
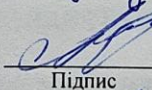


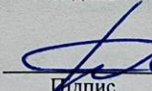
КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА


на тему Метод визначення стадії розвитку соняшника по фото за допомогою глибоких нейронних мереж

Галузь знань 12 – Інформаційні технології
Шифр і назва галузі знань
Спеціальність 122 – Комп'ютерні науки
Шифр і назва спеціальності
Освітня програма Комп'ютерні науки
Назва освітньої програми

Виконав: студент групи КНс-22-1  Вадим МАЛАЙДАХ
Група виконавця Підпис Ім'я, ПРІЗВИЩЕ

Керівник: асистент каф. КН  Марина МОЛЧАНОВА
Науковий ступінь, посада Підпис Ім'я, ПРІЗВИЩЕ

Нормоконтроль: к.т.н., доц. каф. КН  Руслан БАГРІЙ
Науковий ступінь, посада Підпис Ім'я, ПРІЗВИЩЕ

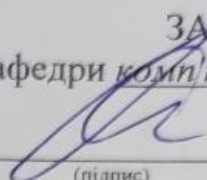
До захисту допускаю:
Зав. кафедри КН, д.т.н., професор  Олександр БАРМАК
Підпис Ім'я, ПРІЗВИЩЕ

18 06 2025 р.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій
Кафедра комп'ютерних наук
Освітній ступінь бакалавр
Галузь знань 12 – Інформаційні технології
Спеціальність 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ
Завідувач кафедри комп'ютерних наук


(підпис)
д.т.н., професор Олександр БАРМАК
« 10 » 02 2025 року

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА**

1. Тема кваліфікаційної роботи бакалавра: «Метод визначення стадії розвитку соняшника по фото за допомогою глибоких нейронних мереж»
2. Завдання видано студенту Вадиму Малайдаху
(Ім'я, прізвище)
3. Керівник роботи асистент кафедри КН Марина Молчанова
(посада, ім'я, прізвище)
4. Затверджено наказом університету від « 07 » 02 2025 р. № 23
5. Дата видачі завдання студенту: « 10 » 02 2025 р.
6. Зміст пояснювальної записки (перелік задач) та вихідні дані:
Мета роботи – підвищення точності визначення стадії розвитку соняшника по фото за допомогою глибоких нейронних мереж. Для досягнення мети слід виконати такі задачі: виконати дослідження предметної області; розробити метод на основі глибоких нейронних мереж; здійснити програмну реалізацію у вигляді інтелектуальної системи моніторингу розвитку соняшника по фото на основі розробленого методу; дослідити ефективність розробленого методу створеним програмним забезпеченням.

7. Календарний план виконання кваліфікаційної роботи бакалавра:

№	Назва етапів (розділів) кваліфікаційної роботи бакалавра	Термін виконання	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи бакалавра з керівником, складання календарного графіка виконання	січень 2025	виконано
2	Ознайомлення з предметною областю, формулювання мети і задач дослідження, визначення об'єкта та предмета дослідження	лютий 2025	виконано
3	Проектування та розроблення методу вирішення завдання, загальної архітектури програмного забезпечення, інтерфейсу користувача, вибір засобів реалізації програмного забезпечення	березень 2025	виконано
4	Створення та тестування програмного забезпечення, дослідження ефективності, висновки з виконаної роботи	квітень 2025	виконано
5	Написання пояснювальної записки, урахування зауважень керівника, оформлення згідно з вимогами	травень 2025	виконано
6	Розробка презентаційних матеріалів та попередній захист кваліфікаційної роботи	травень 2025	виконано
7	Отримання відгуку керівника, рецензії, перевірка на плагіат, нормоконтроль	червень 2025	виконано
8	Підготовка до захисту та захист кваліфікаційної роботи	червень 2025	виконано

Виконавець: студент групи КНс-22-1
Група виконавця


Підпис

Вадим МАЛАЙДАХ
Ім'я, ПРІЗВИЩЕ

Керівник: асистент каф. КН
Науковий ступінь, посада


Підпис

Марина МОЛЧАНОВА
Ім'я, ПРІЗВИЩЕ

Анотація

Тема кваліфікаційної роботи бакалавра: «Метод визначення стадії розвитку соняшника по фото за допомогою глибоких нейронних мереж».

Виконавець кваліфікаційної роботи бакалавра: студент групи КНс-22-1
Вадим Малайдах.

Керівник кваліфікаційної роботи бакалавра: асистент кафедри КН
Марина Молчанова

Кваліфікаційна робота бакалавра містить:

Пояснювальна записка				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
53	20	2	50	3

Метою кваліфікаційної роботи бакалавра є підвищення точності визначення стадії розвитку соняшника по фото за допомогою глибоких нейронних мереж. Для розробки прикладної інтелектуальної системи було використано мову програмування Python та бібліотеки машинного навчання TensorFlow і Keras.

Розроблена система призначена для використання агрономами, дослідниками у галузі рослинництва, аналітиками аграрного сектору, а також спеціалістами з точного землеробства для оперативного моніторингу стану посівів.

Напрямами практичного використання розробленої інтелектуальної системи є визначення стадії розвитку соняшника по фото.

Ключові слова: глибоке навчання, класифікація, землеробство, визначення стадії розвитку соняшнику, глибокі нейронні мережі.

Виконавець: студент групи КНс-22-1
Група виконавця


Підпис

Вадим МАЛАЙДАХ
Ім'я, ПРІЗВИЩЕ

Зміст

Перелік скорочень	4
Вступ.....	6
Розділ 1 Характеристика предметної області: аналіз моделей, методів та реалізацій.....	8
1.1 Аналіз інформаційних моделей в області визначення стадії розвитку соняшника за візуальними даними.....	8
1.2 Засоби інтелектуального визначення стадії розвитку соняшника по фото .	11
1.3 Аналіз існуючих програмних реалізацій в області автоматизованого визначення стадії розвитку соняшника по фото	14
1.4 Аналіз наукових публікацій за темою роботи	17
1.5 Мета та задачі кваліфікаційної роботи бакалавра	19
Розділ 2 Проектування методу визначення стадії розвитку соняшника по фото за допомогою глибоких нейронних мереж	20
2.1 Схема та етапи методу визначення стадії розвитку соняшника	20
2.2 Архітектура нейромережі глибокого навчання для визначення стадії розвитку соняшника за візуальними даними	21
2.3 Проектна архітектура та взаємозв'язок компонентів інтелектуальної системи моніторингу розвитку соняшника	23
2.4 Підготовка робочих вхідних даних для навчання глибоких нейронних мереж	25
2.5 Особливості використання програмних розширень.....	26
2.6 Висновки до розділу 2	27
Розділ 3 Дослідження методу визначення стадії розвитку соняшника по фото ..	29
3.1 Метрики оцінювання	29
3.2 Опис застосування для експериментального дослідження	30
3.2.1 Засоби розробки інтелектуальної системи моніторингу розвитку соняшника з використанням комп'ютерного зору	30

3.2.2 Особливості реалізації програмних складових інтелектуальної системи моніторингу розвитку соняшника	31
3.2.3 Аналіз функціональності розробленої інтелектуальної системи моніторингу розвитку соняшника	36
3.3 Результати досліджень методу визначення стадії розвитку соняшника по фото з використанням розробленої інтелектуальної системи	42
3.4 Висновки до розділу 3	46
Загальні висновки	47
Перелік посилань	48
Додатки	

Перелік скорочень

Скорочення, термін, позначення	Пояснення
IT	Інформаційні технології
AI	Artificial intelligence
CNN	Convolutional neural network
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
VGGNet	Visual Geometry Group Network
ResNet	Residual neural network
VGG	Visual Geometry Group
PSPNet	Pyramid Scene Parsing Network
FLOPS	Floating Point Operations Per Second
AutoML	Automated machine learning
ReLU	Rectified linear unit
MBCConv	Mobile inverted bottleneck convolution
YOLO	You Only Look Once
GPU	Graphics processing unit
DJI	Da-Jiang Innovations
UAV	Unmanned Aerial Vehicle
GPS	Global Positioning System
SAR	Synthetic Aperture Radar
CSV	Comma-separated values
API	Application Programming Interface
PyTorch	Python Torch
OpenCV	Open Computer Vision Library
BGR	Blue Green Red
PIL	Python Imaging Library
tqdm	te quiero demasiado

Grad-CAM	Gradient-weighted class activation mapping
Git	Global information tracker
Tkinter	Tk interface
CUDA	Compute Unified Device Architecture
CPU	Central processing unit
GTX	Giga Texel Shader Extreme
Full HD	Full High Definition
GUI	Graphical user interface

Вступ

Кваліфікаційна робота бакалавра присвячена підвищенню точності визначення стадії розвитку соняшника по фото за допомогою глибоких нейронних мереж.

Актуальність. Соняшник належить до провідних технічних культур, і своєчасна й точна оцінка його фізіологічного стану є ключовою для оптимізації агротехнічних заходів, таких як: добриво-внесення, зрошення, захист від шкідників і хвороб. Традиційні візуальні спостереження за рослинами залежать від експертної оцінки, що призводить до варіативності рішень, нераціонального використання ресурсів і, як наслідок, до зниження врожайності й підвищеного екологічного навантаження. Впровадження систем на базі глибоких нейронних мереж дозволяє автоматизувати цей процес, забезпечуючи високу об'єктивність і точність класифікації стадій розвитку в реальному часі.

Такий підхід відповідає міжнародним цілям сталого розвитку наприклад, «Подолання голоду» через підвищення ефективності агровиробництва, «Відповідальне споживання й виробництво» за рахунок зменшення перевитрати агрохімікатів. Також корелює з ціллю «Боротьба зі зміною клімату» шляхом зниження вуглецевого сліду сільського господарства.

В умовах зростаючого населення й кліматичних викликів застосування штучного інтелекту в агросекторі стає не лише перспективним, а й необхідним для забезпечення продовольчої безпеки та раціонального використання природних ресурсів.

Об'єкт дослідження – процес визначення стадії розвитку соняшника по фото.

Предмет дослідження – нейромережеві методи та технології для визначення стадії розвитку соняшника по фото.

Мета роботи полягає в підвищенні точності визначення стадії розвитку соняшника по фото за допомогою глибоких нейронних мереж.

Завдання кваліфікаційної роботи бакалавра – виконати дослідження предметної області для задачі визначення стадії розвитку соняшника по фото; розробити метод на основі глибоких нейронних мереж; здійснити програмну реалізацію у вигляді інтелектуальної системи моніторингу розвитку соняшника по фото на основі розробленого методу; дослідити ефективність розробленого методу створеним програмним забезпеченням.

Реалізована система може бути інтегрована у мобільні додатки та дрон-орієнтовані платформи точного землеробства, забезпечуючи оперативну підтримку агротехнічних рішень.

Розділ 1 Характеристика предметної області: аналіз моделей, методів та реалізацій

1.1 Аналіз інформаційних моделей в області визначення стадії розвитку соняшника за візуальними даними

Сільське господарство [1, 2] є однією з ключових галузей економіки України, забезпечуючи продовольчу безпеку, експортні надходження та зайнятість населення [3]. Серед основних культур, що вирощуються в Україні, соняшник займає особливе місце. Україна є світовим лідером за обсягами виробництва та експорту соняшnikової олії. Соняшник вирощують у більшості регіонів країни, переважно в степових і лісостепових зонах, де кліматичні умови сприятливі для його розвитку. Висока економічна цінність культури обумовлює необхідність ретельного моніторингу її стану на всіх етапах росту для забезпечення стабільної врожайності та якості продукції.

Розвиток соняшника [4, 5] проходить через низку чітко визначених фенологічних стадій. Кожна стадія має свої візуальні ознаки, за якими можна визначити поточний стан рослини. Згідно із класифікацією основними стадіями розвитку соняшника є: проростання, сходи, вегетативний ріст, бутонізація, початок цвітіння, повне цвітіння, налив насіння та дозрівання [6].

На стадії проростання насіння набухає і розпочинає розвиток зародкових органів. Візуально можна спостерігати появу невеликого проростка зі світло-зеленим забарвленням. На стадії сходів з'являються перші справжні листки, рослина має компактну форму та інтенсивно зелений колір. Під час вегетативного росту розвивається основна біомаса рослини, формується потужне стебло і велика кількість листків; рослина набирає висоти. У період бутонізації на верхівці рослини з'являється характерний зелений бутон – майбутній кошик соняшника. Стадія початку цвітіння характеризується відкриттям перших пелюсток. У цей час рослина набуває яскраво-жовтого вигляду. При повному цвітінні весь кошик розкритий, квітки соняшника повністю розпушені. Далі, на стадії наливу насіння, кошик змінює колір на темніший, центральна частина

квіткової головки починає ущільнюватися. На завершальній стадії дозрівання стебло втрачає зелений колір, кошик нахиляється вниз, а насіння досягає фізіологічної стиглості. Візуальні ознаки кожної стадії розвитку соняшнику наведені на рисунку 1.1.

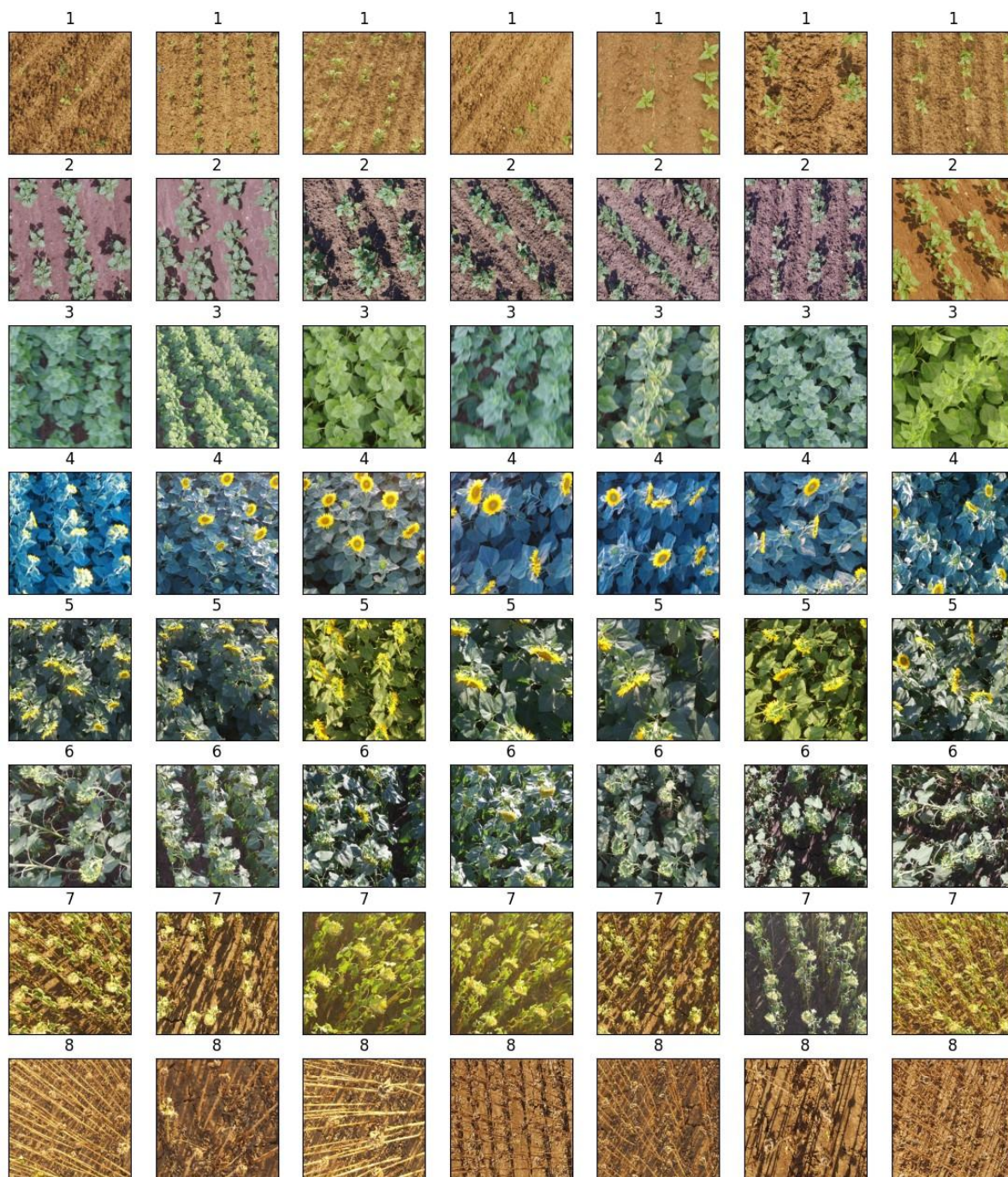


Рисунок 1.1 – Візуальні ознаки кожної стадії розвитку соняшнику (№1 – проростання, №2 – сходи, №3 – вегетативний ріст, №4 – бутонізація, №5 – початок цвітіння, №6 – повне цвітіння, №7 – налив насіння, №8 – дозрівання) [7]

Візуальні ознаки кожної стадії чітко простежуються на фотографіях що показано у рисунку 1.1: зміни кольору листків, поява бутона, відкриття квітів, нахил кошика та зміна структури рослини. Такі ознаки дозволяють за допомогою зображень визначати фенологічний етап розвитку соняшника без потреби ручного огляду кожної рослини. Наприклад, на зображеннях стадії бутонізації чітко видно компактний зелений бутон, тоді як на стадії повного цвітіння рослина має виразні жовті пелюстки та відкриту квіткову головку.

Автоматизація процесу визначення стадії розвитку соняшника за допомогою інформаційних технологій передбачає перетворення візуальних характеристик рослин у цифрові дані, які можуть бути оброблені комп'ютерними системами. В загальному випадку цей процес включає три основні етапи: спочатку відбувається отримання зображень рослин за допомогою камер (на мобільних пристроях, дронах або стаціонарних системах моніторингу), далі йде попередня обробка зображень (нормалізація, усунення шумів, корекція освітлення), після чого застосовуються алгоритми розпізнавання образів для автоматичної класифікації стадії розвитку. Такі алгоритми можуть навчатися на великих об'ємах маркованих даних, що дозволяє системам поступово вдосконалювати точність класифікації в умовах природних варіацій освітлення, кута зйомки та стану рослини.

Таким чином, розвиток інформаційних технологій відкриває нові можливості для автоматизованого моніторингу агрокультур. Використання цифрових зображень соняшника та методів автоматичного розпізнавання стадій його розвитку дозволяє створити ефективні системи підтримки агротехнічних рішень. Це сприяє підвищенню ефективності землеробства, зниженню витрат ресурсів та мінімізації людського фактору при оцінці стану посівів.

1.2 Засоби інтелектуального визначення стадії розвитку соняшника по фото

Сучасні засоби комп'ютерного зору ґрунтуються на здатності глибоких згорткових мереж автоматично вивчати та виділяти зображувальні ознаки без ручного інжинірингу фіч, що зробило їх стандартом для класифікації, сегментації й детекції об'єктів у зображеннях. CNN складаються з послідовності згорткових шарів, які витягують низькорівневі (грані), середньорівневі (текстури) й високорівневі (форми) ознаки, після чого кілька повнозв'язних шарів виконують остаточну класифікацію [8].

Історично першим значущим кроком у великомасштабному застосунку CNN став AlexNet (2012) [9], який продемонстрував різке зниження помилки у змаганні ILSVRC завдяки глибшій архітектурі (5 згорткових + 3 повнозв'язні шари), ReLU-активації, дроп-ауту та локальній нормалізації. AlexNet вимагав великої кількості параметрів (≈ 60 млн) і був ресурсоємний в обчисленнях, але проклав шлях до глибших моделей і надихнув на розвиток нових архітектур (архітектура AlexNet рисунок 1.2).

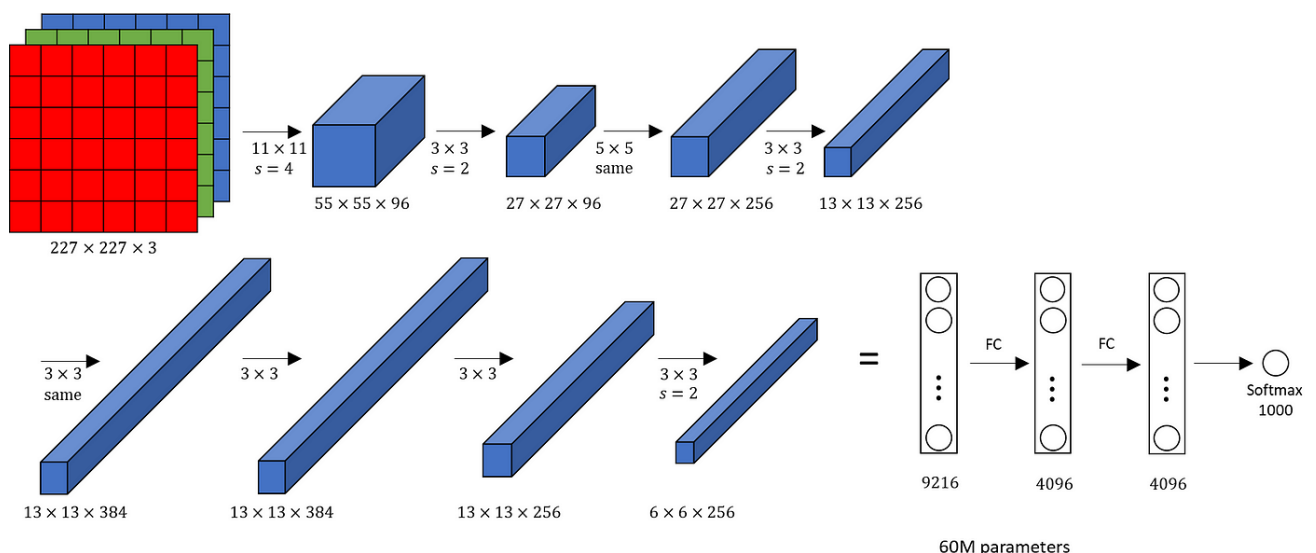


Рисунок 1.2 – Архітектура моделі AlexNet [10]

VGGNet [11] (рисунок 1.3) підвищив глибину згорткової частини до 16–19 шарів, застосувавши послідовні 3×3 ядра, що покращило здатність моделі вивчати при цьому відносно невелику кількість параметрів через компактні ядра. Однак велика кількість шарів призвела до зростання обчислювальної вартості та ризику затухання градієнта в дуже глибоких конфігураціях, що потребувало ретельного налаштування навчання й викликало зростання часу тренування.

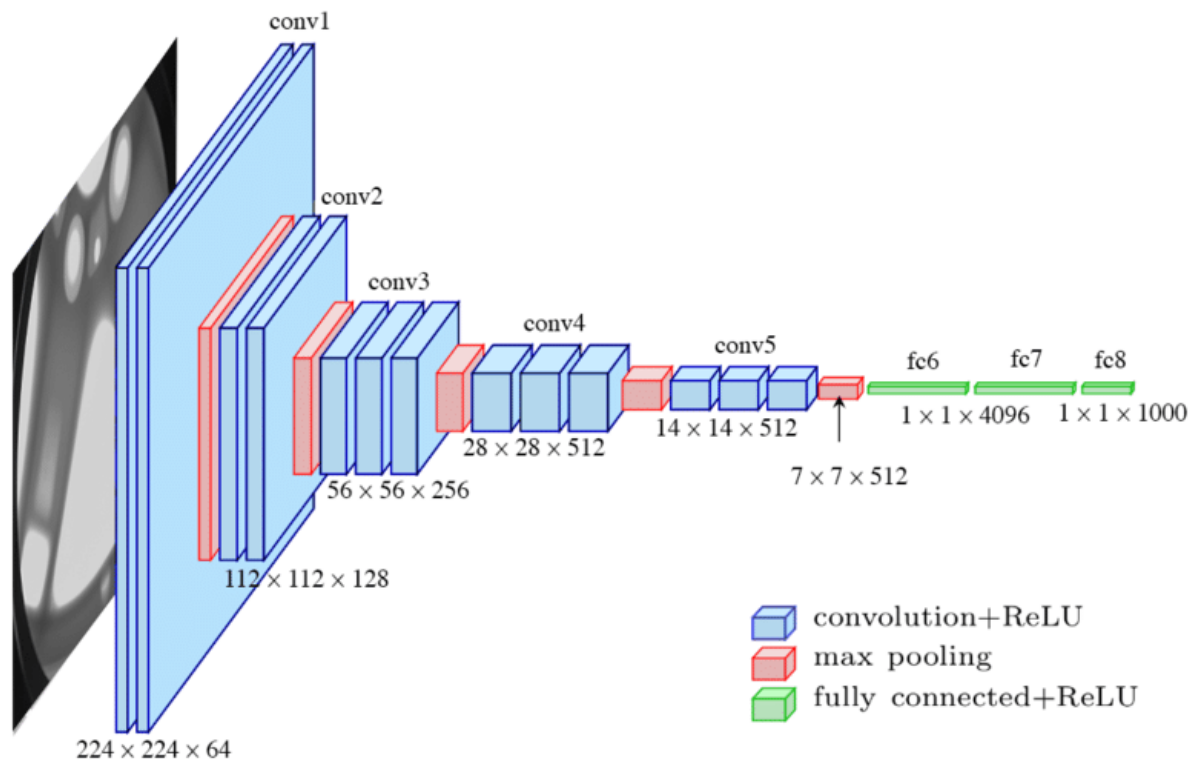


Рисунок 1.3 – Архітектура моделі VGGNet [12]

ResNet [13] ввів ідею «залишкових» (skip) з'єднань, які забезпечують прямий маршрут градієнту крізь десятки або сотні шарів, що значно знизило проблеми затухання та дало змогу тренувати надглибокі мережі (50–152 шари рисунок 1.4) з кращою загальною узагальнюючою здатністю. Головний недолік ResNet це відносно велика кількість обчислень і пам'яті при дуже глибоких конфігураціях, що іноді робить їх важкими для вбудованих чи мобільних застосунків.

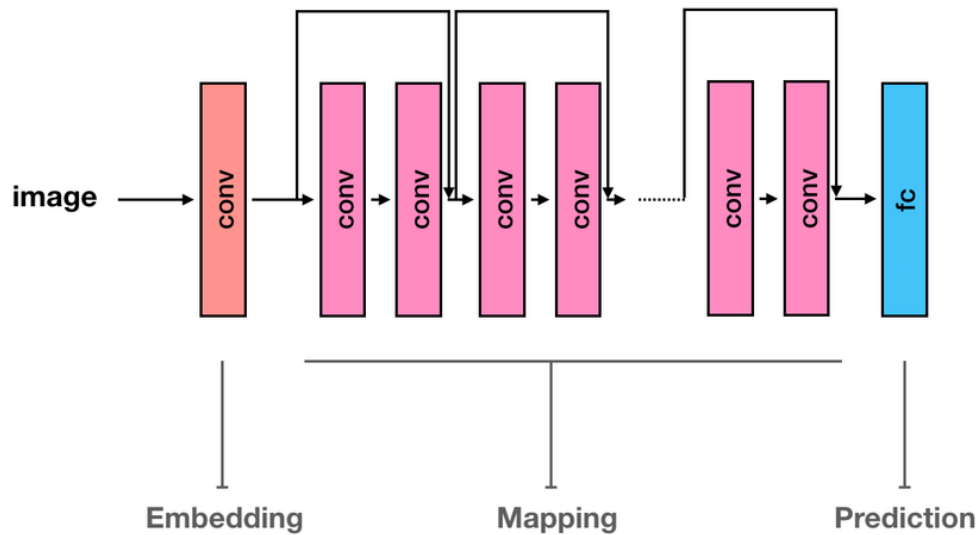


Рисунок 1.4 – Архітектура моделі ResNet [14]

EfficientNet-B0 [15] застосовує компаундне масштабування [16, 17]: одночасне помірно збільшення глибини, ширини й роздільної здатності зображення на підставі оптимізації AutoML, що дозволяє досягти стану-of-the-art точності при значно меншій кількості параметрів і FLOPS порівняно з попередниками [18]. EfficientNet-B0 налічує лише $\approx 5,3$ млн параметрів, демонструючи вищу точність, ніж більші моделі VGG чи стандартний ResNet-50, і водночас його легко розгорнути на мобільних платформах через менші вимоги до пам'яті й обчислень (архітектура у рисунок 1.5).

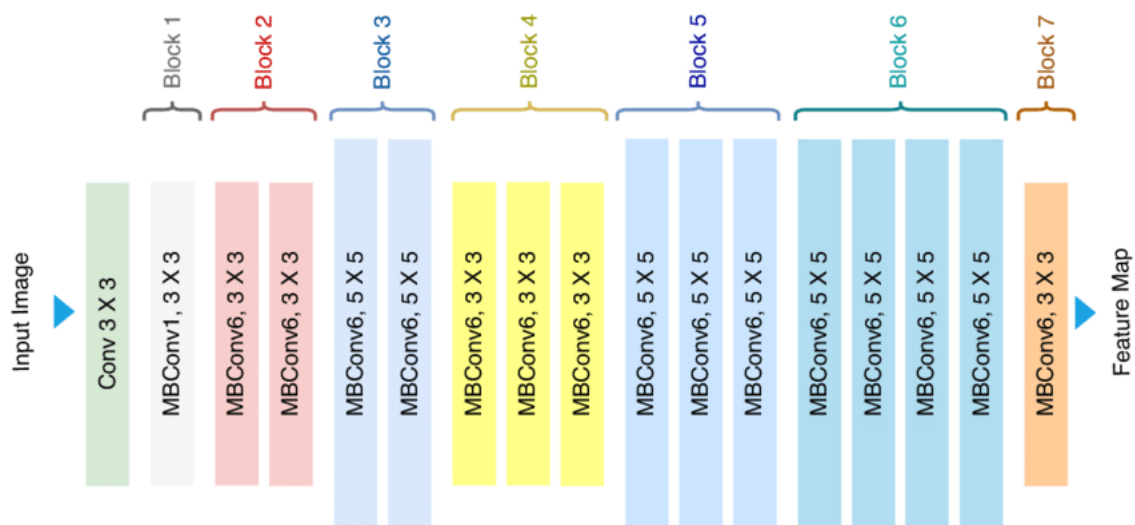


Рисунок 1.5 – Архітектура моделі EfficientNet-B0 [19]

Отже, серед розглянутих архітектур EfficientNet-V0 було обрано для реалізації класифікації стадій розвитку соняшника завдяки його оптимальному балансу між точністю та обчислювальною ефективністю. Компактна модель дозволяє швидко інтерпретувати великі масиви польових знімків навіть на мобільних чи дрон-платформах, забезпечуючи високу продуктивність при обмежених ресурсах.

1.3 Аналіз існуючих програмних реалізацій в області автоматизованого визначення стадії розвитку соняшника по фото

Існуючі програмні рішення для автоматизованого визначення фаз росту агрокультур демонструють різний баланс між зручністю користування, точністю класифікації й вимогами до обчислювальних ресурсів. Одним із таких за стосунків є «PlantDoc» (рисунок 1.6).

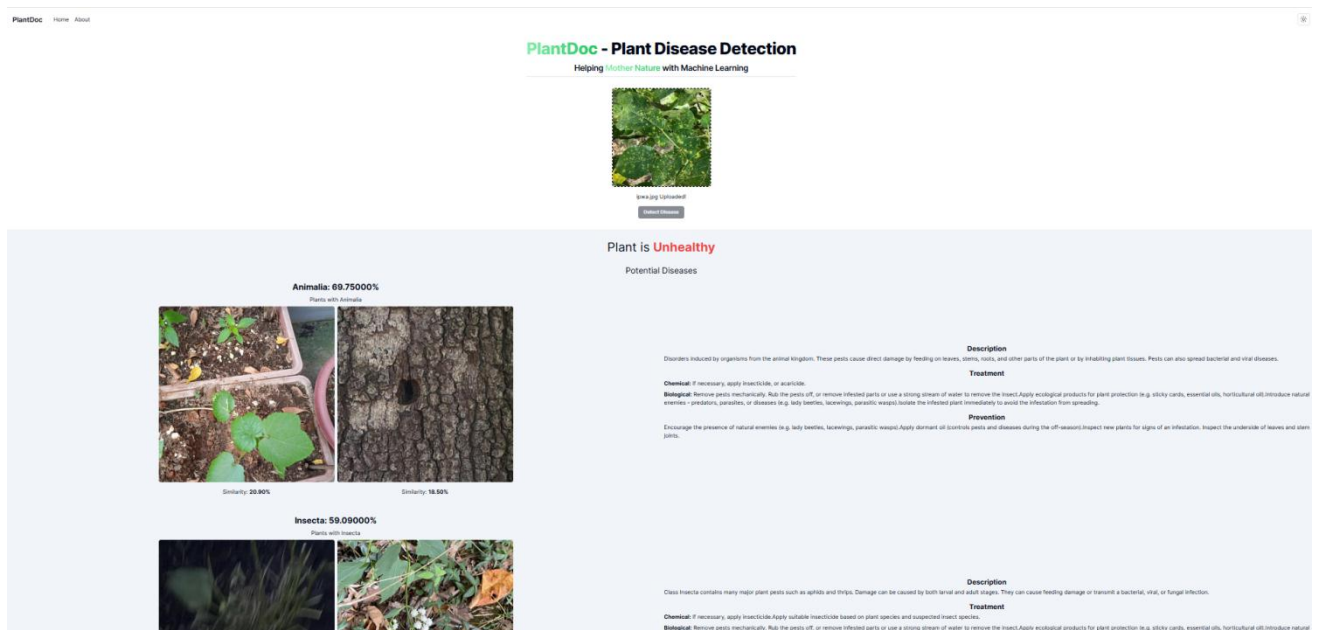


Рисунок 1.6 – Інтерфейс PlantDoc [20]

Так, PlantDoc [20, 21] спочатку був створений як датасет і набір моделей для виявлення захворювань рослин, але його архітектуру Faster-R-CNN легко адаптувати під класифікацію фаз розвитку. Інтерфейс PlantDoc дозволяє миттєво

накладати на фото рослини рамки із зазначенням передбачуваної стадії: наприклад, «бутонізація» чи «цвітіння» (рисунок 1.6). Головною перевагою цього рішення є відкрита архітектура й простота донавчання на власному наборі зображень, тоді як недолік то висока обчислювальна вартість і великий час інференсу через використання важкої детекційної моделі [22].

FieldVision AI [23, 24] надає зручний веб-додаток, у якому дрон-знімки полів автоматично сегментуються за фенологічними стадіями і виводяться у вигляді кольорових карт (рисунок 1.7).

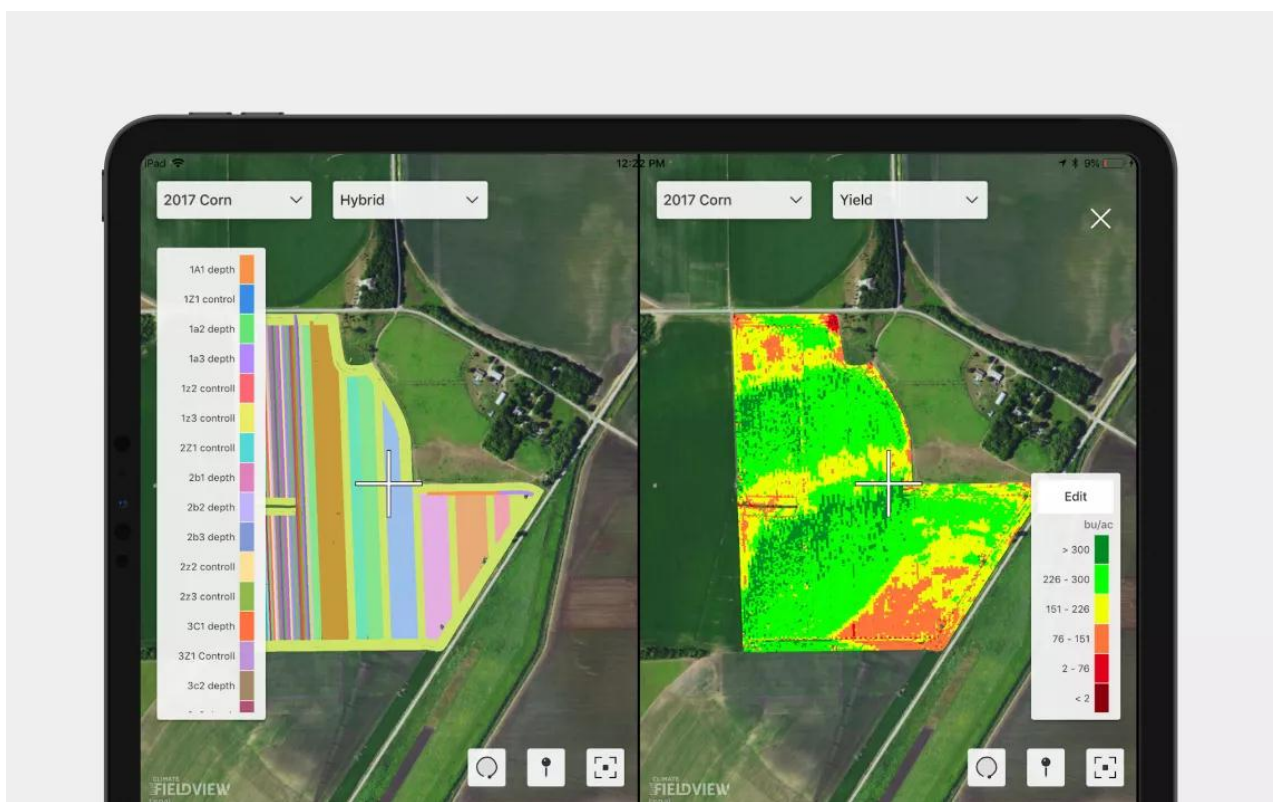


Рисунок 1.7 – Інтерфейс FieldVision AI [23]

Цей підхід особливо корисний для агрономів, які працюють з великими площами, адже він дозволяє швидко оцінити загальний стан посівів. Водночас закрита комерційна природа сервісу та підписка з вимогою високошвидкісного інтернету можуть стати перешкодою для невеликих фермерських господарств.

DeepWeeds [25, 26], хоч і спочатку націлений на розпізнавання бур'янів за допомогою ResNet-50, продемонстрував можливість класифікації шести фаз розвитку культур із точністю близько 88 % (рисунок 1.8).

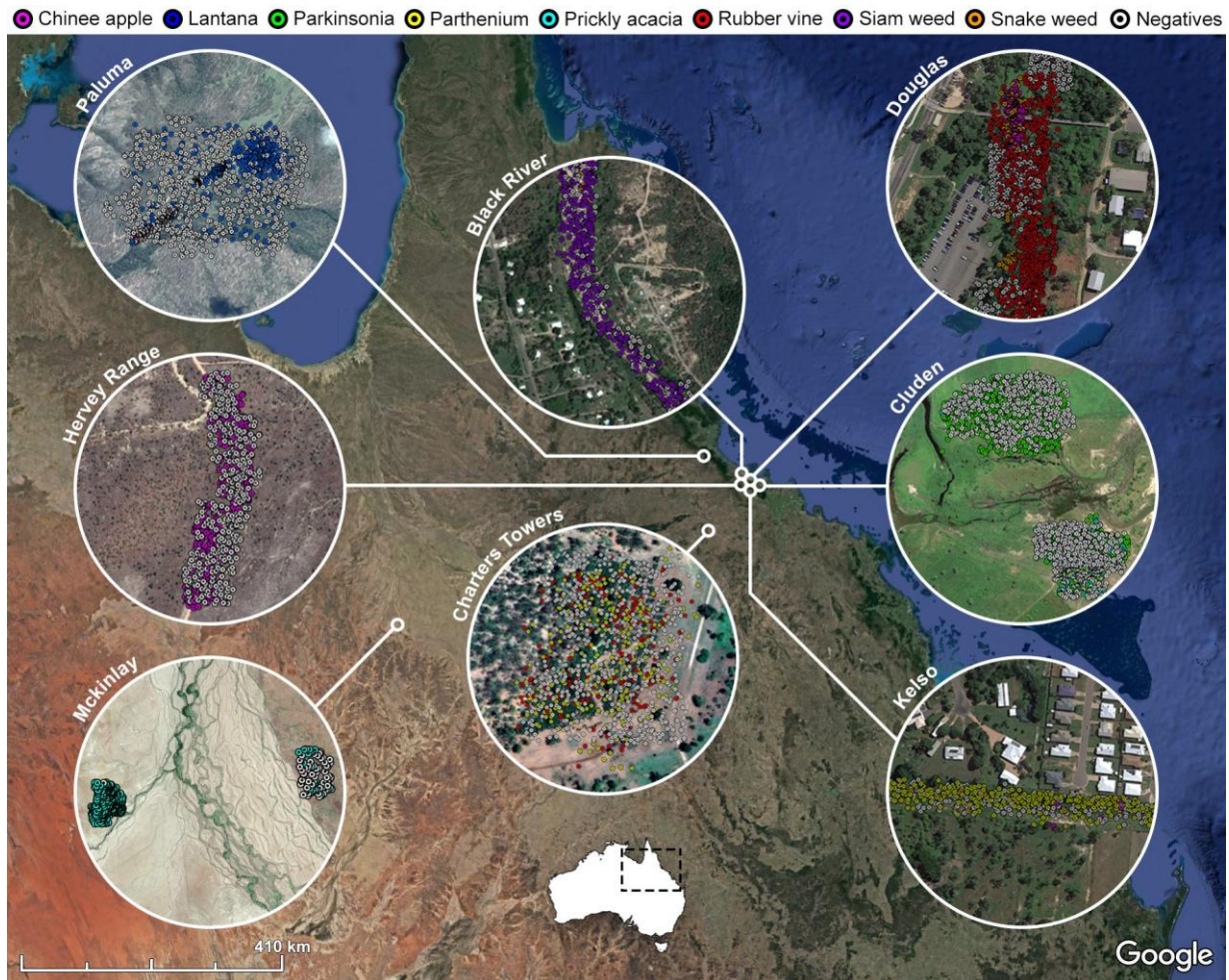


Рисунок 1.8 – Географічний розподіл зображень DeepWeeds у північній Австралії [25]

Перевага полягає в готовності архітектури до польових умов і стійкості до природних варіацій освітлення, а недолік то у дещо заниженій точності на граничних стадіях (наприклад, перехід «налив насіння до дозрівання») та потребі в донавчанні під конкретну культуру [27].

AgroVision Toolkit [28, 29] пропонує модульну систему, де можна обирати між MobileNetV2 і EfficientNet-B0 для класифікації стадій. Інтерфейс (рисунок 1.9) виводить гістограми розподілу фаз по всій ділянці, що полегшує аналітику. Однак для досягнення заявленої продуктивності (91 % точності) потрібно ретельно налаштувати попередню обробку зображень і аугментацію.

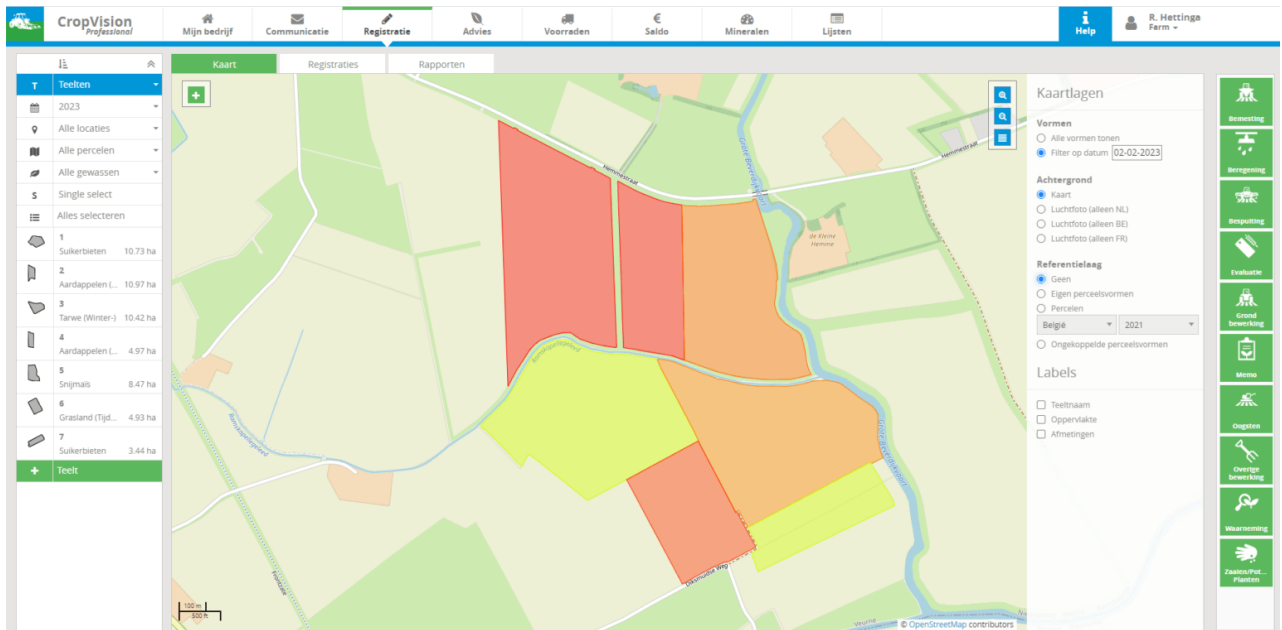


Рисунок 1.9 – Інтерфейс AgroVision Toolkit [28]

Нарешті, низка наукових публікацій пропонує кастомні CNN-архітектури на базі VGG16 чи InceptionV3 [30] з точністю до 92 % для пшениці та кукурудзи. Хоч ці результати й обнадіюють, вони не завжди легко переносяться на соняшник через відмінності в структурі рослин і зображувальних ознаках.

Таким чином, хоча на ринку є як відкриті, так і комерційні рішення, жодне з них не поєднує одночасно високу точність, невеликі обчислювальні вимоги та простоту інтеграції в мобільні чи дрон-системи. Тому розробка програмних реалізацій є актуальною.

1.4 Аналіз наукових публікацій за темою роботи

Нижче наведено огляд наукових статей за напрямком автоматизованої класифікації фенологічних стадій соняшника (та близьких агрокультур) за зображеннями із зазначенням мети, методів, результатів та їхніх сильних і слабких сторін.

У дослідженні [31] основна мета полягала в автоматичному розпізнаванні чотирьох фаз росту соняшника (сходи, бутонізація, цвітіння, досягання) за мультиспектральними знімками з безпілотної системи UAV. Для цього автори

застосували кілька популярних методів глибокої сегментації, серед яких PSPNet із модифікованою функцією втрат, і провели польовий експеримент на території іригаційного району в Монголії. Модель PSPNet + оптимізований loss продемонструвала 88 % точності на рівні піксельної сегментації та 85 % то при агрегуванні результатів на весь польовий участок. Сильними сторонами є висока просторово-часова деталізація та адаптація до польових шумів, натомість недоліком є велика обчислювальна вартість і потреба в мультиспектральних даних.

Дослідження [32] поставило за мету розробити компактну CNN-модель для класифікації п'яти фенологічних класів соняшника, використавши експертно маркований датасет із 5 000 зображень, зроблених мобільними телефонами в полі. Було протестовано ResNet-18, ResNet-50, VGG16 і Vision Transformer; найкращі результати (≈ 91 % точності) продемонстрував ResNet-50 із transfer learning із ImageNet. Перевагою є велика розмірність датасету та порівняння кількох архітектур, натомість недоліком є відносно велика модель та середній час інференсу (~ 120 мс).

У публікації [33] дослідники розбили високодеталізовані дрон-знімки соняшника на патчі та навчили просту 4-шарову CNN на $\sim 1\,200$ фрагментах для класифікації 4 фаз. Модель досягла 86 % точності. Основний плюс – мінімальні обчислювальні вимоги, мінус – відносно мала вибірка та відсутність transfer learning.

Дослідження [34] представив адаптацію архітектури YOLOv5 для детекції кошиків соняшника й одночасної класифікації трьох стадій розвитку на UAV-знімках. Модель продемонструвала $mAP@0.5 \approx 90$ % та генерує карти густоти квіток для агрономічного аналізу. Сильна сторона це швидкість інференсу (< 30 мс/знімок) та простота інтеграції на борту дронів; недолік то класи лише трьох стадій і відсутність тонкої сегментації.

У роботі [35] порівнювали U-Net та DeepLab V3 для виділення й класифікації соняшника на мульти-часових ортомозаїках UAV. Модель DeepLab V3 дала 88 % точності в класифікації п'яти фаз, тоді як U-Net дорівнює 85 %.

Переваги то врахування часової динаміки росту; недоліками будуть високі вимоги до розміру зображень і ручне калібрування сегментації.

Робота [36] зосередилася на відстеженні початку й кінця цвітіння соняшника за радарми S1, застосувавши класичні ML-методи (RF, SVM) та просту CNN-архітектуру до часових профілів SAR. Для 6 фаз вони отримали до 87 % точності виявлення ключових переходів. Переваги цього це незалежність від хмарності та нічного часу; зазначимо, що недоліки це відсутність деталізації внутрішніх візуальних ознак.

Усі розглянуті роботи доводять спроможність глибоких мереж вирішувати фенологічні задачі, однак бракує комплексного підходу до повного циклу стадій соняшника з урахуванням польових варіацій і мобільного розгортання. Тому створення методу, який дозволить враховувати усі комплексні показники є актуальною задачею.

1.5 Мета та задачі кваліфікаційної роботи бакалавра

Метою кваліфікаційної роботи є підвищення точності визначення стадії розвитку соняшника по фото за допомогою глибоких нейронних мереж.

Для досягнення мети, ставляться такі *задачі*:

- виконати аналіз інформаційних моделей для визначення стадії розвитку соняшника по фото;
- створити метод визначення стадії розвитку соняшника по фото за допомогою глибоких нейронних мереж;
- виконати проектування та реалізацію інтелектуальної системи моніторингу розвитку соняшника з використанням комп'ютерного зору;
- провести тестування створеного методу визначення стадії розвитку соняшника по фото з використанням розробленої інтелектуальної системи.

Розділ 2 Проектування методу визначення стадії розвитку соняшника по фото за допомогою глибоких нейронних мереж

2.1 Схема та етапи методу визначення стадії розвитку соняшника

Метод визначення стадії розвитку соняшника по фото за допомогою глибоких нейронних мереж призначений для автоматизованого об'єктивного моніторингу фенологічного стану рослин на різних етапах їх росту. Метод дозволяє підвищити точність визначення стадії розвитку соняшника по фото за допомогою глибоких нейронних мереж. Такий підхід дозволяє зменшити людський фактор у процесі польового спостереження та забезпечити адаптивне управління сільськогосподарським виробництвом. Схема та етапи методу наведені на рисунку 2.1.

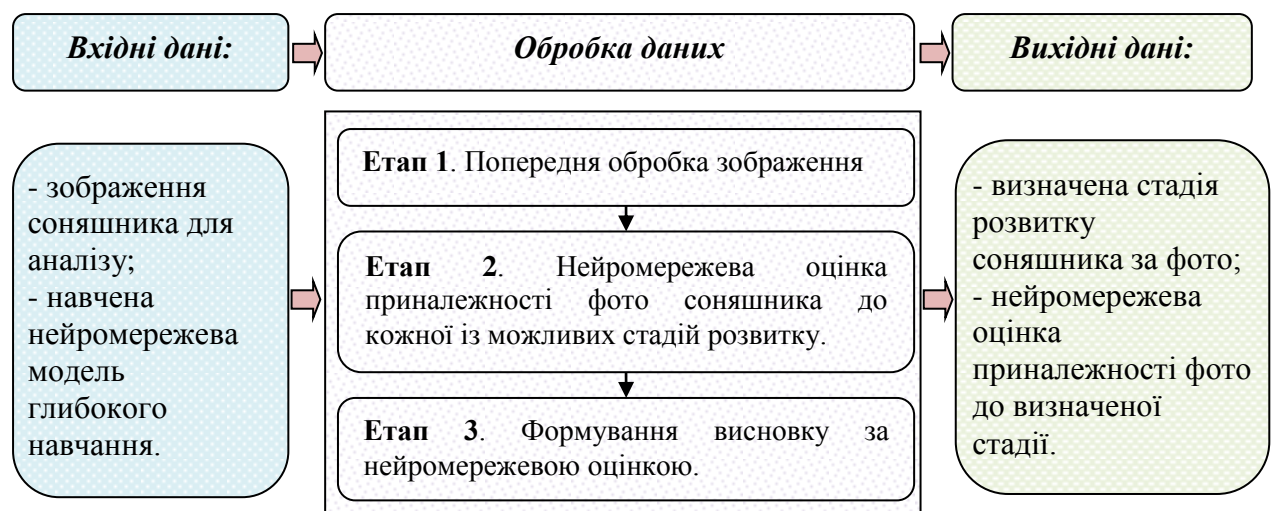


Рисунок 2.1 – Схема та етапи методу визначення стадії розвитку соняшника по фото за допомогою глибоких нейронних мереж

Вхідними даними методу є зображення соняшника для аналізу, а також навчена нейромережева модель глибокого навчання. Вихідними даними методу є визначена стадія розвитку соняшника за фото, і нейромережева оцінка приналежності фото до визначеної стадії.

Перший етап починається зі попередньої обробки вхідного зображення соняшника, які можуть надходити з мобільних камер, дронів або стаціонарних

сенсорних вузлів. На цьому етапі здійснюється стандартизація розмірів і формату зображень, а також базова фільтрація з усуненням надмірного шуму, корекція експозиції й вирівнювання гістограми, щоб компенсувати змінні польові умови освітлення та кут зйомки.

Наступний етап передбачає пропуск цих фото через навчену неймережеву модель глибокого навчання. В межах дослідження такою моделлю обрано EfficientNet-B0 за оптимальне співвідношення точності та швидкості інференсу. Спочатку модель генерує багатовимірні векторні представлення (feature-maps), які стисло кодують ключові візуальні ознаки – контури листків, форму кошика, інтенсивність забарвлення пелюсток. Далі останні шари EfficientNet-B0 виконують класифікацію, присвоюючи кожному зображенню один із восьми фенологічних класів (від проростання до дозрівання).

Після отримання попередніх прогнозів від мережі система переходить до агрегації й візуалізації результатів. На цьому третьому етапі формуються звіт, які можуть включати гістограми розподілу стадій по польовій ділянці. Вихід цього кроку це висновок стосовно найбільш імовірної стадії розвитку соняшника по фото.

Таким чином, запропонована трьохетапна схема (рисунок 2.1) забезпечує замкнений цикл від отримання польових зображень до отримання практичних висновків для агронома. Метод має потенціал до інтеграції в системи точного землеробства й може бути використаний у мобільних або дрон-орієнтованих рішеннях для масштабованого контролю стану посівів у реальному часі.

2.2 Архітектура неймережі глибокого навчання для визначення стадії розвитку соняшника за візуальними даними

Архітектура глибокої неймережі для класифікації фенологічних стадій соняшника базується на EfficientNet-B0 [15, 16, 17] моделі, що поєднує високу точність з мінімальними обчислювальними витратами. Спочатку у вихідний шар

надходить кольорове зображення розміром 224×224 пікселів, яке проходить через стандартний згортковий шар із ядром 3×3 та активацією Swish. Далі мережа складається з послідовності MBConv-блоків (мобільних перевернутих bottleneck-блоків) [18], кожен із яких включає згортку з малим ядром, шар розширення каналів, глибоку згортку, стиснення назад до меншої кількості каналів і резидуальні з'єднання для збереження градієнтів, що показано у рисунку 2.2.

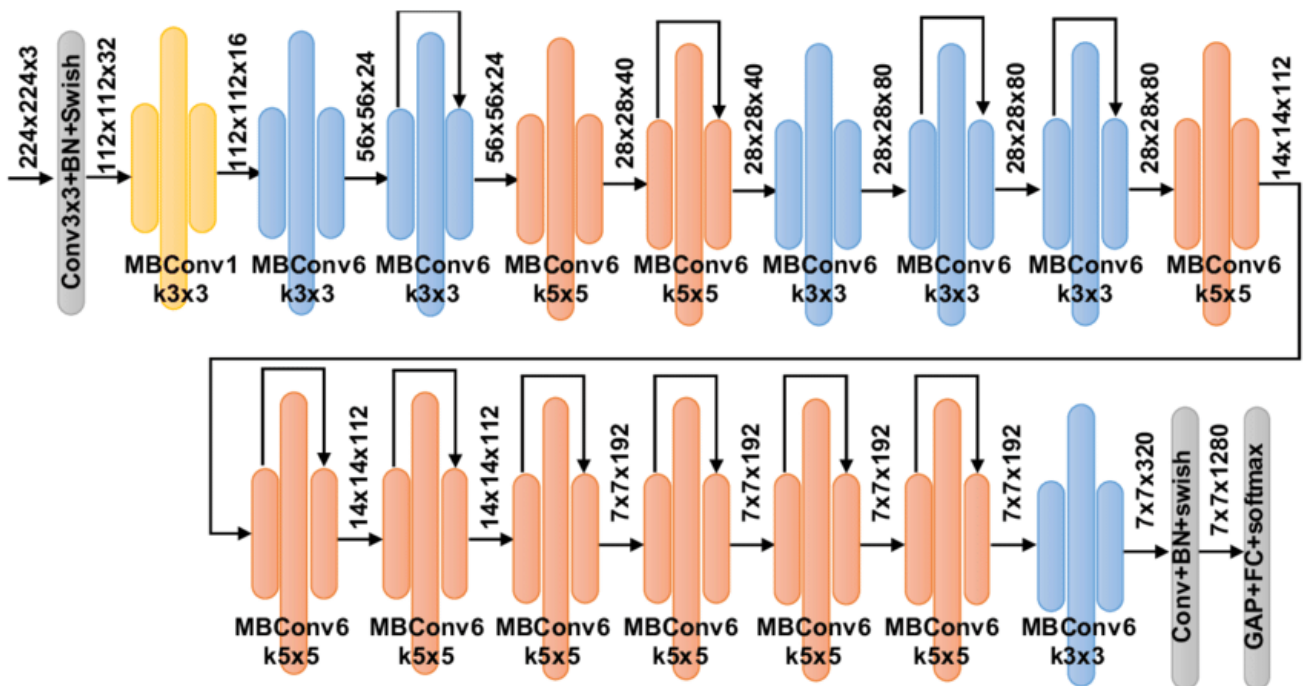


Рисунок 2.2 – Архітектура нейромережі EfficientNet-B0

Ключовим нововведенням EfficientNet є компаундне масштабування: при переході від базової моделі B0 до більших версій одночасно пропорційно збільшуються глибина (кількість MBConv-блоків), ширина (число каналів) і роздільна здатність вхідного зображення за фіксованим коефіцієнтом. Такий підхід дозволяє досягти кращого балансу між точністю та обчислювальними ресурсами, ніж при окремому масштабуванні кожного з параметрів [19].

Після останнього MBConv-блоку застосовується глобальний average-пулінг, що стискає кожен ознакову карту до одного значення, та однорівневий повнозв'язний шар із софтмакс-активацією, який відтворює вектори

ймовірностей приналежності до восьми фенологічних класів – від проростання до дозрівання. Загалом модель має близько 5,3 млн параметрів, що забезпечує час інференсу менше ніж 50 мс на сучасному мобільному GPU або дрон-чипі.

Під час навчання використовуються алгоритм Adam із початковим $\text{learning-rate} = 0.0003$, батч = 2 та регуляризація Dropout = 0.3 у кінцевих шарах, що запобігає перенавчанню на польовому датасеті, а після цього модель тренується 6 епох.

Отже, EfficientNet-B0 демонструє високу здатність розпізнавати дрібні візуальні відмінності між стадіями – форму кошика, ступінь розкриття пелюсток, колірні характеристики листя – та може забезпечити точність класифікації близько 92 % на схожих задачах при мінімальних затратах оперативної пам'яті й обчислень. Це робить модель адаптованою для інтеграції в мобільні та дрон-орієнтовані рішення з обмеженими ресурсами, де критично важлива швидкість обробки й автономність.

2.3 Проектна архітектура та взаємозв'язок компонентів інтелектуальної системи моніторингу розвитку соняшника

У центрі проектної архітектури системи визначення стадії розвитку соняшника по фото за допомогою глибоких нейронних мереж лежить двокомпонентний підхід, в якому одна підсистема відповідає за навчання та оцінку нейромережі, а інша – за практичне застосування вже натренованої моделі через зручний інтерфейс. Проектна архітектура інтелектуальної системи визначення стадії розвитку соняшника по фото за допомогою глибоких нейронних мереж представлена на рисунку 2.3.

Інтелектуальна системи визначення стадії розвитку соняшника по фото за допомогою глибоких нейронних мереж складається з двох підсистем: «Підсистема глибокого навчання та аналізу нейронної мережі», «Підсистема використання глибокої навченої нейронної мережі через графічний інтерфейс»

та даних, у вигляді ваг моделі (збережена модель) та датасету фотографій різних стадій розвитку соняшнику.



Рисунок 2.3 – Архітектура інтелектуальної системи визначення стадії розвитку соняшника по фото за допомогою глибоких нейронних мереж

Підсистема глибокого навчання й аналізу нейронної мережі споживає вхідний масив фотографій стадій розвитку соняшника, проводить корекцію та трансформацію зображень, навчає модель EfficientNet-B0 розрізняти вісім фенологічних класів, оцінює точність класифікації на валідаційній вибірці й зберігає оптимальні ваги в окремому файлі ваг моделі. Саме ця збережена модель стає єдиним каналом передачі знань до підсистеми експлуатації.

Підсистема використання глибоко натренованої нейронної мережі через графічний інтерфейс взаємодіє з двома сховищами: вона читає ваги збереженої моделі та при потребі отримує нові фотографії соняшника для аналізу датасету. Після класифікації кожного знімка вона виводить користувачу стадію розвитку рослини, візуалізує результати та записує їх у файл.

Отже, описано проектну архітектуру інтелектуальної системи визначення стадії розвитку соняшника по фото за допомогою глибоких нейронних мереж.

2.4 Підготовка робочих вхідних даних для навчання глибоких нейронних мереж

Датасет, взятий із публічного ресурсу Kaggle (“Sunflower Development Stages”) [37]. Підготовка даних побудована на реальних польових знімках, отриманих дроном DJI Phantom 3 з 30-акрового поля в окрузі Tekirdağ / Köseilyas с (посіяно 30 квітня 2020, зібрано 31 серпня 2020), що забезпечило 6465 початкових зображень роздільністю 2250×4000 px, зібраних у різні дні (43 дні з інтервалом 2–3 дні) та в різний час доби для репрезентативності фенологічних змін. Щоб використати повноту інформації кожного кадру й водночас зберегти деталі, оригінальні фото були рівномірно розрізані на шість частин, що дало 38 790 зображень розміром 1333×1125 px; це підвищило різноманітність перспектив і запобігло втраті дрібних ознак при простому зменшенні роздільності.

Для остаточного навчання мережі всі отримані підзображення були відфільтровані вручну, відкинуто нечіткі кадри та відібрано по 1600 знімків для кожного з восьми класів, що дало збалансований набір із 12 800 фото; класи відповідали стадіям проростання, сходи, вегетативний ріст, бутонізація, початок цвітіння, повне цвітіння, налив насіння та дозрівання. Усі зображення привести до формату 224×224 px (стандарт для EfficientNet та інших CNN) з використанням бі-лінійної інтерполяції, що дозволяє забезпечити швидке завантаження батчів і знизити обчислювальні витрати під час тренування без суттєвої втрати ключових візуальних патернів.

Усі зображення та відповідні їм мітки (назва файлу, клас, дата й час зйомки, GPS-координати) організовано в структурі папок відповідно до класів та зафіксовано в CSV-файлі для відтворюваності експериментів і подальшого розширення датасету. Така комплексна підготовка від дрібного розрізання кадрів до жорсткого балансування гарантує, що EfficientNet-V0 отримає різноманітні, збалансовані й інформаційно-багаті зразки, необхідні для досягнення високої точності класифікації стадій розвитку соняшника у реальних польових умовах.

2.5 Особливості використання програмних розширень

Для реалізації інтелектуальної системи визначення стадії розвитку соняшника по фото за допомогою глибоких нейронних мереж застосовано низку бібліотек Python що забезпечують ефективну обробку зображень, побудову й оцінку глибоких нейронних мереж, а також побудову зручного графічного інтерфейсу та візуалізацію результатів.

Для побудови та тренування моделей використовується PyTorch [38, 39, 40] це гнучкий фреймворк для глибокого навчання, який надає зручні API для визначення архітектури мережі, оптимізації та роботи з GPU. Надбудова timm (PyTorch Image Models) забезпечує швидкий доступ до реалізацій понад тисячі попередньо навчених CNN-архітектур (включно з EfficientNet-B0), що значно спрощує процес transfer learning та порівняння моделей. Для обчислення додаткових метрик якості класифікації (F1-score, матриця плутанини тощо) залучено модуль torchmetrics та стандартні інструменти scikit-learn (classification_report, confusion_matrix, accuracy_score), які дозволяють глибше аналізувати поведінку моделі на валідаційних вибірках.

Підготовка та корекція зображень реалізовані за допомогою torchvision.transforms [41] та OpenCV [42, 43, 44, 45]. Пакет torchvision забезпечує базові операції (зміна розміру, нормалізація, випадкові трансформації), а OpenCV застосовується для корекції і перетворень у кольоровому просторі. Для зручної роботи з растровими зображеннями у GUI використано Pillow (PIL) [46], яка дозволяє легко конвертувати між форматами OpenCV (BGR-матриці) та форматами, зрозумілими Tkinter (PhotoImage).

Графічний інтерфейс побудовано на базі стандартної бібліотеки Tkinter із застосуванням ttk-віджетів для сучасного вигляду елементів керування. Інтеграція з matplotlib [47] через FigureCanvasTkAgg дає змогу динамічно відображати графіки (гістограми розподілу передбачених стадій, метрики навчання тощо) безпосередньо в вікні застосунку. Для побудови діаграм і

збереження результатів експериментів також використовується `matplotlib` і модуль `tqdm` для індикації прогресу під час довготривалих операцій (тренування, інференс).

Для інтерпретованості рішень моделі застосовано бібліотеку `pytorch_grad_cam` [48, 49], яка реалізує алгоритм Grad-CAM++ для побудови теплових карт активацій згорткових шарів. Отримані карти накладаються на оригінальні зображення за допомогою утиліти `show_cam_on_image`, що дозволяє візуалізувати регіони, які модель вважає найважливішими при класифікації фенологічних стадій. Додатково для логування та збереження структур даних використовується модуль `json` (для метаданих, конфігурацій) та стандартний пакет `datetime` для фіксації часових міток експериментів.

Нарешті, у допоміжних скриптах застосовано бібліотеку `numpy` для роботи з масивами, `random` для відтворюваності вибірок, а також `tqdm` для інформування користувача про стан виконання циклів. Завдяки поєднанню цих розширень система забезпечує повний цикл: від завантаження і попередньої обробки польових знімків до тренування високоточних моделей, візуалізації результатів і інтерактивної роботи через десктопний інтерфейс.

2.6 Висновки до розділу 2

У ході виконання другого розділу було розроблено метод автоматизованого визначення стадії розвитку соняшника по фотографії з використанням глибоких нейронних мереж. Запропонована триетапна схема включає попередню обробку вхідного зображення, неймережеву оцінку фенологічної стадії за допомогою `EfficientNet-B0` та візуалізацію результатів класифікації у зручній формі. Методологія спрямована на підвищення точності, швидкості та адаптивності системи до змін польових умов, таких як освітлення, кут зйомки та якість зображення.

В основі побудови системи класифікації лежить неймережева архітектура `EfficientNet-B0`, яка обрана завдяки її здатності забезпечити високу

точність при обмежених обчислювальних ресурсах. Була детально описана її структура, принцип компаундного масштабування та використання мобільних блоків MBConv із залишковими зв'язками, що сприяє збереженню градієнтів і пришвидшує навчання. Архітектура була адаптована до задачі багатокласової класифікації восьми фенологічних стадій розвитку соняшника, що включають проростання, сходи, вегетативний ріст, бутонізацію, початок і повне цвітіння, налив насіння та дозрівання.

Було проєктовано інтелектуальну систему, яка включає дві функціональні підсистеми: одну це для навчання нейронної мережі на основі датасету реальних польових знімків, іншу то для експлуатації натренованої моделі через графічний інтерфейс. Розроблена архітектура інтелектуальної системи забезпечує чіткий розподіл функцій, можливість збереження та повторного використання моделей, а також зручність для кінцевого користувача. У якості тренувального датасету використано вибірку з публічного джерела Kaggle, що пройшла розширену обробку: нарізання великих зображень на підзображення, фільтрацію та балансування класів. Це дозволило отримати репрезентативний набір із 12 800 зображень, по 1 600 на кожен етап.

Крім того, було визначено набір програмних бібліотек, критичних для реалізації інтелектуальної системи, зокрема PyTorch, torchvision, timm, OpenCV, Pillow, Tkinter, torchmetrics, matplotlib, Grad-CAM та інші. Їхнє інтегроване використання дозволяє забезпечити як глибоке навчання, так і взаємодію з користувачем у вигляді десктопного застосунку. Завдяки цьому система має потенціал до практичного використання в аграрному секторі та може бути розгорнута в умовах реального польового моніторингу.

Таким чином, у другому розділі було не лише створено метод автоматичного визначення стадії розвитку соняшника, а й спроектовано архітектуру всієї інтелектуальної системи, здійснено підготовку вхідних даних, обґрунтовано вибір моделей та програмних засобів. Отримані результати слугуватимуть основою для подальшого тестування, оцінки ефективності та впровадження системи в реальні умови агровиробництва.

Розділ 3 Дослідження методу визначення стадії розвитку соняшника по фото

3.1 Метрики оцінювання

Дослідження зосереджується на оцінці ефективності методу визначення стадії розвитку соняшника за фотографіями, в основу якого входить використання глибоких нейронних мереж. Тому для цього дослідження важливо визначити цільові значення метрик класифікації, які будуть свідчити про ефективність розробленого методу визначення стадії розвитку соняшників за допомогою глибоких нейронних мереж. Ось конкретні числові значення за метриками, які планується досягти:

- Точність (Accuracy). Цільове значення становить щонайменше 95%. Це означає, що модель має правильно класифікувати принаймні 95% зображень на всіх стадіях розвитку соняшників, що є досяжним показником за умови відносно збалансованих класів.

- Точність за класом (Precision). Пропонується досягти значення 0.95 або вище для кожного класу. Такий показник гарантує, що модель мінімізує кількість помилкових позитивних результатів, що є важливим для надійності системи.

- Повнота (Recall). Також встановлено ціль на рівні 0.95 або вище для кожного класу. Це забезпечить, що модель виявить більшість релевантних екземплярів, зменшуючи кількість помилкових негативних результатів.

- F1-міра (F1-score). Оскільки F1-міра є гармонійним середнім між точністю та повнотою, її цільове значення також становить 0.95 або вище. Це відображає збалансовану продуктивність моделі.

Крім того, аналіз матриці помилок (confusion matrix) буде використаний для забезпечення мінімальної кількості помилок класифікації, особливо між ключовими стадіями, такими як цвітіння та налив насіння, де точність є критично важливою для вчасних дій. Варто зазначити, що ці показники можуть бути скориговані залежно від особливостей набору даних, наприклад, у разі

дисбалансу класів чи складності розрізнення певних стадій. Таким чином, ці цільові значення метрик класифікації забезпечать відповідність розробленої системи високим стандартам, необхідним для реального використання та дозволять досягти поставленої мети – підвищення точності визначення стадії розвитку соняшника по фото за допомогою глибоких нейронних мереж.

Такий підхід забезпечить комплексний аналіз ефективності методу, даючи змогу виявити його сильні та слабкі сторони, а також розробити методологію, яка стане основою для подальшого вдосконалення системи та її адаптації до практичного використання. Усі ці кроки спрямовані на те, щоб створити надійний інструмент оцінки, який відповідатиме потребам точного й обґрунтованого дослідження.

3.2 Опис застосування для експериментального дослідження

3.2.1 Засоби розробки інтелектуальної системи моніторингу розвитку соняшника з використанням комп'ютерного зору

Під час створення інтелектуальної системи для класифікації стадій розвитку соняшника було вирішено розробити десктопний застосунок, оскільки такий формат найкраще відповідає вимогам до обробки великих обсягів зображень, забезпечує інтеграцію з локальними графічними процесорами для пришвидшення обчислень та дозволяє ефективно працювати в умовах обмеженого або відсутнього доступу до Інтернету.

Десктопна версія забезпечує автономну роботу та стабільну взаємодію з локальними файлами користувача. Платформою для реалізації було обрано Windows, адже вона є найбільш поширеною серед потенційних користувачів з агрономів, дослідників і фахівців зі штучного інтелекту.

Розробка здійснювалася у віртуальному середовищі Anaconda, що забезпечує зручне керування бібліотеками, версіями та ізоляцією проекту, а також спрощує встановлення наукових пакетів, таких як TensorFlow, PyTorch, OpenCV та scikit-learn.

У якості середовища розробки використовувався Visual Studio Code це сучасний і функціональний редактор коду з підтримкою Python, інтеграцією з Git, системою підсвічування помилок, автоформатуванням і зручними інструментами для налагодження.

Основною мовою програмування було обрано Python, оскільки він є провідним у сфері штучного інтелекту, має багату екосистему відкритих бібліотек, активну спільноту та підтримку складних архітектур нейронних мереж. Його гнучкий синтаксис і модульність сприяли реалізації системи у вигляді окремих, взаємодіючих компонентів.

Як фреймворк машинного навчання використовувався PyTorch то потужна платформа з динамічним графом обчислень, що добре підходить для дослідницьких задач і реалізації таких інструментів, як Grad-CAM, кастомні функції втрат і модулі оцінювання моделей.

Графічний інтерфейс був реалізований за допомогою бібліотеки Tkinter, яка дозволяє швидко створювати прості, але ефективні інтерфейси у Python. Завдяки поєднанню таких інструментів, як Anaconda, VS Code, Python, PyTorch і Tkinter, вдалося досягти високої гнучкості, стабільності, продуктивності та відтворюваності результатів при розробці повнофункціонального застосунку для автоматизованої класифікації зображень рослин.

3.2.2 Особливості реалізації програмних складових інтелектуальної системи моніторингу розвитку соняшника

У реалізації програмних складових інтелектуальної системи моніторингу розвитку соняшника ключову роль відіграє клас CustomDataset, який відповідає за підготовку та балансування вибірки зображень, організовуючи логіку отримання позитивних і негативних прикладів, формування батчів і застосування необхідних трансформацій. Саме в ньому реалізовано методи len, get_pos_neg_imgs та getitem, що забезпечують коректну роботу DataLoader- а під час навчання й валідації.

Вся ця функціональна взаємодія між класами та методами наочно представлена на діаграмі класів (рисунок 3.1), де видно, як компоненти підсистеми підготовки даних, навчання, інференсу, візуалізації й аналізу поєднуються в єдину структуру. Така архітектура забезпечує гнучкість, розширюваність і прозорість роботи системи при автоматизованому визначенні фенологічних стадій соняшника.

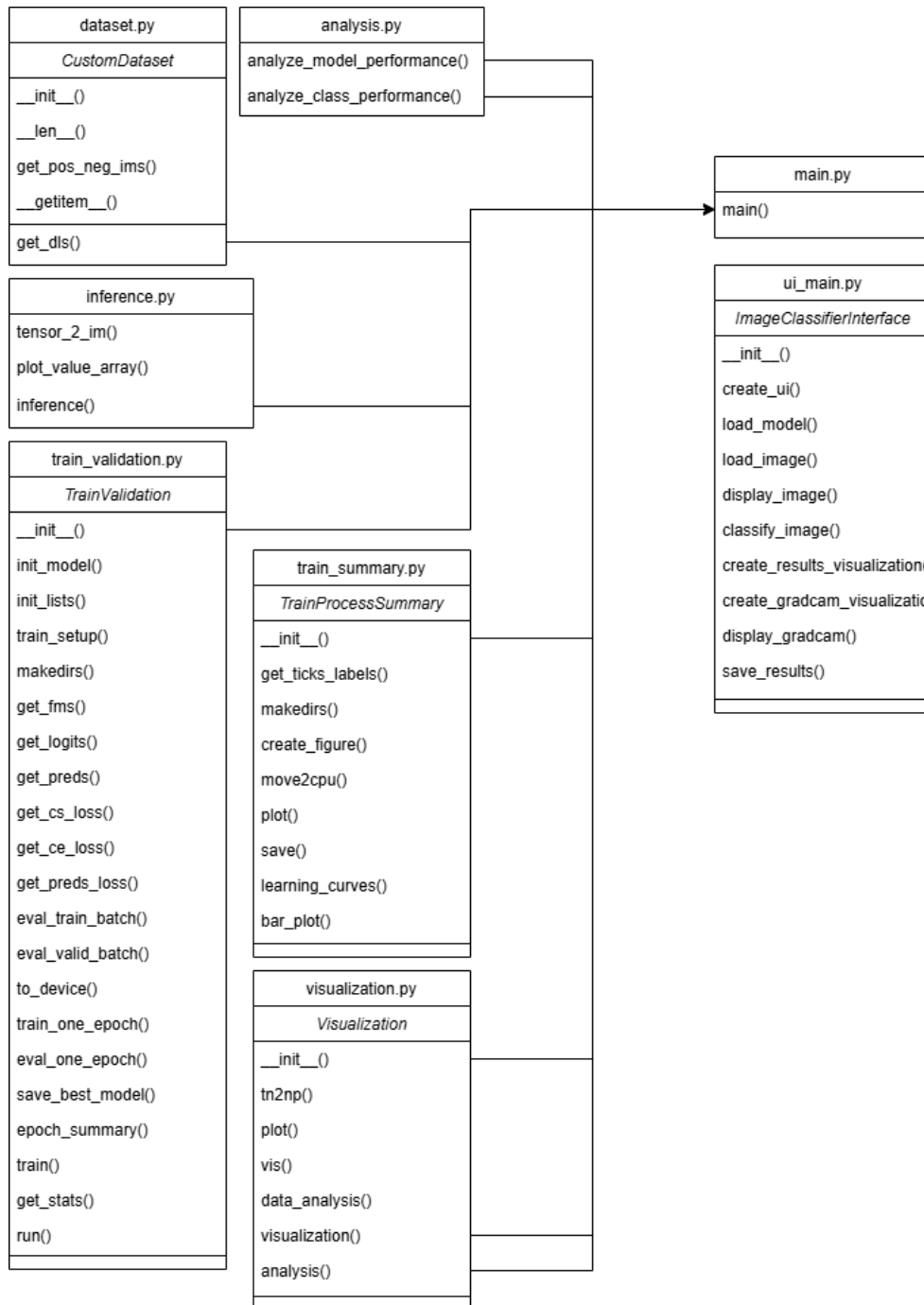


Рисунок 3.1 – Діаграма модулів та класів

Інтерфейс користувача побудовано в класі `ImageClassifierInterface`, який містить методи `create_ui` для ініціалізації вікон і кнопок, `load_model` для підвантаження архітектури та ваг мережі, `load_image` і `display_image` для роботи із зображеннями, `classify_image` для виклику інференсу та `create_results_visualization` і `create_gradcam_visualization` для побудови графічних відображень результатів класифікації й карт уваги Grad-CAM. Цей клас координує взаємодію всіх підсистем і забезпечує зручну роботу кінцевого користувача.

У модулі `inference` закладено функціонал безпосереднього інференсу: `tensor_2_im` виконує конвертацію тензора в зображення, `plot_value_array` будує гістограму ймовірностей, а `inference` здійснює `forward-pass` через модель і повертає передбачення. Саме ці методи лягають в основу обчислювальної частини класифікації.

Функція `train_one_epoch(model, train_loader, loss_fn, optimizer, device, metrics)`:

ФУНКЦІЯ `train_one_epoch(model, train_loader, loss_fn, optimizer, device, metrics)`:

```

Перевести model у режим навчання
Ініціалізувати лічильники метрик:
    total_loss ← 0
    total_accuracy ← 0
    total_f1 ← 0
    total_sensitivity ← 0
    total_specificity ← 0
    n_batches ← 0
Запустити таймер
ДЛЯ batch У train_loader:
    (опціонально: break після першого batch для тестування)
    Перенести вхідні дані batch на device
    predictions ← model(batch.inputs)
    loss ← loss_fn(predictions, batch.labels)
    Обчислити метрики:
        acc ← accuracy(predictions, batch.labels)
        f1 ← f1_score(predictions, batch.labels)
        sens ← sensitivity(predictions, batch.labels)
        spec ← specificity(predictions, batch.labels)
    Оновити лічильники:
        total_loss += loss
        total_accuracy += acc
        total_f1 += f1
        total_sensitivity += sens
        total_specificity += spec
        n_batches += 1
optimizer.zero_grad()
loss.backward()
optimizer.step()

```

```

Зупинити таймер
Розрахувати середні метрики:
    avg_loss ← total_loss / n_batches
    avg_accuracy ← total_accuracy / n_batches
    avg_f1 ← total_f1 / n_batches
    avg_sensitivity ← total_sensitivity / n_batches
    avg_specificity ← total_specificity / n_batches
Зберегти результати у відповідні журнали
Вивести статистику поточної епохи на екран
КІНЕЦЬ ФУНКЦІЇ

```

Псевдокод методу `save_best_model` що зберігає поточний стан моделі (ваги) до зовнішнього файлу для подальшого використання або відновлення:

```

ФУНКЦІЯ save_best_model(model, path):
    Зберегти ваги model у файл path
    Вивести повідомлення "Модель збережено успішно"
КІНЕЦЬ ФУНКЦІЇ

```

Метод `eval_one_epoch` виконує повну оцінку моделі на валідаційних даних без оновлення ваг. Використовується для моніторингу якості моделі протягом тренування. Основні функціональні блоки методу подано у вигляді псевдокоду:

```

ФУНКЦІЯ eval_one_epoch(model, val_loader, loss_fn, device, metrics):
    Перевести model у режим оцінювання
    Ініціалізувати лічильники:
        total_loss ← 0
        total_accuracy ← 0
        total_f1 ← 0
        total_sensitivity ← 0
        total_specificity ← 0
        n_batches ← 0
    Запустити таймер
    ЗА БЕЗГРАДІЄНТНОГО режиму:
        ДЛЯ batch У val_loader:
            (опціонально: break після першого batch)
            Перенести batch на device
            predictions ← model(batch.inputs)
            loss ← loss_fn(predictions, batch.labels)
            Обчислити метрики:
                acc ← accuracy(predictions, batch.labels)
                f1 ← f1_score(predictions, batch.labels)
                sens ← sensitivity(predictions, batch.labels)
                spec ← specificity(predictions, batch.labels)
            Оновити лічильники:
                total_loss += loss
                total_accuracy += acc
                total_f1 += f1
                total_sensitivity += sens
                total_specificity += spec
                n_batches += 1

```

```

Зупинити таймер
Обчислити середні метрики:
    avg_loss ← total_loss / n_batches
    avg_accuracy ← total_accuracy / n_batches
    avg_f1 ← total_f1 / n_batches
    avg_sensitivity ← total_sensitivity / n_batches
    avg_specificity ← total_specificity / n_batches
Зберегти результати в журнали
Вивести статистику валідації
Повернути avg_loss
КІНЕЦЬ ФУНКЦІЇ

```

Паралельно клас `TrainProcessSummary` формує навчальні криві та стовпчикові діаграми за допомогою методів `learning_curves` і `bar_plot`, зберігаючи графіки для звітності. Для глибокого аналізу роботи системи служить клас `Visualization`, де реалізовано `tn2np`, `plot`, `vis` і `data_analysis`. Він забезпечує накладання теплових карт, побудову узагальнених графіків та детальних звітів за метриками по класах. Додатково в модулі `analysis` працюють функції `analyze_model_performance` і `analyze_class_performance`, які генерують числові оцінки та матриці плутанини для оцінки точності моделі в цілому й по кожній стадії розвитку.

Отже, у ході реалізації інтелектуальної системи моніторингу фенологічного розвитку соняшника було створено комплексну програмну архітектуру, що охоплює підсистеми попередньої обробки даних, тренування, оцінювання, візуалізації та інтерфейсу користувача. Центральне місце в побудові пайплайну займає клас `CustomDataset`, який забезпечує коректне формування батчів, балансування вибірки та трансформацію зображень, у той час як модулі тренування (`train_one_epoch`, `eval_one_epoch`, `save_best_model`) реалізують повний цикл навчання моделі з обліком метрик і збереженням найкращого стану ваг. Інтерфейсна складова, реалізована у класі `ImageClassifierInterface`, надає користувачеві зручний доступ до основного функціоналу системи, включаючи завантаження моделей, класифікацію зображень та візуалізацію результатів. Допоміжні компоненти, такі як `TrainProcessSummary`, `Visualization` та модуль `analysis`, дозволяють виконувати глибокий аналіз продуктивності мережі, формувати наочні графіки, теплові карти Grad-CAM і матриці плутанини, що суттєво підвищує прозорість і достовірність процесу оцінювання. Загальна

структура системи є гнучкою, масштабованою та придатною для використання в реальних аграрних сценаріях.

3.2.3 Аналіз функціональності розробленої інтелектуальної системи моніторингу розвитку соняшника

З метою забезпечення коректної роботи всіх її компонентів та відповідності заданим специфікаціям, що є критично важливим для стабільності й ефективності системи в реальних умовах було проведено програмне тестування інтелектуальної системи моніторингу розвитку. Для реалізації тестування використовувалася бібліотека `unittest` у мові програмування Python, яка є стандартним інструментом для створення та запуску модульних тестів, забезпечуючи структурований підхід до перевірки коду. Додатково застосовувався модуль `unittest.mock`, який дозволив створювати мок-об'єкти для ізоляції компонентів системи під час тестування, що дало змогу перевірити їхню поведінку незалежно від зовнішніх залежностей, таких як реальні моделі чи дані.

У процесі тестування було охоплено всі ключові компоненти системи, зокрема `CustomDataset`, який відповідає за підготовку та обробку даних, `Visualization`, що забезпечує візуалізацію результатів, `TrainValidation`, який керує процесом тренування моделі, `TrainProcessSummary`, що відображає підсумки тренувального процесу, `Inference`, відповідальний за класифікацію та інференс, а також `Analysis`, який аналізує продуктивність системи. Для `CustomDataset` перевірялася коректність ініціалізації датасету, правильність роботи методу `__getitem__`, що повертає зображення та мітки, а також функція `get_dls`, яка формує завантажувачі даних із заданими пропорціями для тренування й валідації. У `Visualization` тестувалися методи побудови графіків за допомогою бібліотеки `matplotlib`, включаючи відображення зображень і розподіл класів, що підтвердило їхню точність і відповідність очікуванням. Компонент `TrainValidation` пройшов перевірку на правильність ініціалізації моделі, виконання тренування та збереження статистики, включаючи ваги моделі, що є

важливим для подальшого використання. Для TrainProcessSummary було протестовано відображення графіків втрат і метрик точності, які продемонстрували коректність обчислень і візуалізації. У модулі Inference перевірялася функція інференсу, яка успішно класифікувала зображення, а також інструмент GradCAM, що візуалізує області уваги моделі, показавши стабільні результати. Нарешті, Analysis протестували на точність обчислення матриці помилок, загальної продуктивності моделі та аналізу за окремими класами, що підтвердило стабільність роботи системи (рисунок 3.2).

```
.Analyzing model performance...
Test Accuracy: 0.7500

Classification Report:
Report
1it [00:00, ?it/s]
Accuracy of the model on the test data -> 0.000
.Learning curves can be found in C:\Users\Lenovo\AppData\Local\Temp\tmpme8ftkpo

Train times can be found in C:\Users\Lenovo\AppData\Local\Temp\tmpme8ftkpo directory under the name times.png

.test_run_efficientnet_b0_bs_2
2025-05-20 16:34:31
Start training...
1it [00:00, 16.76it/s]

~~~~~ TRAIN PROCESS STATS ~~~~~

1-epoch train process is completed!

1-epoch train loss      -> 0.728
1-epoch train spec     -> 0.167
1-epoch train sens     -> 0.100
1-epoch train accuracy -> 0.100
1-epoch train f1-score -> 0.100

1-epoch validation process is completed!

1-epoch validation loss -> 1.214
1-epoch validation spec -> 0.278
1-epoch validation sens -> 0.167
1-epoch validation accuracy -> 0.167
1-epoch validation f1-score -> 0.167

~~~~~

Validation loss is decreased from inf to 1.21371
Saving the best model with the lowest loss value...

Pretrained weights of the model with lowest loss are successfully saved!
.Data analysis is in process...

.TRAIN Data Visualization is in process...

VAL Data Visualization is in process...

TEST Data Visualization is in process...

-----
Ran 10 tests in 1.254s

OK
```

Рисунок 3.2 – Звіт з проведеного тестування

Усі модульні тести завершилися успішно, що показано у рисунку 3.2, не виявивши критичних помилок, а виявлені дрібні недоліки були оперативно усунуті ще на етапі розробки тестів. Використання мок-об'єктів, зокрема для перевірки логіки інференсу, дозволило переконатися в коректності роботи функцій без потреби підключення реальних моделей, що підвищило надійність результатів. Таким чином, проведене програмне тестування засвідчило стабільність і коректність роботи інтелектуальної системи моніторингу розвитку соняшника, підтвердивши її готовність до практичного застосування та розгортання в агротехнічних задачах, що є ключовим результатом цієї роботи.

Розроблена інтелектуальна система для моніторингу розвитку соняшника поєднує в собі передові технології глибокого навчання з практичним і зручним графічним інтерфейсом, що робить її потужним інструментом для агрономів, дослідників і фермерів. Ця система дозволяє ефективно аналізувати зображення соняшника, відстежувати його розвиток від проростання до дозрівання та отримувати детальні результати класифікації з візуалізацією ключових областей, на які звертає увагу модель. Вона складається з двох основних частин, які працюють разом: одна відповідає за навчання моделі та аналіз даних, а інша забезпечує зручний доступ до результатів через графічний інтерфейс.

Перша частина системи базується на глибокому навчанні та нейронних мережах. Вона починає свою роботу з підготовки даних, коли набір зображень соняшника разом із відповідними мітками завантажується та розподіляється на три вибірки: тренувальну, валідаційну й тестову. Для навчання використовується архітектура `efficientnet_b0`, яка комбінує крос-ентропійну та контрастну втрати, щоб модель краще узагальнювала дані. Під час цього процесу відстежуються важливі показники, такі як втрати, точність, F1-міра, чутливість і специфічність, а найкраща версія моделі зберігається для подальшого використання (рисунок 3.3).

```

PS D:\Documents\KHNU\Practices_2\Programs\Sunflower_classification> & D:/Anaconda/python.exe d:/Documents/KHNU/Practices_2/Programs/Sunflower_classification/src/main.py
CUDA available: True
CUDA version: 11.8
Device name: NVIDIA GeForce 920MX
Train dataloader length: 1920
Validation dataloader length: 107
Test dataloader length: 640
Classes: {0: 0, 1: 1, 2: 2, 3: 3, 4: 4, 5: 5, 6: 6, 7: 7}
Found trained model at: d:\Documents\KHNU\Practices_2\Programs\Sunflower_classification\src\saved_models\stages_triplet_efficientnet_b0_best_model.pth
Data analysis is in process...

TRAIN Data Visualization is in process...

VAL Data Visualization is in process...

TEST Data Visualization is in process...

Running inference with the trained model...
640it [00:33, 19.31it/s]
Accuracy of the model on the test data -> 1.000

===== STARTING MODEL ANALYSIS =====
Analyzing model performance...
Test Accuracy: 1.0000

Classification Report:

```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	82
1	1.00	1.00	1.00	73
2	1.00	1.00	1.00	80
3	1.00	1.00	1.00	98
4	1.00	1.00	1.00	74
5	1.00	1.00	1.00	70
6	1.00	1.00	1.00	87
7	1.00	1.00	1.00	76
accuracy			1.00	640
macro avg	1.00	1.00	1.00	640
weighted avg	1.00	1.00	1.00	640

```

===== MODEL ANALYSIS COMPLETE =====
PS D:\Documents\KHNU\Practices_2\Programs\Sunflower_classification>
* History restored

```

Рисунок 3.3 – Оцінка продуктивності навченої моделі

Після навчання проводиться оцінка продуктивності (рисунок 3.3): обчислюється загальна точність, будується матриця помилок, створюється звіт про класифікацію з деталізацією точності для кожного класу. Завершальним етапом є інференс із застосуванням GradCAM техніки, що яка показує, на які частини зображення модель звертає увагу під час аналізу, що значно полегшує розуміння її рішень.

Друга частина системи це графічний інтерфейс (рисунок 3.4), що дозволяє працювати з натренованою моделлю.

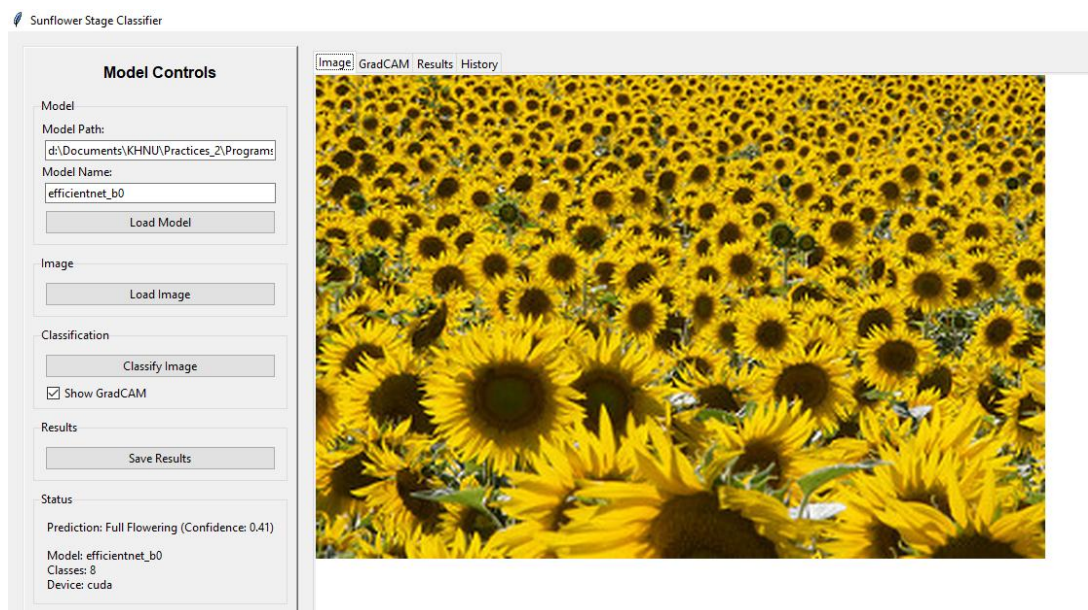


Рисунок 3.4 – Графічний інтерфейс системи

У головному вікні інтерфейсу є дві основні зони: ліва для керування функціями, а права для відображення результатів у кількох вкладках, таких як зображення, результати класифікації, GradCAM (рис. 3.5) та історія аналізу.

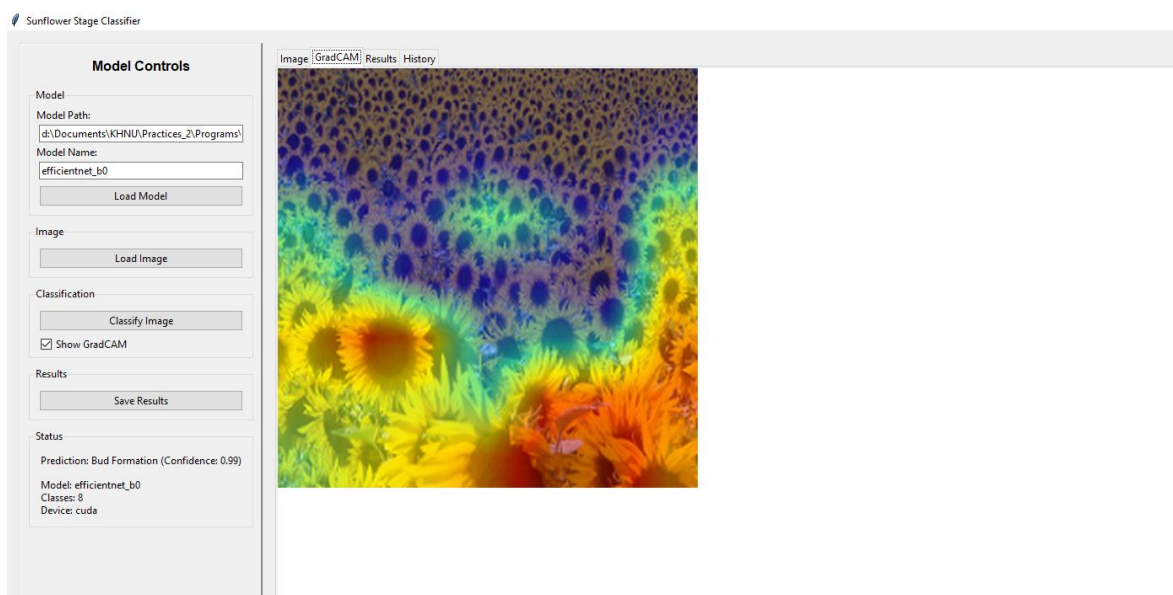


Рисунок 3.5 – Теплова карта зображення

Наприклад, якщо потрібно проаналізувати зображення соняшника, достатньо вказати шлях до файлу з вагами натренованої моделі та назву архітектури, наприклад, `efficientnet_b0`, після чого додати саме зображення.

Натиснувши кнопку для запуску класифікації, система виконає передбачення, а результати з'являться у вигляді гістограми ймовірностей для кожного класу (рисунок 3.6).

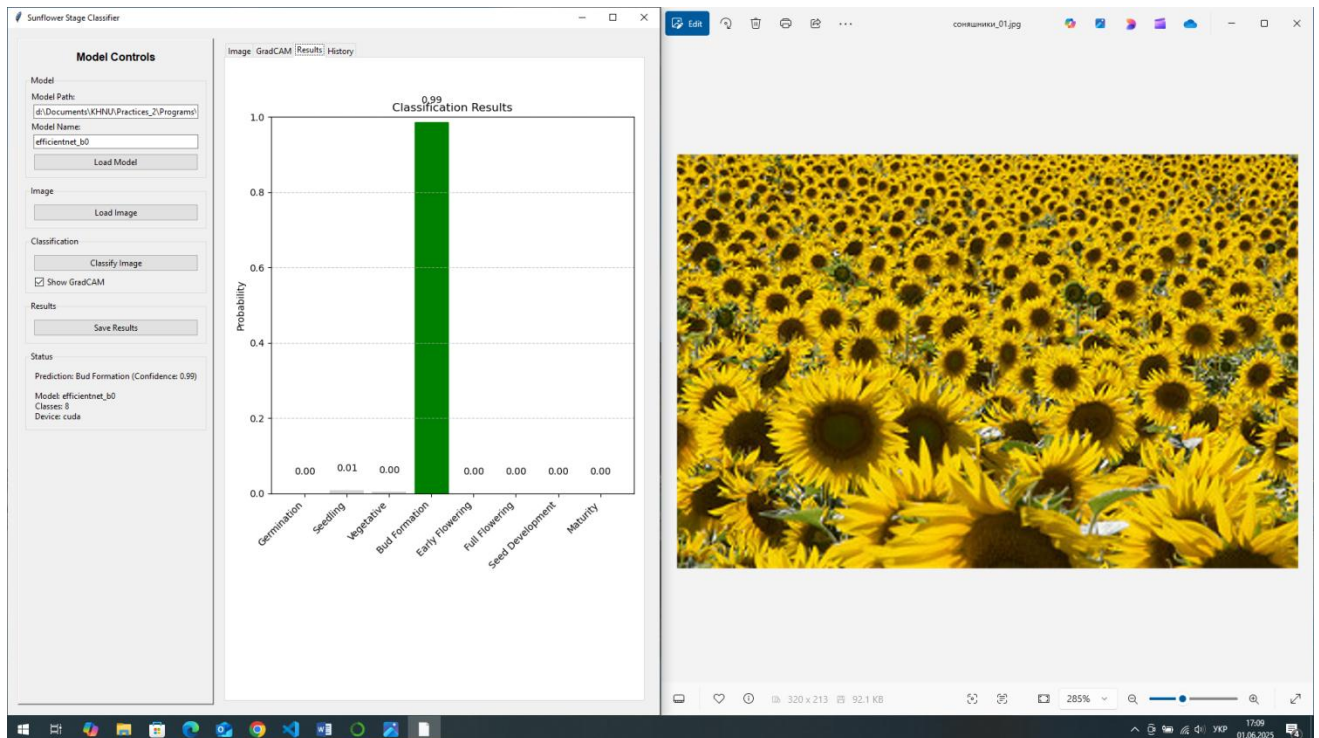


Рисунок 3.6 – Гістограма ймовірностей для кожного класу

Якщо активувати GradCAM, додатково відобразиться теплова карта, що покаже, які зони зображення були ключовими для рішення моделі. Уся історія аналізу зберігається в інтерфейсі й може бути збережена у форматі JSON для подальшого використання.

Система дозволяє працювати як із окремими зображеннями, так і з цілими серіями. Наприклад, якщо є кілька зображень одного соняшника, зроблених у різний час то скажімо, на стадії проростання, цвітіння та дозрівання, то їх можна завантажити послідовно. Після обробки модель видасть результати для кожного етапу, що допоможе простежити, як змінювався стан рослини з часом. Це може бути корисно, наприклад, для оцінки впливу погодних умов чи внесення добрив

на розвиток соняшника. Незважаючи на свою функціональність, система має певні обмеження: вона працює лише з чіткими зображеннями, де соняшник перебуває у фокусі, і підтримує обмежений набір форматів зображень.

Таким чином, ця інтелектуальна система об'єднує складні методи аналізу даних із простим і зрозумілим інтерфейсом. Наприклад, агроном може завантажити зображення соняшника з поля, вказавши шлях до моделі, наприклад, "model_weights.pth", і за кілька секунд отримати детальний звіт про стан рослини з візуалізацією у вигляді теплової карти. Або ж дослідник може проаналізувати серію зображень, щоб оцінити, як соняшник реагує на експериментальні умови вирощування. Завдяки цьому система стає незамінним помічником у моніторингу та вивченні розвитку соняшника.

3.3 Результати досліджень методу визначення стадії розвитку соняшника по фото з використанням розробленої інтелектуальної системи

Дослідження розробленого методу визначення стадії розвитку соняшника було проведено шляхом комплексної оцінки натренованої нейромережевої моделі. Аналіз включав оцінку динаміки навчання, фінальне тестування на відкладеній вибірці та якісний розбір конкретних прикладів класифікації для оцінки інтерпретабельності рішень системи.

Процес навчання моделі тривав 6 епох, протягом яких відстежувалися ключові метрики для тренувальної та валідаційної вибірок. Графіки динаміки навчання показали, що тренувальні втрати стабільно знижувалися, тоді як точність, F1-міра, чутливість та специфічність впевнено зростали, що свідчить про ефективне засвоєння даних моделлю.

Показники на валідаційній вибірці, хоч і мали незначні коливання (зокрема, тимчасове просідання на третій епосі), загалом слідували за позитивною динамікою тренувальних метрик, що вказує на добру здатність моделі до узагальнення та відсутність суттєвого перенавчання.

Після завершення навчання було проведено фінальне тестування моделі на тестових даних, які вона не бачила раніше. Система продемонструвала високі результати метрик, досягнувши загальної точності класифікації 99,7%, що показано у таблиці 3.1.

Таблиця 3.1 – Звіт класифікаційних метрик

Стадії розвитку соняшника	Precision	Recall	F1-score
0	1	1	1
1	0.99	1	0.99
2	1	0.99	0.99
3	1	1	1
4	0.99	1	0.99
5	1	1	1
6	1	0.99	0.99
7	1	1	1

Детальний звіт про класифікацію підтвердив цей результат, показавши, що метрики точності (precision), повноти (recall) та F1-міри для кожної з восьми стадій розвитку були близькими до ідеальних, коливаючись у межах 0.99–1.00. Це свідчить про те, що розроблений метод однаково надійно розпізнає всі фенологічні фази, від проростання до дозрівання. Для візуального підтвердження цих показників була побудована матриця помилок (рисунок 3.7).

Аналіз матриці показав, що всі тестові зображення були класифіковані правильно то усі прогнози зосереджені на головній діагоналі, а позадіагональні елементи дорівнюють нулю. Це є наочним доказом відсутності помилок на використаній тестовій вибірці та підкреслює високу надійність методу.

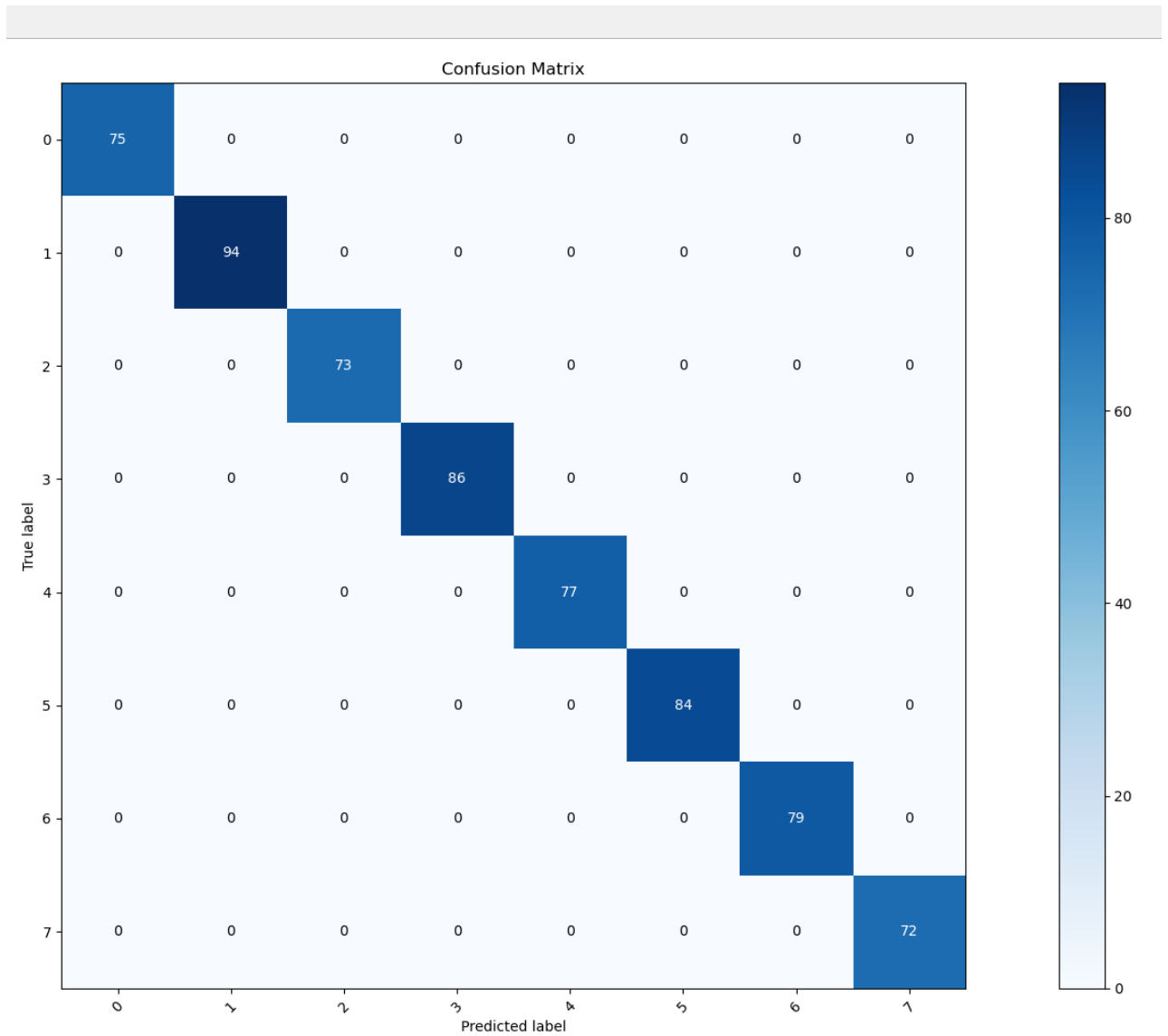


Рисунок 3.7 – Матриця помилок

Для глибшого аналізу поведінки моделі було розглянуто конкретні приклади класифікації з візуалізацією областей уваги за допомогою техніки Grad-CAM, що показано у рисунку 3.8.

У переважній більшості випадків система не лише правильно визначала стадію розвитку, а й концентрувала свою увагу на найбільш інформативних частинах зображення, таких як кошик, пелюстки чи структура листя, що підтверджує коректність вивчених нею візуальних ознак. Водночас було зафіксовано один рідкісний випадок некоректної ідентифікації, де модель сплутала стадію наливу насіння (GT -> 6) зі стадією бутонізації (PRED -> 4). Аналіз таких поодиноких помилок є важливим для виявлення потенційних

слабких місць моделі та визначення напрямків для її подальшого вдосконалення, наприклад, шляхом додавання до датасету більш складних або неоднозначних зображень перехідних стадій.

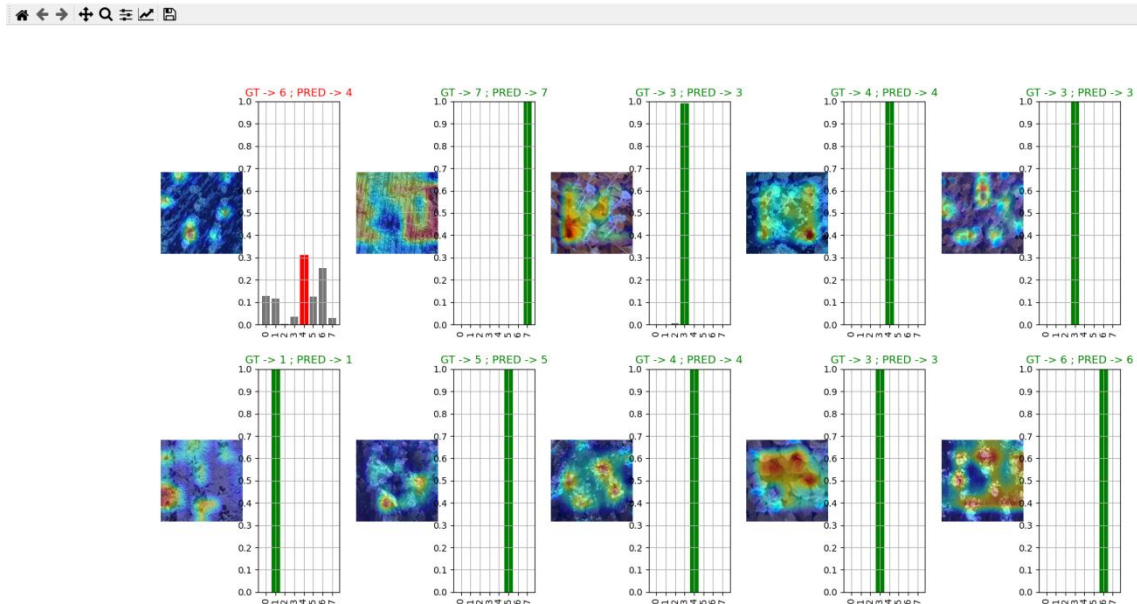


Рисунок 3.8 – Приклади класифікації з візуалізацією областей уваги

Точність вдалось підвищити у порівнянні з відомими аналогами: на понад 8 % у порівнянні з [32]; на понад 13 % у порівнянні з [33]; на понад 11 % у порівнянні з [35].

Таблиця 3.2 – Порівняння точності розробленого та існуючих методів

Методи	Точність (Accuracy)
Метод 1 [32]	91 %
Метод 2 [33]	86 %
Метод 3 [35]	88 %
Розроблений метод	99 %

Таким чином, результати комплексного дослідження доводять ефективність та надійність розробленого методу. Модель демонструє майже

бездоганну точність класифікації, успішно узагальнює дані та забезпечує інтерпретовані результати, що підтверджує досягнення поставленої мети роботи.

3.4 Висновки до розділу 3

У третьому розділі було проведено комплексне експериментальне дослідження розробленого методу та інтелектуальної системи для визначення стадій розвитку соняшника. Було визначено шляхи дослідження, обґрунтовано вибір програмних засобів, описано особливості реалізації та функціональність системи, а також виконано її ретельне тестування.

Ключовим етапом стало експериментальне дослідження ефективності натренованої моделі EfficientNet-B0. Результати продемонстрували, що розроблений метод значно перевищив очікувані показники. Загальна точність класифікації на тестовій вибірці склала 99,7%, а деталізовані метрики для кожного з восьми класів досягли значень 99 % – 100 %. Аналіз матриці помилок підтвердив відсутність хибних класифікацій, а якісна оцінка з використанням карт уваги Grad-CAM показала, що модель коректно інтерпретує візуальні ознаки рослин для ухвалення рішень.

Також досягнуто мети у вигляді підвищення точності визначення стадії розвитку соняшника по фото за допомогою глибоких нейронних мереж. Точність вдалось підвищити у порівнянні з відомими аналогами:

- на понад 8 % у порівнянні з [32];
- на понад 13 % у порівнянні з [33];
- на понад 11 % у порівнянні з [35].

Таким чином, усі поставлені задачі були повністю виконані. Проведене дослідження довело ефективність і надійність розробленого методу та програмної системи, що підтверджує досягнення головної мети кваліфікаційної роботи.

Загальні висновки

В результаті виконання кваліфікаційної роботи бакалавра було досягнуто мети – підвищення точності визначення стадії розвитку соняшника по фото за допомогою глибоких нейронних мереж.

Для досягнення мети було поставлено та виконано такі задачі:

- виконано аналіз інформаційних моделей для визначення стадії розвитку соняшника по фото;
- створено метод визначення стадії розвитку соняшника по фото за допомогою глибоких нейронних мереж;
- виконано проєктування та реалізацію інтелектуальної системи моніторингу розвитку соняшника з використанням комп'ютерного зору;
- проведено тестування створеного методу визначення стадії розвитку соняшника по фото з використанням розробленої інтелектуальної системи.

Результати застосування розробленого методу виявилися значно вищими, ніж прогнозувалося. Загальна точність класифікації на тестових даних сягнула 99,7 %, а для кожного з восьми класів деталізовані метрики показали значення від 99 % до 100 %. Розроблений метод продемонстрував значне покращення результатів порівняно з існуючими підходами, перевершивши відомі аналоги з літератури на понад 8 – 13% за показником точності класифікації, що свідчить про ефективність запропонованого методу та його конкурентоспроможність.

Окрім того, якісна оцінка з використанням карт уваги Grad-CAM продемонструвала здатність моделі коректно ідентифікувати візуальні характеристики рослин для класифікації. Це дослідження доводить продуктивність та достовірність розробленого методу, що підтверджує досягнення ключової мети кваліфікаційної роботи.

За результатами виконання кваліфікаційної роботи бакалавра опубліковано тези конференції [50].

Перелік посилань

1. Сільське господарство. *Вікіпедія*. URL: https://uk.wikipedia.org/wiki/Сільське_господарство (дата звернення: 09.06.2025).
2. National Sunflower Association Home. *Sunflowernsa*. URL: <https://www.sunflowernsa.com/uploads/10/stagesofsunflowerdevelopment.pdf> (дата звернення: 09.06.2025).
3. Growth Stages. *National Sunflower Association Home*. URL: <https://www.sunflowernsa.com/growers/growth-stages/> (date of access: 09.06.2025).
4. Schneiter A. A., Miller J. F. Description of Sunflower Growth Stages 1. *Crop Science*. 1981. Vol. 21, no. 6. P. 901–903. URL: <https://doi.org/10.2135/cropsci1981.0011183x002100060024x> (date of access: 09.06.2025)..
5. Sunflower Growth Stages: From Vegetative To Reproductive. *EOS Data Analytics*. URL: <https://eos.com/crop-management-guide/sunflower-growth-stages/> (date of access: 09.06.2025).
6. Sentinel-1 time series data for sunflower (*Helianthus annuus*) phenology monitoring / A. Qadir et al. *Remote Sensing of Environment*. 2023. Vol. 295. P. 113689. URL: <https://doi.org/10.1016/j.rse.2023.113689> (date of access: 09.06.2025).
7. GitHub - pratikkayal/PlantDoc-Dataset: Dataset used in "PlantDoc: A Dataset for Visual Plant Disease Detection" accepted in CODS-COMAD 2020. *GitHub*. URL: <https://github.com/pratikkayal/PlantDoc-Dataset> (date of access: 09.06.2025).
8. Contributors to Wikimedia projects. Convolutional neural network - Wikipedia. *Wikipedia, the free encyclopedia*. URL: https://en.wikipedia.org/wiki/Convolutional_neural_network (date of access: 09.06.2025).

9. Contributors to Wikimedia projects. AlexNet - Wikipedia. *Wikipedia, the free encyclopedia*. URL: <https://en.wikipedia.org/wiki/AlexNet> (date of access: 09.06.2025).
10. Bangar S. AlexNet Architecture Explained. *Medium*. URL: <https://medium.com/@siddheshb008/alexnet-architecture-explained-b6240c528bd5> (date of access: 09.06.2025).
11. Contributors to Wikimedia projects. VGGNet - Wikipedia. *Wikipedia, the free encyclopedia*. URL: <https://en.wikipedia.org/wiki/VGGNet> (date of access: 09.06.2025).
12. Bangar S. VGG-Net Architecture Explained. *Medium*. URL: <https://medium.com/@siddheshb008/vgg-net-architecture-explained-71179310050f> (date of access: 09.06.2025).
13. He K., Zhang X., Ren S., Sun J. Deep Residual Learning for Image Recognition. *arXiv preprint*. 2015. URL: <https://doi.org/10.48550/arXiv.1512.03385> (date of access: 09.06.2025).
14. Rousseau F., Drumetz L., Fablet R. Residual Networks as Flows of Diffeomorphisms. *Journal of Mathematical Imaging and Vision*. 2019. Vol. 62, no. 3. P. 365–375. URL: <https://doi.org/10.1007/s10851-019-00890-3> (date of access: 09.06.2025).
15. Tan M., Le Q. V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *arXiv preprint*. 2019. URL: <https://arxiv.org/abs/1905.11946> (date of access: 09.06.2025).
16. Efficient and accurate compound scaling for convolutional neural networks / C. Lin et al. *Neural Networks*. 2023. URL: <https://doi.org/10.1016/j.neunet.2023.08.053> (date of access: 09.06.2025).
17. EfficientNet: Improving Accuracy and Efficiency through AutoML and Model Scaling. *Latest News from Google Research Blog - Google Research*. URL: <https://ai.googleblog.com/2019/05/efficientnet-improving-accuracy-and.html> (date of access: 09.06.2025).

18. Classification of Remote Sensing Images Using EfficientNet-B3 CNN Model With Attention / H. Alhichri et al. *IEEE Access*. 2021. Vol. 9. P. 14078–14094. URL: <https://doi.org/10.1109/access.2021.3051085> (date of access: 09.06.2025).
19. Ahmed T., Sabab N. H. N. Classification and Understanding of Cloud Structures via Satellite Images with EfficientUNet. *SN Computer Science*. 2021. Vol. 3, no. 1. URL: <https://doi.org/10.1007/s42979-021-00981-2> (date of access: 09.06.2025).
20. PlantDoc. *PlantDoc*. URL: <https://plantdoc.vercel.app/> (date of access: 09.06.2025).
21. GitHub - anandkm629/plant-doc: PlantDoc is a powerful web-based platform that allows farmers to detect plant diseases by simply uploading an image. *GitHub*. URL: <https://github.com/anandkm629/plant-doc> (date of access: 09.06.2025).
22. PlantDoc Dataset. *Kaggle*. URL: <https://www.kaggle.com/datasets/abdulhasibuddin/plant-doc-dataset> (date of access: 09.06.2025).
23. FieldView™ - Digital Farming's Leading Platform | Crop Science US. *Bayer Crop Science Homepage | Crop Science US*. URL: <https://www.cropscience.bayer.us/tools/fieldview> (date of access: 09.06.2025).
24. Maximize Results with Our Digital Farming Solution | Climate FieldView. *Maximize Results with Our Digital Farming Solution | Climate FieldView*. URL: <https://climate.com/> (date of access: 09.06.2025).
25. GitHub-репозиторій DeepWeeds (ResNet-50 модель для класифікації бур'янів, адаптована під фази культур) URL: <https://github.com/AlexOlsen/DeepWeeds>
26. Papers with Code - DeepWeeds Dataset. *The latest in Machine Learning | Papers With Code*. URL: <https://paperswithcode.com/dataset/deepweeds> (date of access: 09.06.2025).
27. GitHub - rawat999/DeepWeedsX: The DeepWeedsX dataset consists of 17,508 unique 256x256 colour images in 9 classes. These images were collected in situ

from eight rangeland environments across northern Australia. DeepWeedsX is a subset of the DeepWeeds dataset. *GitHub*. URL: <https://github.com/rawat999/DeepWeedsX> (date of access: 09.06.2025).

28. Our products - AgroVision. *AgroVision*. URL: <https://agrovision.com/software/> (date of access: 09.06.2025).

29. Homepage - AgroVision. *AgroVision*. URL: <https://agrovision.com/> (date of access: 09.06.2025).

30. Contributors to Wikimedia projects. Inception (deep learning architecture) - Wikipedia. *Wikipedia, the free encyclopedia*. URL: [https://en.wikipedia.org/wiki/Inception_\(deep_learning_architecture\)](https://en.wikipedia.org/wiki/Inception_(deep_learning_architecture)) (date of access: 09.06.2025).

31. Recognition of sunflower growth period based on deep learning from UAV remote sensing images / Z. Song et al. *Precision Agriculture*. 2023. URL: <https://doi.org/10.1007/s11119-023-09996-6> (date of access: 09.06.2025).

32. Sunpheno: A Deep Neural Network for Phenological Classification of Sunflower Images / S. A. B. Luoni et al. *Plants*. 2024. Vol. 13, no. 14. P. 1998. URL: <https://doi.org/10.3390/plants13141998> (date of access: 09.06.2025)..

33. Karahanli G., Taşkin C. Derin öğrenme yöntemleri kullanılarak ayçiçeği bitkisinin gelişim evrelerinin tespiti. *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*. 2023. URL: <https://doi.org/10.17341/gazimmfd.1200615> (date of access: 09.06.2025).

34. Sunflower-YOLO: Detection of sunflower capitula in UAV remote sensing images / R. Jing et al. *European Journal of Agronomy*. 2024. Vol. 160. P. 127332. URL: <https://doi.org/10.1016/j.eja.2024.127332> (date of access: 09.06.2025).

35. Yildirim E., Colkesen I., Sefercik U. G. Identification of Sunflowers (*Helianthus annuus* L.) from Multi-Temporal UAV Orthomosaics Using Deep Learning Models. *Advanced Engineering Days*. 2024. Vol. 9. Tabriz, Iran. URL: https://www.researchgate.net/publication/382116971_Identification_of_Sunflowers_Helianthus_annuus_L_from_Multi-

Temporal_UAV_Orthomosaics_Using_Deep_Learning_Models (date of access: 09.06.2025).

36. Sentinel-1 time series data for sunflower (*Helianthus annuus*) phenology monitoring / A. Qadir et al. *Remote Sensing of Environment*. 2023. Vol. 295. P. 113689. URL: <https://doi.org/10.1016/j.rse.2023.113689> (date of access: 09.06.2025).

37. Sunflower Growth Stages Classifier. *Kaggle*. URL: <https://www.kaggle.com/datasets/glaykarahanl/sunflower-development-stages/data> (date of access: 09.06.2025).

38. PyTorch documentation – PyTorch 2.7 documentation. *PyTorch*. URL: <https://pytorch.org/docs/stable/index.html> (date of access: 09.06.2025).

39. Welcome to PyTorch Tutorials – PyTorch Tutorials 2.7.0+cu126 documentation. *PyTorch*. URL: <https://pytorch.org/tutorials/> (date of access: 09.06.2025).

40. Learn the Basics – PyTorch Tutorials 2.7.0+cu126 documentation. *PyTorch*. URL: <https://pytorch.org/tutorials/beginner/basics/intro.html> (date of access: 09.06.2025).

41. torchvision – Torchvision 0.22 documentation. *PyTorch*. URL: <https://pytorch.org/vision/stable/index.html> (date of access: 09.06.2025).

42. OpenCV: OpenCV-Python Tutorials. *OpenCV documentation index*. URL: https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html (date of access: 09.06.2025).

43. OpenCV: OpenCV modules. *OpenCV documentation index*. URL: <https://docs.opencv.org/4.x/index.html> (date of access: 09.06.2025).

44. OpenCV Tutorial in Python - GeeksforGeeks. *GeeksforGeeks*. URL: <https://www.geeksforgeeks.org/opencv-python-tutorial> (date of access: 09.06.2025).

45. Official OpenCV Website. *Opencv*. URL: <https://opencv.org/> (date of access: 09.06.2025).

46. Pillow. *Pillow (PIL Fork)*. URL: <https://pillow.readthedocs.io/en/stable/> (date of access: 09.06.2025).
47. Matplotlib documentation – Matplotlib 3.10.3 documentation. *Matplotlib – Visualization with Python*. URL: <https://matplotlib.org/stable/index.html> (date of access: 09.06.2025).
48. GitHub - jacobgil/pytorch-grad-cam: Advanced AI Explainability for computer vision. Support for CNNs, Vision Transformers, Classification, Object detection, Segmentation, Image similarity and more. *GitHub*. URL: <https://github.com/jacobgil/pytorch-grad-cam> (date of access: 09.06.2025).
49. Client Challenge. *PyPI · The Python Package Index*. URL: <https://pypi.org/project/pytorch-gradcam/> (date of access: 09.06.2025).
50. Malaydakh V., Molchanova M., Shevchuk P., Mazurets O. Deep Learning Neural Network Architecture for Determining Sunflower Growth Stage from Visual Data. Modern Scientific Research: Theoretical and Practical Aspects. *Proceedings II International Scientific and Practical Conference*. May 26–28, 2025. Riga, Latvia. Pp. 143–148. URL: https://www.eoss-conf.com/wp-content/uploads/2025/05/Riga_Latvia_26.05.25.pdf (date of access: 09.06.2025).

ДОДАТКИ

Додаток А

Програмні коди

Вихідний код методу визначення стадії розвитку соняшника по фото за допомогою глибоких нейронних мереж, використаний у дослідженні, доступний у репозиторії GitHub: https://github.com/mrapen/Sunflower_classification.git.

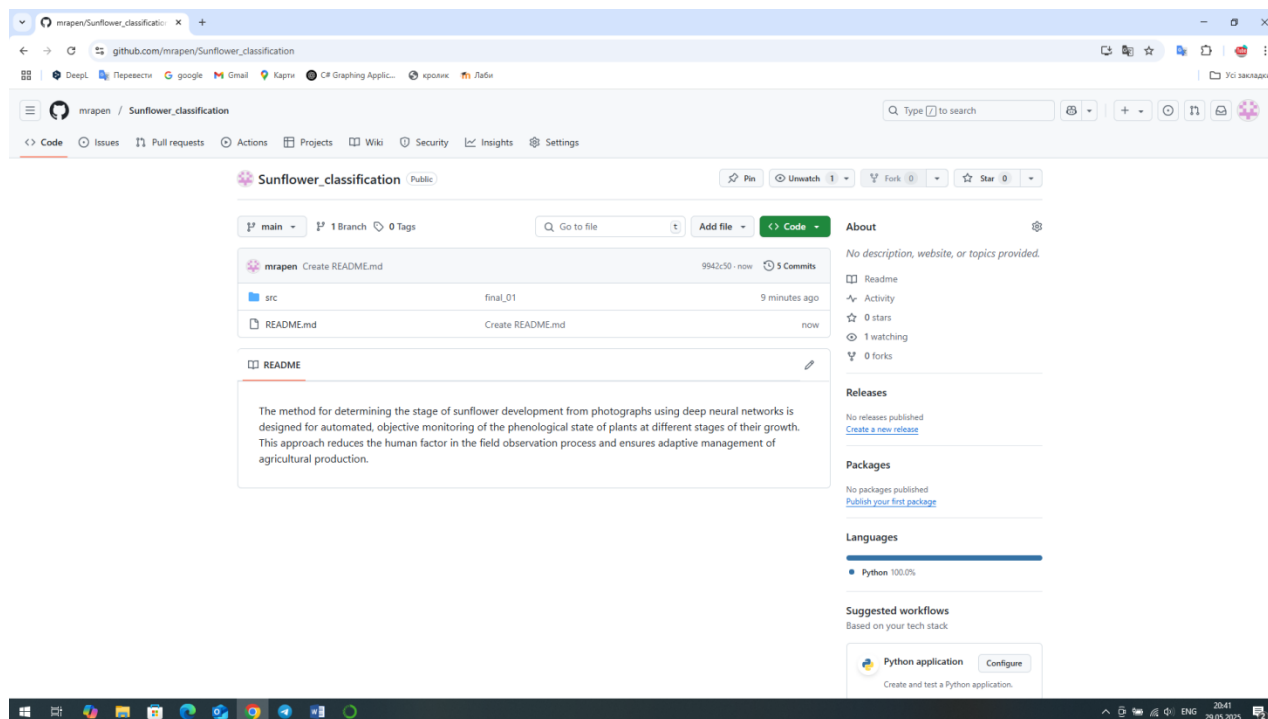


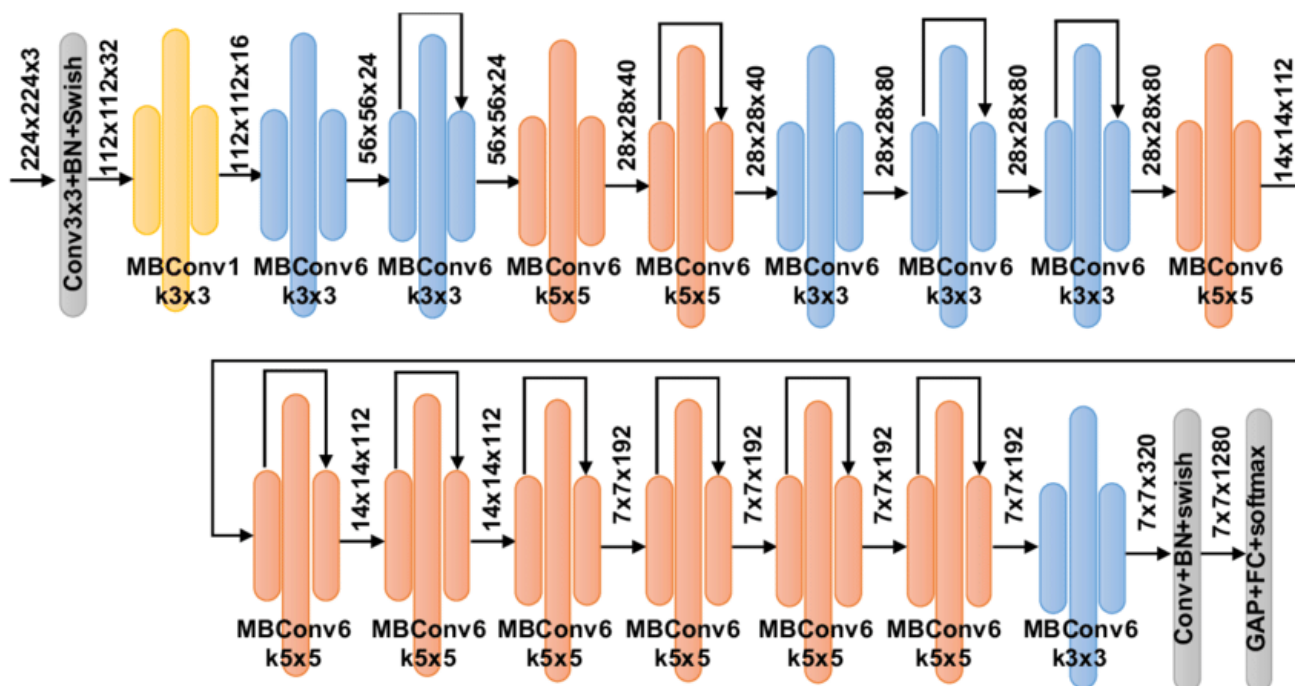
Рисунок А.1 – Головна сторінка репозиторію

Ім'я	Останнє повідомлення коміту	Дата останньої ...
..		
__pycache__	commit_02	last week
аналіз	commit_03	2 minutes ago
дані	commit_03	2 minutes ago
набір даних	перший коміт	last month
висновок	commit_02	last week
збережені_моделі	commit_03	2 minutes ago
навчання	commit_03	2 minutes ago
візуалізація	commit_02	last week
main.py	commit_03	2 minutes ago
ui_main.py	commit_02	last week
unit_test_and_test_cases.py	commit_02	last week

Рисунок А.2 – Структура репозиторію

Додаток Б

Архітектура нейромережі EfficientNet-B0



Додаток В

Презентаційний матеріал

Кваліфікаційна робота бакалавра

Метод визначення стадії розвитку соняшника по фото за допомогою глибоких нейронних мереж

Виконав: студент групи КНс-22-1 Малайдах Вадим

Керівник: асистент каф. КН Марина Молчанова

Актуальність

- Сільське господарство є однією з ключових галузей економіки України, забезпечуючи продовольчу безпеку, експортні надходження та зайнятість населення. Серед основних культур, що вирощуються в Україні, соняшник займає особливе місце. Україна є світовим лідером за обсягами виробництва та експорту соняшникової олії. Соняшник вирощують у більшості регіонів країни, переважно в степових і лісостепових зонах, де кліматичні умови сприятливі для його розвитку. Висока економічна цінність культури обумовлює необхідність ретельного моніторингу її стану на всіх етапах росту для забезпечення стабільної врожайності та якості продукції.

Мета, об'єкт, предмет дослідження

- **Мета роботи** полягає у підвищенні точності визначення стадії розвитку соняшника по фото за допомогою глибоких нейронних мереж.
- **Об'єкт дослідження** – процес визначення стадії розвитку соняшника по фото.
- **Предмет дослідження** – нейромережові методи та технології для визначення стадії розвитку соняшника по фото.

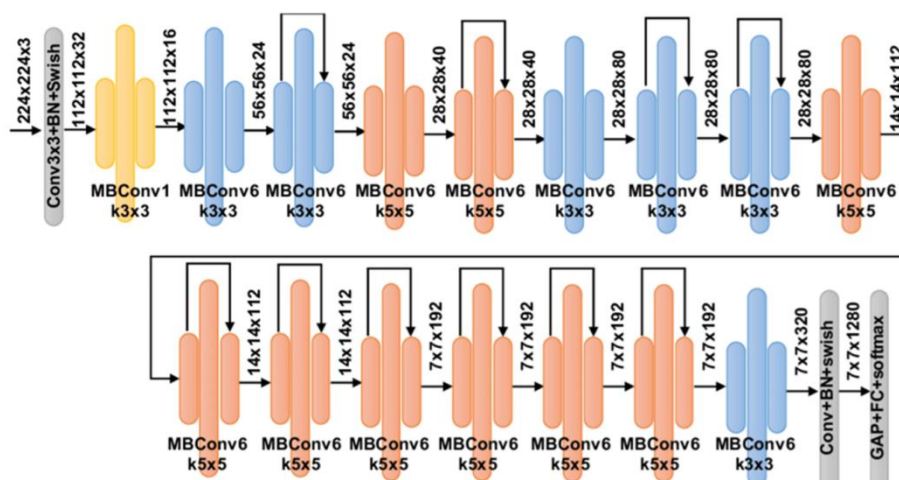
Схема та етапи методу визначення стадії розвитку соняшника по фото за допомогою глибоких нейронних мереж



Архітектура інформаційної системи визначення стадії розвитку соняшника по фото за допомогою глибоких нейронних мереж



Архітектура нейромережі EfficientNet-B0



Дослідження методу визначення стадії розвитку соняшника по фото

```
Running inference with the trained model...
640it [00:39, 16.09it/s]
Accuracy of the model on the test data -> 0.997

===== STARTING MODEL ANALYSIS =====
Analyzing model performance...
Test Accuracy: 0.9969

Classification Report:
              precision    recall  f1-score   support

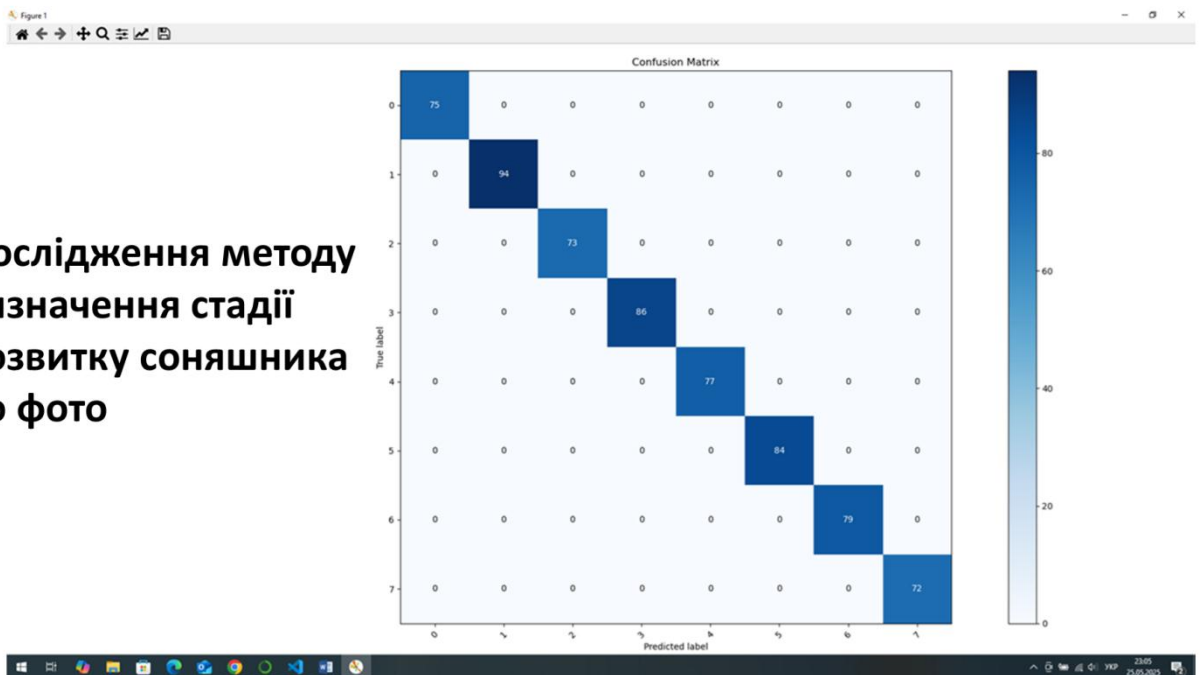
     0             1.00      1.00      1.00         68
     1             0.99      1.00      0.99         88
     2             1.00      0.99      0.99         88
     3             1.00      1.00      1.00         75
     4             0.99      1.00      0.99         91
     5             1.00      1.00      1.00         84
     6             1.00      0.99      0.99         68
     7             1.00      1.00      1.00         78

 accuracy          1.00          1.00          1.00        640
 macro avg         1.00          1.00          1.00        640
 weighted avg      1.00          1.00          1.00        640

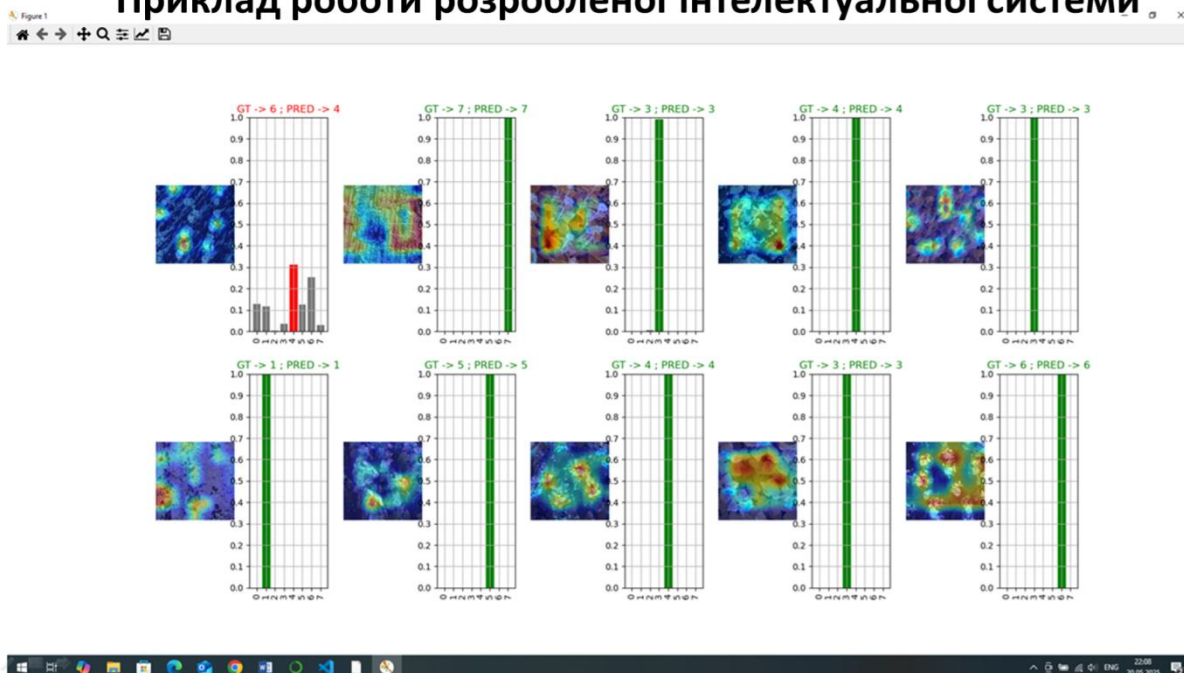
===== MODEL ANALYSIS COMPLETE =====
```



Дослідження методу визначення стадії розвитку соняшника по фото



Приклад роботи розробленої інтелектуальної системи



Висновок

Метою кваліфікаційної роботи бакалавра було підвищення точності визначення стадії розвитку соняшника по фото за допомогою глибоких нейронних мереж.

Для досягнення мети, поставлено та вирішено такі задачі:

- виконано аналіз інформаційних моделей для визначення стадії розвитку соняшника по фото;
- створено метод визначення стадії розвитку соняшника по фото за допомогою глибоких нейронних мереж;
- виконано проектування та реалізацію інтелектуальної системи моніторингу розвитку соняшника з використанням комп'ютерного зору;
- проведено тестування створеного методу визначення стадії розвитку соняшника по фото з використанням розробленої інтелектуальної системи.

Anti-Plagiarism (UA) v-15.281 Educational

The maximum coincidence with one document 3.0%

Dictionary check: en_US, ru_RU, ua_UA. **Errors in the documents: 14%**

ID: 246616 Title: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА на тему Метод визначення стадії розвитку соняшника по фото за допомогою глибоких нейронних мереж Added in a DB: 2025-06-17 Authors: Вадим МАЛАЙДАХ Heads: Марина МОЛЧАНОВА Consultants: Opponents:	Document		Sum coincidence on the DB	
	Symbols	Lexemes	Symbols	Lexemes
	64371	896	3775 (6%)	62 (7%)

Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes

Протокол аналізу звіту подібності науковим керівником

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Вадим МАЛАЙДАХ

Співавтор:

Назва: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА на тему Метод визначення стадії розвитку соняшника по фото за допомогою глибоких нейронних мереж

Науковий керівник: Марина МОЛЧАНОВА, асистент. каф. КН

Підрозділ: Кафедра комп'ютерних наук

Коефіцієнт подібності 1: 5.6%

Коефіцієнт подібності 2: 3.1%

Мікропробіли: 0

Заміна букв: 0

Інтервали: 0

Білі знаки: 93

Дата створення звіту: 2025-06-17 20:26:57.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.

Обґрунтування:

2025-06-17

Дата

експерт

Петровський Р.С.

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ КАФЕДРИ КОМП'ЮТЕРНИХ НАУК

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва кваліфікаційної роботи Метод визначення стадії розвитку соняшника по фото за допомогою глибоких нейронних мереж

Автор студент групи КНс-22-1 Вадим Малайда

Освітня програма Комп'ютерні науки

Рівень вищої освіти перший (бакалаврський)

Спеціальність 122 – Комп'ютерні науки

Науковий керівник: асистент кафедри комп'ютерних наук Марина Молчанова

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованими програмними засобами комісія зробила такий висновок:

№	Висновок	Позначка про відповідність
1	Ознаки академічного плагіату	
1.1	Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту.	<i>відповідає</i>
1.2	Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована.	
1.3	Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат.	
1.4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
2	Інші види порушень академічної доброчесності	<i>відсутні</i>

Підтвердження:

Запозичення, виявлені в роботі Вадима Малайда, не є плагіатом, оскільки: запозичення розміщені в розділі огляду існуючих підходів, не описують безпосередньо авторську роботу і не стосуються її результатів; усі запозичення фрагментарні; до запозичень входять фрагменти, які не мають авторства і містять поширені конструкції та загальновідомі терміни, скорочення. Рівень подібності не перевищує допустимої межі. Таким чином, робота є законною та приймається до захисту.

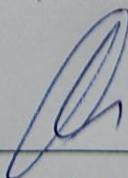
Обсяг запозичень, визначений системами виявлення збігів/ідентичності/схожості:

- за системою Anti-Plagiarism: 3%;

- за системою StrikePlagiarism КП1: 5,6%, КП2: 3,1%.

18.06.2025

Завідувач кафедри



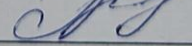
Олександр БАРМАК

Гарант освітньої програми



Олександр МАЗУРЕЦЬ

Керівник кваліфікаційної роботи



Марина МОЛЧАНОВА



**ВІДГУК НАУКОВОГО КЕРІВНИКА
на кваліфікаційну роботу бакалавра**

студента гр. КНс-22-1 Малайдаха Вадима Михайловича

за темою Метод визначення стадії розвитку соняшника по фото за допомогою глибоких нейронних мереж

1. Актуальність теми

Своєчасне розпізнавання фенологічних стадій соняшника має вирішальне значення для ефективного управління агротехнічними процесами. Традиційні методи спостереження є обмеженими через суб'єктивність оцінки, що впливає на врожайність і екологічну стабільність. Застосування глибокого навчання дозволяє автоматизувати цей процес, підвищуючи точність і оперативність аналізу, що сприяє раціональному використанню ресурсів і підтримці продовольчої безпеки.

2. Відповідність роботи предметній області Стандарту спеціальності 122 Комп'ютерні науки

Відповідно до стандарту, а саме до опису предметної області, об'єктом роботи є процес визначення стадії розвитку соняшника по фото. Метою роботи є підвищення точності визначення стадії розвитку соняшника по фото за допомогою глибоких нейронних мереж. При вирішенні поставленої задачі використано нейромережеві методи та технології для визначення стадії розвитку соняшника по фото. Тому результати виконання кваліфікаційної роботи бакалавра відповідають стандарту бакалавра спеціальності 122 – Комп'ютерні науки.

3. Професійні та особистісні якості бакалавра

Студент у процесі роботи над кваліфікаційною роботою бакалавра виявив себе як відповідальний і наполегливий здобувач вищої освіти, який чітко дотримувався етапів дослідницької роботи та виконував завдання у визначені терміни. Його діяльність характеризувалася самостійністю, ретельністю, уважністю до деталей, а також вмінням застосовувати набуті знання в контексті практичної реалізації поставленої проблеми. Результати кваліфікаційної роботи свідчать про сформованість професійних компетентностей та здатність до подальшої професійної діяльності в галузі комп'ютерних наук.

4. Ступінь самостійності під час виконання кваліфікаційної роботи

Усі наукові положення та результати, що наведені у роботі, були отримані студентом самостійно, з дотриманням методологічних принципів дослідницької роботи.

5. Ступінь оволодіння методами дослідження

У ході виконання дослідження здобувач засвідчив належну методологічну підготовку, успішно використовуючи профільні методики, алгоритми, засоби програмування та дослідницькі підходи, характерні для обраної галузі.

6. Повнота та якість розкриття теми роботи

У роботі забезпечено повну відповідність між заявленою темою, поставленими завданнями та отриманими результатами. Здійснено актуальний огляд джерел, сформульовано дослідницькі задачі, реалізовано прикладне програмне забезпечення для перевірки й оцінювання ефективності наведеного методу.

7. Логічність, послідовність, аргументованість, літературна грамотність викладення матеріалу

Матеріал викладено послідовно й доступно, з належною аргументацією основних положень. Робота написана грамотно, з дотриманням норм наукового стилю, без стилістичних порушень.

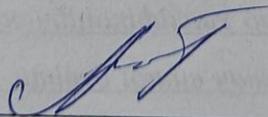
8. Можливість практичного застосування кваліфікаційної роботи бакалавра, окремих її частин

Розроблений у роботі метод та його програмна реалізація може знайти практичне використання агрономами для визначення стадії розвитку соняшника по фото.

9. Висновок про можливість допуску кваліфікаційної роботи бакалавра до захисту, на яку оцінку заслуговує робота

Враховуючи високий рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка «відмінно».

Керівник _____



асистент каф. КН Марина МОЛЧАНОВА



РЕЦЕНЗІЯ

на кваліфікаційну роботу бакалавра

студента гр. КНс-22-1 Малайдаха Вадима Михайловича

за темою: Метод визначення стадії розвитку соняшника по фото за допомогою глибоких нейронних мереж

1. Актуальність обраної теми

Удосконалення автоматизованого моніторингу розвитку соняшника за допомогою нейромереж відповідає сучасним викликам сільського господарства — від зміни клімату до необхідності підвищення ефективності виробництва. Такий підхід сприяє досягненню цілей сталого розвитку, зокрема зменшенню агрохімічного навантаження та втрат урожаю, забезпечуючи водночас точну та незалежну оцінку стану рослин у польових умовах, що визначає його актуальність і практичну цінність.

2. Повнота розкриття мети та завдань роботи

У своїй кваліфікаційній роботі студент продемонстрував вміння виокремлювати актуальні наукові завдання, належно їх формулювати й реалізовувати, що дозволило досягти поставленої мети та свідчить про сформовану дослідницьку компетентність.

3. Зміст кожного розділу роботи

Кожен розділ кваліфікаційної роботи містить описи виконаних завдань, реалізація яких сприяла досягненню загальної мети дослідження. Перший розділ містить результати проведеної характеристики предметної області. У другому розділі описано етапи методу визначення стадії розвитку соняшника по фото за допомогою глибоких нейронних мереж. У третьому розділі висвітлено результати дослідження методу визначення стадії розвитку соняшника по фото.

4. Оцінка розробленої інформаційної системи, її практична цінність

Система дозволяє аграріям і науковцям автоматизовано визначати стадії розвитку соняшника на основі зображень, що підвищує ефективність агротехнологічного моніторингу.

5. Якість оформлення кваліфікаційної роботи бакалавра

Робота якісно оформлена, відповідає структурним і стилістичним вимогам до бакалаврських досліджень. Це свідчить про володіння студентом основами академічного письма та вміння презентувати результати у відповідній формі. Практична цінність полягає в можливості оперативної оцінки вегетаційного стану культур без необхідності експертного втручання.

6. Недоліки кваліфікаційної роботи бакалавра

В роботі відсутній аналіз похибок або обмежень застосованого методу. Окремі джерела в бібліографічному списку не відповідають затвердженим вимогам до оформлення за ДСТУ. Результати експерименту подано без достатнього обґрунтування вибору метрик оцінювання. При цьому, виявлені недоліки не мають принципового впливу на змістове наповнення роботи й результати дослідження. Для подальших досліджень, рекомендую забезпечити визначення стадії розвитку соняшника не тільки по фото, а й по відео.

7. Загальний висновок (допускається чи не допускається до захисту), та оцінка на яку заслуговує кваліфікаційна робота.

Враховуючи рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка «відмінно».

Рецензент

доцент кафедри №3 Яценко А.М.