркалення.

Геометричні перетворення включають випадковий поворот зображення на кут до 20 градусів, що імітує різні кути фотографування листків. Зміщення по горизонталі та вертикалі до 20% імітує неточність кадрування. Масштабування на ±20% моделює різні відстані від камери до об’єкта. Горизонтальне

відзеркалення подвоює кількість унікальних комбінацій, оскільки симетрія рослини природна.

Важливо відзначити, що аугментація застосовується виключно до тренувального набору даних. Валідаційний набір залишається без змін, щоб забезпечити об’єктивну оцінку продуктивності моделі на реальних, неаугментованих даних.

3.3.3 Генератори даних для навчання

Для ефективного завантаження даних використовується клас ImageDataGenerator з бібліотеки Keras. Цей клас реалізує паттерн генератора Python, що дозволяє завантажувати та обробляти зображення батчами безпосередньо під час навчання, замість завантаження всього датасету в оперативну пам’ять одразу.

Для тренувального набору створюється генератор з налаштованою аугментацією та нормалізацією пікселів. Розмір батчу встановлено 32 зображення, що є балансом між швидкістю навчання та стабільністю градієнтного спуску. Для валідаційного набору створюється окремий генератор без аугментації, але з тією ж нормалізацією.

Лістинг 3.1 Використання класу ImageDataGenerator:

```
def create_data_generators():
    train_datagen = ImageDataGenerator(
        rescale=1./255,
        rotation_range=20,
        width_shift_range=0.2,
        height_shift_range=0.2,
        horizontal_flip=True,
        vertical_flip=True,
```

```
    zoom_range=0.2,
    shear_range=0.15,
    fill_mode='nearest'
)

valid_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(
    DATA_DIR,
    target_size=(IMG_SIZE, IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    classes=SELECTED_CLASSES,
    shuffle=True
)

valid_generator = valid_datagen.flow_from_directory(
    VALID_DIR,
    target_size=(IMG_SIZE, IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    classes=SELECTED_CLASSES,
    shuffle=False
)

return train_generator, valid_generator
```

Метод flow_from_directory автоматично визначає класи на основі назв піддиректорій та створює відповідність між індексами класів та їх назвами. Параметр shuffle=True забезпечує випадкове перемішування зображень на

кожній епосі, що покращує якість навчання. Метод також автоматично обробляє перетворення міток у категоріальний формат для багатокласової класифікації.

3.4 Розробка архітектури нейронної мережі

3.4.1 Вибір базової архітектури згорткової мережі

Для розв’язання задачі класифікації захворювань рослин було обрано архітектуру згорткової нейронної мережі, оскільки такі мережі демонструють найкращі результати у задачах комп’ютерного зору. Архітектура побудована на базі MobileNetV2.

Базова архітектура складається з чотирьох згорткових блоків, кожен з яких містить шар згортки, нормалізацію батчу, функцію активації та шар пулінгу.

3.4.2 Структура та параметри моделі

Перший згортковий блок містить 32 фільтри розміром 3×3, що виявляють низькорівневі ознаки такі як краї та текстури. Другий блок подвоює кількість фільтрів до 64, третій збільшує до 128, четвертий – до 256 фільтрів. Така прогресія дозволяє мережі навчитися ієрархії ознак: від простих до складних.

Після кожного згорткового шару застосовується нормалізація батчу, що стабілізує процес навчання та дозволяє використовувати вищі швидкості навчання. Функція активації ReLU вносить нелінійність у модель. Шари MaxPooling 2×2 зменшують просторову розмірність вдвічі, що знижує кількість параметрів та обчислювальну складність.

Після згорткових блоків застосовується шар глобального середнього пулінгу, що перетворює карти ознак у вектори фіксованої довжини. Далі йде Dense шар з 512 нейронами, dropout 0.5 для регуляризації, та фінальний Dense шар з 10 нейронами та softmax активацією для багатокласової класифікації.

Загальна кількість параметрів моделі становить 2594634, що є оптимальним балансом між експресивністю та схильністю до перенавчання.

3.4.3 Функції втрат та метрики оцінювання

Як функцію втрат було обрано категоріальну крос-ентропію, що є поширеним вибором для задач багатокласової класифікації. Ця функція вимірює різницю між передбаченим розподілом ймовірностей та істинним розподілом. Мінімізація крос-ентропії еквівалентна максимізації логарифмічної правдоподібності правильного класу.

Як оптимізатор використовувався Adam з початковою швидкістю навчання 0.001. Adam поєднує переваги адаптивної швидкості навчання та імпульсу, що забезпечує швидку збіжність навіть на складних функціях втрат. Для оцінки якості моделі використовувалися дві метрики: точність (accuracy) та top-3 точність.

Метрика accuracy показує частку правильно класифікованих зображень, коли передбачений клас з найвищою ймовірністю збігається з істинним класом. Метрика top-3 accuracy вважає передбачення правильним, якщо істинний клас знаходиться серед трьох класів з найвищими ймовірностями. Це корисно для розуміння впевненості моделі та виявлення схожих класів захворювань.

3.5 Реалізація процесу навчання моделі

3.5.1 Налаштування гіперпараметрів

Процес навчання моделі конфігурувався з наступними гіперпараметрами: кількість епох – 30, розмір батчу – 32, початкова швидкість навчання – 0.001, розмір вхідного зображення – 224×224 пікселі. Ці значення були обрані на основі типових практик для задач класифікації зображень середньої складності.

Розмір батчу 32 є компромісом між стабільністю градієнтного спуску та швидкістю навчання. Більші батчі дають більш стабільні градієнти, але вимагають більше пам’яті та можуть призвести до гіршої генералізації. Менші батчі додають шум у процес оптимізації, що іноді допомагає уникнути локальних мінімумів, але сповільнюють навчання.

Початкова швидкість навчання 0.001 є стандартним значенням для оптимізатора Adam. Надто велика швидкість навчання може призвести до нестабільності та розбіжності процесу навчання, тоді як надто мала швидкість значно уповільнює збіжність. У процесі навчання швидкість автоматично знижувалася при досягненні плато за допомогою callback ReduceLROnPlateau.

3.5.2 Використання callback-функцій

Для контролю процесу навчання та його оптимізації використовувалися три callback-функції: ModelCheckpoint, ReduceLROnPlateau та EarlyStopping. Кожна з них відповідає за певний аспект управління навчанням та допомагає досягти найкращих результатів.

Лістинг 3.2 Параметри callback-функцій:

```
callbacks = [
    ModelCheckpoint(
        MODEL_DIR / 'best_model.h5',
        monitor='val_accuracy',
        save_best_only=True,
        verbose=1
    ),
    EarlyStopping(
        monitor='val_loss',
        patience=5,
```

```
        restore_best_weights=True,
        verbose=1
    ),
    ReduceLROnPlateau(
        monitor='val_loss',
        factor=0.5,
        patience=3,
        min_lr=1e-7,
        verbose=1
    )
]
```

ModelCheckpoint автоматично зберігає модель після кожної епохи, якщо валідаційна точність покращилася. Параметр *save_best_only=True* гарантує, що зберігається лише найкраща версія моделі, що економить дисковий простір. Це особливо важливо, оскільки модель може показувати відхилення у продуктивності через певну випадковість процесу навчання.

ReduceLROnPlateau відстежує валідаційну точність та зменшує швидкість навчання в 2 рази, якщо метрика не покращується протягом 5 епох. Це дозволяє моделі робити більш точні кроки при наближенні до оптимуму. *EarlyStopping* зупиняє навчання, якщо валідаційна точність не покращується протягом 10 епох, що допомагає уникнути перенавчання та економить обчислювальні ресурси.

3.5.3 Оптимізація на GPU

Для прискорення процесу навчання використовувався графічний процесор NVIDIA GeForce RTX 3050 Ti Laptop GPU з підтримкою DirectML. DirectML є міжплатформовим API від Microsoft, що забезпечує апаратне прискорення машинного навчання на GPU від різних виробників через DirectX 12.

Конфігурація GPU виконувалася на початку скрипту за допомогою TensorFlow PluggableDevice API. Використання GPU дозволило досягти швидкості навчання приблизно 110 – 170 секунд на епоху залежно від складності обчислень. Це приблизно в 10 – 15 разів швидше порівняно з навчанням на CPU, що критично важливо при експериментуванні з різними архітектурами та гіперпараметрами.

TensorFlow автоматично розподіляє обчислення між CPU та GPU, розміщуючи на GPU операції, що підтримують прискорення, такі як матричні множення, згортки та активаційні функції. Управління пам’яттю здійснюється динамічно з використанням параметра force_memory_growth, що дозволяє TensorFlow виділяти пам’ять GPU за потребою замість резервування всієї доступної пам’яті одразу.

3.6 Аналіз результатів навчання моделі

3.6.1 Динаміка зміни точності по епохах

Аналіз поепохової динаміки точності дозволяє оцінити якість процесу навчання та виявити потенційні проблеми. На першій епосі модель досягла тренувальної точності 71,03% та валідаційної точності 83,23%, що свідчить про успішну ініціалізацію ваг мережі. Висока початкова валідаційна точність вказує на те, що задача класифікації є відносно простою для обраної архітектури.

Протягом перших 10 епох спостерігалося інтенсивне зростання обох метрик (рис. 3.5): тренувальна точність зросла з 71,03% до 84,85%, валідаційна – з 83,23% до 89,74%.

Така динаміка є типовою для фази активного навчання, коли модель швидко адаптується до особливостей датасету. Важливо відзначити, що валідаційна точність постійно випереджала тренувальну на 5 – 8%, що вказує на відсутність перенавчання на ранніх етапах.

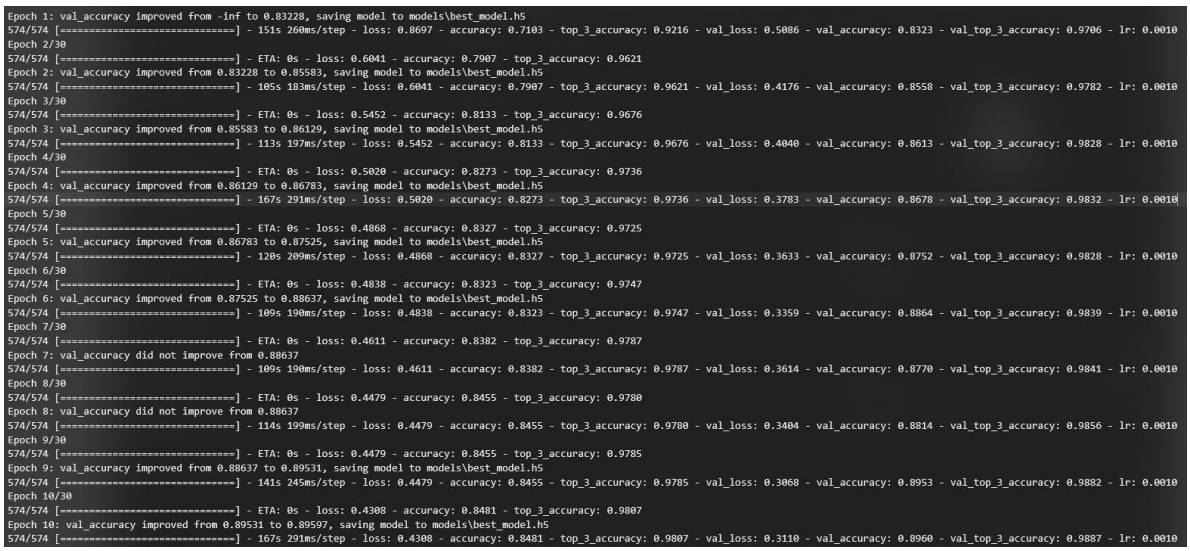

Рисунок 3.5 – Процес навчання перших десяти епох

В епохах 11 – 20 темп зростання сповільнився: тренувальна точність досягла 86,78%, валідаційна – 89,79% (рис. 3.6).

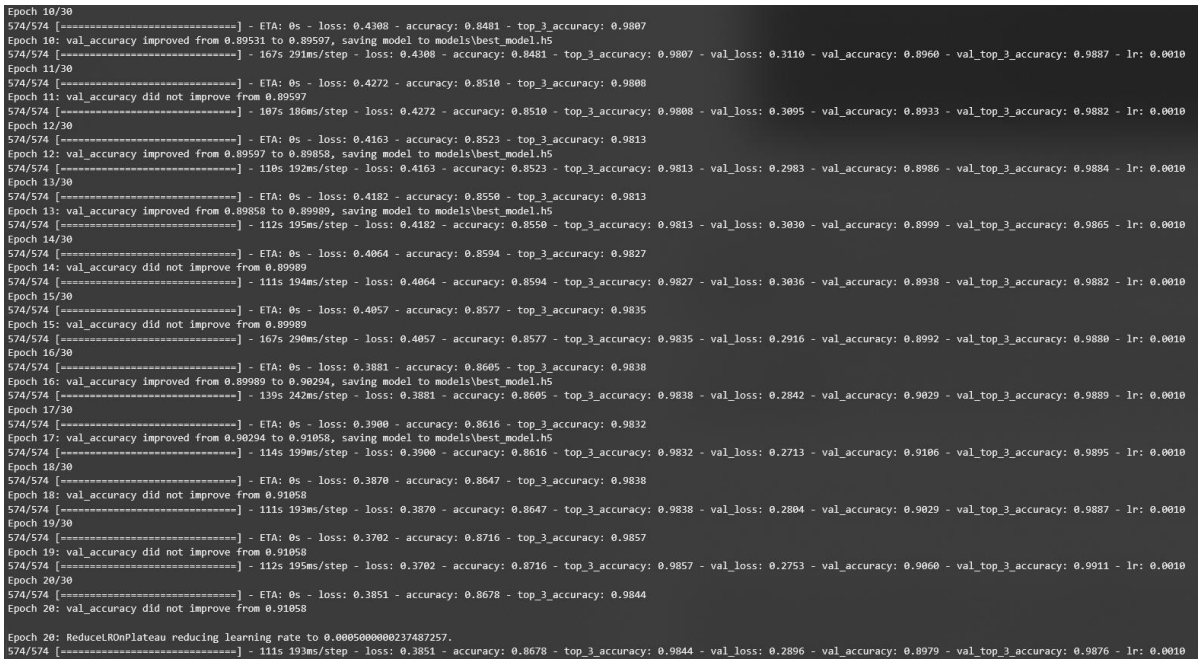

Рисунок 3.6 – Процес навчання від 11 до 20 епох

На епосі 20 спрацювала callback-функція *ReduceLROnPlateau* (рис. 3.6-3.7), що знизила швидкість навчання з 0.001 до 0.0005. Це спричинило новий етап покращення: на епосі 27 валідаційна точність досягла максимуму 92,10%. Фінальна модель на епосі 30 показала тренувальну точність 88,68% та

валідаційну 92,65% (рис. 3.7), що є відмінним результатом для задачі класифікації 10 класів.

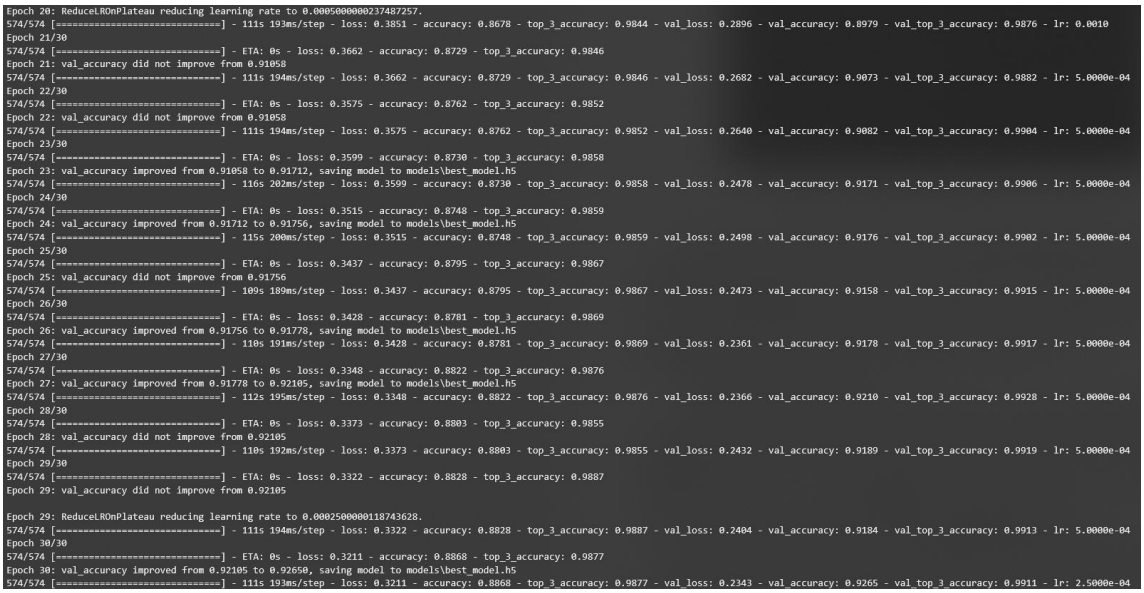

Рисунок 3.7 – Процес навчання від 21 до 30 епох

3.6.2 Аналіз функції втрат

Функція втрат є прямим показником того, наскільки впевнено модель робить передбачення. На першій епосі тренувальна втрата становила 0,8697, валідаційна – 0,5086. Різниця між цими значеннями вказує на те, що модель швидше навчилася на валідаційних даних, можливо через більшу репрезентативність або меншу складність валідаційного набору.

Протягом навчання обидві метрики втрат монотонно зменшувалися, що підтверджує успішну оптимізацію. Тренувальна втрата знизилася з 0,8697 до 0,3211 – це зменшення на 63%, валідаційна – з 0,5086 до 0,2343 – це зменшення на 54%. Швидше зменшення тренувальної втрати у відносних показниках є нормальним, оскільки модель має прямий доступ до цих даних під час навчання.

Важливою характеристикою є те, що валідаційна втрата залишалася нижчою за тренувальну протягом усього процесу навчання. Це нетиповий, але позитивний сценарій, що свідчить про хорошу генералізацію моделі.

Можливими причинами можуть бути ефективна аугментація даних на тренувальному наборі, що робить його складнішим для навчання, або більша репрезентативність валідаційного набору. Відсутність розбіжності між тренувальною та валідаційною втратами вказує на відсутність перенавчання.

3.6.3 Оцінка якості навчання та виявлення перенавчання

Перенавчання є однією з основних проблем у машинному навчанні, коли модель починає запам’ятовувати тренувальні дані замість виявлення загальних закономірностей. Основним індикатором перенавчання є зростання різниці між тренувальною та валідаційною метриками. У даному випадку спостерігається протилежна ситуація: валідаційна точність (92,65%) перевищує тренувальну (88,68%) на 3,97%, що свідчить про відсутність перенавчання (рис. 3.8).

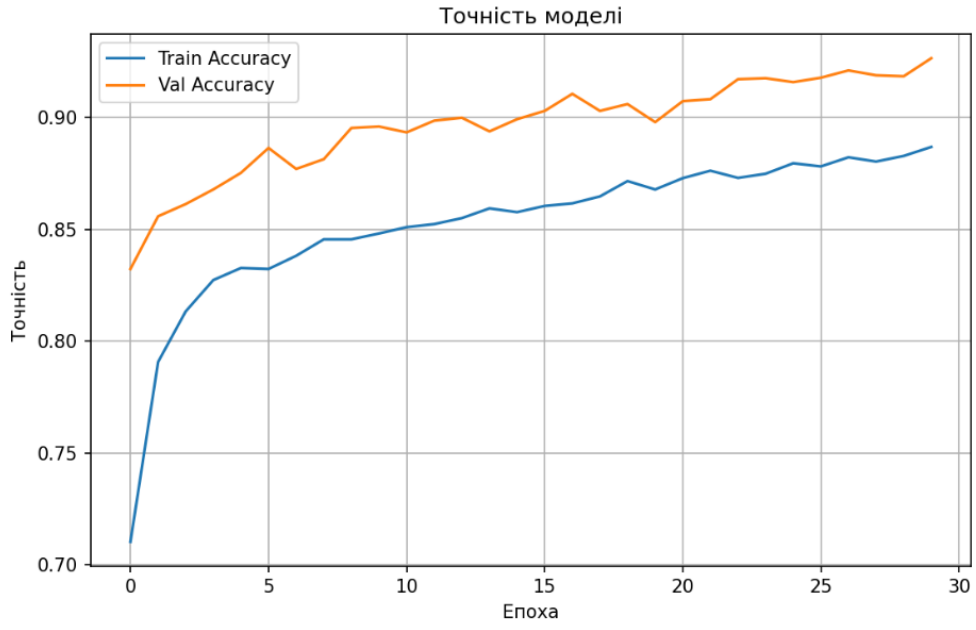

Рисунок 3.8 – Графік залежності точності моделі залежно від епохи

Аналіз графіків навчання підтверджує стабільність процесу оптимізації (рис. 3.8, 3.9). Валідаційна точність не показує ознак деградації або високої варіативності у другій половині навчання, що було б характерним для

перенавчання. Натомість спостерігається плавне зростання з періодичними плато, на яких спрацьовувала зменшення швидкості навчання (рис. 3.8).

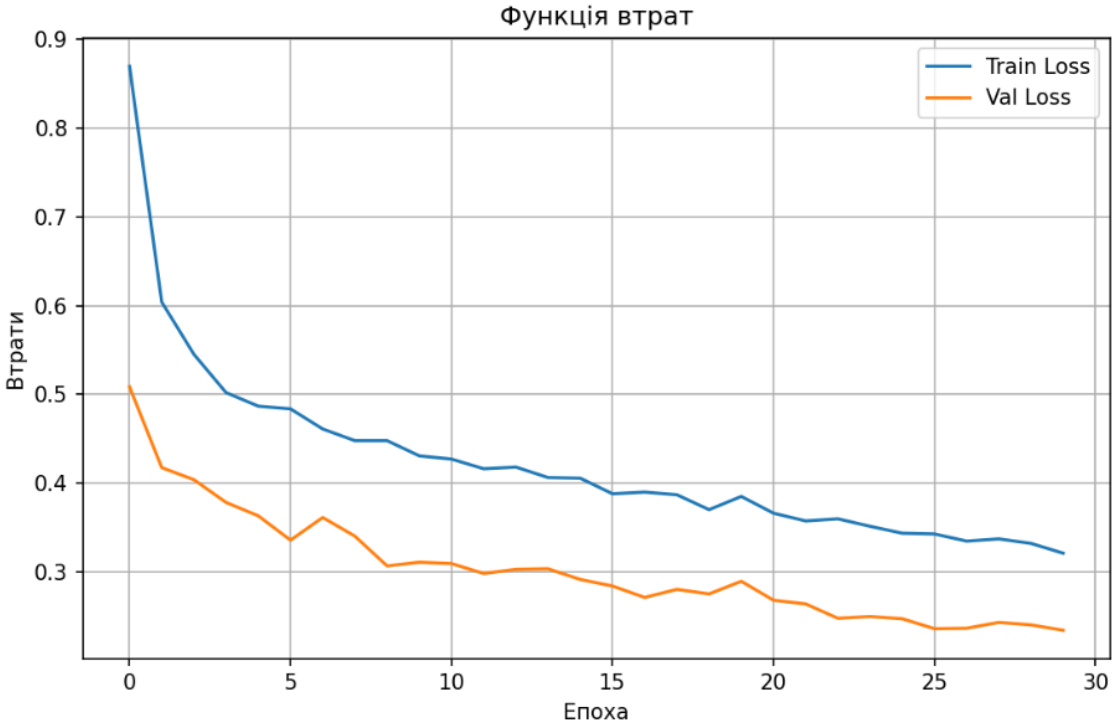

Рисунок 3.9 – Графік залежності втрат моделі залежно від епохи

Використані техніки регуляризації виявилися ефективними: dropout 50% у повнозв'язаному шарі запобіг ко-адаптації нейронів, нормалізація батчу стабілізувала розподіл активацій, аугментація даних збільшила різноманітність тренувального набору. Комбінація цих методів дозволила моделі навчитися узагальненим ознакам захворювань замість запам'ятовування специфічних зображень.

3.6.4 Фінальні показники ефективності моделі

Фінальна модель демонструє високу продуктивність у класифікації захворювань томатів. Валідаційна точність 92,65% означає, що модель правильно ідентифікує захворювання у більш ніж 9 з 10 випадків. Для

порівняння, baseline модель, що завжди передбачає найчастіший клас, досягла б точності лише близько 10%, оскільки класи відносно збалансовані.

Метрика top-3 точності на валідаційному наборі становить 99,11%, що вказує на впевненість моделі у своїх передбаченнях. Це означає, що у 99% випадків правильний клас знаходиться серед трьох найбільш ймовірних варіантів. Така висока top-3 точність робить систему корисною для експертів-агрономів, які можуть переглянути топ-3 передбачень та прийняти остаточне рішення.

Тренувальна точність 88,68% нижча за валідаційну, що є позитивним сигналом про здатність моделі до генералізації. Це також вказує на те, що модель не досягла граничної продуктивності на тренувальному наборі, тобто є потенціал для подальшого навчання. Однак, враховуючи вже високу валідаційну точність, подальше навчання може призвести до перенавчання, тому 30 епох виявилося оптимальним вибором.

Аналіз часу інференсу показав, що модель здатна обробляти одне зображення за 20 – 30 мілісекунд на GPU, що дозволяє класифікувати близько 30 – 50 зображень на секунду. Така швидкість є достатньою для практичного застосування у реальних умовах, наприклад, для мобільного додатку або веб-сервісу діагностики захворювань.

3.7 Реалізація модуля інференсу та користувацького інтерфейсу

3.7.1 Завантаження навченої моделі

Модуль інференсу реалізовано у вигляді окремого скрипту, що завантажує навчену модель з файлу HDF5 та виконує класифікацію нових зображень. Використання формату HDF5 дозволяє зберегти не лише ваги моделі, але й повну архітектуру мережі, що спрощує процес розгортання та виключає необхідність повторного визначення структури моделі в коді.

При завантаженні моделі також зчитується JSON-файл з метаданими, що містить список класів захворювань та їх українські переклади. Це дозволяє відобразити зрозумілі користувачеві назви захворювань замість технічних ідентифікаторів. Метадані також містять інформацію про розмір вхідного зображення, що гарантує коректну попередню обробку нових зображень.

3.7.2 Обробка вхідних зображень

Перед класифікацією вхідне зображення проходить ту ж саму попередню обробку, що і тренувальні дані. Спочатку зображення завантажується з використанням бібліотеки Pillow та конвертується у RGB формат для забезпечення консистентності. Далі зображення масштабується до розміру 224×224 пікселі за допомогою білінійної інтерполяції.

Після масштабування зображення конвертується у NumPy масив та нормалізується діленням на 255, що переводить значення пікселів у діапазон [0, 1]. Це критично важливо, оскільки модель навчалася на нормалізованих даних. Нарешті, масив розширюється додатковим виміром для формування батчу розміром 1, оскільки TensorFlow очікує вхідні дані у форматі: batch_size, height, width, channels.

3.8 Результати роботи програми на листях рослин

Процес роботи програми розпочинається з ініціалізації системи діагностики. При запуску застосунок завантажує попередньо навчену нейронну мережу з файлу моделі, що зберігається у директорії проєкту. Одночасно з моделлю завантажується супутній конфігураційний файл, який містить інформацію про класи захворювань, їхні україномовні назви та параметри обробки зображень. Цей етап є критично важливим для коректної роботи всієї

системи, оскільки саме від якості натренованої моделі залежить точність подальшої діагностики.

Після успішної ініціалізації програма очікує від користувача вказівки шляху до файлу зображення, яке потрібно проаналізувати. Зображення передається як аргумент командного рядка при запуску скрипту (рис. 3.10). Система виконує перевірку існування вказаного файлу та його доступності для читання. У разі відсутності файлу або помилок доступу програма повідомляє користувача про проблему та завершує роботу з відповідним кодом помилки.

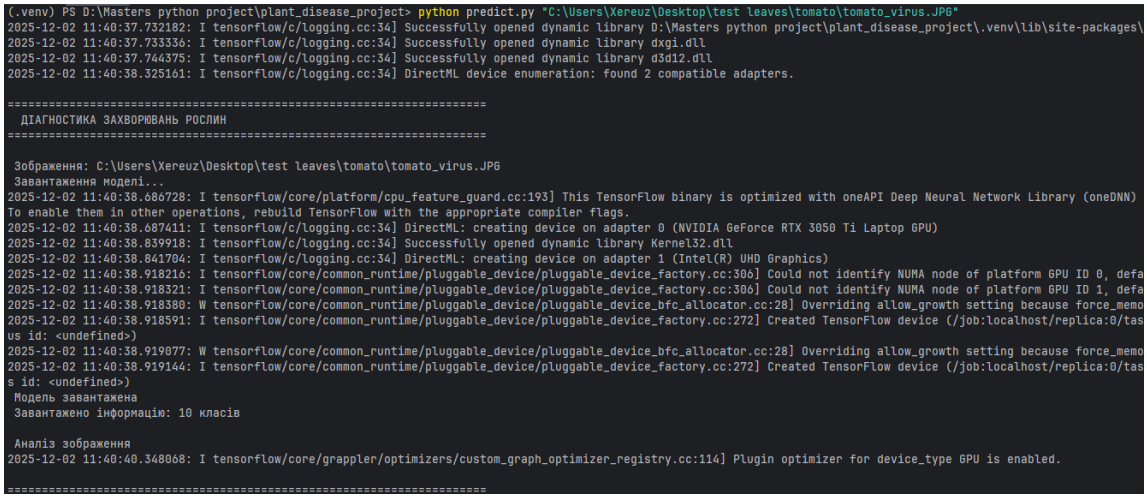

Рисунок 3.10 – Результат процесу завантаження зображення для діагностики захворювань

Етап попередньої обробки зображення включає кілька послідовних операцій. Спочатку програма відкриває файл зображення та конвертує його у стандартний RGB колірний простір, що гарантує однорідність вхідних даних незалежно від формату оригінального файлу. Далі виконується зміна розміру зображення до фіксованих параметрів, які відповідають вимогам вхідного шару нейронної мережі. Цей крок є необхідним, оскільки модель була натренована на зображеннях фіксованого розміру і потребує узгодженості розмірності вхідних даних. Після зміни розміру виконується нормалізація пікселів, що приводить значення інтенсивності кольорів до діапазону, оптимального для роботи нейронної мережі. Завершальним кроком підготовки є додавання додаткової

розмірності до масиву даних, що формує пакет розміром один елемент, як того вимагає архітектура моделі.

Процес безпосереднього розпізнавання відбувається шляхом передачі підготовленого зображення на вхід навченої нейронної мережі. Модель виконує прямий прохід через всі свої шари, обчислюючи активації нейронів та формуючи на виході вектор ймовірностей для кожного з можливих класів захворювань. Кожне значення у цьому векторі представляє обчислену моделлю ймовірність належності проаналізованого зображення до відповідного класу захворювання.

Результати передбачення програма обробляє та структурує для зручного представлення користувачу (рис. 3.11). Система визначає клас з найвищою ймовірністю як основний діагноз та формує список топ-5 найімовірніших варіантів.

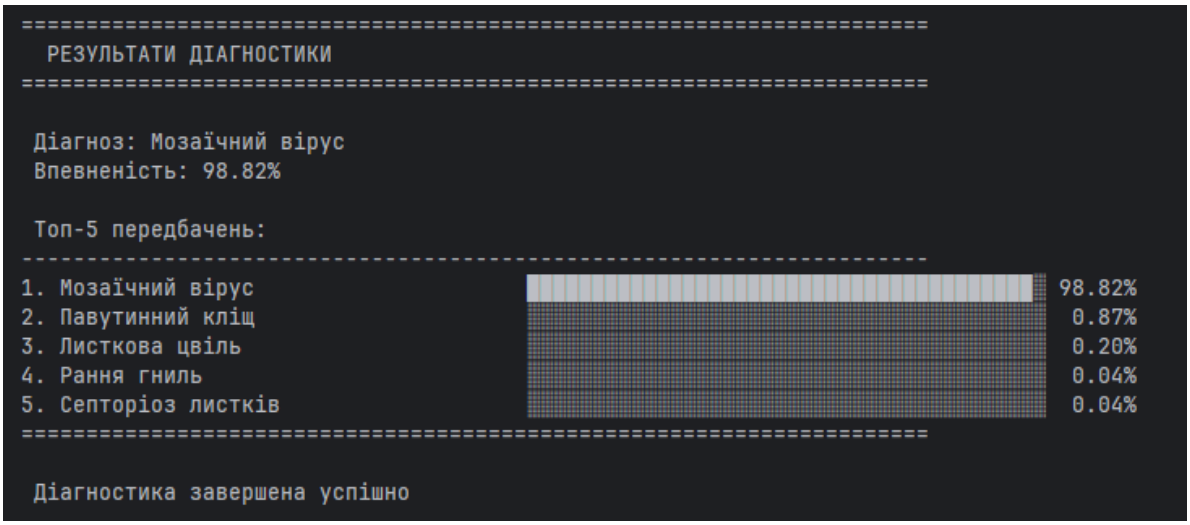

Рисунок 3.11 – Вивід результатів діагностики захворювання рослин

Виведення результатів на екран реалізовано у структурованому та візуально зрозумілому форматі. Програма відображає основний діагноз з найвищою ймовірністю, супроводжуючи його процентним показником впевненості моделі. Нижче представлено список альтернативних діагнозів у порядку зменшення ймовірності, кожен з яких супроводжується текстовою шкалою-індикатором для візуальної оцінки рівня впевненості (рис. 3.11). Така форма представлення дозволяє користувачу не лише отримати найбільш

імовірний діагноз, але й оцінити наявність альтернативних варіантів та їхню відносну ймовірність.

При аналізі зображення листя томату ураженого бактеріальною хворобою (рис. 3.12), програма діагностує бактеріальні плями, та впевнена у результаті діагностики на 96,65% (рис. 3.13). Серед топ-5 передбачень програма також виділяє: ранню гниль (2,44%), септоріоз листків (0,49%), цільова пляма (0,2%) та пізню гниль (0,11%).

Рисунок 3.12 – Вхідне зображення листя ураженого бактеріальною хворобою

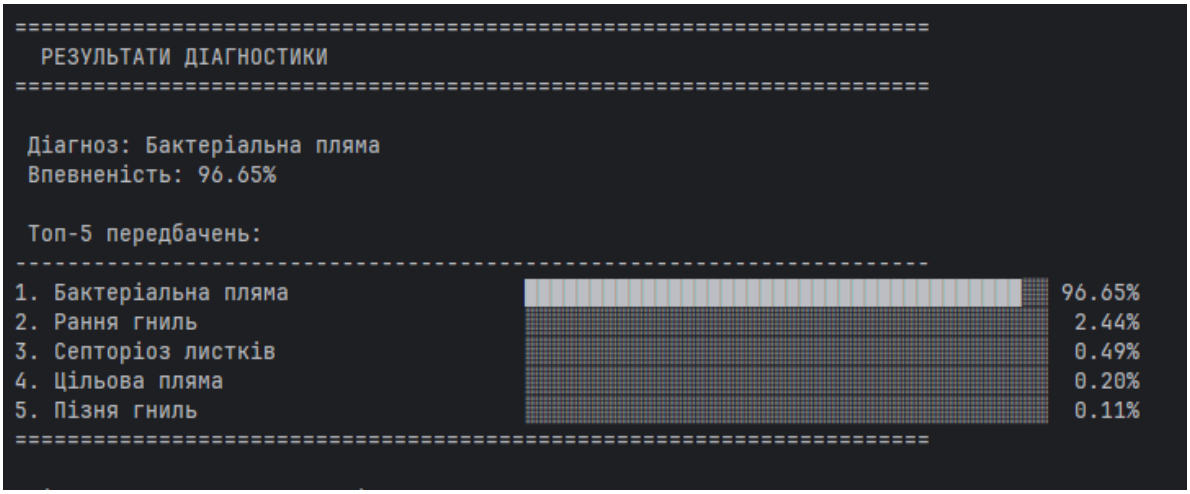

Рисунок 3.13 – Результат діагностики листя ураженого бактеріальною хворобою

Більш неоднозначним результатом аналізу можна назвати результат аналізу іншого листя, яке так само уражено бактеріальною хворобою, але має не настільки чітке її вираження (рис. 3.14).

Рисунок 3.14 – Листя що уражене бактерією та має нечітке вираження симптому

У результаті програма показує впевненість у своєму діагнозі на 69,91% (рис. 3.15). Це може вказувати на невпевненість у діагнозі, особливо враховуючи, що другий за ймовірністю діагноз має 14,82% ймовірності на думку програми. Однак варто зауважити, що подібна поведінка роботи програми є очікуваною, оскільки вираження хвороби на цьому тестовому зразку неяскраво виражене, а одже ускладнює чітке твердження про походження захворювання.

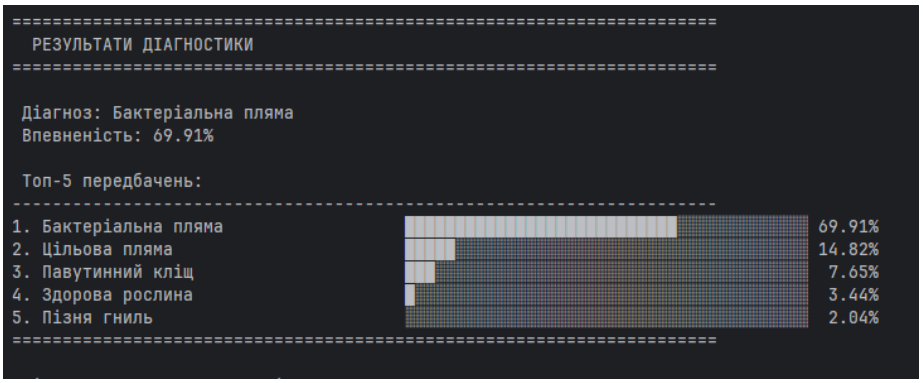

Рисунок 3.15 – Результат діагностики листя що уражене бактерією та має нечітке вираження симптому

ВИСНОВКИ

У кваліфікаційній роботі досліджено та реалізовано метод розпізнавання зображень для діагностики захворювань рослин на основі глибокого навчання. Виконано комплекс теоретичних досліджень та практичних робіт, спрямованих на створення ефективної системи автоматичної класифікації фітопатологій.

Проведено детальний аналіз предметної області, який виявив критичні обмеження традиційних методів діагностики захворювань рослин. Встановлено, що візуальна діагностика, мікроскопічний аналіз та молекулярно-генетичні методи характеризуються високою трудомісткістю, значними витратами часу, необхідністю спеціалізованого обладнання та кваліфікованого персоналу, що робить їх недостатньо ефективними для швидкого прийняття рішень та моніторингу великих сільськогосподарських площ. Це обґрунтувало доцільність застосування методів комп’ютерного зору та штучного інтелекту для автоматизації процесу діагностики.

Виконано комплексний огляд сучасних підходів до розв’язання задачі розпізнавання захворювань рослин. Порівняльний аналіз показав, що класичні методи комп’ютерного зору з ручним проєктуванням ознак досягають точності 70 – 92% на контрольованих датасетах, однак їхня продуктивність критично погіршується в реальних умовах – падіння на 25 – 40% при зміні освітлення або фону [43]. Згорткові нейронні мережі з Transfer Learning забезпечують 96 – 99% точності на лабораторних датасетах та 88 – 95% у польових умовах при значно вищій стійкості до варіацій умов зйомки [44 – 46]. Дослідження еволюції CNN архітектур від AlexNet через VGG, GoogLeNet, ResNet до сучасних EfficientNet та MobileNet виявило ключові інновації: residual connections для навчання глибоких мереж, depthwise separable convolutions для обчислювальної ефективності, compound scaling для оптимального балансу параметрів. Встановлено, що Transfer Learning є критично важливим для практичних застосувань, зменшуючи вимоги до даних у 5 – 10 разів, що дало можливість

визначити оптимальні архітектурні рішення для проєктування системи діагностики [47, 48].

На основі проведеного аналізу спроєктовано та реалізовано програмне забезпечення для діагностики захворювань томатів. Розроблено модульну архітектуру з окремими модулями підготовки даних, навчання моделі та інференсу. Реалізовано власну архітектуру згорткової нейронної мережі з чотирма згортковими блоками загальною кількістю 2,6 мільйона параметрів. Застосовано ефективні техніки регуляризації: dropout (50%), нормалізацію батчу, інтенсивну аугментацію даних, що запобігло перенавчанню моделі та забезпечило високу здатність до генералізації.

Проведено експериментальне дослідження ефективності розробленого методу. Виконано навчання моделі на 30 епохах з використанням GPU для прискорення обчислень, що дозволило досягти швидкості 110 – 170 секунд на епоху. Фінальна модель продемонструвала високі показники класифікації: валідаційна точність 92,65%, top-3 точність 99,11%. Встановлено відсутність перенавчання: валідаційна точність перевищила тренувальну на 4%, що свідчить про відмінну генералізацію моделі. Швидкість обробки зображень на рівні 30 – 50 зображень на секунду є достатньою для практичного застосування в режимі реального часу.

Розроблено функціональний модуль інференсу з користувацьким інтерфейсом, який забезпечує завантаження навченої моделі у форматі HDF5 з метаданими класів та коректну попередню обробку вхідних зображень. Реалізовано збереження результатів у лог-файли та обробку помилок, що дозволило створити готове до практичного використання програмне забезпечення.

Наукова новизна роботи полягає у розробці та експериментальному дослідженні ефективної архітектури згорткової нейронної мережі з оптимізованим балансом глибини, кількості параметрів та застосуванням комплексу сучасних технік регуляризації, що забезпечило високу точність

класифікації захворювань томатів при відсутності перенавчання та швидкості обробки, достатній для практичних застосувань.

Практична цінність роботи полягає у створенні готової до застосування системи діагностики захворювань томатів, яка може бути інтегрована у мобільні додатки або веб-сервіси для аграріїв. Розроблене програмне забезпечення дозволяє значно прискорити процес діагностики з кількох годин до кількох секунд, підвищити точність класифікації порівняно з візуальним оглядом та забезпечити доступність інструментів діагностики для фермерів без спеціальної фітопатологічної освіти.

Результати роботи апробовано у вигляді 2 тез доповідей під час IX Міжнародної студентської наукової конференції [49] та IX Міжнародної мультидисциплінарної студентської наукової конференції «Розвиток суспільства та науки в умовах цифрової трансформації» [50].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Ashurov, A. Y., Al-Gaashani, M. S., Samee, N. A., Alkanhel, R., Atteia, G., Abdallah, H. A., & Saleh Ali Muthanna, M. (2025). Enhancing plant disease detection through deep learning: a Depthwise CNN with squeeze and excitation integration and residual skip connections. *Frontiers in Plant Science*, *15*, 1505857.

2. Pal, C., Karmakar, S., Mukherjee, I., & Chakrabarti, P. P. (2025). A lightweight and explainable CNN model for empowering plant disease diagnosis. *Scientific Reports*, *15*(1), 30720.

3. Krishna, M. S., Machado, P., Otuka, R. I., Yahaya, S. W., Neves dos Santos, F., & Ihianle, I. K. (2025). Plant Leaf Disease Detection Using Deep Learning: A Multi-Dataset Approach. *J*, *8*(1), 4.

4. Sujatha, R., Krishnan, S., Chatterjee, J. M., & Gandomi, A. H. (2025). Advancing plant leaf disease detection integrating machine learning and deep learning. *Scientific Reports*, *15*(1), 11552.

5. Shoaib, M., Sadeghi-Niaraki, A., Ali, F., Hussain, I., & Khalid, S. (2025). Leveraging deep learning for plant disease and pest detection: a comprehensive review and future directions. *Frontiers in Plant Science*, *16*, 1538163.

6. Jung, M., Song, J. S., & Shin, A. Y. *Construction of deep learning-based disease detection model in plants. Sci. Rep. 13, 7331 (2023)*.

7. González-Briones, A., Florez, S. L., Chamoso, P., Castillo-Ossa, L. F., & Corchado, E. S. (2025). Enhancing Plant Disease Detection: Incorporating Advanced CNN Architectures for Better Accuracy and Interpretability. *International Journal of Computational Intelligence Systems*, *18*(1), 120.

8. Natarajan, S., Chakrabarti, P., & Margala, M. (2024). Robust diagnosis and meta visualizations of plant diseases through deep neural architecture with explainable AI. *Scientific Reports*, *14*(1), 13695.

9. Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. *Frontiers in plant science*, *7*, 215232.

10. Tang, Z., Yang, J., Li, Z., & Qi, F. (2020). Grape disease image classification based on lightweight convolution neural networks and channelwise attention. *Computers and Electronics in Agriculture*, *178*, 105735.

11. Kobylin, O., Vyskrebentseva, S., & Petrova, R. (2019). Обробка даних, що містять пропуски в задачах кластеризації. *Системи управління, навігації та зв'язку. Збірник наукових праць*, *5*(57).

12. Mashtalir, V., Ruban, I., & Levashenko, V. (Eds.). (2019). Advances in Spatio-Temporal Segmentation of Visual Data (Vol. 876). Springer Nature.

13. Kobylin, O. A., Gorokhovatskyi, V. O., Tvoroshenko, I. S., & Peredrii, O. O. (2020). The application of non-parametric statistics methods in image classifiers based on structural description components. Telecommunications and Radio Engineering, 79(10).

14. Gorokhovatskyi, V., & Tvoroshenko, I. (2023). Identification of visual objects by the search request. *International scientific symposium «INTELLIGENT SOLUTIONS-S»* (pp. 25-27).

15. Daradkeh, Y. I., Gorokhovatskyi, V., Tvoroshenko, I., Gadetska, S., & Al-Dhaifallah, M. (2023). Statistical data analysis models for determining the relevance of structural image descriptions. IEEE Access, 11, 126938-126949.

16. Gorokhovatskyi V., Tvoroshenko I., Kobylin O., and Vlasenko N. (2023) Search for visual objects by request in the form of a cluster representation for the structural image description, Advances in Electrical and Electronic Engineering, 21(1), pp. 19-27.

17. Kobylin, O. A., Gorokhovatskyi, V. O., Tvoroshenko, I. S., & Peredrii, O. O. (2020). The application of non-parametric statistics methods in image classifiers based on structural description components. Telecommunications and Radio Engineering, 79(10).

18. Ibrahim, D. Y., Gorokhovatskyi, V., Tvoroshenko, I., & Mujahed, A. D. (2022). Classification of Images Based on a System of Hierarchical Features.

19. Rabotiahov, A., Kobylin, O., Dudar, Z., & Lyashenko, V. (2018, February). Bionic image segmentation of cytology samples method. In *2018 14th International*

Conference on Advanced Trends in Radioelecrtronics, Telecommunications and Computer Engineering (TCSET) (pp. 665-670). IEEE.

20. Kobylin, O., & Lyashenko, V. (2016). Contrast Modification as a Tool to Study the Structure of Blood Components.

21. Kinoshenko, D., Kobylin, O., Mashtalir, S., & Stolbovyi, M. (2019, March). Metric video retrieval speedup by irrelevant data elimination. In Eleventh International Conference on Machine Vision (ICMV 2018) (Vol. 11041, pp. 176-183). SPIE.

22. Gorokhovatskyi, V. A., Rusakova, N., & Tvoroshenko, I. S. (2020). The application of image analysis methods and predicate logic in applied problems of magnetic monitoring. Telecommunications and Radio Engineering, 79(20).

23. Daradkeh, Y. I., & Tvoroshenko, I. (2020). Technologies for making reliable decisions on a variety of effective factors using fuzzy logic. International Journal of Advanced Computer Science and Applications, 11(5).

24. Gorokhovatskyi, V. O., Tvoroshenko, I. S., & Vlasenko, N. V. (2020). Using fuzzy clustering in structural methods of image classification. Telecommunications and Radio Engineering, 79(9).

25. Yakovleva, O., & Nikolaieva, K. (2020). Research of descriptor based image normalization and comparative analysis of SURF, SIFT, BRISK, ORB, KAZE, AKAZE descriptors. Advanced Information Systems, 4(4), 89-101.

26. Daradkeh, Y. I., & Tvoroshenko, I. (2020). Technologies for making reliable decisions on a variety of effective factors using fuzzy logic. International Journal of Advanced Computer Science and Applications, 11(5).

27. Pomazan, V., Tvoroshenko, I., & Gorokhovatskyi, V. (2023). Development of an application for recognizing emotions using convolutional neural networks. *International Journal of Academic Information Systems Research*, 7(7), (pp. 25-36).

28. Lyashenko V., Kobylin O., Selevko O. (2020) Wavelet Analysis and Contrast Modification in the Study of Cell Structures Images. International Journal of Advanced Trends in Computer Science and Engineering. 9(4). – 4701-4706.

29. Oleg, K., Sergii, M., & Mykhailo, S. (2017, October). Video Clustering via Multidimensional Time-Series Analysis. In *Proceedings of the 9th International Conference on Information Management and Engineering* (pp. 60-63). ACM.

30. Кобилін, О. А., & Творошенко, І. С. (2021). Методи цифрової обробки зображень.

31. Yakovleva, O., Kovtunenko, A., Liubchenko, V., Honcharenko, V., & Kobylin, O. (2023). Face Detection for Video Surveillance-based Security System. In COLINS (3) (pp. 69-86).

32. Yakovleva, O., Slyusar, V., Kushnir, O., & Sabovchyk, A. (2021). New trends in scientific and technological revolution (STR) and transformation of science and education systems in the paradigm of sustainable development. In E3S Web of Conferences (Vol. 277, p. 06006). EDP Sciences.

33. Bodyanskiy, Y., Vynokurova, O., Kobylin, I., & Kobylin, O. (2016). Adaptive fuzzy clustering of short time series with unevenly distributed observations in Data Stream Mining tasks. *Information Technology and Management Science*, *19*(1), 23-28.

34. Gorokhovatskyi, V., Tvoroshenko, I., & Olena, Y. (2024). Transforming image descriptions as a set of descriptors to construct classification features. *Indonesian Journal of Electrical Engineering and Computer Scienc*, 33 (1), (pp. 113-125).

35. Kuzminska, O., Mazorchuk, M., Morze, N., & Kobylin, O. (2019, June). Digital learning environment of ukrainian universities: The main components to influence the competence of students and teachers. In International Conference on Information and Communication Technologies in Education, Research, and Industrial Applications (pp. 210-230). Springer, Cham.

36. Кобилін, О. А., & Путятіна, О. Є. (2024). Знешумлення зображень, зіпсованих дробовим шумом, у реальному часі. Системи обробки інформації, (1 (176), 46-51.

37. Gorokhovatskyi , V., Chmutov , Y., Tvoroshenko , I., & Kobylin , O. (2025). Reducing computational costs by compressing the structural description in image

classification methods. Advanced Information Systems, 9(1), 5–12. https://doi.org/10.20998/2522-9052.2025.1.01

38. Кобилін, О., Вечірська, І., Афанасьєв, А. (2024). Аналіз існуючих моделей глибинного навчання в задачах обробки природної мови. Information Technology: Computer Science, Software Engineering and Cyber Security, 3, 63–76, https://doi.org/10.32782/IT/2024-3-7

39. Кобилін, О., Вечірська, І., Кравченко, О. (2024). Порівняння нейронних мереж типу RNN та LSTM. Information Technology: Computer Science, Software Engineering and Cyber Security, 3, 97–107, https://doi.org/10.32782/IT/2024-3-10

40. Vechirska, I., Kobylin, O., Prokopiev , S., Vechirska, A., & Kucherenko, M. (2022). Building a logical network for solving the problem of car rental by means algebra of finite predicates. Computer Systems and Information Technologies, (2), 78–87. https://doi.org/10.31891/csit-2022-2-9

41. Ji, M., Zhang, L., & Wu, Q. (2020). Automatic grape leaf diseases identification via UnitedModel based on multiple convolutional neural networks. *Information Processing in Agriculture*, *7*(3), 418-426.

42. Khan, A. I., Quadri, S. M. K., Banday, S., & Shah, J. L. (2022). Deep diagnosis: A real-time apple leaf disease detection system based on deep learning. *computers and Electronics in Agriculture*, *198*, 107093.

43. Jiang, F., Lu, Y., Chen, Y., Cai, D., & Li, G. (2020). Image recognition of four rice leaf diseases based on deep learning and support vector machine. *Computers and Electronics in Agriculture*, *179*, 105824.

44. Barbedo, J. G. A. (2019). Plant disease identification from individual lesions and spots using deep learning. *Biosystems engineering*, *180*, 96-107.

45. Too, E. C., Yujian, L., Njuki, S., & Yingchun, L. (2019). A comparative study of fine-tuning deep learning models for plant disease identification. *Computers and Electronics in Agriculture*, *161*, 272-279.

46. Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. *Computers and electronics in agriculture*, *145*, 311-318.

47. Tm, P., Pranathi, A., SaiAshritha, K., Chittaragi, N. B., & Koolagudi, S. G. (2018, August). Tomato leaf disease detection using convolutional neural networks. In *2018 eleventh international conference on contemporary computing (IC3)* (pp. 1-5). IEEE.

48. Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Computers and electronics in agriculture*, *147*, 70-90.

49. Свістельник Д. О., Кобилін О. А. (2025). Застосування згорткових нейронних мереж для діагностики захворювань рослин: аналіз методів класифікації зображень листя. Актуальні питання розвитку галузей науки: збірник наукових праць з матеріалами VI Міжнародної наукової конференції.

50. Свістельник Д. О., Кобилін О. А. (2025). Виявлення патологій рослин на основі аналізу зображень: порівняння методів Machine Learning та Deep Learning. IX Міжнародна мультидисциплінарна студентська наукова конференція «Розвиток суспільства та науки в умовах цифрової трансформації».