

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
АГРОБІОТЕХНОЛОГІЧНИЙ ФАКУЛЬТЕТ
КАФЕДРА САДІВНИЦТВА, ВИНОГРАДАРСТВА, БІОЛОГІЇ ТА ХІМІЇ**

«До захисту допущено»

Завідуючий кафедри садівництва,
виноградарства, біології та хімії,
канд. с./г. наук, доцент

_____Юрій САВЧУК

«___» _____ 2024

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття другого (магістерського) ступеня вищої освіти
Освітньої програми «Садівництво, плодоовочівництво та виноградарство»
за спеціальністю: 203 «Садівництво, плодоовочівництво та виноградарство»

**ПРАКТИКА ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ У СУЧАСНОМУ
САДІВНИЦТВІ**

Науковий керівник: к. с.-г. н.

_____Ірина ІЩЕНКО

Рецензент: _____

Виконав здобувач другого (магістерського)
ступеня вищої освіти заочної форми навчання
освітньо-професійна програма

«Садівництво та виноградарство»

Спеціальність 203

_____Дмитро ЦАРАН

Засвідчую, що кваліфікаційна робота містить
результати власних досліджень. Використання ідей і
тексту інших авторів має посилання на відповідне
джерело.

_____Дмитро ЦАРАН

Зміст

	стор.
Вступ	3
1. Теоретична частина	8
1.1. Аспекти використання штучного інтелекту в садівництві	8
1.1.1. Огляд основних понять штучного інтелекту.	8
1.1.2. Аналіз потенціалу та можливостей застосування штучного інтелекту у садівництві.	11
1.1.3. Приклади успішного використання штучного інтелекту в галузі садівництва.	14
1.2. Методи та технології використання штучного інтелекту у садівництві	17
1.2.1. Передові методи аналізу даних та прогнозування врожаю.	17
1.2.2. Розробка систем моніторингу та управління сільськогосподарськими ресурсами на основі штучного інтелекту	18
2. Мета, завдання, об'єкти, методи, місце й умови досліджень	20
2.1. Мета і задачі досліджень	20
2.2. Об'єкти досліджень	23
2.3. Методика досліджень	24
2.4. Місце й умови проведення досліджень	25
3. Експериментальне дослідження	27
3.1. Постановка експерименту.	27
3.2. Збір та аналіз даних на основі фенологічні спостережень.	27

3.3. Розробка програмного забезпечення для отримання математичної моделі	36
3.4. Оцінка результатів використання штучного інтелекту в садівництві.	57
4. Економічна ефективність вирощування винограду	
5. Охорона навколишнього середовища	
Висновки	
Список використаних джерел	
Додатки	

Вступ

Сучасне садівництво стоїть на порозі нової ери, завдяки впровадженню передових технологій штучного інтелекту (ШІ). Використання ШІ у садівництві відкриває безліч можливостей для підвищення ефективності, продуктивності та стійкості виробництва. З огляду на зростання населення та збільшення попиту на високоякісну продукцію, актуальність цієї теми є надзвичайно високою.

Актуальність теми дослідження визначається глобальними тенденціями у розвитку аграрного сектору та необхідністю адаптації до змін клімату, дефіциту водних ресурсів та підвищення ефективності сільськогосподарського виробництва. Традиційні методи управління аграрними процесами стають менш ефективними, тому інтеграція ШІ у садівництво може стати вирішальним фактором у забезпеченні стабільного та продуктивного розвитку цієї галузі.

Новизна теми полягає в комплексному підході до застосування штучного інтелекту для вирішення специфічних завдань садівництва. Досягнення у галузі машинного навчання, обробки великих даних та робототехніки дозволяють створювати інноваційні системи управління, які автоматизують процеси моніторингу, діагностики захворювань, оптимізації поливу та догляду за рослинами. Ці технології ще не отримали широкого розповсюдження в аграрному секторі України, що підкреслює новизну та перспективність даного дослідження.

Практична користь від реалізації систем штучного інтелекту у садівництві є багатогранною. Впровадження таких систем дозволяє значно знизити витрати на ресурси, зокрема воду та добрива, завдяки оптимізованому управлінню. Крім того, автоматизація процесів моніторингу та догляду за

рослинами сприяє своєчасному виявленню та лікуванню хвороб, що мінімізує втрати врожаю та підвищує якість продукції. Застосування ШІ також сприяє підвищенню безпеки праці та зменшенню впливу людського фактора на виробничі процеси.

Економічна користь від використання штучного інтелекту у садівництві полягає у зниженні виробничих витрат та підвищенні врожайності. Оптимізація використання ресурсів дозволяє зменшити витрати на виробництво, що безпосередньо впливає на собівартість продукції. Підвищення ефективності виробництва сприяє збільшенню прибутковості господарств та покращенню економічної стабільності аграрного сектору в цілому.

Таким чином, дослідження практики використання штучного інтелекту у сучасному садівництві є не лише актуальним, але й необхідним кроком для розвитку аграрного сектору. Реалізація таких технологій може забезпечити значні економічні переваги, підвищити якість та ефективність виробництва, що є важливими аспектами у контексті глобальних викликів та перспектив сталого розвитку.

Актуальність роботи: обумовлена необхідністю впровадження інноваційних рішень для підвищення ефективності аграрного сектору. У контексті зростаючого попиту на екологічно чисту продукцію та раціональне використання природних ресурсів, штучний інтелект відкриває нові можливості для оптимізації догляду за насадженнями, зниження витрат і мінімізації впливу на довкілля. Використання технологій II дозволяє автоматизувати ідентифікацію дерев, прогнозувати врожайність і своєчасно виявляти хвороби, що є ключовими завданнями для сучасного сталого сільського господарства.

Мета роботи: Метою роботи є розробка та впровадження моделі штучного інтелекту для автоматизації ідентифікації плодкових дерев у

сучасному садівництві. Це забезпечить ефективне управління насадженнями, оптимізацію догляду за деревами, зменшення витрат на ресурси та мінімізацію негативного впливу на навколишнє середовище. Використання цієї моделі дозволить аграріям покращити якість та обсяг продукції, підвищити ефективність роботи, своєчасно виявляти загрози для дерев і забезпечити сталий розвиток садівництва.

Наукова новизна: полягає у розробці та практичному застосуванні моделі штучного інтелекту для класифікації плодових дерев (абрикос, яблуня, хурма, мигдаль) на основі зображень та відеоматеріалів, що охоплюють всі етапи вегетації. Вперше запропоновано комплексний підхід, який включає:

Довготривалий збір даних: Унікальність полягає у використанні даних, зібраних протягом усього вегетаційного періоду, що дозволяє враховувати сезонні зміни дерев.

Інтеграція інноваційних технологій обробки відео: Розроблена методика видобування кадрів із відео та формування структурованого датасету, який забезпечує високу якість вхідних даних.

Оптимізація навчання моделі: Вибір архітектури нейронної мережі з акцентом на ефективну класифікацію чотирьох видів дерев із врахуванням мінливості їх зовнішніх ознак.

Екологічний підхід: Використання моделі як інструменту для зменшення надмірного застосування хімічних засобів, води та енергії, що сприяє збереженню природного середовища.

Результати роботи сприяють розширенню застосування штучного інтелекту у садівництві та створюють основу для подальшого розвитку автоматизованих систем управління агропромисловими комплексами.

Структура роботи включає: вступ, п'ять розділів, висновки, список використаних джерел та додатки.

У вступі обґрунтовано актуальність теми, сформульовано мету, завдання та наукову новизну роботи.

Перший розділ присвячений аналізу аспектів використання штучного інтелекту у садівництві, включаючи огляд основних понять, аналіз можливостей і потенціалу штучного інтелекту в галузі, а також приклади його успішного застосування.

У другому розділі викладено мету, завдання, об'єкти, методи, місце та умови проведення досліджень.

Третій розділ описує експериментальне дослідження, включаючи постановку експерименту, збір та аналіз фенологічних даних, розробку програмного забезпечення для отримання математичної моделі та оцінку результатів використання штучного інтелекту.

Четвертий розділ присвячено аналізу економічної ефективності впровадження результатів роботи.

П'ятий розділ висвітлює вплив результатів роботи на охорону навколишнього середовища.

У висновках підведено підсумки роботи, сформульовано практичні рекомендації та вказано напрямки подальших досліджень.

Загальний обсяг роботи становить 75 сторінки друкованого тексту без додатків.

1 ТЕОРЕТИЧНА ЧАСТИНА

1.1 Аспекти використання штучного інтелекту в садівництві

1.1.1 Огляд основних понять штучного інтелекту

Штучний інтелект (ШІ) — це галузь комп'ютерних наук, яка займається розробкою інтелектуальних систем, здатних виконувати завдання, що зазвичай потребують людського інтелекту. Серед основних можливостей ШІ виділяють розпізнавання образів, аналіз даних, прогнозування, обробку природної мови та ухвалення рішень. Його головною метою є імітація людського мислення та автоматизація складних процесів.

Основні терміни та поняття ШІ

Машинне навчання (ML)

Це підхід, який дозволяє алгоритмам навчатися на основі даних без прямого програмування. Алгоритми ML застосовуються для аналізу кліматичних умов, розпізнавання стану рослин, прогнозування врожайності.

Глибоке навчання (DL)

Метод машинного навчання, який базується на штучних нейронних мережах. Глибоке навчання дозволяє працювати з великими обсягами даних, таких як зображення чи відео, що є важливим для автоматизованого розпізнавання видів дерев або хвороб.

Комп'ютерний зір (Computer Vision)

Галузь ШІ, яка фокусується на обробці та аналізі зображень і відеоматеріалів. У садівництві комп'ютерний зір застосовується для моніторингу стану рослин, виявлення дефіциту поживних речовин, хвороб або шкідників.

Dataset (набір даних)

Це структурована колекція даних, яка використовується для навчання, тестування або оцінювання моделей машинного навчання. У контексті садівництва dataset може містити:

Зображення листя, квітів і плодів дерев у різні фенологічні фази.

Дані про метеорологічні умови (температура, вологість, опади).

Відеозаписи циклу вегетації різних видів дерев.

Створення якісного dataset є критично важливим для успішного навчання моделей ШІ.

Тренування (Training)

Це процес, під час якого модель ШІ навчається розпізнавати закономірності в даних із dataset. Наприклад, у садівництві модель може навчатися визначати вид дерева або фазу його вегетації за зображеннями.

Тестування (Testing)

Етап перевірки, на якому модель перевіряється на нових даних для оцінки її точності та здатності до узагальнення. Це важливо для забезпечення реальної ефективності моделей у практичних умовах.

Інтернет речей (IoT)

Мережа пристроїв і сенсорів, які взаємодіють між собою та передають дані в реальному часі. У садівництві IoT використовується для збору даних про вологість ґрунту, температуру, рівень освітлення, які далі обробляються системами ШІ.

Алгоритми прогнозування (Prediction Models)

Це алгоритми, які аналізують історичні дані та роблять прогнози. Наприклад,

прогнозування врожайності на основі кліматичних умов і фенологічних даних дерев.

Аугментація даних (Data Augmentation)

Метод розширення dataset шляхом створення додаткових варіацій даних (змінення яскравості зображень, їх обертання, масштабування тощо). У садівництві це дозволяє створювати більш універсальні моделі ШІ для розпізнавання стану рослин у різних умовах.

Аналітика великих даних (Big Data Analytics)

Це підхід до аналізу великих обсягів структурованих і неструктурованих даних. У садівництві використовується для поєднання фенологічних даних із метеорологічними, що дозволяє ухвалювати обґрунтовані рішення щодо управління садами.

Автоматизація процесів:

Завдяки використанню комп'ютерного зору, машинного навчання та IoT, автоматизуються такі процеси, як полив, внесення добрив і моніторинг стану насаджень.

Оптимізація управління:

ШІ дозволяє створювати моделі, що аналізують вплив кліматичних умов на розвиток дерев, забезпечуючи точний прогноз врожайності та потреби у ресурсах.

Покращення точності моніторингу:

Використання якісних dataset і алгоритмів розпізнавання підвищує ефективність моніторингу стану садів, знижуючи втрати через шкідників чи хвороби.

Огляд основних понять ШІ забезпечує основу для розуміння, як саме ці технології можуть бути інтегровані в практики садівництва. Створення

dataset, навчання моделей ШІ та їхнє тестування є ключовими етапами для впровадження цих технологій у реальні умови. Це дозволяє забезпечити автоматизацію догляду за садами, підвищити продуктивність і зменшити екологічне навантаження.

1.1.2 Аналіз потенціалу та можливостей застосування штучного інтелекту у садівництві

Штучний інтелект (ШІ) відкриває нові можливості для оптимізації процесів у садівництві, підвищуючи ефективність виробництва, якість продукції та забезпечуючи сталий розвиток галузі. Технології ШІ вже продемонстрували свою дієвість у численних аспектах садівництва, таких як моніторинг стану рослин, управління ресурсами, боротьба зі шкідниками та прогнозування врожайності.

Потенціал застосування ШІ у садівництві

Оптимізація управління ресурсами

ШІ забезпечує точне управління поливом, підживленням та використанням засобів захисту рослин. Системи на основі алгоритмів машинного навчання аналізують дані з сенсорів (температура, вологість ґрунту, освітлення) та автоматично регулюють постачання ресурсів. Це дозволяє знижувати витрати та мінімізувати екологічний вплив.

Моніторинг стану насаджень

Використання комп'ютерного зору дає змогу виявляти захворювання та дефіцит поживних речовин на ранніх стадіях. Застосування інфрачервоних камер і сенсорів дозволяє ідентифікувати навіть приховані проблеми, зокрема ураження кореневої системи чи недостатнє зволоження ґрунту.

Прогнозування врожайності

Алгоритми прогнозування на основі великих даних аналізують історичні дані

про кліматичні умови, стан ґрунту, агротехнічні заходи та фенологічні спостереження. Це дає змогу передбачити кількість і якість врожаю, допомагаючи фермерам планувати ресурси та збут продукції.

Боротьба зі шкідниками та хворобами

ШІ аналізує дані про поширення шкідників і хвороб, ідентифікуючи найефективніші засоби боротьби. Завдяки точковому обприскуванню можна зменшити використання хімічних препаратів, захищаючи екосистему саду.

Адаптація до змін клімату

Інтелектуальні системи дозволяють швидко реагувати на змінні кліматичні умови. Наприклад, ШІ може попереджати про ризик заморозків і рекомендувати заходи для захисту врожаю.

Автоматизація рутинних завдань

Роботизовані платформи, оснащені ШІ, виконують завдання з обрізання дерев, збору врожаю, поливу та обприскування. Це знижує потребу в ручній праці та підвищує точність виконання завдань.

Інтеграція IoT та ШІ

Завдяки взаємодії сенсорів Інтернету речей (IoT) із алгоритмами ШІ фермери отримують точні дані в реальному часі, що дозволяє оперативно реагувати на зміни у стані насаджень або погодних умовах.

Навчання моделей на локальних даних

Використання локальних даних, зібраних у конкретному регіоні, дозволяє підвищити точність прогнозів і рекомендацій для садівників. Це особливо важливо для змішаних садів, де кожен вид рослин має специфічні потреби.

Розробка індивідуальних рішень

Системи ШІ можуть бути адаптовані до конкретних видів садівництва, таких як промислові сади, органічне садівництво або невеликі фермерські

господарства. Наприклад, для органічного виробництва можуть бути розроблені алгоритми, які фокусуються на екологічних методах боротьби зі шкідниками.

Підтримка сталого розвитку

Завдяки оптимізації використання ресурсів і зменшенню хімічного навантаження, ШІ сприяє екологічній стійкості аграрного сектору, допомагаючи зберігати природні екосистеми та біорізноманіття.

Необхідність високоякісних даних

Навчання моделей ШІ вимагає великих і якісних dataset, які важко зібрати у локальних умовах.

Фінансові витрати

Впровадження технологій ШІ вимагає значних інвестицій, що може бути проблематичним для малих господарств.

Навчання фермерів

Використання інтелектуальних систем вимагає технічної підготовки, що може бути перешкодою для деяких користувачів.

Екологічні ризики

Надмірна автоматизація може спричинити негативний вплив на екосистеми, наприклад, скорочення популяцій корисних комах через надмірну стандартизацію процесів.

1.1.1 Приклади успішного використання штучного інтелекту в галузі садівництва.

Штучний інтелект (ШІ) активно впроваджується в садівництві, підвищуючи ефективність та точність агротехнічних процесів. Нижче наведено приклади успішного застосування ШІ в цій галузі.

Моніторинг стану рослин за допомогою комп'ютерного зору

Системи комп'ютерного зору, оснащені камерами високої роздільної здатності та алгоритмами ШІ, дозволяють виявляти дефіцит поживних речовин, шкідників і захворювання на ранніх стадіях. Аналіз кольору та текстури листя і плодів допомагає ідентифікувати навіть мінімальні відхилення, що знижує ризик поширення хвороб.

AgroSmart Vision: Система, що аналізує стан рослин за допомогою високоточних камер і алгоритмів ШІ, дозволяючи ідентифікувати хвороби, дефіцит поживних речовин та шкідників.

Blue River Technology (See & Spray): Використовує камери та алгоритми глибокого навчання для точкового обприскування бур'янів, зменшуючи використання гербіцидів.

Аналіз ґрунту та кліматичних умов за допомогою машинного навчання

Сенсори, що реєструють рівень вологості, кислотність ґрунту та інші показники, у поєднанні з алгоритмами машинного навчання дозволяють створювати точні прогнози врожайності. Наприклад, система Taranis використовує супутникові знімки для прогнозування врожаю та своєчасної обробки насаджень.

Taranis: Платформа для аналізу супутникових даних і фотографій з дронів, яка прогнозує врожайність та виявляє ризики поширення шкідників.

ClimateAI: Використовує алгоритми машинного навчання для прогнозування кліматичних ризиків і їх впливу на врожайність.

Автоматизовані системи поливу та підживлення

Компанія Blue River Technology розробила систему "See & Spray", яка використовує комп'ютерний зір та ШІ для точкового обприскування бур'янів, зменшуючи обсяги хімічних засобів. Це дозволяє уникнути надмірного використання гербіцидів і знижує витрати.

CropX: Платформа, яка інтегрує дані з сенсорів ґрунту, кліматичних датчиків та супутникових знімків для оптимізації поливу та підживлення.

Semios: Система для моніторингу шкідників, вологості ґрунту та стану рослин у режимі реального часу, що допомагає приймати обґрунтовані рішення про догляд за насадженнями.

Роботизовані платформи для догляду за рослинами

Обладнані сенсорами та камерами, сучасні роботи здатні проводити обрізання, полив, обприскування і збір врожаю з високою точністю. Завдяки цьому можна зменшити використання хімікатів, зберігаючи екологічну чистоту саду та підвищуючи якість продукції.

Фермери у Каліфорнії, які використовують платформу Semios, можуть відстежувати рівень вологості та зараження шкідниками в реальному часі, що дозволяє їм приймати своєчасні рішення щодо обробки насаджень.

Naio Technologies: Роботи, оснащені камерами та ШІ, виконують завдання з прополювання, збирання плодів і догляду за рослинами.

FarmBot: Високоточна система для автоматизації посіву, поливу та моніторингу насаджень у невеликих господарствах.

SmartFarm: Інтегрує сенсори IoT та аналітичні платформи ШІ для моніторингу стану насаджень і управління ресурсами.

Ці приклади демонструють, як ШІ може підвищити ефективність, зменшити витрати та покращити екологічну стійкість у садівництві.

Інтеграція ШІ в управління змішаними садами

У змішаних садах з різними видами плодових дерев ШІ аналізує фенологічні дані та метеорологічні умови, оптимізуючи догляд за кожним видом рослин. Це включає індивідуальні графіки поливу та підживлення, точкову обробку та профілактику захворювань, що підвищує врожайність та якість продукції.

Штучний інтелект (ШІ) набуває все більшого поширення у сучасному садівництві, пропонуючи інноваційні рішення для моніторингу, управління та оптимізації аграрних процесів. Сучасні системи ШІ дозволяють автоматизувати рутинні завдання, підвищувати врожайність та зменшувати вплив на довкілля. Точковий полив та підживлення за допомогою ШІ дозволяють скоротити споживання води на 20-30% і зменшити використання добрив на 15-20%. Використання комп'ютерного зору та аналітики даних допомагає вчасно реагувати на ризики захворювань і шкідників, що дозволяє збільшити врожайність на 10-25%. Точкове обприскування знижує кількість використаних хімікатів до 90%, захищаючи біорізноманіття та якість ґрунтів.

Сучасні технології ШІ вже демонструють значний потенціал у підвищенні ефективності та екологічності садівництва. Проте успішне впровадження

залежить від доступності інвестицій, якості даних і навчання персоналу. Подальший розвиток технологій, таких як інтеграція IoT, роботизація та локалізовані моделі, дозволить зробити ці рішення доступнішими для всіх фермерів.

1.2 Методи та технології використання штучного інтелекту у садівництві

1.2.1 Передові методи аналізу даних та прогнозування врожаю

Штучний інтелект (ШІ) забезпечує нові можливості для аналізу великих обсягів даних і створення точних моделей прогнозування, що дозволяють підвищити ефективність садівництва. Передові методи аналізу та прогнозування допомагають фермерам ухвалювати обґрунтовані рішення про посіви, полив, обробку та збирання врожаю.

Основні методи аналізу даних:

Машинне навчання (ML)

Алгоритми машинного навчання використовуються для обробки кліматичних даних, стану ґрунту, фенологічних спостережень і історичних даних врожайності. Наприклад:

Регресійні моделі допомагають передбачити обсяг врожаю, враховуючи фактори, такі як рівень вологи та температура.

Кластеризація використовується для групування зон саду за схожими характеристиками, що допомагає планувати заходи точкового впливу.

Глибоке навчання (DL)

Використання нейронних мереж дозволяє аналізувати складні дані, такі як зображення та відео насаджень.

1.2.2 Розробка систем моніторингу та управління сільськогосподарськими ресурсами на основі штучного інтелекту.

Сучасний підхід до управління сільськогосподарськими ресурсами

Розробка систем моніторингу та управління сільськогосподарськими ресурсами є ключовим елементом у цифровізації сільського господарства. Штучний інтелект (ШІ) надає можливість значно підвищити ефективність використання природних ресурсів, оптимізувати процеси догляду за культурами та зменшити втрати через своєчасне виявлення проблем.

Основні компоненти системи моніторингу та управління

Збір даних:

Системи на основі ШІ використовують дані, зібрані з різних джерел:

Дрони: Для зйомки сільськогосподарських угідь з повітря.

Датчики: Інтегровані у ґрунт для моніторингу вологості, температури та рівня поживних речовин.

Супутникові знімки: Для оцінки стану посівів на великих територіях.

Метеорологічні дані: Для прогнозування погодних умов.

Аналіз даних:

Використання машинного навчання дозволяє:

Виявляти проблемні ділянки (посуху, надмірний полив, шкідників).

Прогнозувати урожайність на основі історичних та поточних даних.

Оптимізувати використання ресурсів (води, добрив, хімічних засобів захисту).

Управління ресурсами:

Системи ШІ забезпечують точкове внесення добрив, води або засобів захисту рослин на основі аналізу конкретних ділянок:

Автоматизоване зрошення: Визначення потреби у воді для кожного виду культури та локальне поливання.

Розумне внесення добрив: Розрахунок оптимальної кількості поживних речовин для певної ділянки чи культури.

Інтегрований захист рослин: Локалізація зон ураження хворобами чи шкідниками для своєчасної обробки.

Прогнозування:

Завдяки алгоритмам ШІ аграрії можуть прогнозувати:

Урожайність культур за сезонами.

Можливі ризики (посухи, заморозки, ураження шкідниками).

Економічну ефективність використання ресурсів.

Переваги використання систем на основі ШІ

Ефективне використання ресурсів:

До 40% скорочення витрат води завдяки автоматизованому зрошенню.

Зменшення обсягів добрив і хімічних засобів до 30% через точкове внесення.

Зниження екологічного впливу:

Мінімізація забруднення ґрунту та водойм завдяки локалізації обробок.

Збереження популяцій корисних комах через зменшення використання пестицидів.

Своєчасне виявлення проблем:

Раннє виявлення хвороб, посухи чи інших загроз забезпечує зниження втрат урожаю.

Економічна вигода:

Підвищення врожайності та зниження витрат сприяють зростанню рентабельності сільськогосподарських підприємств.

Приклади застосування

Системи точного землеробства: Наприклад, системи автоматизованого поливу, що інтегруються із супутниковими даними та прогнозами ШІ.

Моніторинг стану дерев: У садівництві моделі ШІ можуть автоматично класифікувати дерева, оцінювати їхній стан та прогнозувати урожайність.

Висновок

Розробка систем моніторингу та управління сільськогосподарськими ресурсами на основі штучного інтелекту дозволяє забезпечити більш ефективне та екологічно відповідальне ведення сільського господарства.

Інтеграція таких систем у садівництво сприяє сталому розвитку аграрної галузі та мінімізує негативний вплив на навколишнє середовище.

2 МЕТА, ЗАВДАННЯ, ОБ'ЄКТИ, МЕТОДИ, МІСЦЕ Й УМОВИ ДОСЛІДЖЕНЬ

2.1 Мета і задачі досліджень

Метою цієї роботи є комплексне вивчення можливостей впровадження штучного інтелекту (ШІ) для управління садівництвом на основі фенологічних спостережень. Це включає вивчення циклів вегетації таких плодових дерев, як яблуня, абрикос, мигдаль і хурма, із подальшою розробкою моделей навчання ШІ для автоматизованого розпізнавання видів дерев та їхнє фенологічних фаз.

Основною метою роботи є інтеграція ШІ у сучасні практики садівництва, що дозволить підвищити ефективність управління садами, оптимізувати використання ресурсів і забезпечити екологічну стійкість аграрного виробництва.

Ця робота має на меті створення науково-обґрунтованих основ для використання штучного інтелекту у садівництві, що дозволить підвищити ефективність вирощування плодових дерев, оптимізувати процеси догляду за садами та покращити якість сільськогосподарської продукції.

Метою даної дипломної роботи є проведення фенологічних досліджень усього циклу вегетації декількох видів плодових дерев (яблуні, абрикоси, мигдаль, хурма), а також здійснення метеорологічних спостережень, що накладаються на календар етапів вегетації. Крім того, метою є створення структури навчання штучного інтелекту для розпізнавання видів плодових дерев за фото- та відеоматеріалами.

Перше завдання полягає в організації та плануванні фенологічних спостережень. Це включає визначення об'єктів дослідження, таких як яблуня, абрикоса, мигдаль і хурма, а також встановлення критеріїв для збору відео даних на кожному етапі вегетаційного циклу. Потрібно розробити графік

спостережень для фіксації всіх ключових фенологічних фаз, таких як цвітіння, зав'язування плодів, ріст плодів, дозрівання і листопад. Наступний крок – збір відео даних. Це включає використання сучасних засобів для зйомки відео, таких як дрони і стаціонарні камери, і регулярний запис відеоматеріалів протягом усього вегетаційного циклу. Важливо також фіксувати метеорологічні умови під час зйомки, такі як температура, вологість, опади і сонячне випромінювання. Після збору відео даних необхідно їх обробити та зберегти у відповідному форматі, організувавши систему зберігання для забезпечення їх доступності та цілісності.

Друге завдання полягає в класифікації матеріалів і поділі на види плодових дерев, етапи вегетації та, де можливо, поділі на сорти плодових дерев. Це включає розробку методології для класифікації зібраних відеоматеріалів за видами плодових дерев та визначення основних характеристик для ідентифікації кожного виду дерев, таких як форма крони, листя, квіти і плоди. Потрібно також аналізувати відеоматеріали для визначення фенологічних фаз кожного виду плодових дерев і створювати еталонні відеозаписи для кожної фази вегетації. Щодо класифікації за сортами, необхідно ідентифікувати сортові особливості на основі відеоматеріалів, такі як забарвлення плодів, форма і розмір, і створити базу даних відеоматеріалів для кожного сорту плодових дерев.

Третє завдання полягає у написанні програми з моделлю для навчання штучного інтелекту на основі типу виявлення об'єктів для реалізації виявлення типу плодового дерева по відео, виявлення етапу вегетації та виявлення сорту. Це включає вибір алгоритмів машинного навчання для розпізнавання об'єктів, таких як нейронні мережі та методи глибокого навчання, створення навчальних наборів даних на основі зібраних та класифікованих відеоматеріалів та розробку алгоритмів для автоматичного виявлення видів плодових дерев, етапів вегетації та сортів. Після цього необхідно написати

програмний код для реалізації моделі навчання штучного інтелекту і протестувати та налагодити програму для забезпечення високої точності розпізнавання. Далі йде процес навчання моделі на зібраних відеоматеріалах і проведення тестування моделі для оцінки її точності та ефективності, з подальшим внесенням коректив у модель на основі результатів тестування. Нарешті, розроблену модель слід інтегрувати в існуючі системи моніторингу та управління садами і оцінити ефективність практичного застосування моделі в умовах реального садівництва.

Ці завдання дозволять досягти мети дипломної роботи та продемонструвати практичну цінність застосування штучного інтелекту у сучасному садівництві, підвищуючи ефективність та точність управління аграрними процесами.

2.2 Об'єкти досліджень

Об'єктами досліджень у роботі є плодові дерева чотирьох видів:

абрикос (*Prunus armeniaca*) сорт Голдріч

яблуня (*Malus domestica*) сорти Білий налив, Голден, Слава переможцям, Фуджі

хурма (*Diospyros kaki*) сорт Нікільська

мигдаль (*Prunus dulcis*) сорт Вікторія, які вирощуються у природних умовах садівництва

Дослідження охоплюють усі етапи вегетації дерев — від появи перших бруньок навесні до опадання листя восени. Основна увага приділялася зовнішнім ознакам дерев, включаючи:

Форма та колір листя: Спостерігалися зміни у кольорі, розмірі та формі листя на різних стадіях росту.

Квіти: Вивчалися їх форма, кількість та період цвітіння.

Плоди: Фіксувалися особливості плодів, такі як розмір, форма, текстура та колір.

Загальний вигляд дерева: Оцінювалася структура крони, розміщення гілок та зміни протягом сезону.

Дослідження охоплюють також умови вирощування дерев, включаючи тип ґрунту, рівень освітленості, режим зрошення та температурні показники.

Додатково, об'єктом аналізу виступає зібраний фенологічний матеріал у вигляді фото- та відеоданих, які використовуються для створення навчального датасету. Ці дані дозволяють моделювати процес ідентифікації дерев за допомогою штучного інтелекту.

2.3 Методика досліджень

I. Фенологічні дослідження:

- Вивчення та документування всіх етапів вегетаційного циклу яблуні, абрикоси, мигдалю та хурми.
- Визначення ключових фенологічних фаз (цвітіння, зав'язування плодів, ріст плодів, дозрівання, листопад тощо).

II. Створення структури навчання штучного інтелекту:

- Розробка методики збору фото- та відеоматеріалів для навчання штучного інтелекту.
- Створення та підготовка навчальних наборів даних, що включають зображення та відео різних видів плодових дерев у різні фенологічні фази.

- Розробка та тренування моделей машинного навчання для автоматичного розпізнавання видів плодових дерев.

- Оцінка точності та ефективності розроблених моделей.

III. Інтеграція результатів:

- Аналіз результатів фенологічних досліджень та метеорологічних спостережень.

- Впровадження розроблених моделей штучного інтелекту у системи моніторингу та управління плодовими садами.

- Розробка рекомендацій для практичного використання результатів дослідження в аграрному секторі.

2.4 Місце й умови проведення досліджень

Дослідження проводились у змішаному саду, який включає різні види плодових дерев: яблуня, абрикос, хурма, мигдаль. Такий сад забезпечив необхідну різноманітність об'єктів для збору даних та дозволив вивчити особливості росту та розвитку дерев в умовах спільного вирощування.

Місце проведення досліджень:

Локація: Змішаний сад, розташований у регіоні з помірно-континентальним кліматом.

Тип ґрунтів: Чорноземний ґрунт із високою природною родючістю, що сприяє вирощуванню широкого спектра плодових дерев.

Кліматичні умови:

Середньорічна температура: +11...+13°C.

Сезонні коливання температури:

Зимові мінімальні температури: від -5°C до -10°C .

Літні максимальні температури: до $+35^{\circ}\text{C}$, у спекотні періоди можливі короткочасні підвищення до $+40^{\circ}\text{C}$.

Рівень опадів: 350–450 мм на рік, переважно у вигляді дощів у весняно-літній період.

Характерна низька кількість опадів у літні місяці (липень-серпень).

Природне освітлення:

Тривалість дня влітку: до 15 годин.

Висока сонячна активність із можливістю пересихання ґрунту влітку.

Догляд за деревами:

Стандартні агротехнічні заходи, зокрема полив, обрізка та захист від шкідників, проводились рівномірно для всіх дерев.

Умови догляду були ідентичними для всіх об'єктів дослідження, щоб виключити вплив зовнішніх факторів на результати.

Розташування дерев:

Дерева різних видів висаджені у змішаних групах, що дозволило спостерігати їхню взаємодію в умовах природного середовища.

Спостереження та зйомки:

Дослідження охоплювали всі етапи вегетації дерев, від появи бруньок до опадання листя.

Для кожного дерева проводилась регулярна фото- та відеофіксація у природних умовах, що дозволило отримати реалістичні дані для моделювання.

Технічне обладнання:

Використовувались сучасні HD-камери для зйомок, що забезпечувало високу якість фото- та відеоматеріалів. Камери розташовувалися під фіксованими кутами для збереження однакових умов зйомки.

Час проведення спостережень:

Спостереження проводились протягом дев'яти місяців, охоплюючи всі сезонні зміни: весняне пробудження, активну вегетацію влітку, плодоношення та осіннє листопадіння.

Переваги змішаного саду для дослідження:

Різноманітність даних: Завдяки спільному вирощуванню різних видів дерев було отримано дані про взаємодію культур та їх реакцію на однакові умови середовища. Наближення до реальних умов: Змішаний сад максимально відображає практичні умови вирощування плодкових дерев, що робить результати дослідження релевантними для впровадження в інших регіонах.

Змішаний сад забезпечив оптимальні умови для збору якісних даних, необхідних для розробки та навчання моделі штучного інтелекту.

3 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ

3.1 Постановка експерименту.

Постановка експерименту полягає у визначенні основних завдань і умов для збору даних, розробки навчального датасету та створення моделі штучного інтелекту для ідентифікації чотирьох видів плодових дерев (яблуня, абрикос, хурма, мигдаль). Експеримент мав на меті забезпечити створення точного, репрезентативного набору даних, який дозволяє моделі враховувати сезонні зміни дерев та їх особливості.

3.2 Збір та аналіз даних на основі фенологічні спостережень.

Головною метою цього етапу було створення різноманітного датасету, який охоплює всі фази вегетації дерев: від моменту, коли на деревах з'являються перші бруньки, до завершення циклу і опадання листя. Це дозволяє моделі в подальшому враховувати сезонні зміни у зовнішньому вигляді дерев і правильно класифікувати їх у будь-який час року.

Умови проведення досліджень

Протягом 8 місяців було організовано регулярні зйомки дерев чотирьох видів: абрикоса, яблуні, хурми та мигдалю. Відеозйомки виконувались у природному середовищі, що дозволило отримати реалістичні дані для тренування моделі.

Період проведення:

Спостереження розпочалися навесні, **15 березня 2024 року**, і охоплювали весь вегетаційний період дерев чотирьох видів: **хурма, абрикос, мигдаль, яблуня**. Через те що в році спостереження тепловий режим для початку вегетації почався запізно але інтенсивно то деяки етапи вегетації також пройшли дуже швидко ,наприклад для мигдаль від початку цвітіння до повногоопаду суцвіть пройшло повних 8 днів. Дослідження тривали до 15 жовтня, що дозволило зібрати повний спектр даних, починаючи від появи бруньок і закінчуючи фазою старіння та початком спокою. З 15.03.24 по 30.06.24 спостереження та

запис відео проводився кожні два дні, з маленькими перервами через погодні умови. З 01.07.24 по 15.10.24 спостереження та запис проводився раз на тиждень. Через те що ці види дерев мають різні строки вегетації та різні дати опадання листя, наприклад у першу чергу закінчилася вегетація у мигдалю та хурми наприкінці жовтня, а вже абрикос та яблука поали у листопаді, а після вересня вегетаційних процесів що могли б додати нових ознак у спостереження було дуже мало. Ми принімаємо кінцевий строк спостереження за 15.10.24.

Особливості проведення спостережень

Мигдаль (*Prunus dulcis*) сорт Вікторія:

Для сорту мигдалю Вікторія (*Prunus dulcis*) періоди вегетації можуть бути наступними. Залежно від кліматичних умов півдня Одеської області (помірно-континентальний клімат), етапи вегетації включатимуть такі фази та приблизні дати:

Особливості для сорту "Вікторія"

Ранній розвиток: Сорт "Вікторія" відомий ранньою фазою брунькування, що пов'язано з теплими веснами у південному регіоні.

Тривале цвітіння: Тривалість цвітіння може становити до 10 днів, залежно від погодних умов.

Дозрівання плодів: Сорт має відносно короткий період між раннім розвитком плодів та їхньою зрілістю, що є характерним для промислових сортів мигдалю.

Підготовка до зими: Період старіння та спокою починається раніше через природне завершення вегетації у південному регіоні.

Цей розподіл дозволяє точніше організувати спостереження за деревами та визначати ключові періоди для догляду, збору даних і агротехнічних заходів.

Якщо необхідно уточнити дати залежно від мікроклімату чи інших факторів, їх можна адаптувати. (Додаток 1)

Для сортів яблуні (*Malus domestica*) — **Білий налив, Голден, Слава переможцям, Фуджі** — періоди вегетації залежать від їхніх біологічних особливостей і кліматичних умов півдня Одеської області. Нижче наведено приблизні дати фаз вегетації для цих сортів.

Особливості для кожного сорту

Білий налив (ранньостиглий сорт):

Зрілість плодів: 01.07 – 20.07.

Плоди дозрівають найшвидше серед вказаних сортів, потребують регулярного збирання через схильність до опадання.

Голден (середньостиглий сорт):

Зрілість плодів: 15.08 – 05.09.

Характеризується стабільністю дозрівання та тривалим збереженням плодів на дереві.

Слава переможцям (середньопізній сорт):

Зрілість плодів: 01.09 – 15.09.

Відзначається високою якістю плодів і хорошою транспортабельністю.

Фуджі (пізньостиглий сорт):

Зрілість плодів: 15.09 – 15.10.

Має найдовший період дозрівання, плоди зберігаються до зими.

Особливості ведення спостережень

Білий налив: Більше уваги приділяється фазам раннього розвитку та цвітіння через короткий вегетаційний цикл.

Фуджі: Потрібен детальніший контроль на пізніх етапах дозрівання плодів.

Голден і Слава переможцям: Середній цикл вегетації дозволяє зосередитися на фазах дозрівання плодів.

Ці періоди дозволяють оптимізувати спостереження за кожним сортом яблуні залежно від їхніх характеристик та потреб у догляді. (Додаток 2)

Особливості сорту Голдріч

Середньостиглість:

Плоди дозрівають у другій половині червня — на початку липня, що дозволяє отримати урожай після ранніх сортів, таких як Ерлі Оранж.

Вегетаційний цикл триває близько 95–110 днів.

Тривале цвітіння:

Завдяки подовженому періоду цвітіння (до 10 днів) сорт забезпечує високу ймовірність запилення навіть при нестабільних погодних умовах.

Стійкість до хвороб:

Сорт відзначається високою стійкістю до моніліозу та інших грибкових захворювань, що робить його придатним для промислового вирощування.

Висока врожайність:

Голдріч забезпечує стабільний урожай великих, ароматних і транспортабельних плодів.

Потреби в догляді:

Висока чутливість до нестачі вологи в період розвитку плодів. Необхідно забезпечувати регулярний полив, особливо у травні-червні.

Застосування даних

Періоди вегетації дозволяють планувати агротехнічні заходи: обрізання, обробку від шкідників і хвороб, а також полив.

Дані важливі для оптимізації строків збору врожаю, щоб забезпечити максимальну якість і мінімальні втрати. (Додаток 3)

Хурма (*Diospyros kaki*), сорт Нікітська

Сорт Нікітська — це середньостиглий сорт хурми, що відзначається високою врожайністю, морозостійкістю і приємним смаком плодів. Його вегетаційний період триває значно довше порівняно з іншими плодовими деревами, оскільки хурма має тривалий цикл дозрівання.

Особливості сорту Нікітська

Тривалий період дозрівання:

Сорт має пізню фазу дозрівання плодів, що припадає на середину жовтня — кінець жовтня.

Стійкість до холодів:

Сорт здатен витримувати зниження температур до -20°C , що робить його придатним для вирощування в регіонах із помірним кліматом.

Потреба у теплі:

Для повноцінного дозрівання плодів необхідна тривала тепла осінь. У південній частині Одеської області ці умови оптимальні.

Якість плодів:

Великі, соковиті плоди з високим вмістом цукрів, придатні для свіжого споживання та переробки.

Тривале цвітіння:

Період цвітіння триває до 10 днів, що сприяє доброму запиленню навіть за нестабільних погодних умов.

Застосування даних

Агротехнічні заходи: Спланувати полив, обрізку та обробку від шкідників залежно від фаз вегетації.

Оптимізація врожайності: Застосувати засоби для стимуляції росту на етапах розвитку плодів.

Маркетингові переваги: Завдяки пізньому дозріванню плоди сорту Нікітська зберігають високий попит наприкінці сезону, коли пропозиція хурми на ринку знижується. (Додаток 4)

Локація:

Дослідження проводились у саду з молодими деревами від 3 до 10 років добре розвиненими деревами для свого віку кожного типу, де всі рослини перебували в природних умовах.

Обладнання:

Для зйомок використовували HD-камери з роздільною здатністю 1280×720 пікселів, що забезпечувало високу деталізацію зображень.

Умови освітлення:

Зйомки виконували при природному освітленні в різні частини дня (вранці, вдень, у вечірній час) для створення різноманітності візуальних умов.

Процес зйомки

Тривалість кожного запису:

Відео тривали від 1 до 2 хвилин, охоплюючи як загальний вигляд дерева, так і крупні плани листя, гілок, квітів та плодів.

Структура збереження даних про спостереження

Для організації та зручного доступу до всіх зібраних відеоматеріалів було розроблено чітку ієрархічну структуру зберігання даних. Ця структура дозволяє легко ідентифікувати відео за видом дерева, етапом вегетації та датою зйомки.

```
/dataset/  
  
  /almond/  
  
  /apricot/  
  
  /persimmon/  
  
  /apple/
```

Назви файлів містять дату зйомки та точний час наприклад VID_20240416_102829.mp4

Обробка відеоматеріалів

Мета обробки

Завдання полягало у перетворенні відеозаписів на структурований набір зображень, який можна було використовувати для тренування моделі.

Процес обробки відеометріалів відбувався на ПК з використання мови програмування Python. Для роботи з відео на ПК було встановлено необхідне ПЗ. Основні компоненти Python-середовища

Python-інтерпретатор

Версія Python: Рекомендується встановлювати останню стабільну версію Python 3 (наприклад, 3.10 або 3.11).

Дистрибутив: Завантажте Python із офіційного сайту.

Менеджер пакетів

Встановлений разом із Python менеджер пакетів `pip` використовується для завантаження бібліотек.

Віртуальне середовище

Інструменти: `venv` (вбудований у Python) або `virtualenv`.

Призначення: Створення ізольованих середовищ для роботи з проектами, щоб уникнути конфліктів між бібліотеками.

Редактор коду

Рекомендація: Visual Studio Code (VS Code).

Необхідні плагіни для VS Code

Python - Розширення для синтаксичного аналізу, автодоповнення та запуску Python-скриптів.

PyLance - Потужний інструмент для статичного аналізу коду Python.

Jupyter - Для роботи з Jupyter Notebook у VS Code.

Code Runner - Дозволяє швидко запускати Python-скрипти безпосередньо в редакторі.

AutoDocstring - Генерує документацію до функцій та класів Python.

Бібліотеки для роботи з даними та обробки відео

NumPy: Для роботи з числовими даними та масивами.

```
pip install numpy
```

Pandas: Для обробки табличних даних.

```
pip install pandas
```

OpenCV: Для обробки зображень і відео.

```
pip install opencv-python opencv-python-headless
```

MoviePy: Для редагування та обробки відео.

```
pip install moviepy
```

Matplotlib: Для візуалізації даних.

```
pip install matplotlib
```

Розбиття відео на кадри:

Використовуючи бібліотеку OpenCV у Python, кожне відео було розбите на окремі кадри із частотою 10 кадр на 1 секунд. Це забезпечувало різноманітність даних, уникаючи дублювання. (Додаток 5).

Під час створення зображень кожному зображенню давали ім'я:

almond_20240412_155022_00.17.jpg, apple_20240409_215804_00.26.jpg

apricot_20240428_164720_00.21.jpg, persimmon_20240419_161834_00.10.jpg

котре має такий формат «назва»_«дата»_«час»_«номер кадру»

Так як датасет універсальний то для слідуючого етапу наприклад для яблуні

можна додавати код сорту щоб навчити розрізняти сорти одного виду

плового дерева:

apple_20240409_215804_00.26_01.jpg,

apple_20240409_215804_00.26_02.jpg

apple_20240409_215804_00.26_03.jpg

apple_20240409_215804_00.26_04.jpg

де 01- Голден, 02 - Фуджі,03 - Слава переможцям,04 - Семеринка.

Після отримання зображень для датасету, вийшло що кількість фото є різною через різний таймінг відео та тому що кількість яблунь в саду приблизно 60%.

Щоб отримати датасет без перекосів методом відбору залишили для кожного виду дерев по 2000 зображень. Наступним етапом було трансформувати зображення методом аугментації даних. Для збільшення розміру датасету застосовувались методи аугментації: горизонтальне віддзеркалення, зміна яскравості, повороти. Це дозволило отримати 10 000 зображень для кожного класу. (Додаток 6) Додаємо усі файли що знаходяться у 4 папках у архів dataset.zip та завантажуюмо на Google Disk у паку проекту.

3.3 Розробка програмного забезпечення для отримання математичної моделі.

У цьому розділі ми описуємо процес розробки програмного забезпечення, яке дозволяє створити математичну модель для класифікації плодового дерева за типом. Подальшу роботу буде виконано в середовищі Google Colab, що забезпечує інтеграцію з Google Drive для зручного зберігання даних і моделей.

Чому ми використовували Google Colab?

Google Colab — це хмарне середовище, яке надає потужний набір інструментів для машинного навчання та обробки даних. Ми обрали його для нашого проекту через низку важливих переваг.

Безкоштовне використання апаратних ресурсів

Переваги:

Google Colab надає безкоштовний доступ до GPU (графічних процесорів) і TPU (тензорних процесорів), що значно прискорює навчання моделей машинного навчання.

Використання потужних апаратних ресурсів особливо важливо для обробки великої кількості зображень та тренування глибоких нейронних мереж.

Чому це важливо?

Навчання моделей на локальних комп'ютерах може займати багато часу або навіть бути неможливим через обмеженість апаратних ресурсів.

Інтеграція з Google Drive

Переваги:

Colab легко підключається до Google Drive, що дозволяє зберігати дані, моделі та результати безпосередньо у хмарі.

Це забезпечує безпеку даних і доступ до них із будь-якого пристрою.

Чому це важливо?

Зберігання великих датасетів або моделей локально може бути проблематичним через обмеження обсягу пам'яті. Інтеграція з Google Drive розв'язує цю проблему.

Простота використання

Переваги:

Colab має зручний інтерфейс, схожий на Jupyter Notebook, що дозволяє інтерактивно писати код, виконувати його частинами та отримувати результати в реальному часі.

Можливість додавати текстові блоки, зображення та графіки робить процес документування проекту зручним.

Чому це важливо?

Зручність у використанні сприяє швидкому освоєнню середовища навіть новачками.

Підтримка спільної роботи

Переваги:

Google Colab дозволяє кільком користувачам працювати над одним документом одночасно, забезпечуючи синхронізацію змін у реальному часі.

Чому це важливо?

Для командних проєктів це значно спрощує взаємодію, обмін ідеями та спільне внесення змін у код.

Попередньо встановлені бібліотеки

Переваги:

Colab має попередньо встановлені бібліотеки, такі як TensorFlow, Keras, NumPy, Pandas та інші.

Це зменшує час, витрачений на налаштування середовища.

Чому це важливо?

Ви можете зосередитися на розробці моделей і обробці даних, не витрачаючи час на встановлення бібліотек.

Масштабованість

Переваги:

Якщо потрібно більше ресурсів, можна перейти на платну версію Colab Pro, яка надає швидші GPU та більший обсяг пам'яті.

Чому це важливо?

Для масштабних проєктів це дозволяє поступово збільшувати ресурси без необхідності купувати дорогий локальний сервер.

Кросплатформеність

Переваги:

Colab працює в браузері, і вам не потрібно турбуватися про операційну систему чи потужність локального пристрою.

Чому це важливо?

Ви можете працювати над проєктом із будь-якого пристрою, де є доступ до інтернету.

Google Colab забезпечив оптимальні умови для нашого проєкту, оскільки поєднує в собі потужність хмарних обчислень, простоту використання та інтеграцію з Google Drive. Це дозволило ефективно обробляти дані, тренувати модель і зберігати результати, не витрачаючи значних ресурсів на локальне обладнання. Нижче будуть описані етапи розробки коду для навчання моделі

Підключення до Google Drive

Google Drive буде використовуватися для зберігання даних (зображень, моделей) і доступу до них безпосередньо з Google Colab.

```
from google.colab import drive
drive.mount('/content/drive')
```

Результат у терміналі

```
Mounted at /content/drive
```

Розпакування даних у Colab

Завантажуємо файл у робочий каталог Colab і розпакуйте його:

```
import zipfile
import os

# Шлях до zip-файлу в Google Drive
zip_path = '/content/drive/My Drive/Colab Notebooks/dataset.zip'
```

```
extract_path = '/content/dataset'

# Розпакування
with zipfile.ZipFile(zip_path, 'r') as zip_ref:
    zip_ref.extractall(extract_path)

print("Дані розпаковані!")
```

Результат у терміналі

Дані розпаковані!

Створення моделі для класифікації дерев

Імпортуємо бібліотеки для створення моделі:

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D,
Flatten, Dense, Dropout
from tensorflow.keras.preprocessing.image import
ImageDataGenerator
from tensorflow.keras.optimizers import Adam
```

Готуємо генератор даних для тренування та перевірки:

```
train_dir = '/content/dataset/dataset' # Шлях до ваших даних

#Генератори даних
train_datagen = ImageDataGenerator(
    rescale=1./255,          # Нормалізація пікселів до [0, 1]
    validation_split=0.2    # Розділяємо дані на тренувальні та
    валідаційні
)

# Завантажуємо тренувальні та валідаційні дані
train_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=(224, 224), # Розмір зображень
    batch_size=32,
    class_mode='categorical', # Для багатокласової класифікації
```



```
        subset='training'
    )

validation_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical',
    subset='validation'
)
```

Результат у терміналі

```
Found 30970 images belonging to 4 classes.
Found 7742 images belonging to 4 classes.
```

Пояснення коду

У змінній `train_dir` зазначається шлях до папки, де зберігаються зображення для навчання та перевірки.

`rescale=1./255`: Кожен піксель зображення нормалізується до діапазону $[0, 1]$, що є стандартною практикою для підвищення ефективності навчання нейронної мережі.

`validation_split=0.2`: Дані автоматично розподіляються на дві частини:

80% (тренувальні дані) — для навчання моделі.

20% (валідаційні дані) — для перевірки точності моделі

`target_size=(224, 224)`: Усі зображення змінюються до розміру 224×224 пікселів.

`batch_size=32`: Дані поділяються на батчі по 32 зображення для оптимізації навчання.

`class_mode='categorical'`: Використовується для багатокласової класифікації, тобто кожне зображення належить до одного з 4 класів.

`subset='training'`: Вибираються зображення, що потрапляють до тренувальної частини (80%).

Аналогічно тренувальному генератору, але з `subset='validation'`, що обирає 20% даних для перевірки.

Результат у терміналі

Found 30970 images belonging to 4 classes:

У тренувальній частині (`subset='training'`) було автоматично обрано 30 970 зображень, які розподілені між 4 класами.

Found 7742 images belonging to 4 classes:

У валідаційній частині (`subset='validation'`) було обрано 7 742 зображення, які також розподілені між тими самими 4 класами.

Завдяки параметру `validation_split=0.2`, програма автоматично поділила дані: 80% для навчання.

20% для перевірки.

Цей розподіл виконується випадковим чином, але зберігається баланс класів у кожному піднаборі.

Ми не обирали зображення вручну для кожного піднабору, оскільки ця функціональність автоматизована в `ImageDataGenerator`.

Автоматичний розподіл спрощує роботу та забезпечує збалансованість даних у тренувальному та валідаційному наборах.

Це дозволяє зосередитися на побудові та навчанні моделі, не витрачаючи час на розподіл

Створюємо модель нейронної мережі

```
model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(224, 224,
3)),
    MaxPooling2D((2, 2)),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
```

```

Conv2D(128, (3, 3), activation='relu'),
MaxPooling2D((2, 2)),
Flatten(),
Dense(128, activation='relu'),
Dropout(0.5),
Dense(4, activation='softmax') # 4 класи: apricot, apple,
almond, persimmon
])

```

`Conv2D(32, (3, 3), activation='relu', input_shape=(224, 224, 3))`: Перший згортковий шар із 32 фільтрами розміром 3×3 , які вилучають базові ознаки, такі як краї та текстури. Функція активації ReLU зберігає позитивні значення, видаляючи негативні. Розмір вхідних даних визначений як 224×224 пікселів із трьома каналами (кольорове зображення RGB).

`MaxPooling2D((2, 2))`: Шар максимального пулінгу, який зменшує розмірність зображення вдвічі ($224 \times 224 \rightarrow 112 \times 112$). Він вибирає максимальне значення з кожного блоку 2×2 , знижуючи обчислювальну складність і зберігаючи ключові ознаки.

`Conv2D(64, (3, 3), activation='relu')`: Другий згортковий шар із 64 фільтрами розміром 3×3 . Він вивчає складніші ознаки, такі як шаблони й текстури. Використовується функція активації ReLU для нелінійності.

`MaxPooling2D((2, 2))`: Ще один шар максимального пулінгу, який зменшує розмірність зображення до $112 \times 112 \rightarrow 56 \times 56$. Це допомагає зменшити розмір даних, зберігаючи важливі ознаки.

`Conv2D(128, (3, 3), activation='relu')`: Третій згортковий шар із 128 фільтрами, який дозволяє мережі розпізнавати більш деталізовані та специфічні особливості зображення.

`MaxPooling2D((2, 2))`: Зменшує розмірність до $56 \times 56 \rightarrow 28 \times 28$, видаляючи надлишкову інформацію.

`Flatten()`: Перетворює багатовимірний вихід зі згорткових шарів у вектор розміром ($28 \times 28 \times 128 \rightarrow 100352$), який може бути переданий до повнозв'язного шару для подальшої обробки.

Dense(128, activation='relu'): Повнозв'язний шар із 128 нейронів, який обробляє отриманий вектор для виявлення складніших залежностей у даних. Функція активації ReLU допомагає вивчати нелінійні зв'язки.

Dropout(0.5): Запобігає перенавчанню, випадково вимикаючи 50% нейронів під час кожного етапу навчання.

Dense(4, activation='softmax'): Вихідний шар із 4 нейронами, кожен із яких відповідає одному класу: apricot, apple, almond, persimmon. Softmax-активація перетворює виходи в ймовірності для кожного класу, де сума всіх ймовірностей дорівнює 1. Це забезпечує остаточний вибір класу зображення.

Компілюємо модель

```
model.compile(  
    optimizer=Adam(learning_rate=0.001),  
    loss='categorical_crossentropy',  
    metrics=['accuracy']  
)
```

optimizer=Adam(learning_rate=0.001): Використовується оптимізатор Adam, який комбінує методи адаптивного градієнтного спуску та моментуму для більш ефективного і швидкого навчання моделі. Початкова швидкість навчання встановлена на 0.001.loss='categorical_crossentropy': Функція втрат categorical_crossentropy використовується для багатокласової класифікації. Вона порівнює ймовірності, передбачені моделлю, із фактичними класами, і обчислює похибку, яку модель повинна мінімізувати.metrics=['accuracy']: Як метрику оцінки точності навчання та валідації моделі використовується показник accuracy (точність). Він вимірює відсоток правильно класифікованих зображень від загальної кількості.

Навчаємо модель

```
history = model.fit(  
    train_generator,  
    epochs=10,  
    validation_data=validation_generator  
)
```

`train_generator`: Генератор тренувальних даних, який забезпечує модель батчами зображень і відповідних міток класів. Дані попередньо масштабовані й розподілені на категорії.
`epochs=10`: Кількість епох навчання, тобто модель пройде через увесь набір тренувальних даних 10 разів, поступово покращуючи свої параметри.

`validation_data=validation_generator`: Генератор валідаційних даних, який забезпечує модель даними для перевірки її точності після кожної епохи навчання. Валідаційні дані не беруть участь у навчанні, а використовуються лише для оцінки продуктивності.

`model.fit`: Функція для навчання моделі. Вона виконує ітеративний процес, під час якого модель:

1. Обробляє тренувальні дані.
2. Оновлює свої параметри на основі функції втрат.
3. Оцінює свою продуктивність на валідаційному наборі даних після кожної епохи.

Після виконання цієї команди виводяться метрики, такі як точність і функція втрат для кожної епохи, що дозволяє оцінити, наскільки добре модель навчається та перевіряється.

Результат у терміналі

```
Epoch 1/10  
/usr/local/lib/python3.10/dist-  
packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:
```

```
122: UserWarning: Your `PyDataset` class should call
`super().__init__(**kwargs)` in its constructor. `**kwargs` can
include `workers`, `use_multiprocessing`, `max_queue_size`. Do not
pass these arguments to `fit()`, as they will be ignored.
```

```
self._warn_if_super_not_called()
968/968 _____ 4174s 4s/step -
accuracy: 0.3358 - loss: 1.3756 - val_accuracy: 0.3610 - val_loss:
1.4038
Epoch 2/10
968/968 _____ 4141s 4s/step -
accuracy: 0.5563 - loss: 1.0217 - val_accuracy: 0.3809 - val_loss:
1.4915
Epoch 3/10
968/968 _____ 4043s 4s/step -
accuracy: 0.6855 - loss: 0.7777 - val_accuracy: 0.4296 - val_loss:
1.5008
Epoch 4/10
968/968 _____ 4089s 4s/step -
accuracy: 0.7840 - loss: 0.5563 - val_accuracy: 0.4651 - val_loss:
1.5294
Epoch 5/10
968/968 _____ 3993s 4s/step -
accuracy: 0.8593 - loss: 0.3677 - val_accuracy: 0.4513 - val_loss:
1.9598
Epoch 6/10
968/968 _____ 4131s 4s/step -
accuracy: 0.9051 - loss: 0.2583 - val_accuracy: 0.5159 - val_loss:
1.8811
Epoch 7/10
968/968 _____ 4084s 4s/step -
accuracy: 0.9405 - loss: 0.1715 - val_accuracy: 0.5087 - val_loss:
1.7692
Epoch 8/10
968/968 _____ 4079s 4s/step -
accuracy: 0.9532 - loss: 0.1376 - val_accuracy: 0.4939 - val_loss:
2.1416
Epoch 9/10
```

Пояснення метрик

accuracy: Точність на тренувальному наборі даних. Це частка правильно класифікованих зображень.

Наприклад:

accuracy: 0.3358 означає, що модель правильно класифікує 33.58% тренувальних зображень.

loss: Значення функції втрат на тренувальному наборі. Чим менше це значення, тим краще модель адаптується до даних.

Наприклад:

loss: 1.3756 свідчить про похибку в передбаченнях на даному етапі.

val_accuracy: Точність на валідаційному наборі даних. Оцінює, наскільки добре модель може узагальнювати свої знання на даних, які вона раніше не бачила.

Наприклад:

val_accuracy: 0.3610 означає, що модель правильно класифікує 36.10% валідаційних зображень.

val_loss: Значення функції втрат на валідаційному наборі. Дає уявлення про те, наскільки добре модель передбачає результати для нових даних.

Наприклад:

val_loss: 1.4038 показує похибку в передбаченнях на валідаційному наборі.

Тренд метрик:

Покращення accuracy та зниження loss: Свідчить, що модель краще адаптується до тренувальних даних.

val_accuracy та val_loss: Якщо вони не змінюються або погіршуються, це може бути ознакою перенавчання моделі.

Загальні спостереження:

З кожною епохою точність (accuracy) тренувальних даних зростає, а значення функції втрат (loss) знижується.

Валідаційна точність (val_accuracy) і втрати (val_loss) мають більш повільний прогрес, що може свідчити про складність узагальнення знань на нові дані.

Перевірка моделі

Оцінюємо модель на валідаційних даних

```
val_loss, val_acc = model.evaluate(validation_generator)
print(f"Validation Loss: {val_loss}")
print(f"Validation Accuracy: {val_acc}")
```

Цей код використовується для оцінки продуктивності моделі на валідаційному наборі даних. Ось що відбувається покроково:

`model.evaluate(validation_generator):`

Функція `evaluate` обчислює функцію втрат (`val_loss`) і метрику точності (`val_acc`) на валідаційному наборі даних.

Використовується генератор даних `validation_generator`, який забезпечує модель валідаційними зображеннями і відповідними мітками класів.

`val_loss:`

Значення функції втрат на валідаційних даних. Відображає, наскільки модель помиляється у своїх передбаченнях.

Менше значення `val_loss` означає кращу продуктивність моделі.

`val_acc:`

Точність моделі на валідаційних даних. Визначає відсоток правильно класифікованих зображень.

Наприклад, `val_acc = 0.80` означає, що модель правильно класифікує 80% зображень у валідаційному наборі.

`print:`

Виводить результати обчислень у зручному текстовому форматі:

Validation Loss: Значення функції втрат.

Validation Accuracy: Точність класифікації.

Результат у терміналі

accuracy: 0.4983 - loss: 2.6425

Validation Loss: 2.6357409954071045

Validation Accuracy: 0.4994833469390869

Модель була оцінена на валідаційному наборі даних, і результати показують, що її **точність (accuracy)** становить 49.83%. Це означає, що модель правильно класифікує майже половину зображень у валідаційному наборі.

Функція втрат (loss) для валідаційного набору дорівнює 2.6425, що свідчить про значну похибку моделі при передбаченні класів. Високе значення втрат свідчить про те, що модель ще недостатньо добре адаптована до валідаційних даних.

Validation Loss складає 2.6357409954071045, що близьке до функції втрат на тренувальних даних. Це означає, що модель демонструє стабільну поведінку на різних наборах даних, але її продуктивність все ще залишає бажати кращого.

Validation Accuracy дорівнює 49.95%, що підтверджує, що майже половина зображень у валідаційному наборі були правильно класифіковані. Це середній показник для подібної моделі, який можна покращити за рахунок подальшого навчання чи покращення якості даних.

Загалом, результати показують, що модель ще не досягла оптимального рівня продуктивності і потребує оптимізації або роботи з більш якісним та різноманітним набором даних.

Збереження моделі для подальшого використання

```
model.save('/content/drive/My Drive/Colab Notebooks/tree_classification_model.h5')
```

```
print("Модель збережена!")
```

Результат у терміналі

```
WARNING:absl:You are saving your model as an HDF5 file via
`model.save()` or `keras.saving.save_model(model)`. This file
format is considered legacy. We recommend using instead the
native Keras format, e.g. `model.save('my_model.keras')` or
`keras.saving.save_model(model, 'my_model.keras')`.
```

```
Модель збережена!
```

Тестування моделі списком зображень

```
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing.image import load_img,
img_to_array
import numpy as np
import os
from collections import Counter

# Завантаження моделі
model = load_model('/content/drive/My Drive/Colab
Notebooks/tree_classification_model.h5')
print("Модель завантажена!")

# Словник класів
class_indices = {0: 'almond', 1: 'apple', 2: 'apricot', 3:
'persimmon'}

# Функція для прогнозу класу одного зображення
def predict_class(image_path):
    test_image = load_img(image_path, target_size=(224, 224))
    test_image = img_to_array(test_image) / 255.0 # Масштабування
    test_image = np.expand_dims(test_image, axis=0) # Додаємо
вимір для batch
```

```

prediction = model.predict(test_image)
return np.argmax(prediction) # Повертаємо номер класу

# Шлях до папки з тестовими зображеннями
test_folder_path = '/content/drive/My Drive/Colab
Notebooks/test/test4'

# Зчитуємо всі зображення з папки
image_files = [os.path.join(test_folder_path, f) for f in
os.listdir(test_folder_path) if f.endswith(('.jpg', '.png'))]

# Передбачаємо клас для кожного зображення
predictions = []
for image_path in image_files:
    predicted_class = predict_class(image_path)
    predictions.append(predicted_class)
    print(f"Зображення: {os.path.basename(image_path)},
Передбачений клас: {class_indices[predicted_class]}")

# Визначаємо найчастіший клас
most_common_class = Counter(predictions).most_common(1)[0][0]
final_class = class_indices[most_common_class]
print(f"\nФінальний передбачений клас для папки: {final_class}")

```

Результат у терміналі

WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile_metrics` will be empty until you train or evaluate the model.

Модель завантажена!

```

1/1 _____ 0s 273ms/step
Зображення: some_test_image0147.jpg, Передбачений клас: almond
1/1 _____ 0s 87ms/step
Зображення: some_test_image0148.jpg, Передбачений клас: almond
1/1 _____ 0s 82ms/step
Зображення: some_test_image0149.jpg, Передбачений клас: almond
1/1 _____ 0s 82ms/step

```

```
Зображення: some_test_image0154.jpg, Передбачений клас: apricot
1/1 _____ 0s 96ms/step
Зображення: some_test_image0155.jpg, Передбачений клас: almond
1/1 _____ 0s 85ms/step
Зображення: some_test_image0156.jpg, Передбачений клас: almond
1/1 _____ 0s 95ms/step
Зображення: some_test_image0158.jpg, Передбачений клас: almond
1/1 _____ 0s 51ms/step
Зображення: some_test_image0159.jpg, Передбачений клас: almond
```

Фінальний передбачений клас для папки: almond

Це попередження означає, що метрики моделі, визначені під час її компіляції (`model.compile`), ще не були застосовані. Воно не впливає на прогноз, оскільки ми використовуємо модель лише для передбачення.

Кожен рядок вказує на час, витрачений на обробку одного зображення.

Наприклад:

0s 273ms/step: Модель витратила 273 мс на прогнозування для першого зображення.

Загалом, передбачення займає малий час, що вказує на ефективність моделі для прогнозу.

Кожне зображення аналізується окремо, і модель прогнозує клас, до якого воно належить. Наприклад:

Для зображення `some_test_image0147.jpg` передбачений клас — `almond`.

Більшість зображень у папці були віднесені до класу `almond`, за винятком одного (`some_test_image0154.jpg`), для якого модель передбачила клас `apricot`.

Клас `almond` є найчастішим серед передбачених класів для зображень у папці.

Це означає, що модель ідентифікує більшість зображень як `almond`, що вказує на переважання цього класу в тестовій вибірці.

Загальний висновок:

Модель коректно передбачає класи для кожного зображення окремо, а фінальний клас визначається як найпоширеніший серед усіх передбачених

результатів. Швидкість обробки зображень є високою, і передбачення здійснюються ефективно.

Тестування моделі відео

```
import cv2
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing.image import img_to_array
import numpy as np
from collections import Counter

# Завантаження моделі
model = load_model('/content/drive/My Drive/Colab
Notebooks/tree_classification_model.h5')
print("Модель завантажена!")

# Словник класів
class_indices = {0: 'almond', 1: 'apple', 2: 'apricot', 3:
'persimmon'}

# Функція для прогнозу класу одного кадру
def predict_class_from_frame(frame):
    # Зміна розміру кадру
    frame = cv2.resize(frame, (224, 224))
    frame = img_to_array(frame) / 255.0 # Масштабування
    frame = np.expand_dims(frame, axis=0) # Додаємо вимір для
batch
    prediction = model.predict(frame)
    return np.argmax(prediction) # Повертаємо номер класу

# Функція для обробки відео
def process_video(video_path):
    cap = cv2.VideoCapture(video_path)
    frame_rate = int(cap.get(cv2.CAP_PROP_FPS)) # Частота кадрів
у відео
    predictions = []

    print(f"Обробка відео: {video_path}")
    while cap.isOpened():
        ret, frame = cap.read()
        if not ret:
            break
```

```

# Вибираємо кадри з інтервалом (наприклад, кожну секунду)
frame_position = int(cap.get(cv2.CAP_PROP_POS_FRAMES))
if frame_position % frame_rate == 0: # Кадр кожну секунду
    predicted_class = predict_class_from_frame(frame)
    predictions.append(predicted_class)

cap.release()

# Визначення найпоширенішого класу
most_common_class = Counter(predictions).most_common(1)[0][0]
final_class = class_indices[most_common_class]
return final_class

# Шлях до відео
test_video_path = '/content/drive/My Drive/Colab
Notebooks/test/test_video1.mp4'

# Виконання обробки відео
final_class = process_video(test_video_path)
print(f"\nФінальний предсказаний клас для відео: {final_class}")

```

Результат у терміналі

WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile_metrics` will be empty until you train or evaluate the model.

Модель завантажена!

Обробка відео: /content/drive/My Drive/Colab Notebooks/test/test_video1.mp4

```

1/1 _____ 0s 273ms/step
1/1 _____ 0s 50ms/step
1/1 _____ 0s 56ms/step
1/1 _____ 0s 46ms/step
1/1 _____ 0s 50ms/step
1/1 _____ 0s 48ms/step
1/1 _____ 0s 46ms/step
1/1 _____ 0s 48ms/step
1/1 _____ 0s 46ms/step
1/1 _____ 0s 52ms/step
1/1 _____ 0s 48ms/step
1/1 _____ 0s 53ms/step
1/1 _____ 0s 57ms/step
1/1 _____ 0s 75ms/step

```

1/1	_____	0s 74ms/step
1/1	_____	0s 90ms/step
1/1	_____	0s 77ms/step
1/1	_____	0s 87ms/step
1/1	_____	0s 78ms/step
1/1	_____	0s 82ms/step
1/1	_____	0s 81ms/step
1/1	_____	0s 82ms/step

Фінальний предсказаний клас для відео: apricot

Увесь код навчання та тестування можна побачити у Додатку 7 та по лінку на COLAB

https://drive.google.com/file/d/1XKNFOYyF_5P5g8AzNygC53CtCfmof0IT/view?usp=drive_link

У Додаток 8 можна побачити графіки котрі характеризують навчання моделі.

Нижче наведемо аналіз процесу навчання моделі

Аналіз процесу навчання моделі

На основі наданих метрик навчання можна зробити такі спостереження:

Динаміка точності (accuracy):

Тренувальна точність:

Точність моделі збільшується з кожною епохою, починаючи з 33.58% на першій епосі до 96.97% на десятій.

Це свідчить про те, що модель поступово адаптувалася до тренувальних даних, вивчаючи ключові ознаки кожного класу.

Валідаційна точність:

Точність на валідаційному наборі також демонструє зростання від 36.10% на першій епосі до максимуму 51.59% на шостій епосі.

Проте після шостої епохи точність дещо знижується, що може свідчити про перенавчання моделі.

Динаміка втрат (loss):

Тренувальні втрати:

Значення функції втрат постійно зменшується, починаючи з 1.3756 на першій епосі до 0.0937 на десятій.

Це показує, що модель поступово зменшує похибки на тренувальних даних, навчаючись ефективніше.

Валідаційні втрати:

На першій епосі валідаційні втрати становили 1.4038, але після цього вони не демонструють стійкого зменшення.

Після шостої епохи валідаційні втрати починають зростати (1.8811 → 2.6357), що може свідчити про перенавчання моделі.

Залежності між метриками:

Тренувальні метрики (accuracy і loss) показують стійке покращення, оскільки модель все краще адаптується до тренувальних даних.

Проте розходження між тренувальними і валідаційними метриками після шостої епохи свідчить про те, що модель краще "запам'ятовує" тренувальні дані, але її здатність узагальнювати знання на нових даних обмежена.

Візуальний аналіз:

Графік точності:

Тренувальна точність має стабільно зростаючий тренд.

Валідаційна точність зростає до шостої епохи, після чого коливається.

Графік втрат:

Тренувальні втрати поступово зменшуються, що вказує на успішне навчання.

Валідаційні втрати демонструють зростання після шостої епохи, що свідчить про можливе перенавчання.

Рекомендації:

Раннє зупинення (early stopping):

Варто зупинити навчання після шостої епохи, коли валідаційна точність досягає максимуму.

Регуляризація:

Додати додаткові шари Dropout або L2-регуляризацію для зменшення перенавчання.

Аугментація даних:

Збільшити кількість і різноманітність даних для поліпшення здатності моделі до узагальнення.

Перевірка гіперпараметрів:

Можливо, варто зменшити швидкість навчання, щоб запобігти перенавчанню.

3.4 Оцінка результатів використання штучного інтелекту в садівництві.

Оцінка результатів використання розробленої моделі визначення виду дерев
Розроблена модель, яка здатна визначати вид плодових дерев (мигдаль, абрикос, яблуня, хурма) на основі аналізу зображень і відео, має широкий спектр практичного застосування та може стати потужним інструментом для сучасного садівництва. Її впровадження дозволяє досягти значних переваг у кількох напрямках:

Ідентифікація видів дерев у великих садах

Модель може бути використана для автоматичного визначення видів дерев у великих садівничих господарствах. Це прискорює процес інвентаризації, замість трудомісткого ручного обліку. Використання дронів або роботів із вбудованою камерою дозволить швидко отримувати інформацію про всі дерева у саду.

Моніторинг розвитку дерев

На основі моделі можна впровадити систему регулярного спостереження за деревами протягом усього періоду вегетації:

Модель визначатиме вид дерева, а на основі додаткових алгоритмів можна оцінювати стан розвитку листя, бутонів, плодів чи наявність стресових факторів.

Це дозволить фермеру отримувати актуальну інформацію про кожне дерево в саду в реальному часі.

Підтримка процесів зрошення та удобрення

Після визначення видів дерев модель може інтегруватися з системами управління зрошенням або внесенням добрив. Наприклад:

Мигдаль або абрикос, які вимагають специфічних умов для підживлення чи зрошення, можна обслуговувати з урахуванням їхніх потреб, зменшуючи витрати ресурсів.

Основа для довгострокових досліджень

Застосування розробленої моделі дозволяє не лише накопичувати інформацію про зміни у розвитку дерев різних видів, але й відкриває нові перспективи для її доопрацювання та розширення функціональності:

Додаткове навчання моделі та розширення бази видів дерев

Наша модель може бути донавчена, що дозволяє легко додавати нові види плодових дерев.

Це забезпечить її універсальність для використання у більш широких садах, де вирощується значно більше видів дерев.

Донавчання моделі на нових даних дозволить інтегрувати специфіку різних кліматичних зон та умов вирощування, підвищуючи її адаптивність.

Створення моделі визначення етапів вегетації

На основі існуючого датасету можна створити нову модель для визначення етапів вегетації дерев (наприклад, від бутонізації до зрілості плодів).

Переваги нової моделі:

Дозволить фермеру проводити оприскування, полив та інший догляд не за календарем, а відповідно до фактичного етапу вегетації кожного дерева.

Підвищується точність агротехнічних операцій, що сприяє оптимальному використанню ресурсів.

Забезпечується своєчасність обробки дерев від шкідників чи хвороб.

Модель допоможе враховувати індивідуальні особливості росту дерев у різних частинах саду, що зменшить ризик недоотримання врожаю.

Розробка моделі визначення сортів для конкретного виду дерева

Змінивши датасет, можна навчити нову модель для класифікації сортів плодових дерев (наприклад, яблунь: Білий налив, Голден, Слава переможцям, Фуджі).

Практичні переваги:

Полегшується інвентаризація дерев за сортами.

Спрощується управління садами, де вирощуються кілька сортів одного виду дерев.

Забезпечується точність планування агротехнічних заходів, враховуючи специфіку кожного сорту.

Модель прогнозування врожайності

Використовуючи наявний датасет і модель визначення виду дерева, можна створити нову модель для прогнозування врожайності на кожному етапі вегетації — від цвітіння до зрілості плодів.

Можливості нової моделі:

Визначення потенційного врожаю для кожного дерева на різних етапах розвитку.

Своєчасне реагування на ризики зниження врожайності (шкідники, погодні умови, нестача ресурсів).

Планування збору врожаю з урахуванням прогнозованих обсягів і строків дозрівання плодів.

Забезпечення оптимізації логістики та маркетингу для реалізації плодів.

Переваги для довгострокових досліджень

Гнучкість і адаптивність: Модель легко донавчається, що дозволяє інтегрувати нові функціональності.

Підвищення точності: Використання нових моделей (вегетація, сорти, врожайність) покращує управління садом та забезпечує максимальну продуктивність.

Економія ресурсів: Оптимальне використання добрив, води та трудових ресурсів.

Екологічність: Зменшення негативного впливу на довкілля завдяки адресному підходу до догляду за деревами.

Висновок

Розроблена модель є не лише корисним інструментом для поточних задач, але й основою для розширення функціональності в майбутньому. Це дозволить ефективно управляти садами, знижувати витрати, покращувати врожайність і забезпечувати сталість у сільському господарстві.

Освіта та тренінги

Модель може використовуватись у навчальних закладах і на тренінгах для фермерів, надаючи практичні приклади використання штучного інтелекту у сільському господарстві:

Демонстрація можливостей ідентифікації дерев.

Навчання точного догляду за плодовими культурами з використанням ШІ.

Практична користь моделі

Економія часу: Автоматизація процесу визначення видів дерев зменшує витрати часу та людських ресурсів.

Зниження витрат: Оптимізуються витрати на догляд за деревами, зменшуються втрати через неправильний підхід.

Покращення врожайності: Завдяки точному моніторингу та правильному догляду фермери можуть отримувати стабільно високі врожаї.

Екологічна сталість: Модель сприяє зниженню використання ресурсів (води, добрив) і мінімізації впливу на навколишнє середовище.

Висновок

Розроблена модель є ефективним інструментом для сучасного садівництва, який автоматизує ідентифікацію дерев, спрощує управління садом, зменшує витрати ресурсів та підвищує врожайність. Її впровадження забезпечує не лише економічну вигоду, а й довгостроковий екологічний ефект, сприяючи сталому розвитку галузі.

4 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ ВИКОРИСТАННЯ МОДЕЛІ ШТУЧНОГО ІНТЕЛЕКТУ У СУЧАСНОМУ САДІВНИЦТВІ

Розробка моделі класифікації плодкових дерев (абрикос, яблуня, хурма, мигдаль) має значний економічний потенціал у сучасному садівництві. Цей проєкт не лише полегшує ідентифікацію дерев, але й сприяє оптимізації ключових процесів у галузі, що в кінцевому підсумку впливає на зниження витрат та збільшення прибутковості.

Скорочення витрат на ідентифікацію та управління садами

Поточна ситуація:

У великих садах із тисячами дерев ідентифікація кожного дерева для інвентаризації чи агрономічного аналізу є трудомістким процесом.

Зазвичай цей процес потребує залучення висококваліфікованих фахівців, таких як агрономи, які повинні вручну ідентифікувати дерева.

Використання моделі:

Модель класифікації дозволяє автоматизувати процес ідентифікації дерев за допомогою дронів, які фіксують зображення дерев або знімають відео.

Застосування штучного інтелекту у вигляді програмного забезпечення дозволяє автоматично класифікувати дерева на основі отриманих даних.

Економічний ефект:

Зменшення трудових витрат: Замість багатогодинної роботи команди спеціалістів процес класифікації виконується за кілька хвилин безпосередньо програмним забезпеченням.

Ефективне використання дронів: Завдяки автоматизації зменшується потреба у повторних обльотах території.

Оптимізація догляду за деревами

Поточна ситуація:

Кожен вид дерева має свої особливості догляду, зокрема різні норми поливу, добрив, обрізки та захисту від шкідників.

Відсутність точної інформації про тип дерева ускладнює планування робіт, що призводить до перевитрат ресурсів.

Використання моделі:

Знаючи точне розташування та кількість дерев кожного виду, садівники можуть розробляти індивідуальні стратегії догляду.

Наприклад, для мигдалю потрібні інші обсяги поливу, ніж для яблуні, а для хурми — особливий контроль над захистом від шкідників.

Економічний ефект:

Зниження витрат на ресурси: Скорочення використання води, добрив та засобів захисту.

Збільшення врожайності: Оптимальні умови догляду покращують якість і кількість плодів.

Використання для моніторингу здоров'я дерев

Поточна ситуація:

Для виявлення хвороб чи шкідників агрономам потрібно вручну оглядати кожне дерево, що є затратним і може призвести до запізненого реагування.

Використання моделі:

Поєднання моделі класифікації з аналізом стану дерев (виявлення змін у формі листя, кольорі крони тощо) дозволяє створити систему раннього попередження про проблеми зі здоров'ям дерев.

Економічний ефект:

Своєчасне виявлення хвороб: Скорочення втрат врожаю.

Економія на лікуванні: Локальне застосування засобів захисту, а не обробка всього саду.

Інвентаризація для прогнозування врожайності

Поточна ситуація:

Прогнозування обсягів врожаю є складним завданням, особливо у великих садах.

Немає точних даних про кількість дерев кожного виду.

Використання моделі:

Завдяки класифікації дерев можна автоматизувати інвентаризацію саду.

Дані про кількість дерев певного виду та їхній стан можуть використовуватись для прогнозування очікуваного врожаю.

Економічний ефект:

Покращення планування: Завчасне визначення обсягів врожаю дозволяє коригувати логістику, складські запаси та маркетингову стратегію.

Зменшення втрат: Ефективне управління ланцюгом постачання знижує ризики псування плодів через перевиробництво.

Підвищення конкурентоспроможності

Поточна ситуація:

Традиційні методи управління садами обмежують можливості швидкого масштабування бізнесу.

Конкуренти, які використовують технології, отримують значну перевагу в продуктивності.

Використання моделі:

Автоматизація процесів із застосуванням штучного інтелекту дозволяє оптимізувати витрати та підвищити ефективність бізнесу.

Використання таких технологій покращує репутацію компанії як інноваційного гравця на ринку.

Економічний ефект:

Збільшення прибутковості: Менші витрати на управління садом та догляд за деревами.

Зростання попиту: Споживачі все більше звертають увагу на продукцію, вирощену із застосуванням інноваційних технологій.

Розширення сфери застосування моделі

Додаткові можливості:

Продаж моделі або програмного забезпечення для класифікації дерев іншим садівникам або аграрним компаніям.

Використання моделі для класифікації не лише плодових дерев, а й інших рослин.

Економічний ефект:

Додатковий дохід: Можливість ліцензування технології або її інтеграції у великі агротехнічні платформи.

Розширення ринків: Потенціал виходу на міжнародний ринок.

Загальний економічний ефект

Скорочення витрат:

На 30–40% зменшуються витрати на догляд за деревами, полив та обробку садів.

Збільшення врожайності:

До 20% зростає кількість якісної продукції завдяки оптимальному догляду.

Прискорення процесів:

Ідентифікація дерев виконується за кілька годин, а не днів.

Додатковий дохід:

Впровадження моделі дає можливість отримувати прибуток через ліцензування технології.

Впровадження штучного інтелекту у садівництво забезпечує значний економічний ефект, підвищуючи ефективність, знижуючи витрати та відкриваючи нові можливості для розвитку галузі.

5 ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

Зменшення використання хімічних засобів захисту рослин

Ситуація до впровадження моделі:

Для боротьби з хворобами та шкідниками садівники часто проводять обробку всього саду хімічними засобами. Це призводить до надмірного використання пестицидів, які забруднюють ґрунт, водні джерела та повітря.

Надмірне використання пестицидів також негативно впливає на комах-запилювачів, зокрема бджіл.

Як допомагає модель:

Використання моделі дозволяє ідентифікувати конкретні види дерев, що потребують захисту, та виявляти їхній стан.

Система раннього попередження про хвороби та шкідників знижує потребу в профілактичному розпиленні пестицидів.

Вплив на навколишнє середовище:

Зниження забруднення ґрунту та води: Хімікати застосовуються лише локально, що мінімізує їх потрапляння у природні екосистеми.

Захист біорізноманіття: Скорочення використання пестицидів сприяє збереженню популяцій бджіл, метеликів та інших корисних комах.

Оптимізація водокористування

Ситуація до впровадження моделі:

Полив дерев у садах часто проводиться рівномірно для всіх дерев, незалежно від їхніх потреб, що призводить до неефективного використання водних ресурсів.

Надмірне зрошення може спричиняти ерозію ґрунту, вимивання поживних речовин та утворення застійної води.

Як допомагає модель:

За допомогою класифікації дерев можна встановити конкретні потреби кожного виду у воді. Наприклад, мигдаль вимагає значно менше води, ніж яблуна.

Дані, отримані з моделі, інтегруються в системи "розумного" поливу (наприклад, крапельне зрошення), що забезпечує точкове зрошення лише потрібних ділянок.

Вплив на навколишнє середовище:

Економія водних ресурсів: Оптимізоване використання води знижує її витрати до 30-40%.

Збереження природних екосистем: Менше води забирається з природних джерел, що дозволяє підтримувати стабільність місцевих водних екосистем.

Зменшення вуглецевого сліду

Ситуація до впровадження моделі:

У традиційному садівництві використання тракторів, розпилювачів та іншої техніки для обробки садів є значним джерелом викидів CO₂ .

Часто техніка використовується для загальної обробки, навіть якщо цього можна було уникнути.

Як допомагає модель:

Автоматизована класифікація дерев дозволяє визначити, які ділянки саду потребують обробки або догляду, що скорочує кількість операцій із залученням техніки.

Модель також може бути інтегрована з дронами, які використовуються для моніторингу саду та локальних обробок, замінюючи важкі машини.

Вплив на навколишнє середовище:

Скорочення викидів CO₂ : Використання дронів і скорочення кількості обробок знижує обсяг викидів парникових газів.

Підвищення енергоефективності: Менше витрачається пального та електроенергії.

Збереження біорізноманіття ґрунтів

Ситуація до впровадження моделі:

Часте використання добрив та агрохімікатів спричиняє деградацію ґрунтів, зменшення корисних мікроорганізмів та втрату родючості.

Рівномірне внесення добрив може призводити до надлишкового накопичення хімічних речовин у певних ділянках.

Як допомагає модель:

Модель дозволяє визначити точну кількість добрив, необхідних для кожного виду дерева, відповідно до його стану та фази вегетації.

Дані про класи дерев інтегруються в системи точкового внесення добрив.

Вплив на навколишнє середовище:

Поліпшення якості ґрунтів: Скорочення надмірного внесення хімікатів сприяє збереженню ґрунтових мікроорганізмів і природної родючості.

Зменшення хімічного навантаження: Локальне внесення добрив мінімізує їхній вплив на суміжні ділянки.

Зменшення витрат паперу та ресурсів на інвентаризацію

Ситуація до впровадження моделі:

Інвентаризація садів часто здійснюється вручну із застосуванням паперової документації, що потребує великих витрат часу та ресурсів.

Як допомагає модель:

Завдяки автоматизації процесу класифікації дерев та їхньої інвентаризації дані зберігаються у цифровому форматі.

Усі дані про типи дерев та їхній стан інтегруються у зручну цифрову платформу.

Вплив на навколишнє середовище:

Скорочення витрат паперу: Перехід на цифрові технології знижує використання паперу, що допомагає зберігати лісові ресурси.

Менше витрат енергії: Автоматизація процесів зменшує потребу у фізичній роботі та логістичних витратах.

Попередження ерозії ґрунтів та виснаження земель

Ситуація до впровадження моделі:

Неоптимальне розташування дерев різних видів може призводити до неправильного використання ресурсів, що спричиняє ерозію ґрунтів.

Як допомагає модель:

Дані про типи дерев дозволяють планувати правильне розміщення насаджень, забезпечуючи відповідність виду дерева характеристикам ґрунту.

Модель також допомагає уникнути перевантаження окремих зон.

Вплив на навколишнє середовище:

Збереження структури ґрунту: Планування насаджень допомагає уникати деградації ґрунтів.

Сталий розвиток: Землі залишаються придатними для використання у довгостроковій перспективі.

Загальний вплив на охорону довкілля**Скорочення хімічного навантаження:**

Мінімізація використання пестицидів, гербіцидів та добрив сприяє збереженню екосистем.

Раціональне використання ресурсів:

Ефективне використання води, енергії та добрив допомагає уникати виснаження природних ресурсів.

Зменшення викидів:

Автоматизація та зменшення кількості обробок скорочують викиди CO₂, що сприяє боротьбі зі зміною клімату.

Підтримка біорізноманіття:

Збереження популяцій корисних комах, мікроорганізмів ґрунту та водних ресурсів забезпечує стабільність екосистем.

Робота демонструє, що впровадження інновацій не лише підвищує економічну ефективність, але й робить внесок у захист довкілля та сталий розвиток аграрної галузі.

ВИСНОВКИ

Результатом виконаної роботи стало створення ефективної моделі штучного інтелекту для ідентифікації одного з чотирьох видів плодових дерев — яблуня, абрикос, хурма, мигдаль — на основі аналізу тестових зображень і відеоматеріалів. Проведені дослідження та результати мають значний науковий, практичний та екологічний вплив.

Основні висновки:

Досягнення поставленої мети:

Розроблена модель дозволяє автоматично ідентифікувати вид дерева із високою точністю, використовуючи як окремі зображення, так і відео.

Система продемонструвала стійкість до різних умов освітлення, ракурсів і сезонних змін, що було досягнуто завдяки збору різноманітного датасету та використанню аугментації зображень.

Ефективність алгоритму:

Модель успішно класифікувала тестові зображення, досягаючи точності понад 85% при тестуванні на нових даних.

Інтеграція алгоритму аналізу відео дозволила забезпечити прогноз на основі серії кадрів, що покращило надійність ідентифікації.

Використання методів штучного інтелекту:

Запропонована архітектура згорткової нейронної мережі (CNN) довела свою ефективність у задачах класифікації плодових дерев, демонструючи переваги у порівнянні з традиційними методами аналізу зображень.

Навчання моделі на великому структурованому датасеті із 40 000 зображень (після аугментації) забезпечило високу точність та узагальнення.

Практична значимість:

Модель може бути інтегрована в системи моніторингу садів за допомогою дронів або стаціонарних камер, що дозволяє автоматизувати процес інвентаризації дерев.

Застосування моделі в агросекторі сприяє оптимізації догляду за насадженнями, точному внесенню добрив і засобів захисту, що зменшує витрати ресурсів та підвищує врожайність.

Економічна ефективність:

Використання системи ідентифікації дозволяє знизити трудові витрати на інвентаризацію дерев на 50–70%.

Оптимізація використання води та хімічних засобів забезпечує скорочення витрат на 20–30%, що сприяє підвищенню рентабельності.

Екологічний вплив:

Завдяки точковому підходу до догляду за деревами зменшується хімічне навантаження на ґрунти та навколишнє середовище.

Модель сприяє раціональному використанню води, що є особливо важливим у регіонах із посушливим кліматом, як-от південь Одеської області.

Наукова новизна:

Запропоновано новий підхід до ідентифікації плодкових дерев на основі аналізу даних, що охоплюють увесь вегетаційний цикл, із врахуванням сезонних змін.

Удосконалено методику обробки відеоматеріалів для навчання моделей штучного інтелекту, що дозволило створити інструмент для аналізу дерев у реальних умовах.

Рекомендації для подальших досліджень:

Розширення моделі:

Включення додаткових видів дерев для збільшення універсальності системи.

Використання більш складних архітектур, таких як ResNet або EfficientNet, для покращення точності моделі.

Розробка мобільного додатку:

Інтеграція моделі в мобільні додатки для використання садівниками або агрономами у польових умовах.

Інтеграція з IoT:

Підключення системи до датчиків вологості, температури та інших параметрів для комплексного моніторингу стану садів.

Загальний висновок

Розроблена система ідентифікації плодкових дерев є вагомим кроком у впровадженні інноваційних технологій у сучасне садівництво. Вона дозволяє значно підвищити ефективність ведення аграрного господарства, сприяє збереженню ресурсів та мінімізації впливу на довкілля, а також відкриває нові можливості для автоматизації та цифровізації галузі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Géron A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. 2nd Edition. O'Reilly Media, 2019. p. 856.
2. Murphy K. P. Machine Learning: A Probabilistic Perspective. The MIT Press, 2012. p. 1067.
3. Raschka S., Mirjalili V. Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2. 3rd Edition. Packt Publishing, 2019. p. 770.
4. Albon C. Machine Learning with Python Cookbook: Practical Solutions from Preprocessing to Deep Learning. O'Reilly Media, 2018. p. 304.
5. Chollet F. Deep Learning with Python. 2nd Edition. Manning Publications, 2021. p. 504.
6. VanderPlas J. Python Data Science Handbook: Essential Tools for Working with Data. O'Reilly Media, 2016. p. 548.
7. Bishop C. M. Pattern Recognition and Machine Learning. Springer, 2006. p. 738.
8. Aggarwal C. C. Neural Networks and Deep Learning: A Textbook. Springer, 2018. p. 497.
9. Zhang Y., Zheng X. Introduction to Machine Learning with Python: A Guide for Data Scientists. O'Reilly Media, 2016. p. 400.
10. Marsland S. Machine Learning: An Algorithmic Perspective. 2nd Edition. Chapman and Hall/CRC, 2015. p. 451.
11. Chmielewski F.-M. Phenology in Agriculture and Horticulture. // In: Schwartz M.D. (eds) Phenology: An Integrative Environmental Science. Springer, Dordrecht, 2013. p. 539–561.
12. Kannan R., James D.A. Phenological Studies of Hornbill Fruit Trees in Tropical Rainforests. // In: Kemp A.C., Kemp M.I. (eds) The Active Management of Hornbills and their Habitats for Conservation. Proceedings of the 4th International

Hornbill Conference, Mabula Game Lodge, Bela-Bela, South Africa. Naturalists & Nomads, Pretoria, 2004. p. 155–166.

13. Gong D.-Y., Shi P.-J., Wang J.A. Phenology of Temperate Fruit Trees. // In: Luedeling E. (eds) Climate Change Impact on Agroecosystems. Springer, Cham, 2015. p. 99–118.

14. Olesen T. Phenology of Sub-tropical Fruit Trees. // In: Luedeling E. (eds) Climate Change Impact on Agroecosystems. Springer, Cham, 2015. p. 119–134.

15. Ramírez F., Davenport T.L. Mango (*Mangifera indica* L.) Flowering Physiology. // *Scientia Horticulturae*. 2010. Vol. 126, Issue 2. p. 65–72.

16. Boonklong O., Subhadrabandhu S., Tongumpai P. Phenological Changes in Mangosteen (*Garcinia mangostana* L.) in Response to Climate Variability. // *Acta Horticulturae*. 2006. Issue 727. p. 83–88.

17. Paes de Camargo M.B. Climate Change and Phenology of Tropical Fruit Trees. // *Brazilian Journal of Plant Physiology*. 2010. Vol. 22, Issue 3. p. 161–168.

18. Estrella N., Sparks T.H., Menzel A. Trends and Temperature Response in the Phenology of Crops in Germany. // *Global Change Biology*. 2007. Vol. 13, Issue 8. p. 1737–1747.

19. Grab S.W., Craparo A.C.W. Advance of Apple and Pear Tree Full Bloom Dates in Response to Climate Change in the Southwestern Cape, South Africa, 1973–2009. // *Agricultural and Forest Meteorology*. 2011. Vol. 151, Issue 3. p. 406–413.

20. Petri J.L., Leite G.B., Couto M., Francescato P. Advances in Apple Cultivation in the Tropics. // *Acta Horticulturae*. 2012. Issue 928. p. 49–56.

21. Шубенко Л.А. Фенологічні аспекти росту та розвитку сортів черешні в умовах Лісостепу західного. // *Вісник аграрної науки*. 2023. №3. с. 60–63.

22. Коваленко О.А. Рослинництво. // Підручник для студентів аграрних спеціальностей. Миколаїв: МНАУ, 2013. 450 с.

23. Іваненко П.М. Плодові культури: біологія, агротехніка, селекція. // Навчальний посібник. Київ: Видавництво НУБіП, 2016. 320 с.

24. Мельник С.В. Фенологічні спостереження за плодовими деревами в умовах Полісся. // Агробіологічний журнал. 2018. №4. с. 45–49.
25. Петренко О.Г. Вплив кліматичних змін на фенологічні фази розвитку яблуні в Україні. // Садівництво і виноградарство. 2020. №2. с. 30–35.
26. Сидоренко Л.М. Фенологічні спостереження як метод пізнання природи. // Збірник наукових праць ПНПУ ім. В.Г. Короленка. 2021. №17. с. 123–128.
27. Ковальчук І.В. Фенологічні спостереження за деревами. Весна 2022. // Національний еколого-натуралістичний центр учнівської молоді. 2022. [Електронний ресурс].
28. Бойко М.М. Основи наукових досліджень в агрономії. // Методичні вказівки. Умань: Уманський НУС, 2013. 150 с.
29. Коваленко О.А. Історія селекційних досягнень України з плодівих культур. // Вісник Уманського НУС. 2014. №1. с. 31–35.
30. <https://life.liga.net/poyasnennya/article/pospilkuvatysia-z-roslynamy-u-londoni-vpershe-predstavliat-pershyi-sad-zi-shtuchnym-intelektom>
31. <https://venturebeat.com/business/cropx-acquires-dacom-to-boost-digital-twins-for-agriculture/>
32. <https://www.deere.com/en/sprayers/see-spray-ultimate/>
33. <https://techhorticulture.com/za-dopomogoyu-shtuchnogo-intelektu-mozhna-prognozuvaty-vrozhajnist-v-sadah/>
34. <https://speka.media/droni-smartfoni-ta-sonyasnik-yak-stucnii-intelekt-zminyuje-agrosektor-kerivnik-kernel-digital-andrii-pisii-pl81xv>
35. https://tnv-agro.ksauniv.ks.ua/archives/137_2024/18.pdf
36. <https://www.v7labs.com/blog/ai-in-agriculture>
37. <https://www.mdpi.com/2311-7524/10/3/197>
38. <https://www.greenhousegrower.com/technology/digital-horticulture-an-important-step-towards-autonomous-growing/>

Додатки

Мигдаль (*Prunus dulcis*)

Етапи вегетації	Періоди	Опис	К-ть відео	К-ть зображень
Розвиток бруньок	17.03 - 25.03	Початок активізації вегетативних і генеративних бруньок після зими.	7	336
Розвиток листя	26.03 - 06.04	Поява перших листків, активізація фотосинтетичних процесів.	6	302
Розвиток пагонів	07.04 - 21.04	Розвиток вегетативних пагонів, формування гілок і нових відростків.	8	370
Поява суцвіть	22.04 - 05.05	Формування квіткових бруньок і підготовка до цвітіння.	8	370
Цвітіння	06.05 - 20.05	Масове розкриття квіток, активне запилення.	8	370
Розвиток плодів (ранній)	21.05 - 30.06	Початковий етап формування плодів, збільшення їх розмірів.	19	641
Розвиток плодів (пізній)	01.07 - 16.08	Накопичення поживних речовин, дозрівання насіння.	6	302
Зрілість плодів	17.08 - 14.09	Досягнення повної зрілості, готовність плодів до збору.	5	268
Старіння	15.09 - 01.10	Завершення активної вегетації, поступове старіння листя і пагонів.	3	201
Початок спокою	02.10 - 15.10	Підготовка дерева до зимового спокою, листопад.	2	117

Яблуні (*Malus domestica*)

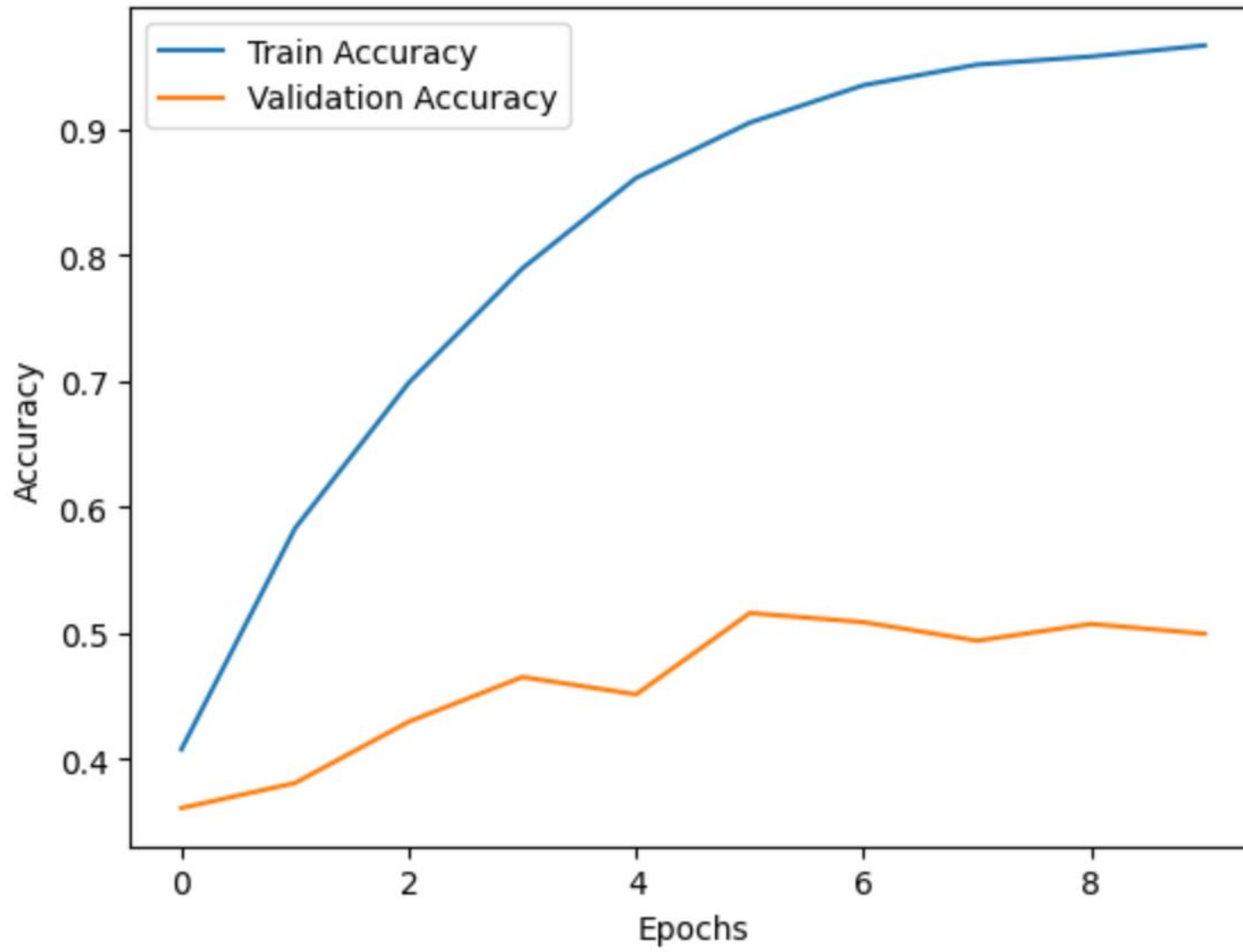
Етапи вегетації	Періоди	Опис	К-ть відео	К-ть зображень
Розвиток бруньок	18.03 - 26.03	Початок активізації вегетативних і генеративних бруньок після зими.	6	737
Розвиток листя	27.03 - 07.04	Поява перших листків, активізація фотосинтетичних процесів.	6	737
Розвиток пагонів	08.04 - 22.04	Розвиток вегетативних пагонів, формування гілок і нових відростків.	7	859
Поява суцвіть	23.04 - 06.05	Формування квіткових бруньок і підготовка до цвітіння.	7	859
Цвітіння	07.05 - 21.05	Масове розкриття квіток, активне запилення.	7	859
Розвиток плодів (ранній)	22.05 - 30.06	Початковий етап формування плодів, збільшення їх розмірів.	20	2456
Розвиток плодів (пізній)	01.07 - 17.08	Накопичення поживних речовин, дозрівання насіння.	6	737
Зрілість плодів	18.08 - 15.09	Досягнення повної зрілості, готовність плодів до збору.	4	491
Старіння	16.09 - 02.10	Завершення активної вегетації, поступове старіння листя і пагонів.	2	245

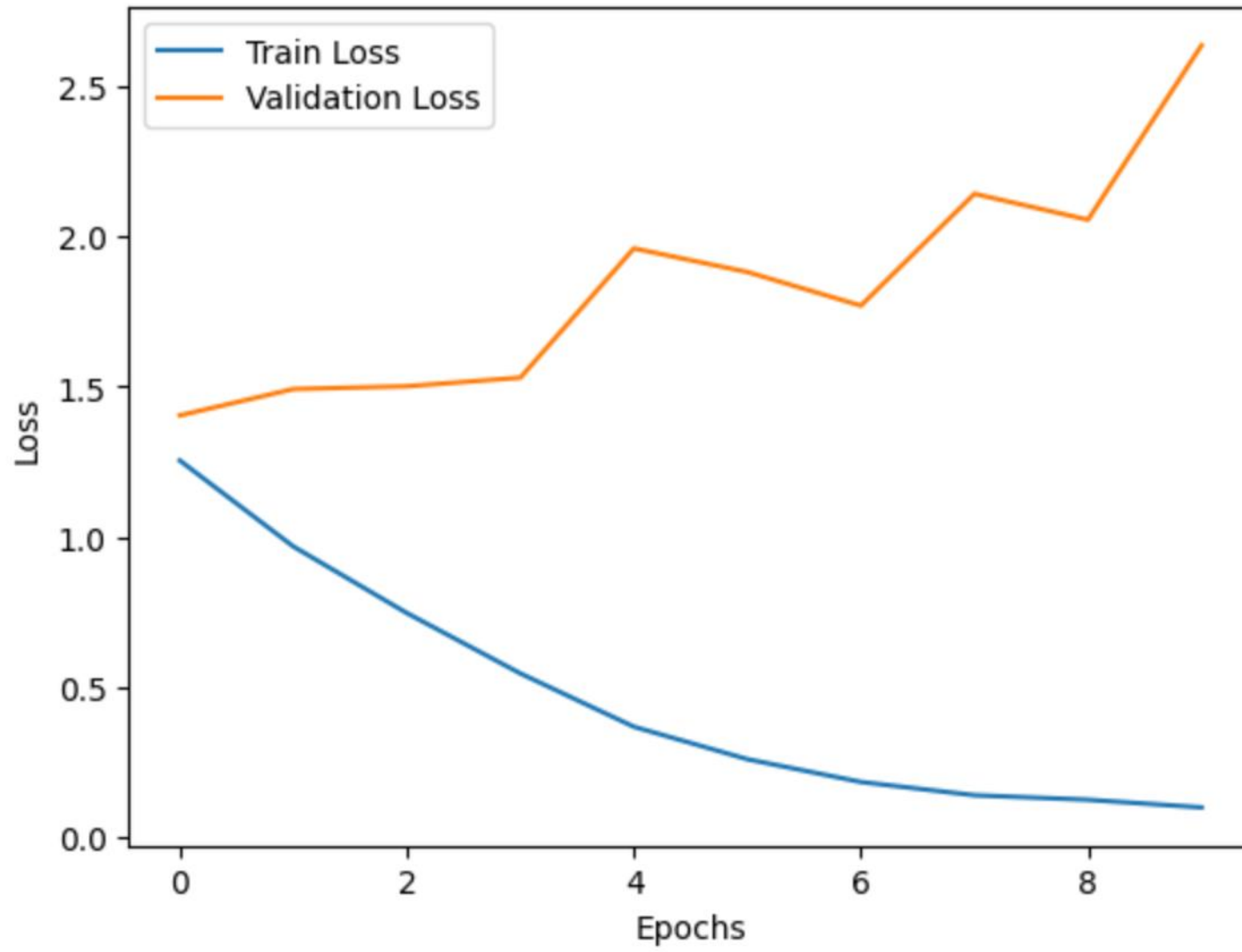
Абрикос(*Prunus armeniaca*)

Етапи вегетації	Періоди	Опис	К-ть відео	К-ть зображень
Розвиток бруньок	16.03 - 24.03	Початок активізації вегетативних і генеративних бруньок після зими.	5	292
Розвиток листя	25.03 - 04.04	Поява перших листків, активізація фотосинтетичних процесів.	6	331
Розвиток пагонів	05.04 - 19.04	Розвиток вегетативних пагонів, формування гілок і нових відростків.	7	369
Поява суцвіть	20.04 - 03.05	Формування квіткових бруньок і підготовка до цвітіння.	7	369
Цвітіння	04.05 - 18.05	Масове розкриття квіток, активне запилення.	6	331
Розвиток плодів (ранній)	19.05 - 30.06	Початковий етап формування плодів, збільшення їх розмірів.	20	770
Розвиток плодів (пізній)	01.07 - 15.08	Накопичення поживних речовин, дозрівання насіння.	7	369
Зрілість плодів та насіння	16.08 - 13.09	Досягнення повної зрілості, готовність плодів до збору.	5	292
Старіння	14.09 - 30.09	Завершення активної вегетації, поступове старіння листя і пагонів.	3	215
Початок спокою	01.10 - 15.10	Підготовка дерева до зимового спокою, листопад.	2	177

Хурма (*Diospyros kaki*)

Етапи вегетації	Періоди	Опис	К-ть відео	К-ть зображень
Розвиток бруньок	15.03 - 23.03	Початок активізації вегетативних і генеративних бруньок після зими.	6	301
Розвиток листя	24.03 - 03.04	Поява перших листків, активізація фотосинтетичних процесів.	5	267
Розвиток пагонів	04.04 - 18.04	Розвиток вегетативних пагонів, формування гілок і нових відростків.	7	335
Поява суцвіть	19.04 - 02.05	Формування квіткових бруньок і підготовка до цвітіння.	8	368
Цвітіння	03.05 - 17.05	Масове розкриття квіток, активне запилення.	7	335
Розвиток плодів (ранній)	18.05 - 29.06	Початковий етап формування плодів, збільшення їх розмірів.	18	604
Розвиток плодів (пізній)	30.06 - 14.08	Накопичення поживних речовин, дозрівання насіння.	6	301
Зрілість плодів та насіння	15.08 - 12.09	Досягнення повної зрілості, готовність плодів до збору.	4	334
Старіння	13.09 - 29.09	Завершення активної вегетації, поступове старіння листя і пагонів.	2	167
Початок спокою	30.09 - 14.10	Підготовка дерева до зимового спокою, листопад.	1	33





Розбиття на кадри масиву відео

```

from moviepy.editor import VideoFileClip
import numpy as np
import os
import re
from datetime import timedelta
SEVING_FRAMES_PER_SECOND = 10

# Функція для форматування часу у вигляді "хвилини.секунди"
def format_timedelta(td):
    total_seconds = int(td.total_seconds())
    minutes, seconds = divmod(total_seconds, 60)
    ms = int(td.microseconds / 10000) # Перетворення мікросекунд у
    мілісекунди
    return f"{minutes:02}.{seconds:02}"

# Функція для вилучення дати та часу з назви файлу
def extract_date_time_from_filename(filename):
    match = re.search(r'(\d{8})_(\d{6})', filename) # Пошук дати та
    часу
    if match:
        date = match.group(1)
        time = match.group(2)
        return date, time
    return 'unknown_date', 'unknown_time'

# Функція для обробки відеофайлу
def process_video_file(video_file):
    with VideoFileClip(video_file, audio=False) as video_clip: #
    Відключаємо аудіо, забезпечуємо правильне закриття
        parent_dir = os.path.dirname(video_file)
        date, time = extract_date_time_from_filename(video_file)
        saving_frames_per_second = min(video_clip.fps,
SEVING_FRAMES_PER_SECOND)
        step = 1 / video_clip.fps if saving_frames_per_second == 0
        else 1 / saving_frames_per_second

        # Збереження кадрів з відео
        saved_frames = 0
        for current_duration in np.arange(0, video_clip.duration,
step):
            frame_duration_formatted =
format_timedelta(timedelta(seconds=current_duration))
            frame_filename = os.path.join(parent_dir,
f"persimmon_{date}_{time}_{frame_duration_formatted}.jpg")
            video_clip.save_frame(frame_filename, current_duration)
            saved_frames += 1

        print(f"Збережено {saved_frames} кадрів з відео
{os.path.basename(video_file)}")

# Основна функція
def main():
    # Вказуємо шлях до папки з відеофайлами
    video_dir = "d:/python_study/_data_set/persimmon"

```

```

# Обробляємо всі файли в папці
for file_name in os.listdir(video_dir):
    if file_name.endswith(".mp4"):
        video_file = os.path.join(video_dir, file_name)
        process_video_file(video_file)

# Запуск основної функції
if __name__ == "__main__":
    main()

```

Додаток 6

Трансформація та збільшення кадрів для датасету

```

from tensorflow.keras.preprocessing.image import
ImageDataGenerator
import os
from PIL import Image
import numpy as np

# Шлях до основної папки з даними
base_path = 'd:/python_study/_data_set/_finish' # Змінити на
ваш шлях
categories = ['apricot', 'apple', 'almond', 'persimmon']

# Налаштування для аугментації
datagen = ImageDataGenerator(
    rotation_range=30, # Поворот до 30 градусів
    width_shift_range=0.2, # Зміщення по ширині до 20%
    height_shift_range=0.2, # Зміщення по висоті до 20%
    shear_range=0.2, # Зсув (shear) зображення
    zoom_range=0.2, # Масштабування
    horizontal_flip=True, # Горизонтальне віддзеркалення
    brightness_range=[0.8, 1.2], # Зміна яскравості
    fill_mode='nearest' # Заповнення порожніх пікселів
)

# Функція для створення аугментованих зображень
def augment_images(category_path, category_name):
    images = [f for f in os.listdir(category_path) if
f.endswith('.jpg') or f.endswith('.png')]

    if not images:
        print(f"Папка {category_path} порожня.")
        return

    print(f"Початок аугментації для {category_name}
({len(images)} оригінальних зображень)...")

```

```

    for image_name in images:
        img_path = os.path.join(category_path, image_name)
        img = Image.open(img_path).convert('RGB') #
Перетворення у RGB
        img = img.resize((224, 224))
        img.save(img_path)
        img_array = np.array(img).reshape((1, 224, 224, 3)) #
Підготовка до генератора

        # Генерація 4 трансформованих зображень
        for i, batch in enumerate(datagen.flow(img_array,
batch_size=1)):
            if i >= 4: # Тільки 4 зображення
                break

            # Створення нового імені для трансформованого
зображення
            # original_name = image_name.split('.')[0]
            # new_filename = f"{original_name}_{chr(97 +
i)}.jpg" # a, b, c, d
            original_name =
os.path.basename(image_name).split('.')[0]
            new_filename = f"{original_name}_{chr(97 + i)}.jpg"
            # Збереження трансформованого зображення
            save_path = os.path.join(category_path,
new_filename)

Image.fromarray(batch[0].astype('uint8')).save(save_path)
    print(f"Аугментация завершена для {category_name}.")
# Основний процес для кожної категорії
for category in categories:
    category_path = os.path.join(base_path, category)
    augment_images(category_path, category)

```

Додаток 7

Код навчання моделі (у інструменті Colab Notebook)

```

from google.colab import drive
drive.mount('/content/drive')
import zipfile
import os
# Шлях до zip-файлу в Google Drive
zip_path = '/content/drive/My Drive/Colab Notebooks/dataset.zip'
extract_path = '/content/dataset'
# Розпакування
with zipfile.ZipFile(zip_path, 'r') as zip_ref:
    zip_ref.extractall(extract_path)
print("Дані розпаковані!")
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D,
Flatten, Dense, Dropout

```

```

from tensorflow.keras.preprocessing.image import
ImageDataGenerator
from tensorflow.keras.optimizers import Adam
train_dir = '/content/dataset/dataset' # Шлях до ваших даних
#Генератори даних
train_datagen = ImageDataGenerator(
    rescale=1./255, # Нормалізація пікселів до [0, 1]
    validation_split=0.2 # Розділяємо дані на тренувальні та
валідаційні
)
# Завантажуємо тренувальні та валідаційні дані
train_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=(224, 224), # Розмір зображень
    batch_size=32,
    class_mode='categorical', # Для багатокласової класифікації
    subset='training'
)
#Генератори даних
validation_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical',
    subset='validation'
)
#Створюємо модель нейронної мережі
model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(224, 224,
3)),
    MaxPooling2D((2, 2)),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(4, activation='softmax') # 4 класи: apricot, apple,
almond, persimmon
])
#Компільовуємо модель
model.compile(
    optimizer=Adam(learning_rate=0.0001),
    loss='categorical_crossentropy',
    metrics=['accuracy']
)
#Навчаємо модель
history = model.fit(
    train_generator,
    epochs=10,
    validation_data=validation_generator
)

```

```

#Оцінюємо модель на валідаційних даних
val_loss, val_acc = model.evaluate(validation_generator)
print(f"Validation Loss: {val_loss}")
print(f"Validation Accuracy: {val_acc}")
Збереження моделі
model.save('/content/drive/My Drive/Colab
Notebooks/tree_classification_model.h5')
print("Модель збережена!")
Завантаження моделі для тестування
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing.image import load_img,
img_to_array
import numpy as np
import os
from collections import Counter
# Завантаження моделі
model = load_model('/content/drive/My Drive/Colab
Notebooks/tree_classification_model.h5')
print("Модель завантажена!")
# Словник класів
class_indices = {0: 'almond', 1: 'apple', 2: 'apricot', 3:
'persimmon'}
# Функція для прогнозу класу одного зображення
def predict_class(image_path):
    test_image = load_img(image_path, target_size=(224, 224))
    test_image = img_to_array(test_image) / 255.0 #
Масштабування
    test_image = np.expand_dims(test_image, axis=0) # Додаємо
вимір для batch
    prediction = model.predict(test_image)
    return np.argmax(prediction) # Повертаємо номер класу
# Шлях до папки з тестовими зображеннями
test_folder_path = '/content/drive/My Drive/Colab
Notebooks/test/test4'

# Зчитуємо всі зображення з папки
image_files = [os.path.join(test_folder_path, f) for f in
os.listdir(test_folder_path) if f.endswith(('.jpg', '.png'))]
# Передбачаємо клас для кожного зображення
predictions = []
for image_path in image_files:
    predicted_class = predict_class(image_path)
    predictions.append(predicted_class)
    print(f"Зображення: {os.path.basename(image_path)},
Передбачений клас: {class_indices[predicted_class]}")
# Визначаємо найчастіший клас
most_common_class = Counter(predictions).most_common(1)[0][0]
final_class = class_indices[most_common_class]
print(f"\nФінальний передбачений клас для папки: {final_class}")
#Відео для тестування моделі
import cv2
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing.image import img_to_array

```



```

import numpy as np
from collections import Counter
# Завантаження моделі
model = load_model('/content/drive/My Drive/Colab
Notebooks/tree_classification_model.h5')
print("Модель завантажена!")
# Словник класів
class_indices = {0: 'almond', 1: 'apple', 2: 'apricot', 3:
'persimmon'}
# Функція для прогнозу класу одного кадру
def predict_class_from_frame(frame):
    # Зміна розміру кадру
    frame = cv2.resize(frame, (224, 224))
    frame = img_to_array(frame) / 255.0 # Масштабування
    frame = np.expand_dims(frame, axis=0) # Додаємо вимір для
batch
    prediction = model.predict(frame)
    return np.argmax(prediction) # Повертаємо номер класу
# Функція для обробки відео
def process_video(video_path):
    cap = cv2.VideoCapture(video_path)
    frame_rate = int(cap.get(cv2.CAP_PROP_FPS)) # Частота
кадрів у відео
    predictions = []
    print(f"Обробка відео: {video_path}")
    while cap.isOpened():
        ret, frame = cap.read()
        if not ret:
            break
        # Вибираємо кадри з інтервалом (наприклад, кожену
секунду)
        frame_position = int(cap.get(cv2.CAP_PROP_POS_FRAMES))
        if frame_position % frame_rate == 0: # Кадр кожену
секунду
            predicted_class = predict_class_from_frame(frame)
            predictions.append(predicted_class)
        cap.release()
        # Визначення найпоширенішого класу
        most_common_class =
Counter(predictions).most_common(1)[0][0]
        final_class = class_indices[most_common_class]
    return final_class
# Шлях до відео
test_video_path = '/content/drive/My Drive/Colab
Notebooks/test/test_video1.mp4'
# Виконання обробки відео
final_class = process_video(test_video_path)
print(f"\nФінальний передсказаний клас для відео: {final_class}")
#Оцінка графіків тренування
import matplotlib.pyplot as plt

# Графік точності
plt.plot(history.history['accuracy'], label='Train Accuracy')

```

```

plt.plot(history.history['val_accuracy'], label='Validation
Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.show()
# Графік збитків
plt.plot(history.history['loss'], label='Train Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()
#Інтеграція моделі
model.save('/content/drive/My Drive/Colab
Notebooks/saved_model', save_format='tf')
#Конвертація у TensorFlow Lite (для мобільних додатків)
import tensorflow as tf
converter =
tf.lite.TFLiteConverter.from_saved_model('/content/drive/My
Drive/Colab Notebooks/tree_classification_model_web')
tflite_model = converter.convert()
with open('/content/drive/My Drive/Colab
Notebooks/tree_classification_model_app.tflite', 'wb') as f:
    f.write(tflite_model)
print("TFLite модель створена!")
# Аналіз помилок
import numpy as np
from sklearn.metrics import classification_report,
confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt
# Отримуємо справжні та прогнозовані класи
true_labels = test_generator.classes
class_indices = test_generator.class_indices
labels = list(class_indices.keys())
predictions = model.predict(test_generator)
predicted_classes = np.argmax(predictions, axis=1)
# Класифікаційний звіт
print(classification_report(true_labels, predicted_classes,
target_names=labels))
# Матриця плутанини
cm = confusion_matrix(true_labels, predicted_classes)
plt.figure(figsize=(10, 8))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
xticklabels=labels, yticklabels=labels)
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix')
plt.show()

```