

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій
(повне найменування факультету)

Кафедра «Системний аналіз та обчислювальна математика»
(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавра
(ступінь вищої освіти)

на тему «Дослідження розвитку рослини методами машинного навчання»
(назва теми)

Виконав: студент 4 курсу, групи КНТ-811сп

Спеціальності 124 Системний аналіз
(код і найменування спеціальності)

Освітня програма (спеціалізація)
Інтелектуальні технології та прийняття рішень
в складних системах

ВОЙТЕНКО О. Д.
(ПРИЗВИЩЕ та ініціали)

Керівник ШИРОКОРАД Д. В.
(ПРИЗВИЩЕ та ініціали)

Рецензент ДУМІН О. М.
(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

Кафедра «Системний аналіз та обчислювальна математика»

Ступінь вищої освіти бакалавр

Спеціальність 124 Системний аналіз

(код і найменування)

Освітня програма (спеціалізація) Інтелектуальні технології та прийняття рішень в складних системах

(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри «Системний аналіз

та обчислювальна математика»

Еліна ТЕРЕЩЕНКО

« 18 » червня 2024 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ВОЙТЕНКО Олександр Дмитрович

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження розвитку рослини методами машинного навчання

керівник проєкту (роботи) к.ф.-м.н., доц. ШИРОКОРАД Дмитро Вікторович,

(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від « 12 » квітня 2024 року № 139

2. Строк подання студентом проєкту (роботи) 18 червня 2024 р.

3. Вихідні дані до проєкту (роботи) фотографії рослин на різних стадіях розвитку

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) дослідити можливість використання моделей машинного навчання для визначення площі зеленої частини рослини; використовуючи надані фотографії донавити мережу для кращої сегментації.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)

слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1	ШИРОКОРАД Д.В., доцент каф. САОМ	27.04.24	25.05.24
2	ШИРОКОРАД Д.В., доцент каф. САОМ	27.04.24	01.06.24
3	ШИРОКОРАД Д.В., доцент каф. САОМ	27.04.24	10.06.24
Нормоконтроль	ШИРОКОРАД Д.В., доцент каф. САОМ	17.06.24	18.06.24

7. Дата видачі завдання « 15 » 04 2024 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Формулювання мети та основних завдань дипломної роботи	20.03.24	
2	Аналіз літератури та джерел	06.05.24	
3	Вибір моделі YOLOv8 для визначення зеленої частини рослини	13.05.24	
4	Fine-tuning моделі на власному наборі даних	27.05.24	
5	Конвертування вихідних даних нейронної мережі	03.06.24	
6	Попередній захист дипломної роботи	10.06.24	
7	Захист дипломної роботи	18.06.24	

Студент(ка)

(підпис)

Олександр ВОЙТЕНКО

(Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

(підпис)

Дмитро ШИРОКОРАД

(Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра:
64 с., 1 табл., 22 рис., 28 джерел.

Об'єкт дослідження – використання алгоритмів машинного навчання в агропромисловості.

Предмет дослідження – розвиток рослин методами машинного навчання.

Мета роботи – дослідити можливість використання моделей машинного навчання для визначення площі зеленої частини рослини.

Актуальність роботи – автоматизація спостережень за розвитком рослин завжди була затребованою в сільському господарстві через великі засівні площі та необхідність швидкого реагування на відхилення від нормального розвитку. Сьогодні проблема стає навіть більш актуальною через складність або неможливість потрапити на певні ділянки.

Досліджено сучасні методи використання штучного інтелекту та комп'ютерного зору в сільському господарстві. Було перевірено, налаштовано та донавчено модель YOLOv8 на спеціально підготовленому наборі даних.

РОСЛИНА, МАШИННЕ НАВЧАННЯ, СЕГМЕНТАЦІЯ, YOLO,
КОМП'ЮТЕРНИЙ ЗІР

ЗМІСТ

ЗАВДАННЯ.....	2
РЕФЕРАТ.....	4
ВСТУП.....	6
1 ТЕОРЕТИЧНІ ВІДОМОСТІ ТА ПОСТАНОВКА ЗАВДАННЯ	8
1.1 Необхідність оцінки стану рослин в сучасному світі.....	8
1.2 Роль Big Data та штучного інтелекту в дослідженні розвитку рослин	13
1.3 Нейронні мережі, їх застосування, принципи роботи та вплив на сучасну агрономію	18
1.4 Нейронна мережа YOLOv8, її застосування та особливості.....	22
1.5 Огляд аналогів	26
1.6 Висновки з розділу	30
2 ДОСЛІДЖЕННЯ КОМП'ЮТЕРНО-ІНТЕГРОВАНИХ ТЕХНОЛОГІЙ ДЛЯ РЕАЛІЗАЦІЇ ЗАВДАННЯ	32
2.1 Обґрунтування вибору мови програмування.....	32
2.2 Обґрунтування вибору середовища розробки	36
2.3 Огляд використаних бібліотек.....	40
2.4 Огляд використаних наборів даних.....	43
2.5 Висновки з розділу	45
3 ВИКОРИСТАННЯ НЕЙРОМЕРЕЖІ ДЛЯ РОБОТИ З ЗЕЛЕНОЮ ЧАСТИНОЮ РОСЛИНИ.....	47
3.1 Вибір моделі YOLOv8 для визначення зеленої частини рослини	47
3.2 Fine-tuning моделі на власному наборі даних	52
3.3 Конвертування вихідних даних нейронної мережі	57
3.4 Висновки з розділу	58
ВИСНОВКИ	60
ПЕРЕЛІК ПОСИЛАНЬ	63

ВСТУП

Автоматизація спостережень за розвитком рослин завжди була затребуваною в сільському господарстві через великі засівні площі та необхідність швидкого реагування на відхилення від нормального розвитку. Сьогодні проблема стає навіть більш актуальною через складність або неможливість потрапити на певні ділянки. Використання сучасних методів машинного навчання для аналізу розвитку рослин дозволяє значно підвищити ефективність і точність моніторингу, а також оптимізувати процеси вирощування та збору врожаю. Тема є надзвичайно актуальною в контексті сучасних технологічних можливостей та потреб агропромислового комплексу.

Сучасні дослідження свідчать про значний потенціал застосування штучного інтелекту та комп'ютерного зору в аграрній сфері. Особливу увагу науковці приділяють нейронним мережам, їх архітектурі, принципам роботи та впливу на різні аспекти сільського господарства. Використання таких технологій дозволяє значно підвищити ефективність і точність аналізу стану рослин, виявлення хвороб та пошкоджень, а також оптимізацію процесів вирощування та збору врожаю. Зокрема, автоматизація процесів моніторингу стану рослин та їхнього розвитку за допомогою методів обробки зображень дозволяє агрономам швидко і точно ідентифікувати ознаки хвороб, стресів або дефіциту поживних речовин, що дозволяє оперативно вживати заходів для усунення проблем і зменшення їхнього впливу на врожайність.

Метою роботи є дослідження можливості використання моделей машинного навчання для визначення площі зеленої частини рослини. Для досягнення цієї мети були поставлені наступні завдання:

1. Проаналізувати сучасні методи застосування машинного навчання в аграрній сфері.
2. Оцінити архітектури нейронних мереж та їх вплив на ефективність сільськогосподарських процесів.

3. Протестувати моделі машинного навчання на реальних даних у вигляді фото рослин.

4. Оцінити точність та ефективність запропонованих моделей у задачах моніторингу рослин.

Наукова новизна роботи полягає у впровадженні передових методів машинного навчання для аналізу розвитку рослин. Зокрема, було перевірено, налаштовано та донавчено модель YOLOv8 на спеціально підготовленому наборі даних. Процес включав використання платформи Roboflow для розмітки даних і подальше навчання моделі на цих даних. Після навчання модель показала значне покращення у якості анотацій зображень, що свідчить про ефективність підходу та можливість застосування цієї моделі для практичних завдань у галузі агрономії. Також було виявлено, що правильна підготовка набору даних є критично важливою для точності моделей машинного навчання, а також важливість ретельного вибору інструментів та методів обробки даних для забезпечення високої ефективності моделей.

Практична значущість роботи полягає у можливості використання розроблених моделей для автоматизації процесів у сільському господарстві. Зокрема, застосування моделі YOLOv8 для моніторингу росту рослин дозволить агрономам автоматизувати спостереження за розвитком рослин, що сприяє оперативному прийняттю рішень для покращення врожайності. Крім того, моделі машинного навчання можуть бути використані для визначення оптимального часу збору врожаю, що допомагає фермерам мінімізувати втрати та підвищити ефективність виробництва. Застосування алгоритмів обробки великих даних у поєднанні з штучним інтелектом дозволить створювати точні моделі прогнозування врожайності, виявляти патерни в розвитку рослин та передбачати вплив різних факторів на їхній ріст. Це дозволить фермерам приймати більш обґрунтовані рішення та оптимізувати використання ресурсів.

1 ТЕОРЕТИЧНІ ВІДОМОСТІ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1 Необхідність оцінки стану рослин в сучасному світі

Актуальність оцінки стану пошкодження і розвитку рослин у сучасному світі є надзвичайно важливою з кількох причин. Насамперед, це пов'язано із зростаючими викликами, які стоять перед сільським господарством через глобальні зміни клімату. Зміна температурних режимів, підвищення частоти та інтенсивності екстремальних погодних явищ, таких як посухи, повені та шторми, значно впливають на стан рослинних культур, їхню продуктивність та стійкість до захворювань. У цих умовах моніторинг і своєчасна оцінка стану рослин набувають критичного значення для забезпечення продовольчої безпеки та ефективного управління агроекосистемами.

Крім того, сучасні методи оцінки стану рослин дозволяють виявляти початкові стадії захворювань та пошкоджень (рис. 1.1 [1]), що дає змогу здійснювати превентивні заходи з метою мінімізації втрат врожаю. Зокрема, впровадження високотехнологічних засобів, таких як дистанційне зондування Землі, використання дронів та супутникових знімків, дозволяє фермерам отримувати детальну інформацію про стан їхніх полів у режимі реального часу. Це не лише покращує оперативність реагування на потенційні загрози, але й сприяє оптимізації використання ресурсів, таких як вода, добрива та засоби захисту рослин.



Рисунок 1.1 – Приклад початкової стадії захворювання рослини [1]

У сучасних умовах також зростає значення інтеграції інформаційних технологій у процес оцінки стану рослин. Використання великих даних (Big Data), машинного навчання та штучного інтелекту відкриває нові можливості для більш точного прогнозування розвитку рослин та їхньої реакції на змінні умови навколишнього середовища. Завдяки цьому стає можливим більш раціональне планування сільськогосподарських операцій та підвищення ефективності агровиробництва.

Суттєвий внесок у розвиток методів оцінки стану рослин вносять і біотехнологічні дослідження. Генетичні маркери, молекулярна діагностика та інші біотехнологічні підходи дозволяють не лише швидко ідентифікувати патогени та виявляти генетичні характеристики рослин, але й розробляти нові сорти, стійкі до хвороб та несприятливих умов середовища. Усе це сприяє зростанню продуктивності та стабільності сільськогосподарських систем, що є надзвичайно важливим у контексті глобального зростання населення та потреби у забезпеченні продовольством.

Також варто зазначити, що оцінка стану рослин має важливе значення для охорони навколишнього середовища та збереження біорізноманіття. Сучасні методи моніторингу дозволяють оцінювати вплив сільськогосподарської діяльності на екосистеми, виявляти деградацію ґрунтів, забруднення водних ресурсів та інші екологічні проблеми. Це дає можливість приймати обґрунтовані рішення щодо впровадження сталих практик землеробства, зменшення негативного впливу агровиробництва на довкілля та сприяння збереженню екосистемних послуг.

У контексті глобалізації та міжнародної торгівлі оцінка стану рослин також набуває значення з точки зору забезпечення фітосанітарної безпеки. Поширення карантинних шкідників і хвороб рослин може мати значні економічні наслідки, а тому швидке і точне виявлення таких загроз є ключовим для захисту національних агровиробників та запобігання втратам у міжнародній торгівлі сільськогосподарською продукцією.

У сучасних умовах також важливо враховувати аспект соціальної значущості оцінки стану рослин. У багатьох регіонах світу сільське господарство є основним джерелом засобів для існування значної частини населення. Тому забезпечення здоров'я та продуктивності рослин безпосередньо впливає на економічний добробут та соціальну стабільність цих регіонів.

Основні типи пошкоджень, які трапляються з рослинами, класифікуються на біотичні та абіотичні [2], а також можна ще додати механічні та фізіологічні. Кожен з цих типів має свої особливості та причини виникнення.

Механічні пошкодження виникають внаслідок дії зовнішніх фізичних факторів. Це можуть бути пошкодження, спричинені вітром, градом, снігом, дощем, а також механічні ушкодження під час агротехнічних заходів, таких як оранка, посів чи збирання врожаю. До цієї групи також відносяться пошкодження від диких тварин або комах, які можуть фізично пошкоджувати рослини, розриваючи або обгризаючи їхні частини.

Фізіологічні пошкодження виникають через порушення нормальних фізіологічних процесів у рослинах. Вони можуть бути викликані дефіцитом або надлишком води, неправильним живленням, надмірним або недостатнім освітленням, а також несприятливими температурними умовами. Наприклад, дефіцит води призводить до зневоднення та в'янення рослин, а надмірне зволоження може спричинити гниття коренів. Недостатнє або надмірне освітлення впливає на процес фотосинтезу, що, в свою чергу, відображається на рості та розвитку рослин.

Біотичні пошкодження пов'язані з дією живих організмів – патогенних мікроорганізмів (грибів, бактерій, вірусів), комах-шкідників, нематод, а також паразитичних рослин. Грибкові інфекції, такі як борошниста роса, фітофтороз та іржа, можуть сильно впливати на здоров'я рослин, викликаючи різноманітні симптоми, від некрозу тканин до повного загибелі рослини. Бактеріальні хвороби, такі як бактеріоз та в'янення, також можуть спричиняти значні пошкодження рослин, в той час як вірусні інфекції, такі як мозаїчні захворювання, можуть уповільнювати ріст та знижувати продуктивність рослин.

Комахи-шкідники, такі як попелиці, кліщі та гусениці, безпосередньо пошкоджують рослини, споживаючи їхні тканини або висмоктуючи соки, що призводить до ослаблення рослин та зниження їхньої врожайності.

Абіотичні пошкодження спричинені несприятливими умовами навколишнього середовища. Вони можуть бути викликані екстремальними температурами, як високими, так і низькими, хімічними речовинами, такими як пестициди, гербіциди та добрива, а також забрудненням повітря, ґрунту та води. Наприклад, високі температури можуть спричинити тепловий шок, тоді як низькі температури можуть викликати морозобоїни або загибель тканин рослин. Хімічні пошкодження виникають внаслідок неправильного застосування агрохімікатів або високого вмісту токсичних речовин у ґрунті та воді, що призводить до фітотоксичності.

Кожен з цих типів пошкоджень має свої специфічні ознаки та симптоми, які можуть варіюватися залежно від виду рослин та умов середовища. Знання характерних ознак різних типів пошкоджень є критично важливим для своєчасної діагностики та впровадження ефективних заходів захисту рослин. Застосування комплексного підходу до ідентифікації та оцінки пошкоджень, включаючи візуальні спостереження, лабораторні дослідження та використання сучасних технологій, таких як дистанційне зондування та молекулярні методи, дозволяє більш точно визначати причини пошкоджень та розробляти ефективні стратегії їхнього попередження та контролю.

Історія дослідження стану рослин та їх пошкоджень охоплює тривалий період, починаючи від давніх часів до сучасних високотехнологічних підходів. У стародавні часи люди здебільшого покладалися на спостереження і накопичений досвід для визначення здоров'я рослин та боротьби зі шкідниками і хворобами. Сільськогосподарські культури спостерігалися переважно емпіричними методами, що включали просте візуальне оцінювання та порівняння стану рослин в різних умовах.

Античні цивілізації, такі як Єгипетська та Римська, внесли значний вклад у розвиток агрономічних знань. Відомо, що вони використовували методи

іригації, обробки ґрунту та знання про ротацію культур для покращення врожаїв. Проте, систематичних досліджень стану рослин ще не існувало.

Середньовіччя було періодом стагнації у науках, включаючи агрономію, проте з появою Відродження спостерігається відродження інтересу до природознавства. Важливим кроком у розвитку науки про рослини став винахід мікроскопа в XVII столітті, який дозволив Антоні ван Левенгуку та іншим вченим відкривати світ мікроорганізмів. Це, в свою чергу, сприяло розумінню ролі патогенів у хворобах рослин.

У XVIII та XIX століттях почалися більш систематичні дослідження в галузі ботаніки та агрономії. Вчені, такі як Карл Лінней, зробили значний внесок у класифікацію рослин [3], що дало змогу краще розуміти їхню біологію та взаємодію з навколишнім середовищем. У цей період також розвивалися методи дослідження ґрунтів, і з'явилися перші теорії про живлення рослин, які пізніше були підтверджені роботами Юстуса фон Лібіха [4].

З розвитком хімії та біології у XIX столітті з'явилися методи аналізу хімічного складу рослин, що дозволило краще розуміти фізіологічні процеси і реакції на різні стресові фактори. Зокрема, вивчення фотосинтезу та обміну речовин відкрило нові горизонти в дослідженні росту і розвитку рослин.

У XX столітті розвиток генетики та молекулярної біології відкрив нові можливості для дослідження рослин на рівні генів і клітин. Завдяки роботам таких вчених, як Грегор Мендель та Барбара Макклінток, стало можливим розуміння спадковості та мутацій, що значно вплинуло на селекцію та створення нових сортів рослин, стійких до хвороб та шкідників.

Середина та друга половина XX століття стали періодом активного впровадження хімічних засобів захисту рослин, таких як пестициди та гербіциди. Хоча вони значно підвищили врожайність та зменшили втрати від шкідників і хвороб, їх надмірне використання спричинило екологічні проблеми та стійкість шкідників до хімічних препаратів. Це викликало необхідність пошуку нових методів оцінки стану рослин та боротьби з їхніми пошкодженнями.

З кінця XX століття та на початку XXI століття, розвиток інформаційних технологій та біотехнологій радикально змінив підходи до дослідження стану рослин. Впровадження дистанційного зондування, зокрема використання супутників і дронів, дозволило здійснювати моніторинг великих площ сільськогосподарських угідь в режимі реального часу. Це забезпечило можливість оперативного виявлення проблем, таких як захворювання чи дефіцит елементів живлення.

Сучасні методи дослідження включають також використання великих даних (Big Data), машинного навчання та штучного інтелекту для аналізу стану рослин. Біотехнологічні методи, такі як генетична інженерія та редагування геному, дозволяють створювати рослини з покращеними характеристиками стійкості до стресів, хвороб та шкідників.

1.2 Роль Big Data та штучного інтелекту в дослідженні розвитку рослин

Big Data (великі дані) — набори інформації (як структурованої, так і неструктурованої) настільки великих розмірів, що традиційні способи та підходи (здебільшого засновані на інструментах бізнес-аналітики та системах управління базами даних) не можуть бути до них застосовані [5]. У сфері дослідження розвитку рослин, Big Data включає широкий спектр інформації, отриманої з різних джерел, таких як супутникові знімки, дані з дронів, метеорологічні спостереження, дані сенсорів, генетична інформація та інші біологічні дані. Цей підхід дозволяє здійснювати більш комплексний та точний аналіз стану рослин, їхнього розвитку та взаємодії з навколишнім середовищем.

Одним з ключових аспектів застосування Big Data в агрономії є можливість збору і аналізу даних в режимі реального часу. Завдяки цьому аграрії можуть отримувати актуальну інформацію про стан своїх полів, що дозволяє оперативно реагувати на зміни і приймати обґрунтовані рішення щодо догляду за рослинами.

Наприклад, за допомогою сенсорів, встановлених у полі, можна відслідковувати вологість ґрунту, температуру, рівень освітленості та інші параметри, що впливають на ріст рослин. Ці дані можуть бути використані для оптимізації зрошення, внесення добрив та інших агротехнічних заходів.

Big Data також має важливе значення для генетичних досліджень та селекції рослин. Збір і аналіз геномних даних дозволяє ідентифікувати гени, відповідальні за стійкість до захворювань, посуху, високих чи низьких температур. Використовуючи ці дані, селекціонери можуть створювати нові сорти рослин з покращеними характеристиками, які краще адаптовані до змін клімату та інших стресових факторів. Такі підходи дозволяють значно прискорити процес селекції порівняно з традиційними методами.

Ще один важливий аспект використання Big Data в агрономії — це моделювання і прогнозування. Застосовуючи математичні моделі і статистичний аналіз великих обсягів даних, можна прогнозувати врожайність, поширення хвороб і шкідників, вплив змін клімату на розвиток рослин. Це дозволяє фермерам і агрономам планувати свої дії заздалегідь, що сприяє зниженню ризиків і підвищенню ефективності виробництва.

Використання Big Data також сприяє розвитку прецизійного землеробства, яке базується на індивідуальному підході до кожного окремого поля або навіть ділянки поля. Це включає оптимізацію посівних площ, управління водними ресурсами, точне дозування добрив і засобів захисту рослин, що дозволяє зменшити негативний вплив на навколишнє середовище і підвищити стійкість агроєкосистем.

Загалом, інтеграція Big Data в дослідження розвитку рослин відкриває нові горизонти для підвищення продуктивності та сталості сільського господарства. Використання великих даних дозволяє більш точно і ефективно управляти рослинними ресурсами, що є критично важливим у контексті зростаючого населення планети та змін клімату.

У свою чергу, штучний інтелект — це імітація процесів людського інтелекту машинами, особливо комп'ютерними системами [6]. Це включає

обробку природної мови, розпізнавання образів, прийняття рішень, машинне навчання та інші форми аналізу даних. Штучний інтелект використовує алгоритми, моделі та статистичні методи для аналізу великих обсягів даних, виявлення закономірностей та прогнозування.

У контексті дослідження розвитку рослин штучний інтелект має декілька важливих застосувань, які значно впливають на ефективність та точність досліджень.

Перш за все, штучний інтелект дозволяє здійснювати автоматизований моніторинг та аналіз стану рослин (рис. 1.2 [7]). Використовуючи технології комп'ютерного зору та машинного навчання, системи штучного інтелекту можуть обробляти зображення, отримані з дронів, супутників або наземних сенсорів, і автоматично ідентифікувати різні види пошкоджень, захворювань чи стресових умов. Наприклад, алгоритми можуть розпізнавати ознаки хвороб на листках рослин, виявляти шкідників або оцінювати ступінь пошкодження від несприятливих погодних умов.



Рисунок 1.2 – Алгоритми штучного інтелекту визначають ознаки хвороби рослини [7]

Друге важливе застосування штучного інтелекту полягає в оптимізації агротехнічних заходів. Використовуючи дані про стан ґрунту, кліматичні умови та біологічні параметри рослин, алгоритми штучного інтелекту можуть надавати рекомендації щодо оптимального часу для посіву, зрошення, внесення добрив та засобів захисту рослин. Це дозволяє підвищити ефективність використання

ресурсів, зменшити витрати та мінімізувати негативний вплив на навколишнє середовище.

Штучний інтелект також активно використовується в селекції та генетиці рослин. Машинне навчання дозволяє аналізувати великі обсяги геномних даних для виявлення генів, пов'язаних з певними бажаними властивостями, такими як стійкість до хвороб, посухи чи висока врожайність. Це значно прискорює процес селекції нових сортів, оскільки дозволяє точніше прогнозувати результати схрещування та відбирати найбільш перспективні зразки.

Іншим важливим напрямком є моделювання і прогнозування. Використовуючи дані про попередні врожаї, погодні умови та агротехнічні заходи, алгоритми штучного інтелекту можуть створювати моделі, що прогнозують врожайність, розвиток хвороб чи поширення шкідників. Це дозволяє фермерам краще планувати свої дії, знижуючи ризики та підвищуючи стабільність виробництва.

Окрім цього, штучний інтелект може сприяти розвитку автоматизованих систем управління фермерськими господарствами. Такі системи можуть інтегрувати дані з різних джерел, аналізувати їх та автоматично управляти різними аспектами сільськогосподарського виробництва, від зрошення до збору врожаю. Це не лише підвищує продуктивність, але й дозволяє зменшити потребу в ручній праці та підвищити точність виконання агротехнічних операцій.

Штучний інтелект також має значний потенціал для покращення стійкості агросистем до змін клімату. Використовуючи дані про погодні умови, ґрунтові характеристики та біологічні властивості рослин, системи штучного інтелекту можуть розробляти стратегії адаптації, що дозволяють мінімізувати негативний вплив кліматичних змін на сільське господарство.

Таким чином, штучний інтелект відіграє ключову роль у сучасних дослідженнях розвитку рослин, сприяючи підвищенню ефективності, точності та стійкості сільськогосподарського виробництва. Його інтеграція в агрономію відкриває нові можливості для оптимізації процесів вирощування рослин,

селекції нових сортів та управління агроєкосистемами, що є критично важливим у контексті глобальних викликів продовольчої безпеки та змін клімату.

Поєднання штучного інтелекту та Big Data для дослідження пошкоджень та поточного стану дорослості рослини є однією з найсучасніших та перспективних областей в агрономії. Це інтегрування технологій дозволяє здійснювати глибокий аналіз і моніторинг рослин в режимі реального часу, підвищуючи ефективність сільськогосподарського виробництва та стійкість агроєкосистем.

Історія Big Data починається з появою цифрових технологій і накопичення великих обсягів інформації в кінці XX століття. Зі зростанням обчислювальних потужностей і розвитком інтернету з'явилися нові можливості для збору та зберігання даних. У сільському господарстві Big Data стали активно використовуватися з початку 2000-х років, коли сенсори і супутникові технології стали більш доступними та точними.

Історія штучного інтелекту розпочалася набагато раніше, ще в середині XX століття, з розробки перших алгоритмів для обробки інформації та прийняття рішень. Однак справжній прорив у застосуванні штучного інтелекту для агрономії стався на початку XXI століття завдяки розвитку машинного навчання та глибокого навчання. Ці методи дозволили значно підвищити точність і швидкість аналізу великих даних.

Що є дуже важливим для цієї роботи, алгоритми штучного інтелекту та Big Data можуть використовуватися для оцінки пошкоджень рослин. За допомогою комп'ютерного зору та обробки зображень можна автоматично розпізнавати ознаки хвороб, пошкоджень від шкідників або стресів, спричинених несприятливими умовами. Це дозволяє швидко і точно ідентифікувати проблемні ділянки на полях та приймати відповідні заходи для їх усунення.

Таким чином, поєднання штучного інтелекту та Big Data є потужним інструментом для сучасного сільського господарства, що дозволяє забезпечувати

високу врожайність, стійкість до змін клімату та інші важливі аспекти аграрного виробництва.

1.3 Нейронні мережі, їх застосування, принципи роботи та вплив на сучасну агрономію

Штучні нейронні мережі є одним із сучасних інструментів штучного інтелекту. Вони складаються з великої кількості штучних нейронів, які зв'язані між собою за допомогою ваг, що регулюють сили сигналів між нейронами (рис. 1.3 [8]). Ці мережі здатні до самонавчання та адаптації до змін в навколишньому середовищі.

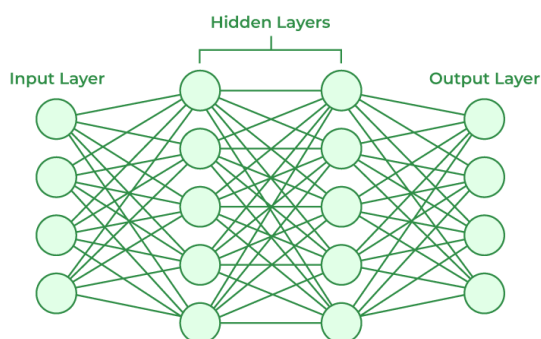


Рисунок 1.3 – Приклад нейронної мережі [8]

Нейронні мережі в основному працюють на основі функціонування біологічних нейронних мереж у мозку людини [9]. Основні компоненти нейронної мережі включають в себе вхідні шари, приховані шари (якщо вони присутні), і вихідний шар. Кожен з цих шарів містить нейрони, які приймають вхідні сигнали, обчислюють їх і, в кінцевому підсумку, видають вихідні значення.

Принцип роботи нейронної мережі полягає у передачі сигналів від одного шару нейронів до іншого через зв'язки між ними [10]. Кожен нейрон у нейронній мережі має вагу, яка визначає, наскільки важливий є вхідний сигнал. Під час

процесу навчання ці ваги оптимізуються з метою мінімізації помилки між фактичними та бажаними вихідними значеннями.

Поточний нейрон обчислює зважену суму своїх вхідних сигналів, додаючи їх між собою та домножуючи на вагу кожного з'єднання (1.1). Після цього до отриманої суми застосовується активаційна функція, яка визначає вихідне значення нейрона. Цей процес повторюється для кожного нейрона в мережі, починаючи з вхідного шару і закінчуючи вихідним.

$$n_i = \sum_j (w_{ji} * n_j), \quad (1.1)$$

де n_i – значення i -го нейрону в поточному шарі;

n_j – значення j -го нейрону в попередньому шарі;

w_{ji} – значення ваги з'єднання від j -го нейрону попереднього шару до i -го нейрону поточного шару.

Активаційна функція в штучній нейронній мережі визначає вихідний сигнал нейрона відповідно до зваженої суми його вхідних сигналів. Ця функція визначає, коли нейрон має передавати сигнал наступному шару мережі. Значення активаційної функції залежить від вхідних сигналів та параметрів самого нейрона.

Приклади активаційних функцій включають у себе сигмоїдальну функцію (1.2), яка перетворює вхідні значення у діапазон від 0 до 1, та функцію ReLU (1.3), яка видає вихідне значення, якщо вхід більше за нуль, інакше вона повертає нуль. Існують також інші типи активаційних функцій, такі як гіперболічний тангенс та softmax.

$$f(x) = \frac{1}{1+e^{-x}}, \quad (1.2)$$

$$f(x) = \max(0, x), \quad (1.3)$$

Активаційні функції є важливим елементом нейронних мереж, оскільки вони дозволяють моделі вчитися та адаптуватися до складних залежностей в даних. Вони допомагають вирішувати проблему лінійної нероздільності та дозволяють нейронній мережі моделювати складні функції та відносини між вхідними та вихідними даними.

Одним з ключових етапів роботи нейронної мережі є навчання. Під час навчання нейронна мережа використовується для аналізу набору навчальних даних і встановлюється оптимальне значення ваг для кожного з'єднання між нейронами. Цей процес зазвичай здійснюється шляхом використання алгоритмів зворотного поширення помилки, які коригують ваги відповідно до різниці між фактичними та передбаченими вихідними значеннями.

Ще однією важливою характеристикою нейронних мереж є їх здатність до адаптації до нових даних. Після завершення процесу навчання мережа може бути застосована для аналізу нових даних та видачі передбачень без необхідності повторного навчання. Така здатність до адаптації робить нейронні мережі ефективними інструментами для вирішення складних задач аналізу даних та прогнозування.

Застосування штучних нейронних мереж в широкому спектрі сфер, включаючи медицину, фінанси, транспорт, а також сільське господарство. В агрономії вони можуть бути використані для аналізу великих обсягів даних про розвиток рослин, виявлення патологій, класифікації шкідників та хвороб, а також для прогнозування врожаю та розробки оптимальних стратегій вирощування рослин.

Одним із переваг штучних нейронних мереж є їхній потенціал для автоматизації та оптимізації процесів, що вимагають складних аналітичних рішень. Вони можуть працювати з великими обсягами даних та швидко виявляти взаємозв'язки та закономірності [11], що дозволяє зробити більш точні та обґрунтовані рішення. Також вони можуть навчатися на нових даних та поступово покращувати свою ефективність та точність.

Незважаючи на свою потужність та широке застосування, штучні нейронні мережі мають кілька недоліків:

- Один з головних недоліків полягає в тому, що навчання нейронних мереж може бути вимогливим з точки зору обчислювальних ресурсів та часу. Особливо при великих розмірах даних та складних структурах мереж, процес навчання може займати значну кількість часу і вимагати потужних обчислювальних ресурсів.

- Також важливо враховувати, що інтерпретація результатів нейронних мереж може бути складною. У багатьох випадках нейронні мережі діють як «чорні скрині», тобто важко розуміти, як саме вони прийшли до своїх висновків або розв'язків. Це може ускладнити процеси прийняття рішень на практиці та використання нейронних мереж у ситуаціях, де важливо мати чітке пояснення рішень.

- Деякі нейронні мережі також можуть страждати від проблеми перенавчання, коли вони показують дуже високу точність на навчальних даних, але погано працюють на нових даних. Це може виникати через надмірну складність моделі, яка може дуже докладно пристосовуватися до навчальних даних, але узагальнюючих навичок недостатньо для ефективного застосування на нових даних.

- Ще однією проблемою є вразливість нейронних мереж до атак, таких як атаки на введення або зміну даних. Деякі типи нейронних мереж можуть бути дуже чутливими до навіть невеликих змін навчальних даних, що може призвести до неправильних передбачень або виконання небезпечних дій.

- Нарешті, однією з основних проблем є важкість визначення оптимальної архітектури нейронної мережі для конкретного завдання. Вибір кількості шарів, кількості нейронів у кожному шарі та інших параметрів може бути складним завданням,

і неправильний вибір може призвести до поганої продуктивності мережі.

Проте, незважаючи на недоліки, штучні нейронні мережі мають свої переваги, завдяки яким їх застосування в дослідженнях стану дорослості рослини є досить обґрунтованим.

Наприклад, штучні нейронні мережі можуть бути використані для раннього виявлення ознак стресу в рослинах, таких як недостатність води, нестача поживних речовин або зараження хворобами. Застосування нейронних мереж дозволяє швидко аналізувати великі обсяги даних, включаючи дані від сенсорів на полях, метеорологічних станцій та супутникові знімки, щоб виявити відхилення від норми та вчасно реагувати на них.

Крім того, штучні нейронні мережі можуть бути ефективним інструментом для визначення оптимальних агротехнічних заходів та стратегій вирощування рослин. Шляхом аналізу великих обсягів даних про врожайність, вплив погодних умов та агротехнічних заходів, нейронні мережі можуть допомогти фермерам та агрономам вирішувати складні завдання планування та управління виробництвом.

Отже, не дивлячись на недоліки, застосування штучних нейронних мереж у дослідженнях стану пошкодженості та поточного стану того, наскільки рослина вже виросла, має значний потенціал, що може використовуватися для покращення продуктивності та стійкості сільськогосподарського виробництва.

1.4 Нейронна мережа YOLOv8, її застосування та особливості

YOLOv8 (You Only Look Once Version 8) – це модель глибокого навчання для об'єктного виявлення в зображеннях і відео (рис. 1.4 [12]). Ця модель відноситься до штучних нейронних мереж, оскільки вона базується на

архітектурі глибоких нейронних мереж, зокрема згорткових нейронних мереж (CNN).



Рисунок 1.4 – логотип нейронної мережі [12]

У свою чергу, згорткові нейронні мережі, що ще називаються Convolutional Neural Networks або CNNs (рис. 1.5 [13]), є типом штучних нейронних мереж, які спеціалізуються на обробці зображень та виявленні в них шаблонів та особливостей. Вони широко використовуються в областях комп'ютерного зору, розпізнавання образів, відеоаналітики та інших сферах, де потрібна обробка великих обсягів візуальних даних.

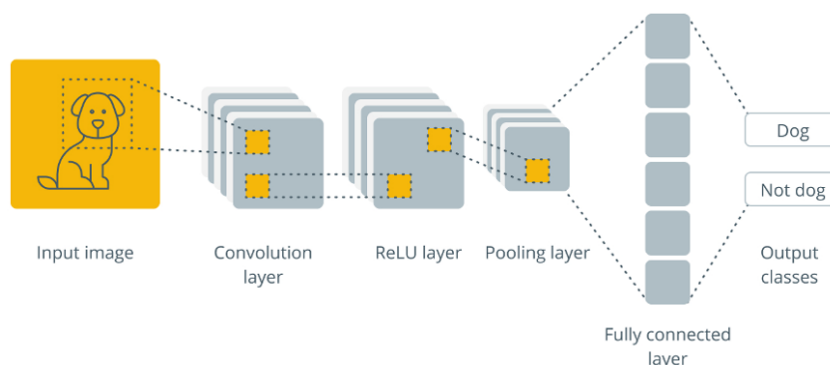


Рисунок 1.5 – Принцип роботи згорткових нейронних мереж [13]

Основна особливість згорткових нейронних мереж полягає в їхній структурі, яка включає так звані "згорткові шари". Ці нейронні шари здійснюють операцію згортки, що дозволяє виявляти різні шаблони та ознаки у зображеннях. Після цього зазвичай використовуються інші типи шарів, такі як пулінг-шари для зменшення розмірності даних та повнозв'язані шари для класифікації або визначення об'єктів на зображенні.

Згортка забезпечує нейронним мережам здатність розпізнавати локальні структури в даних, такі як границі, кольорові плями, текстури тощо.

Суть згортки полягає в тому, що ми беремо ядро (яке також називається фільтром або маскою) та рухаємо його по вхідним даним. Для кожного положення ядра на вхідних даних виконується деяка операція. Зазвичай ця операція - обчислення зваженої суми значень вхідних даних, які перетинаються з ядром. Результат цієї операції утворює одну точку у вихідному зображенні, яке називається картою ознак.

Використання згортки у нейронних мережах дозволяє здійснювати локальну обробку даних, зосереджуючись на малих областях та виявляючи локальні структури. Це допомагає мережі впізнавати шаблони незалежно від їх положення у вхідних даних, що робить її ефективною для різних завдань, таких як розпізнавання образів та виявлення об'єктів на зображеннях.

Згорткові нейронні мережі можуть працювати з різними типами даних, включаючи зображення з різними розмірами та кольоровими просторами. Вони можуть автоматично виявляти та вивчати різні характеристики об'єктів на зображеннях, такі як форма, текстура, контур і колір, що робить їх ефективними для різних завдань обробки зображень.

Принцип роботи згорткових нейронних мереж можна пояснити простими словами так:

Уявімо, що ми маємо зображення з кішкою. Кожен піксель цього зображення має три значення, що відображає інтенсивність кольору (рис 1.6). Згорткова нейронна мережа розглядає це зображення, подібно до того, як ми розглядаємо піксель за пікселем.

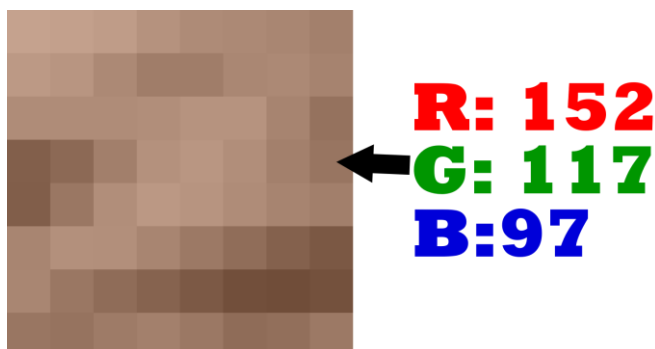


Рисунок 1.6 – Приклад яскравості кольорів пікселю

У нейронній мережі є спеціальні фільтри, які навчаються розпізнавати різні ознаки на зображенні, такі як границі, кольорові плями або текстури. Ці фільтри маленькі, наприклад, вони можуть бути матрицями 3 на 3 або 5 на 5, і вони переміщуються по всьому зображенню.

Під час переміщення фільтра, він згортається з пікселями у маленькій області зображення, обчислюючи зважену суму значень цих пікселів. Це значення стає новим значенням для пікселя у вихідному зображенні. Цей процес відбувається для кожного пікселя у зображенні.

Наприклад, якщо один з фільтрів навчився реагувати на вертикальні лінії, він виділить вертикальні границі на зображенні, визначаючи, де саме на зображенні є такі границі.

У свою чергу, YOLOv8 є однією з останніх версій популярної моделі для об'єктного виявлення в зображеннях і відео. До прикладів його застосування можна віднести системи допомоги водіям для виявлення дорожніх знаків, пішоходів, транспортних засобів та інших об'єктів на дорозі. Це допомагає автономним транспортним засобам у прийнятті рішень та уникненні аварій.

Ще однією галуззю, де можна застосовувати цю модель штучної нейронної мережі може бути медична діагностика. YOLOv8 може бути використана для автоматичного виявлення аномалій та патологій на медичних зображеннях, таких як рентгенівські знімки, МРТ або комп'ютерні томографії. Це може допомогти лікарям у точній та швидкій діагностиці різних захворювань.

В той же час, застосування цієї технології у сфері аграрних досліджень може мати низку важливих переваг:

- Перш за все, YOLOv8 може бути використаний для автоматичного виявлення шкідників та хвороб на рослинах. Це дозволить швидко виявляти будь-які небезпеки для рослин, такі як шкідники, грибкові захворювання або віруси, що можуть завдати шкоди врожаю.

- Крім того, YOLOv8 може допомогти в оцінці поточного стану рослин. Шляхом аналізу зображень рослин можна виявити ознаки стресу, дефіциту поживних речовин або інших проблем, що можуть вплинути на їх розвиток та врожайність.

- Додатково, ця технологія може бути корисною у визначенні ступеня дорослості рослин. Виявлення рівня зрілості дозволить сільськогосподарським фермерам оптимізувати час збирання врожаю та максимізувати виробництво.

В цій роботі, YOLOv8 буде використано для визначення площі зеленої частини рослини.

1.5 Огляд аналогів

Розглядаючи аналоги YOLOv8, слід визначити їх основні характеристики та відмінності, що визначають їхню придатність для конкретних завдань в області об'єктного виявлення.

FastFCN є інноваційною моделлю у сфері сегментації зображень, яка спеціалізується на швидкому та ефективному виділенні областей об'єктів на зображенні. Порівняно з YOLOv8, яка зосереджується на об'єктному виявленні, FastFCN дозволяє визначати піксельні маски об'єктів, що дозволяє отримати більш деталізовану інформацію про границі та структуру об'єктів на зображенні.

Що стосується архітектури, FastFCN використовує швидкі та легкі блоки для розрахунку карт високої роздільної здатності. Це дозволяє моделі швидко та ефективно працювати з великими обсягами даних, що особливо важливо у сучасному об'ємному потоці зображень.

Однією з головних переваг FastFCN є його швидкість. Модель може опрацьовувати зображення швидше за багато інших методів, зберігаючи високу точність сегментації. Це особливо важливо у сферах, де важливо швидко

аналізувати великі обсяги даних, наприклад у медичних аналізах або в галузі автомобільної безпеки.

Крім того, FastFCN може бути ефективно використаний у сфері відеоспостереження, де швидкість обробки даних є критичною для виявлення та відстеження об'єктів у реальному часі.

Одним з основних недоліків FastFCN порівняно з YOLOv8 є його призначення для сегментації зображень, тобто визначення піксельних масок об'єктів на зображенні. Хоча це дозволяє отримати детальнішу інформацію про структуру об'єктів, сам процес сегментації може вимагати більше ресурсів та часу порівняно з об'єктним виявленням, яке використовує YOLOv8.

З іншого боку, YOLOv8 володіє швидкістю та ефективністю обробки даних, що робить його ідеальним для виконання завдань по розпізнаванню дорослості та ступеню пошкодження рослин у великих обсягах даних.

Її здатність швидко та точно виявляти об'єкти на зображеннях дозволяє швидко аналізувати велику кількість зразків та ефективно виявляти характеристики росту та пошкодження рослин.

Отже, для виконання завдань по розпізнаванню дорослості та ступеню пошкодження рослин у великих обсягах даних, YOLOv8 може бути кращим вибором завдяки своїй швидкості та ефективності обробки, порівняно з FastFCN.

Ще одним потенційним аналогом YOLOv8 є Mask R-CNN. Ця модель також відома своєю здатністю до об'єктного виявлення, але додає до цього можливість сегментації зображень [14], що робить її подібною до FastFCN, але з більшою увагою до деталей та точності. Mask R-CNN використовує дві мережі: одну для виявлення об'єктів та інший для генерації піксельних масок областей об'єктів. Це дозволяє отримувати додаткову інформацію про структуру та контури об'єктів на зображенні.

У порівнянні з YOLOv8, Mask R-CNN може бути більш точним та деталізованим інструментом для розпізнавання об'єктів на зображеннях. Проте, він зазвичай працює повільніше. Це може зробити його менш практичним для додатків, де важлива швидкість обробки даних.

Одним з недоліків Mask R-CNN порівняно з YOLOv8 є його складність та обчислювальна витратність. Mask R-CNN вимагає додаткових обчислень для генерації піксельних масок областей об'єктів, що може зробити його гіршим у порівнянні з YOLOv8, який пропонує швидкий та ефективний метод об'єктного виявлення.

З іншого боку, YOLOv8 відома своєю швидкістю та ефективністю обробки даних. Вона здатна швидко та точно виявляти об'єкти на зображеннях, що робить її ідеальним для аналізу великих обсягів даних у реальному часі. Її швидкість та точність можуть зробити її кращим вибором для виконання завдань по дослідженню рослин, особливо у великих агроecosистемах, де потрібно обробляти велику кількість зображень швидко та ефективно.

Поза методами машинного навчання, існують традиційні агрономічні підходи до оцінки пошкоджень та ступеня дорослості рослин, які базуються на фізичних ознаках та оцінках, що проводяться людьми. Одним з них є оцінка фізичних параметрів рослин, таких як висота, розміри листя, товщина стебла тощо. Агрономи традиційно використовують різноманітні інструменти, такі як лінійка, каліпери (рис. 1.7 [15]) та інші мірні засоби, для вимірювання цих параметрів на місці.



Рисунок 1.7 – Приклад каліперу [15]

Крім того, агрономи оцінюють стан рослин за їхнім зовнішнім виглядом, таким як колір та текстура листя, наявність ознак хвороб чи пошкоджень. Ці оцінки можуть бути зроблені шляхом візуального спостереження або за

допомогою спеціалізованих інструментів, таких як мікроскопи або фотокамери з високою роздільною здатністю.

До того ж, агрономи можуть використовувати традиційні методи оцінки росту рослин, такі як вимірювання кількості та розміру плодів чи квітів. Ці дані можуть бути використані для оцінки ступеня дорослості та зростання рослин у часі.

Проте, ці традиційні методи оцінки гірше за методи машинного навчання у деяких аспектах. Однією з основних обмежень є об'єктивність та стандартизованість оцінок. У традиційних методах оцінки, результати можуть значно варіюватися в залежності від суб'єктивності та досвіду оцінювача. Це може призвести до неточностей та недостовірності результатів, особливо при аналізі великої кількості даних.

Крім того, традиційні методи оцінки можуть бути часо- та працезатратними. Вимірювання фізичних параметрів рослин або візуальне спостереження за їх станом може вимагати значної кількості часу та зусиль, особливо у великих агроєкосистемах або дослідницьких проєктах.

Додатково, традиційні методи можуть бути менш ефективними у виявленні складних патернів або взаємозв'язків у даних. Машинне навчання може допомогти виявити приховані зв'язки та патерни, які можуть бути складні для виявлення за допомогою традиційних методів.

Отже, хоча традиційні методи оцінки можуть мати свої переваги та використовуватися в певних ситуаціях, вони часто виявляються менш ефективними та надійними порівняно з методами машинного навчання у сучасних дослідженнях розвитку та пошкоджень рослин.

1.6 Висновки з розділу

В даному розділі було досліджено сучасні методи застосування штучного інтелекту та комп'ютерного зору в аграрній сфері. Особливу увагу було приділено нейронним мережам, їх архітектурі, принципам роботи та впливу на різні аспекти сільського господарства. Використання таких технологій дозволяє значно підвищити ефективність і точність аналізу стану рослин, виявлення хвороб та пошкоджень, а також оптимізацію процесів вирощування та збору врожаю.

Застосування нейронних мереж та технології комп'ютерного зору в сільському господарстві має декілька ключових переваг:

- Перш за все, це можливість автоматизації процесів моніторингу стану рослин та їхнього розвитку. Використовуючи методи обробки зображень, агрономи можуть швидко і точно ідентифікувати ознаки хвороб, стресів або дефіциту поживних речовин, що дозволяє оперативно вживати заходів для усунення проблем і зменшення їхнього впливу на врожайність.

- Крім того, нейронні мережі можуть бути використані для оцінки ступеня дорослості рослин. Ця інформація є важливою для визначення оптимального часу збору врожаю, що допомагає фермерам максимізувати виробництво та мінімізувати втрати. Аналіз зображень рослин за допомогою алгоритмів глибокого навчання дозволяє отримати точні дані про їхній стан та рівень зрілості, що значно підвищує ефективність сільськогосподарських операцій.

- Важливою характеристикою нейронних мереж є їх здатність до адаптації до нових даних. Після завершення процесу навчання, мережа може бути застосована для аналізу нових даних та видачі передбачень без необхідності повторного навчання. Це робить нейронні мережі ефективними інструментами для вирішення складних задач аналізу даних та прогнозування в різних умовах та ситуаціях.

- Застосування нейронних мереж у сільському господарстві також включає в себе використання алгоритмів обробки великих даних (Big Data). Поєднання штучного інтелекту та великих обсягів даних дозволяє створювати точні моделі прогнозування врожайності, виявляти патерни в розвитку рослин та передбачати вплив різних факторів на їхній ріст. Це дозволяє фермерам приймати більш обґрунтовані рішення та оптимізувати використання ресурсів.

Загалом, впровадження нейронних мереж та технології комп'ютерного зору в аграрну сферу відкриває нові можливості для підвищення ефективності та стійкості сільськогосподарських процесів. Ці технології дозволяють забезпечити високу врожайність, зменшити витрати та зусилля, необхідні для догляду за рослинами, а також покращити якість продукції.

2 ДОСЛІДЖЕННЯ КОМП'ЮТЕРНО-ІНТЕГРОВАНИХ ТЕХНОЛОГІЙ ДЛЯ РЕАЛІЗАЦІЇ ЗАВДАННЯ

2.1 Обґрунтування вибору мови програмування

Вибір мови програмування для виконання конкретної задачі важливий з кількох причин:

По-перше, різні мови програмування мають різні особливості та характеристики, які можуть бути більш або менш підходящими для певного типу завдання. Наприклад, деякі мови мають вбудовану підтримку для певних типів обчислень або обробки даних, що робить їх ідеальними для виконання певних завдань.

По-друге, екосистема та інструментарій, що супроводжує мову програмування, може бути важливим фактором у виборі мови. Наприклад, деякі мови мають широкий вибір бібліотек та фреймворків, які полегшують розробку та розвиток програмного забезпечення. Інші можуть мати добре розвинену спільноту користувачів, яка забезпечує підтримку та розвиток мови.

Крім того, важливо враховувати зручність та зрозумілість мови програмування для команди розробників. Якщо команда вже має досвід у використанні певної мови або вона має більше знань та навичок у цій мові, то це може зробити розробку програмного забезпечення більш ефективною та продуктивною.

При виборі мови програмування для реалізації необхідної задачі було докладно оглянуто кілька варіантів, зокрема C++, Java, Python та JavaScript. Після ретельного аналізу об'єктивних та суб'єктивних переваг кожної мови було прийнято рішення вибрати саме Python для цього проєкту. У наступних абзацах будуть розглянуті причини цього вибору та доведено, чому саме ця мова програмування є доцільним вибором.

C++ - це загальноприйнята мова програмування, яка використовується у широкому спектрі сфер, від розробки вбудованих систем до ігор та

високопродуктивного програмного забезпечення [16]. Однією з основних переваг C++ є її висока продуктивність та ефективність. Ця мова дозволяє розробникам працювати на низькому рівні, отримуючи високу швидкість виконання програм. Крім того, C++ є дуже гнучкою мовою, яка дозволяє розробникам використовувати різні стилі програмування, включаючи об'єктно-орієнтоване програмування, процедурне програмування та шаблони.

Іншою важливою перевагою C++ є його велика спільнота користувачів та широка екосистема інструментів та бібліотек. Розробники мають доступ до великої кількості ресурсів та матеріалів для вивчення та розвитку мови. Крім того, існує багато сторонніх бібліотек та фреймворків, які полегшують розробку програмного забезпечення на C++.

Проте, C++ має і свої недоліки:

- Одним з найбільших недоліків є складність мови. C++ може бути складним для вивчення та розуміння, особливо для початківців. Вона має багато складних концепцій та може вимагати від розробників додаткового часу та зусиль для оволодіння мовою.
- Крім того, C++ може бути менш продуктивним у порівнянні з іншими мовами програмування, такими як Python або JavaScript, якщо не використовувати його ефективно.

У свою чергу, Java – це об'єктно-орієнтована мова програмування, яка відома своєю кросплатформеністю та простотою використання [17].

Однією з головних переваг Java є те, що програми, написані на цій мові, можуть працювати на будь-якій операційній системі, яка підтримує віртуальну машину Java (JVM). Це робить Java дуже популярною у сферах розробки програмного забезпечення, яке потрібно виконувати на різних платформах. Крім того, Java має велику кількість сторонніх бібліотек та фреймворків, що полегшують розробку програмного забезпечення та розширюють її можливості.

Ще однією перевагою Java є високий рівень безпеки. Java використовує механізми безпеки на рівні віртуальної машини, що робить програми, написані на цій мові, більш стійкими до атак та вразливостей. Це особливо важливо для

розробки програмного забезпечення, яке працює у веб-середовищі або обробляє конфіденційні дані.

Однак, Java теж не позбавлена недоліків. Великим недоліком є високий рівень витрат пам'яті. Велика кількість вбудованих бібліотек та інструментів може призвести до високого споживання пам'яті, що може стати проблемою для додатків з обмеженими ресурсами або високофункціональних додатків. Крім того, Java, хоча і має підтримку та багато потужних функцій, вона може бути менш ефективною у деяких областях, таких як обробка великих обсягів даних, порівняно з іншими мовами програмування.

Ще однією мовою програмування є JavaScript, яка є високорівневою, інтерпретованою мовою, що використовується для розробки веб-додатків та веб-сторінок [18]. Однією з основних переваг JavaScript є те, що вона є стандартною мовою програмування для вебу, що означає, що вона підтримується всіма сучасними веб-браузерами. Це робить JavaScript надзвичайно популярною серед розробників веб-додатків та дозволяє їм створювати динамічні та інтерактивні веб-сторінки з великою кількістю функцій.

Додатковою перевагою JavaScript є її простота вивчення та використання. JavaScript має простий, лаконічний синтаксис, який допомагає новачкам швидко засвоїти мову. Крім того, велика кількість документації та онлайн-ресурсів робить JavaScript доступною для вивчення та розвитку навичок.

Як й інші мови програмування, JavaScript також має свої недоліки. Однією з таких проблем є відсутність типізації. JavaScript є слаботипізованою мовою програмування, що може призвести до помилок під час розробки та ускладнити розуміння коду, особливо у великих проєктах. Крім того, через свою інтерпретованість, JavaScript може мати деякі проблеми з продуктивністю та швидкістю порівняно з компільованими мовами, такими як C++ або Java.

І останньою розглянутою тут мовою програмування буде Python. Вона є високорівневою, інтерпретованою мовою, яка відома своєю простотою та читабельністю коду [19].

Однією з основних переваг Python є його простота вивчення та використання. Мова має простий, лаконічний синтаксис, що дозволяє новачкам швидко оволодіти нею. Крім того, Python має велику кількість документації та онлайн-ресурсів, що робить її доступною для вивчення та розвитку навичок.

До того ж, Python має велику екосистему та розширюваність. В Python доступні велика кількість сторонніх бібліотек та фреймворків, що полегшують розробку програмного забезпечення та розширюють її можливості. Це робить Python дуже популярним серед розробників та використовується у багатьох галузях, від наукових досліджень до веб-розробки та штучного інтелекту.

Проте, в Python також є недоліки. Однією з найбільших проблем є його відносна повільність порівняно з іншими мовами програмування, такими як C++ або Java. Через те, що Python є інтерпретованою мовою, він може мати деякі проблеми з продуктивністю, особливо у великих проєктах або при обробці великих обсягів даних. Крім того, Python може мати меншу підтримку для деяких високопродуктивних задач, таких як розробка вбудованих систем або робота з низькорівневими операціями.

Якщо звести основні характеристики цих мов програмування, можна отримати значення, які представлені в таблиці 2.1:

Таблиця 2.1 – Коротка характеристика мов програмування

JavaScript	Зазвичай використовується для розробки програм, що виконуються в браузері, оскільки розроблений додаток має запускатися без сторонніх програм.
Python	Простіша мова програмування, що має вбудовану велику кількість алгоритмів та яка вимагає від програміста значно менше знань для створення багатьох програм.
C++	Ця мова програмування є однією з найшвидших для великих обчислень. Крім того, вона пропонує безліч інструментів для розробки.
Java	Дозволяє створювати кросплатформені додатки, що можуть працювати майже на будь-якій системі. Крім того, ця мова є однією з найструктурованіших та об'єктно-орієнтованих.

Отже, для виконання задач з дослідження розвитку рослин методами машинного навчання краще обрати або C++, або Python. Проте, у підсумку було обрано саме мову програмування Python, оскільки вона надає велику кількість інструментів для виконання завдань з використанням штучної нейронної мережі.

2.2 Обґрунтування вибору середовища розробки

В цьому підрозділі буде розглянуто чотири середовища розробки для Python, а саме IDLE, PyCharm, JupyterLab та Visual Studio Code, після чого обґрунтовано вибір саме одного з них.

IDLE (Integrated Development and Learning Environment) це інтегроване середовище розробки для Python, що дозволяє конфігурувати та навчати штучні

нейронні мережі [20]. Головною перевагою IDLE є те, що воно поставляється разом з установкою Python і є абсолютно безкоштовним у використанні. Це робить його доступним для широкого кола розробників та дослідників, які бажають вивчати розробку та навчання нейронних мереж.

Ще однією суттєвою перевагою IDLE є його надзвичайна простота використання та інтуїтивний інтерфейс. Крім того, IDLE включає різноманітні функції, такі як автоматичне вирівнювання, автодоповнення та інші, що сприяють зручній розробці.

Проте, IDLE має свої обмеження. Найбільш важливою проблемою є його обмежена функціональність у порівнянні з іншими інтегрованими середовищами розробки. Наприклад, воно може бути менш потужним у відлагодженні та профілюванні коду або у підтримці розширених функцій, таких як візуалізація даних. Крім того, IDLE може бути менш ефективним для роботи з великими та складними проєктами порівняно з іншими середовищами розробки.

Наступним йде PyCharm, який є інтегрованим середовищем розробки для Python, але надає більше функцій та можливостей порівняно з більш простими середовищами розробки, такими як IDLE [21]. Однією з головних переваг PyCharm є його багатофункціональність та великий набір інструментів для розробки, налаштування та навчання штучних нейронних мереж.

PyCharm має інтегровану підтримку для роботи з різними бібліотеками та фреймворками машинного навчання, такими як TensorFlow, PyTorch, scikit-learn та інші, що полегшує розробку та експериментування з ними.

Крім того, PyCharm має розширені можливості для відлагодження та профілювання коду, що дозволяє розробникам ефективніше вирішувати проблеми та оптимізувати продуктивність програм. Також, PyCharm надає більше можливостей для автоматизації рутинних завдань та підтримки колективної розробки, що робить його ідеальним вибором для великих та складних проєктів.

Проте, одним з недоліків PyCharm є його висока вартість для користувачів. У порівнянні з безкоштовними або дешевшими альтернативами,

такими як IDLE, PyCharm може бути дорожчим у використанні, особливо для студентів чи початківців. Крім того, PyCharm може мати високі системні вимоги, що може бути проблемою для користувачів з менш потужними комп'ютерами. Саме через це, PyCharm не буде використано в ході цієї роботи.

Іншим середовищем розробки може бути JupyterLab, що є інтерактивним та надає можливість працювати з кодом, текстом, графіками та іншими елементами у вигляді "ноутбуків".

Однією з переваг JupyterLab може бути гнучкість та розширюваність. Він підтримує велику кількість мов програмування, включаючи Python, R, Julia та інші, що робить його універсальним інструментом для розробки в різних областях. Крім того, JupyterLab має широкий набір розширень та плагінів, що дозволяє розробникам налаштовувати середовище під свої потреби.

Проте, JupyterLab може мати деякі недоліки порівняно з іншими середовищами розробки. Наприклад, в порівнянні з інтегрованими середовищами розробки, такими як PyCharm, JupyterLab може бути менш потужним у відлагодженні та профілюванні коду. Крім того, JupyterLab може вимагати додаткових зусиль для налаштування та управління проектами порівняно з іншими інструментами.

Нарешті, Visual Studio Code (VS Code) є потужним, безкоштовним редактором коду з відкритим вихідним кодом, який підтримує широкий спектр мов програмування та розширень, що робить його ідеальним вибором для розробки, налаштування та навчання штучних нейронних мереж [22]. Однією з головних переваг VS Code є його розширюваність і гнучкість. Завдяки великій кількості доступних розширень, можна легко додати підтримку для бібліотек машинного навчання, таких як TensorFlow, PyTorch, Keras та інші, що значно полегшує роботу з нейронними мережами.

VS Code також має інтегровану систему контролю версій, зручний інтерфейс для відлагодження коду та потужні можливості для налаштування середовища розробки відповідно до індивідуальних потреб користувача. Крім того, VS Code підтримує інтеграцію з Jupyter Notebook, що дозволяє розробникам

використовувати інтерактивні "ноутбуки" безпосередньо у редакторі. Це поєднує переваги JupyterLab з потужністю та зручністю VS Code.

У порівнянні з PyCharm, VS Code має легший інтерфейс (рис. 2.1) і менше вимог до системних ресурсів, що робить його швидшим і більш зручним для користувачів з менш потужними комп'ютерами. На відміну від IDLE, VS Code надає значно більше функціональних можливостей, таких як відлагодження, автодоповнення коду, інтеграція з системами контролю версій і підтримка численних розширень.

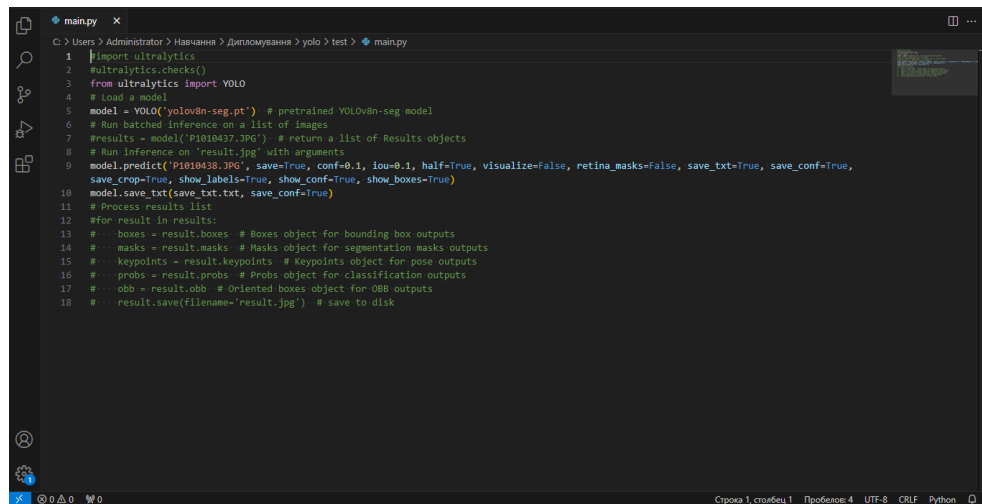


Рисунок 2.1 – Інтерфейс середовища розробки VS Code

Однак, VS Code також має свої недоліки. Хоча він є потужним і гнучким, початківцям може знадобитися час на освоєння всіх його можливостей і налаштувань. Крім того, деякі користувачі можуть відчувати, що він менш інтегрований для специфічних задач машинного навчання порівняно з спеціалізованими середовищами, такими як PyCharm Professional.

Отже, після розгляду кількох варіантів було обрано VS Code завдяки великій кількості зручних інструментів та плагінів, високому рівню професійності та орієнтованості на складні проєкти, а також потужному налагоджувачу коду, який дозволяє ефективно писати великі програми та швидко виправляти помилки.

2.3 Огляд використаних бібліотек

Бібліотека для Python – це набір модулів, які містять готовий код, що можна використовувати в програмі для певних операцій [23]. Модулі в бібліотеці містять функції, класи і змінні, які можуть бути використані для реалізації різних функціональних можливостей, не потребуючи написання коду з нуля. Це значно спрощує процес розробки програмного забезпечення, дозволяючи розробникам зосередитися на логіці додатку, а не на реалізації базових функцій.

Python має велику кількість стандартних бібліотек, які поставляються разом з інтерпретатором, а також багато сторонніх бібліотек, доступних через пакетний менеджер Python, PyPI (Python Package Index).

Використання бібліотек для Python дозволяє значно скоротити час і зусилля, необхідні для розробки програмного забезпечення, забезпечуючи доступ до передових інструментів і алгоритмів, необхідних для проведення складних аналізів і моделювання.

Бібліотека Ultralytics є інструментом для реалізації алгоритмів комп'ютерного зору, зокрема для задач об'єктного детектування та сегментації. Вона відома своєю високою продуктивністю та ефективністю, що робить її популярною серед дослідників і практиків в галузі машинного навчання. Основним продуктом бібліотеки Ultralytics є YOLOv8 (You Only Look Once), яка є однією з найпоширеніших моделей для швидкого та точного детектування об'єктів на зображеннях.

Переваги бібліотеки Ultralytics полягають у її високій швидкості та точності. YOLOv8 здатна в реальному часі обробляти зображення та відео, що є важливим для застосувань, де потрібна миттєва реакція, таких як системи спостереження, автоматизоване водіння та робототехніка. Бібліотека також пропонує простий та інтуїтивно зрозумілий інтерфейс, що полегшує її використання навіть для тих, хто має обмежений досвід у машинному навчанні. Крім того, Ultralytics підтримує багато різних форматів даних та методів

аугментації, що дозволяє легко налаштувати та розширювати моделі для різних задач і наборів даних.

Ще однією значущою перевагою є активна підтримка спільноти та регулярне оновлення бібліотеки. Розробники Ultralytics постійно додають нові функції та оптимізації, що дозволяє користувачам завжди мати доступ до найсучасніших технологій та методів. Це робить бібліотеку надійним вибором для довготривалих проєктів, оскільки користувачі можуть бути впевнені в її актуальності та сумісності з новими версіями інших бібліотек і фреймворків.

Однак бібліотека Ultralytics має і деякі недоліки. Одним з них є відносно висока вимога до апаратних ресурсів. Для досягнення максимальної продуктивності та точності, користувачам може знадобитися доступ до потужних графічних процесорів (GPU), що може бути обмежувальним фактором для тих, хто працює з обмеженими ресурсами. Крім того, хоча інтерфейс бібліотеки є простим у використанні, глибоке налаштування моделей і параметрів може вимагати значних знань і досвіду в області машинного навчання і комп'ютерного зору.

Іншим потенційним недоліком є залежність від зовнішніх бібліотек та фреймворків, таких як PyTorch. Це може створювати проблеми сумісності при використанні Ultralytics в середовищах з суворими вимогами до програмного забезпечення або на спеціалізованих апаратних платформах. Незважаючи на те, що Ultralytics активно підтримує та інтегрується з більшістю сучасних інструментів для машинного навчання, іноді виникають проблеми з сумісністю, які можуть вимагати додаткових зусиль для їх вирішення.

Загалом, бібліотека Ultralytics є потужним інструментом для реалізації задач комп'ютерного зору, пропонуючи високу продуктивність та зручність використання.

PyTorch є однією з провідних бібліотек для глибинного навчання [24], яка широко використовується в дослідницьких і промислових колах. Вона була розроблена та підтримується компанією Facebook AI Research (FAIR) і набула популярності завдяки своїй гнучкості, простоті використання та потужним

інструментам для побудови та тренування нейронних мереж. PyTorch забезпечує динамічний граф обчислень, що дозволяє змінювати його структуру на льоту, що особливо корисно для дослідницьких цілей та експериментування з новими моделями.

Однією з ключових переваг PyTorch є його інтуїтивно зрозумілий і зрозумілий API, який спрощує процес розробки і тренування моделей. PyTorch також підтримує автоматичну обробку градієнтів через механізм автоградієнтів, що значно спрощує процес оптимізації моделей. Ця функція дозволяє користувачам автоматично обчислювати градієнти для параметрів моделі під час зворотного поширення, що є критично важливим для ефективного тренування нейронних мереж. Крім того, PyTorch надає широкий набір готових модулів і функцій для побудови складних архітектур нейронних мереж, включаючи згорткові, рекурентні та трансформерні мережі.

Важливою перевагою PyTorch є його активна спільнота користувачів і розробників. широкий набір готових модулів і функцій регулярно оновлюється новими функціями та покращеннями, а користувачі можуть знайти велику кількість навчальних матеріалів, документації та прикладів коду. Це сприяє швидкому засвоєнню бібліотеки та її ефективному використанню в різних проєктах, включаючи академічні дослідження та комерційні додатки.

Проте PyTorch має і деякі недоліки. Одним з них є порівняно висока вимога до апаратних ресурсів, особливо при тренуванні великих моделей на великих наборах даних. Це може бути проблемою для користувачів з обмеженим доступом до потужних графічних процесорів (GPU) або інших високопродуктивних обчислювальних ресурсів. Крім того, хоча PyTorch забезпечує високу гнучкість і можливість для експериментів, це може призвести до складності в управлінні великими проєктами, де необхідно забезпечити підтримку стабільності та узгодженості коду.

Таким чином, PyTorch є потужним інструментом для глибокого навчання, що поєднує в собі гнучкість, простоту використання та широкий набір функцій. Незважаючи на деякі недоліки, пов'язані з вимогами до апаратних

ресурсів та управління великими проєктами, бібліотека залишається одним з провідних виборів для дослідників та інженерів у галузі машинного навчання та штучного інтелекту.

2.4 Огляд використаних наборів даних

Dataset, або набір даних, є сукупністю структурованих або неструктурованих даних, які використовуються для аналізу, моделювання та інших видів обробки. У контексті машинного навчання, набір даних відіграє критично важливу роль, оскільки саме на його основі алгоритми навчаються, тестуються та оцінюються. Зазвичай, набори даних складаються з великої кількості прикладів (зразків), кожен з яких містить певні характеристики (ознаки або атрибути) і відповідні мітки (цільові змінні), якщо йдеться про кероване навчання. Для дослідження розвитку рослин методами машинного навчання, набір даних може включати такі типи інформації, як зображення рослин, дані про погодні умови, показники ґрунту, дані про зрошення та добрива, а також генетичні дані рослин.

Якість набору даних безпосередньо впливає на точність та ефективність побудованих моделей машинного навчання. Для забезпечення високої якості, дані повинні бути повними, коректними, актуальними та різноманітними. Набір даних може бути розділений на тренувальну, валідаційну та тестову підмножини. Тренувальна підмножина використовується для навчання моделі, валідаційна – для налаштування гіперпараметрів та вибору кращої моделі, а тестова – для оцінки загальної продуктивності моделі на нових, невідомих даних.

У дослідженні розвитку рослин, дані можуть бути зібрані з різних джерел, включаючи супутникові знімки, дрони, наземні сенсори, метеорологічні станції, а також лабораторні аналізи ґрунту і рослин. Кожне джерело може надавати унікальну інформацію, яка разом утворює комплексне уявлення про стан рослин

і фактори, що впливають на їхній розвиток. Для обробки та аналізу таких даних, часто використовують різні методи попередньої обробки, включаючи нормалізацію, масштабування, обробку відсутніх значень, а також методи видалення викидів та зниження розмірності.

Розробка моделей машинного навчання для аналізу наборів даних включає використання різних алгоритмів, таких як лінійна регресія, дерева рішень, випадкові ліси, нейронні мережі та методи глибокого навчання. Ці моделі можуть допомагати у виявленні патернів, прогнозуванні врожайності, виявленні хвороб рослин на ранніх стадіях, оптимізації використання ресурсів і прийнятті агрономічних рішень. Тому правильний вибір та підготовка набору даних є фундаментальним кроком у побудові ефективних моделей машинного навчання, здатних надавати точні та корисні прогнози в контексті розвитку рослин.

У нашому випадку, ми маємо набір даних, який складається з фотографій рослин, зроблених зверху. Ці фотографії зафіксовані на двох різних часових моментах: перша зйомка відбулася на першому тижні, коли рослина ще не досягла повної зрілості (рис. 2.2, а); друга була зроблена на наступному тижні відповідно, коли рослина вже мала більший ріст і розвиток (рис. 2.2, б).



а)

б)

Рисунок 2.2 – Зображення з фотосету

На фотографіях, крім рослин, можуть бути присутні сторонні об'єкти, такі як листя або інші рослини, які не є об'єктом дослідження. Для полегшення роботи нейромережі за потреби маємо донавчити визначати потрібну, щоб модель могла зосередитися тільки на обраній рослині та її розвитку.

Таким чином, правильний вибір та підготовка набору даних є фундаментальним кроком у побудові ефективних моделей машинного навчання, здатних надавати точні та корисні прогнози в контексті розвитку рослин.

2.5 Висновки з розділу

У розділі було визначено, що для розв'язання завдань з дослідження розвитку рослин методом машинного навчання найбільш доцільним є використання мови програмування Python. Це обумовлено її широкими можливостями та великою кількістю інструментів для роботи зі штучними нейронними мережами. Було обґрунтовано вибір середовища розробки, яким став VSCode, завдяки його зручності, інтерактивності та підтримці багатьох корисних функцій для програмування.

Аналіз наданих зображень рослин показав, що правильна підготовка набору даних є критично важливою для точності моделей машинного навчання. Було виявлено, що на фотографіях можуть бути сторонні об'єкти, тому для полегшення роботи нейромережі за потреби маємо донавчити та визначати потрібну, щоб модель могла зосередитися тільки на обраній рослині та її розвитку.

Таким чином, проведений аналіз продемонстрував важливість ретельного вибору інструментів та методів для ефективного застосування технологій машинного навчання у дослідженні розвитку рослин.

3 ВИКОРИСТАННЯ НЕЙРОМЕРЕЖІ ДЛЯ РОБОТИ З ЗЕЛЕНОЮ ЧАСТИНОЮ РОСЛИНИ

3.1 Вибір моделі YOLOv8 для визначення зеленої частини рослини

Для початку маємо встановити YOLOv8 на свій персональний комп'ютер із встановленою на ньому операційною системою Windows 10. Для цього нам потрібно мати встановлений Python. Заходимо на офіційний сайт та завантажуюмо дистрибутив встановлення під Windows [25].

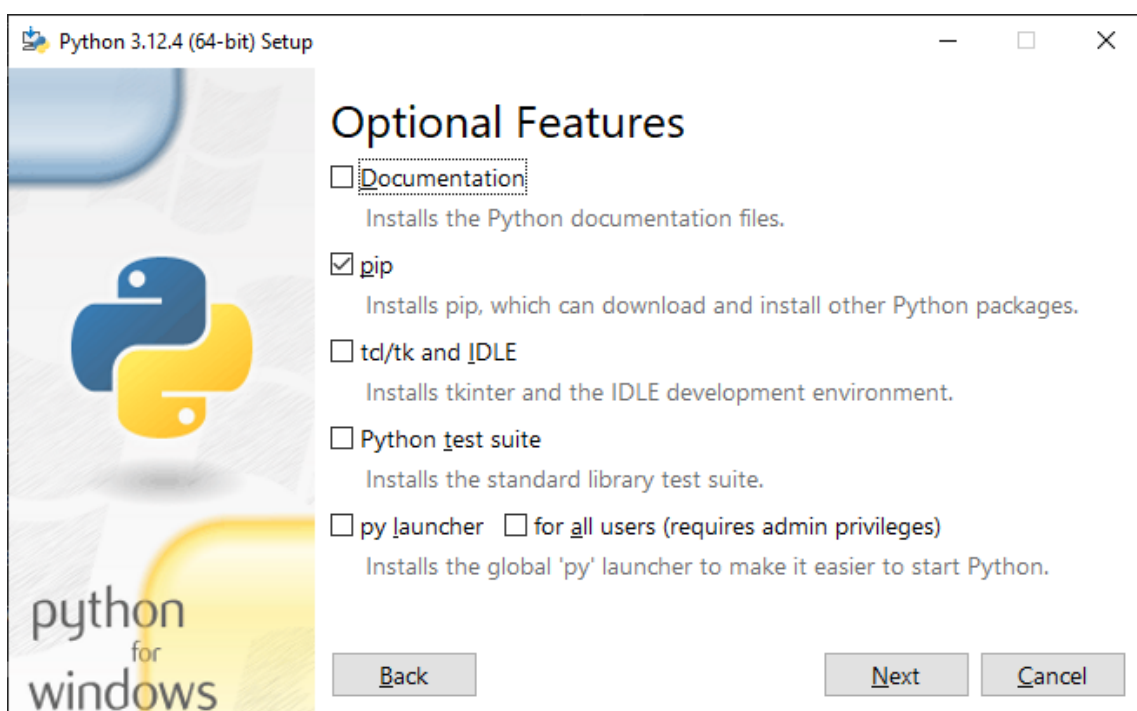


Рисунок 3.1 – Процес встановлення Python

Виконуємо вибіркове встановлення (рис. 3.1) завантаженого дистрибутиву, в нашому випадку це Python версії 3.12.3.

За встановленої мови програмування, маємо можливість скористатись YOLOv8, для чого треба інстальювати пакет Ultralytics. У відкритому терміналі Windows вводимо наступну команду: `pip install ultralytics` [26].

Після завершення завантаження потрібних файлів, яке може тривати достатньо довго, в каталозі AppData поточного користувача Windows будуть

розміщені папки з файлами пакету. Також для роботи YOLOv8 нам потрібно буде у коді нашої програми один раз написати `ultralitics.checks()`, що перевірить залежності, та докачає залишкові файли. Тепер можемо користуватись неймерережею на персональному комп'ютері.

Напишемо базову програму для розпізнавання зеленої частини рослини. Обираємо каталог для нашого проєкту, у ньому розміщуємо фотосет (рис. 3.2) та, безпосередньо, наш файл програми.



Рисунок 3.2 – Фотографія рослини

Написавши код (рис. 3.3), розглянемо структуру:

`import ultralitics` - імпортуємо бібліотеку `ultralitics`

`ultralitics.checks()` - перевіряємо структуру та докачуємо файли

`from ultralitics import YOLO` - імпортуємо детектор об'єктів YOLO

`model = YOLO("yolov8n-seg.pt")` - змінна `model` міститиме ініціалізацію об'єкта класу YOLO з параметром `"yolov8n-seg.pt"`, який вказує на використання моделі YOLO версії 8n з підтримкою сегментації

`results = model.predict(source="P1010437.JPG", conf=0.01, iou=0.01, max_det=300, imgsz=640, retina_masks=True, classes=[58], save=True)` - змінна `results` міститиме результати передбачення моделі YOLO на зазначеному зображенні з врахуванням усіх вказаних параметрів:

- `source` - визначає джерело даних для висновку. Це, в даному випадку, шлях до зображення;
- `conf` - встановлює мінімальний поріг достовірності для виявлення. Об'єкти, виявлені з упевненістю нижче цього порогу, не враховуватимуться. Налаштування цього значення може допомогти зменшити помилкові спрацьовування;
- `iou` - порогове значення перетину через об'єднання (IoU) для немаксимального придушення (NMS). Нижчі значення призводять до меншої кількості виявлень завдяки виключенню блоків, що перекриваються, що корисно для зменшення дублікатів;
- `max_det` - максимально дозволена кількість виявлень для одного зображення. Обмежує загальну кількість об'єктів, які модель може виявити в одному висновку, запобігаючи надмірним виводам;
- `imgsz` - визначає розмір зображення для висновку. Може бути одним цілим числом 640 для зміни розміру квадрата або кортежем (висота, ширина). Правильний розмір може підвищити точність виявлення та швидкість обробки;
- `retina_masks` - використовує маски сегментації високої роздільної здатності, якщо вони доступні в моделі. Це може підвищити якість маски для завдань сегментації, забезпечуючи дрібніші деталі.
- `classes` - фільтрує прогнози до набору ідентифікаторів класів. В нашому випадку буде повернено лише клас з ідентифікатором 58, що визначає на фото рослину;
- `save` - зберігає анотоване зображення у файл.

```
import ultralytics
ultralytics.checks()
from ultralytics import YOLO
model = YOLO("yolov8n-seg.pt")
results = model.predict(source="P1010437.JPG", conf=0.01, iou=0.01, max_det=300, imgsz=640,
retina_masks=True, classes=[58], save=True)
```

Рисунок 3.3 – Код програми для сегментації моделлю YOLOv8

Після виконання програми, у каталозі проєкту, з'явиться анотоване зображення за шляхом "\runs\segment\predict" (рис. 3.4).



Рисунок 3.4 – Анотована фотографія рослини

Як ми можемо бачити, неймережа змогла визначити рослину, обвівши її майже по контуру, але нам потрібно покращити результат. Одним з вагомих варіантів може бути зміна моделі.

YOLOv8 має п'ять моделей сегментації [27]: nano, small, medium, large та extra large. Ми перевірили розпізнавання на моделі nano, тому маємо перевірити

інші варіанти. Для цього треба змінювати модель у рядку `model = YOLO("yolov8n-seg.pt")` на відповідну.

Найкращим варіантом з усіх виявилася модель `medium` (рис. 3.5).



Рисунок 3.5 – Анотована фотографія рослини моделлю `medium`

Як ми бачимо, модель впоралася краще, але маємо ще покращити результат. В результаті рядок у коді виглядає наступним чином: `model = YOLO("yolov8m-seg.pt")`. Для такого варіанту використаємо донавчання та тонке настроювання (Fine-tuning) моделі.

3.2 Fine-tuning моделі на власному наборі даних

Fine-tuning моделі будемо проводити в два етапи: спочатку розмітимо датасет за допомогою Roboflow [28], після чого вже безпосередньо навчимо модель на фотосеті.

Для виконання першого етапу, заходимо на сайт Roboflow, реєструємось, та створюємо проєкт. Завантажуємо зображення фотосету та розмічуємо (рис. 3.6).

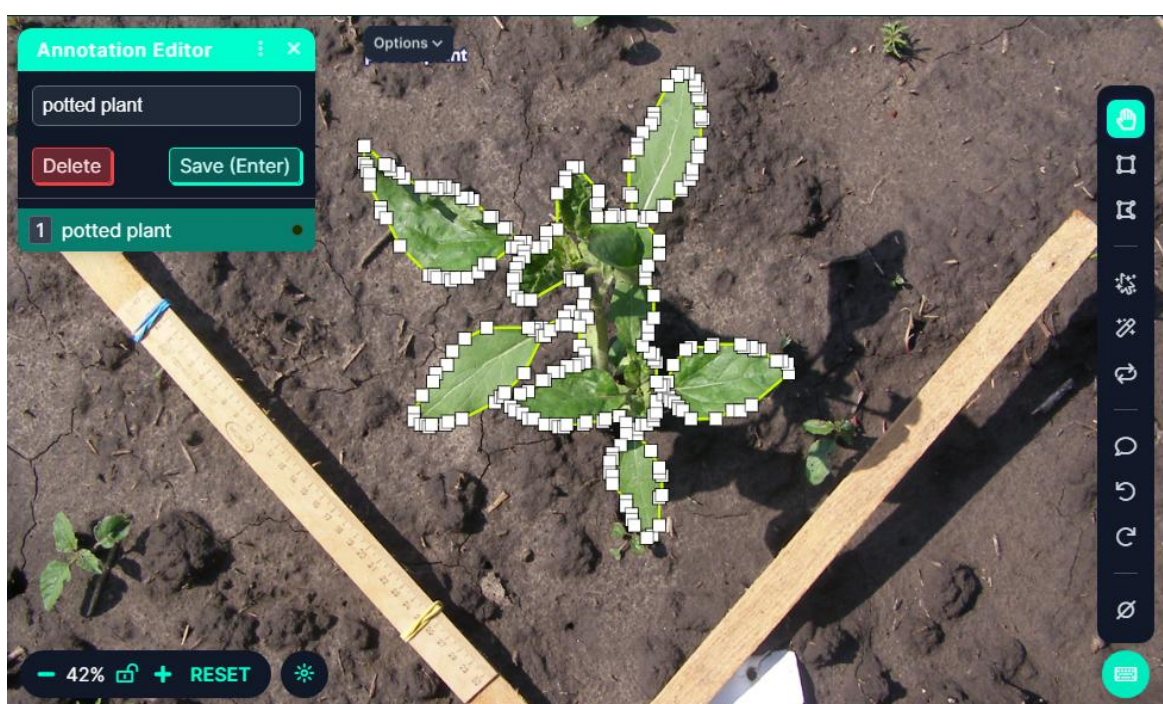


Рисунок 3.6 – Розмічена у Roboflow фотографія рослини

Після розмітки датасету, розподіляємо фотографії між тренувальною, валідаційною та тестовою підмножинами. Для нашої ситуації дублюємо фотографії у перших двох частинах. Також потребуємо завантаження розміченого датасету, для чого на сайті зберігаємо його як готову версію, після чого з'являється можливість експорту. Експортуємо у форматі YOLOv8 та обираємо радіокнопку "download zip to computer" (рис. 3.7).

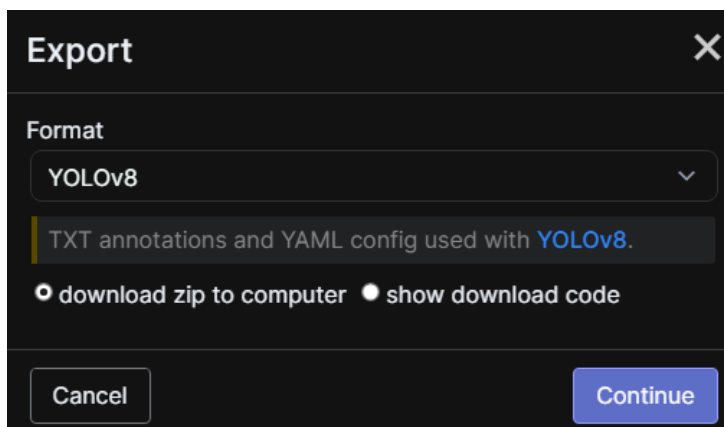


Рисунок 3.7 – Експорт розміченого у Roboflow набору даних

Після завантаження архіву, розпаковуємо вміст до каталогу нашого проєкту, попередньо створивши папку для датасету.

Датасет має таку структуру каталогів: папки `valid` та `train`, що містять у собі по два каталоги `labels` та `images`, також у корені лежить цікавлячий нас файл `data.yaml`. У каталозі `labels` містяться мітки у вигляді текстових файлів з назвою відповідної фотографії, а у папці `images` – зображення фотосету. Файл `data.yaml` має шлях до проєкту на комп'ютері, а також до двох вищеназваних підмножин.

Коли завершено перший етап, переходимо до наступного, а саме, до навчання моделі на нашому розміченому фотосеті. Для цього, для зручності, створимо окрему програму наступним чином (рис. 3.8).

```
from ultralytics import YOLO
model = YOLO("yolov8m-seg.pt")
results = model.train(data="datasets\plant.v1i.yolov8\data.yaml", epochs=100, patience=0, imgsz=640,
save=True)
```

Рисунок 3.8 – Код програми для навчання моделі YOLOv8 на своєму датасеті

Розглянувши структуру, можемо бачити:

- `import ultralytics` - імпорт бібліотеки `ultralytics`;
- `model = YOLO("yolov8m-seg.pt")` - створення моделі YOLOv8 для завантаження попередньо навчених ваг моделі з файлу `"yolov8m-seg.pt"`;

- `results = model.train(data="datasets\plant.v1i.yolov8\data.yaml", epochs=100, patience=0, imgsz=640, save=True)` - рядок для тренування моделі YOLOv8 на даних, які описані у файлі конфігурації з наступними параметрами:

- `data` - шлях до файлу конфігурації набору даних;
- `epochs` - загальна кількість епох навчання. Кожна епоха представляє повний прохід по всьому набору даних;
- `patience` - кількість періодів очікування без покращення показників перевірки перед достроковим припиненням навчання;
- `imgsz` - цільовий розмір зображення для навчання. Усі зображення змінюються до цього розміру перед подачею в модель. Впливає на точність моделі та складність обчислень;
- `save` - дозволяє зберігати контрольні точки навчання та остаточну вагу моделі.

Запустивши програму, отримаємо результат за шляхом `"\runs\segment\train"`. Всередині папка містить каталог `weights` та файли `results.csv` і `args.yaml`. У каталозі `weights` містяться збережені ваги навченої моделі - в нашому випадку це `best.pt` та `last.pt`. Файл `results.csv` має результати метрик якості моделі після тренування. Файл `args.yaml` містить параметри, які використовувалися під час тренування моделі.

Ваги – це потрібні нам навчені моделі. В нашому випадку скористаймося кращою. Для цього повертаємось до файлу програми сегментації моделлю YOLOv8 та внесемо потрібні зміни (рис. 3.9): у рядку `"model = YOLO("runs/segment/train/weights/best.pt")"` визначаємо шлях до нової моделі; а у рядку `"results = model.predict(source="P1010437.JPG", conf=0.01, iou=0.01, max_det=300, imgsz=640, retina_masks=True, save=True)"` прибираємо параметр `classes=[58]`, що вказував на клас з моделі YOLOv8.

```
from ultralytics import YOLO
model = YOLO("runs/segment/train/weights/best.pt")
results = model.predict(source="P1010437.JPG", conf=0.01, iou=0.01, max_det=300, imgsz=640,
retina_masks=True, save=True)
```

Рисунок 3.9 – Код програми для сегментації навченою моделлю

Після виконання програми, у каталозі проєкту з'явиться зображення з анотаціями, доступне за шляхом "`\runs\segment\predict2`" (рис. 3.10).



Рисунок 3.10 – Анотована фотографія рослини навченою моделлю

Для кращого візуального сприйняття розмістимо фотографії поряд та порівняємо зміни (рис. 3.11).



Рисунок 3.11 – Порівняння анотованих фотографій рослини YOLOv8 та навченою моделями

Як можна помітити, результати значно поліпшилися, що демонструє можливість застосування нової моделі для розрахунку площі зеленої частини рослини та визначення її зростання.

Після проробленої роботи, приступимо до визначення площі у пікселях. Для цього нам знадобиться бібліотека PyTorch. Для обчислення потребуємо ще раз внести зміни до нашої програми для сегментації навченою моделлю (рис. 3.12).

```
from ultralytics import YOLO
import torch
model = YOLO("runs/segment/train/weights/best.pt")
results = model.predict(source="P1010437.JPG", conf=0.01, iou=0.01, max_det=300, imgsz=640,
retina_masks=True, save=True)
for i in results:
    print(torch.sum(i.masks.data[0]))
```

Рисунок 3.12 – Код програми для сегментації навченою моделлю з обчисленням площі рослини у пікселях

Розглянемо додані рядки:

- import torch - імпортуємо бібліотеку torch;
- for i in results: - цикл перебирає кожен елемент i в results;
- print(torch.sum(i.masks.data[0])) - вивід та обчислення площі за допомогою знаходження суми пікселів виділеної рослини

Після запуску програми, в кінці отриманого виводу в консолі, маємо наступний рядок: "tensor(270345.)". Цифри в дужках і є наша площа рослини у пікселях.

Для визначення площі рослин на інших фото, потрібно змінити значення аргументу source="P1010437.JPG" на потрібний шлях до обраного фото у рядку з результатами передбачення моделі. Відповідно, після запуску програми отримаємо результат у вигляді нового каталогу з іншим анотованим фото за шляхом "\runs\segment" та на кінці виводу в консолі побачимо рядок з площею рослини.

3.3 Конвертування вихідних даних нейронної мережі

Для більшої зручності використання отриманих результатів у вигляді площі рослини можемо перетворити їх з площі у пікселях на площу у квадратних сантиметрах. Цього можна досягти, скориставшись лінійкою, що фігурує на кожному фото. Визначивши кількість пікселів, що відповідають міліметру, є можливість конвертувати у потрібний нам формат.

Розглянемо цей варіант на одній з фотографій. Для цього відкриємо фото у звичайному графічному редакторі. У даному випадку використаємо Microsoft Paint. Збільшивши зображення до появи пікселів, обираємо інструмент "Олівець" та відмітимо чорним кольором міліметр на фото (рис. 3.13).



Рисунок 3.13 – Довжина міліметра в пікселях у Microsoft Paint

Отримавши інформацію про довжину міліметра з обраного фото, на якому вона дорівнює два пікселя, скористаймося цим знанням для перетворення площі.

Спочатку знайдемо площу пікселя. Для цього скористаємось формулою для обчислення площі квадрата S за його діагоналлю d (3.1).

$$S = \frac{d^2}{2}, \quad (3.1)$$

Діагональ пікселя дорівнює 0,05 см, а це означає, що його площа дорівнює 0,00125 см².

Маємо площу, яка дорівнює 270345 пікс., множимо на 0,00125 см², та в результаті отримуємо приблизно 338 см².

Для отримання аналогічних даних з інших фото буде потрібно зробити такі ж перетворення.

Тепер маємо змогу користуватись даними площі рослини у квадратних сантиметрах для подальших операцій. Прикладом застосування визначеної площі рослини є визначення її дорослості: площа зеленої частини, сфотографована на другому тижні буде більшою. Також можна визначити, на скільки збільшилася площа, обчисливши їх різницю.

Щодо можливих покращень точності обчислень площі зеленої частини: наприклад, навчання моделі розпізнаванню лінійки, може допомогти з обчисленням її довжини у пікселях, що в свою чергу можна порівнювати з її реальною довжиною у сантиметрах для уточнення даних.

3.4 Висновки з розділу

Модель YOLOv8 була успішно налаштована та донавчена на власному наборі даних. Процес включав використання платформи Roboflow для розмітки даних і подальше навчання моделі на цих даних. Налаштування включали розподіл даних на тренувальну та валідаційну підмножини, що дозволило оптимізувати процес навчання моделі та покращити її точність.

Після навчання модель показала значне покращення у якості анотацій зображень, що свідчить про ефективність підходу та можливість застосування цієї моделі для практичних завдань у галузі агрономії.

Розроблена модель може бути використана для автоматизації процесів моніторингу росту рослин, що є важливим для агрономічних досліджень. Отримані результати можуть бути застосовані для подальших досліджень та удосконалення методів аналізу рослинних структур. Проведена робота має значний потенціал для практичного використання в аграрній сфері та може слугувати основою для подальших наукових досліджень.

ВИСНОВКИ

В ході проведеного дослідження було досягнуто значних результатів у застосуванні методів машинного навчання для аналізу розвитку рослин.

Основними результатами дослідження є:

1) Аналіз сучасних методів застосування штучного інтелекту в аграрній сфері:

Було досліджено сучасні методи використання штучного інтелекту та комп'ютерного зору в сільському господарстві. Особливу увагу приділено нейронним мережам, їх архітектурі та впливу на різні аспекти сільськогосподарських процесів. Використання таких технологій дозволяє значно підвищити ефективність і точність аналізу стану рослин, виявлення хвороб та пошкоджень, а також оптимізацію процесів вирощування та збору врожаю.

2) Налаштування моделі YOLOv8:

Модель YOLOv8 була успішно налаштована та донавчена на спеціально підготовленому наборі даних. Процес включав використання платформи Roboflow для розмітки даних і подальше навчання моделі. Після навчання модель показала значне покращення у якості анотацій зображень, що свідчить про ефективність підходу та можливість застосування цієї моделі для практичних завдань у галузі агрономії.

3) Підготовка даних та обробка зображень:

Аналіз наданих зображень рослин показав, що правильна підготовка набору даних є критично важливою для точності моделей машинного навчання. Виявлено, що на фотографіях можуть зустрічатися зайві об'єкти, тому необхідно провести додаткове навчання нейронної мережі для точного визначення обраної рослини та її розвитку, що сприятиме зосередженню моделі на цільовому об'єкті.

Мета роботи полягала у дослідженні можливості використання моделей машинного навчання для визначення площі зеленої частини рослини. Всі поставлені завдання були успішно виконані. Було проведено аналіз сучасних методів застосування машинного навчання в аграрній сфері, протестовано моделі на реальних даних та оцінено їх точність і ефективність.

Об'єктом дослідження було використання алгоритмів машинного навчання в агропромисловості, а предметом дослідження – розвиток рослин методами машинного навчання. Основні властивості об'єкта дослідження включають:

- Здатність нейронних мереж адаптуватися до нових даних і видавати передбачення без необхідності повторного навчання.
- Можливість використання великих даних для створення точних моделей прогнозування врожайності та аналізу стану рослин.
- Висока ефективність автоматизації процесів моніторингу стану рослин та їхнього розвитку за допомогою комп'ютерного зору.

Під час виконання дипломної роботи було застосовано методику налаштування та донавчання моделі YOLOv8 для аналізу розвитку рослин. Було використано платформу Roboflow для розмітки даних та проведено навчання моделі на підготовлених даних. Отримані результати показали високу точність та ефективність моделі для анотацій зображень.

Можливі сфери застосування отриманих результатів включають в себе:

- Автоматизацію процесів моніторингу стану рослин у сільському господарстві.
- Використання моделей для визначення оптимального часу збору врожаю.
- Прогнозування врожайності та аналіз впливу різних факторів на ріст рослин за допомогою великих даних та штучного інтелекту.

Результати роботи відповідають сучасним світовим тенденціям у вирішенні поставленої задачі. Використання машинного навчання та нейронних

мереж у сільському господарстві активно розвивається у всьому світі, і отримані результати свідчать про значний потенціал таких підходів.

Застосування розроблених моделей та використаних методик має велике значення для агропромислового комплексу. Впровадження таких технологій дозволить підвищити ефективність сільськогосподарських процесів, зменшити витрати та покращити якість продукції. Це має значний потенціал для розвитку аграрної сфери та підвищення конкурентоспроможності фермерських господарств.

Подальші дослідження можуть бути спрямовані на:

- Удосконалення моделей машинного навчання для більш точного аналізу стану рослин.
- Розширення наборів даних для навчання моделей з урахуванням різноманітних факторів впливу.
- Застосування розроблених підходів для інших аспектів аграрного виробництва, таких як прогнозування погодних умов та оптимізація використання ресурсів.

Отже, проведена робота має значний науковий і практичний потенціал, відповідає сучасним тенденціям та потребам агропромислового комплексу, і може стати основою для подальших досліджень у цій галузі.

ПЕРЕЛІК ПОСИЛАНЬ

1. Maharachchikumbura S. Occurrence of leaf blotch disease [Електронний ресурс] / S. Maharachchikumbura, N. Adikaram // Ceylon Journal of Science (Biological Sciences). — 2009. — 38(2). — Р. 51-56. — Режим доступу: https://www.researchgate.net/figure/Stages-of-leaf-blotch-disease-in-Ficus-religiosa-at-a-early-stage-and-b-later-stage_fig1_237075355 (дата звернення: 07.06.2024). — Назва з екрана.
2. Injury in plants [Електронний ресурс]. — Режим доступу: https://en.wikipedia.org/wiki/Injury_in_plants (дата звернення: 07.06.2024). — Назва з екрана.
3. Carolus Linnaeu [Електронний ресурс]. — Режим доступу: <https://www.britannica.com/biography/Carolus-Linnaeus> (дата звернення: 07.06.2024). — Назва з екрана.
4. Liebig, J. Instructions for the Chemical Analysis of Organic Bodies / J. von Liebig // Scientific and Medical Knowledge Production, 1796-1918. — Routledge, 2023. — 15 p.
5. Великі дані [Електронний ресурс]. — Режим доступу: https://uk.wikipedia.org/wiki/Великі_дані (дата звернення 07.06.2024). — Назва з екрана.
6. What is artificial intelligence (AI)? [Електронний ресурс]. — Режим доступу: <https://www.techtarget.com/searchenterpriseai/definition/AI-Artificial-Intelligence> (дата звернення: 07.06.2024). — Назва з екрана.
7. Using Deep Learning for Image-Based Plant Disease Detection [Електронний ресурс]. — Режим доступу: <https://www.letsnurture.com/blog/using-deep-learning-for-image-based-plant-disease-detection.html> (дата звернення: 07.06.2024). — Назва з екрана.

8. Artificial Neural Networks and its Applications [Электронный ресурс]. — Режим доступа: <https://www.geeksforgeeks.org/artificial-neural-networks-and-its-applications/> (дата звернения: 07.06.2024). — Назва з екрана.
9. What Is a Neural Network? [Электронный ресурс]. — Режим доступа: <https://www.investopedia.com/terms/n/neuralnetwork.asp> (дата звернения: 07.06.2024). — Назва з екрана.
10. What is a neural network? [Электронный ресурс]. — Режим доступа: <https://www.elastic.co/what-is/neural-network> (дата звернения: 07.06.2024). — Назва з екрана.
11. Artificial Neural Network [Электронный ресурс]. — Режим доступа: <https://www.wallstreetmojo.com/artificial-neural-network/> (дата звернения: 07.06.2024). — Назва з екрана.
12. YOLOv8: A Comprehensive Framework for Object Detection» [Электронный ресурс]. — Режим доступа: <https://medium.com/@beyzaakyildiz/what-is-yolov8-how-to-use-it-b3807d13c5ce> (дата звернения: 07.06.2024). — Назва з екрана.
13. What are convolutional neural networks?» [Электронный ресурс]. — Режим доступа: <https://cointelegraph.com/explained/what-are-convolutional-neural-networks> (дата звернения: 07.06.2024). — Назва з екрана.
14. What is Mask R-CNN? The Ultimate Guide [Электронный ресурс]. — Режим доступа: <https://blog.roboflow.com/mask-rcnn/> (дата звернения: 07.06.2024). — Назва з екрана.
15. Introduction to C++ Libraries [Электронный ресурс]. — Режим доступа: <https://gymbeam.ua/ua/fat-caliper-gymbeam.html> (дата звернения: 07.06.2024). — Назва з екрана.
16. C++ [Электронный ресурс]. — Режим доступа: <https://en.wikipedia.org/wiki/C++> (дата звернения: 08.06.2024). — Назва з екрана.
17. Java (programming language) [Электронный ресурс]. — Режим доступа: [https://en.wikipedia.org/wiki/Java_\(programming_language\)](https://en.wikipedia.org/wiki/Java_(programming_language)) (дата звернения: 08.05.2023). — Назва з екрана.

18. JavaScript [Электронный ресурс]. — Режим доступа: <https://en.wikipedia.org/wiki/JavaScript> (дата звернення: 08.06.2024). — Назва з екрана.

19. Python [Электронный ресурс]. — Режим доступа: [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language)) (дата звернення: 08.05.2023). — Назва з екрана.

20. IDLE [Электронный ресурс]. — Режим доступа: <https://en.wikipedia.org/wiki/IDLE> (дата звернення: 08.06.2024). — Назва з екрана.

21. PyCharm [Электронный ресурс]. — Режим доступа: <https://en.wikipedia.org/wiki/PyCharm> (дата звернення: 08.06.2024). — Назва з екрана.

22. Visual Studio Code [Электронный ресурс]. — Режим доступа: https://en.wikipedia.org/wiki/Visual_Studio_Code (дата звернення: 08.06.2024). — Назва з екрана.

23. Libraries in Python [Электронный ресурс]. — Режим доступа: <https://www.geeksforgeeks.org/libraries-in-python/> (дата звернення: 08.06.2024). — Назва з екрана.

24. PyTorch»[Электронный ресурс]. — Режим доступа: <https://pytorch.org/> (дата звернення: 08.06.2024). — Назва з екрана.

25. Download Python [Электронный ресурс]. — Режим доступа: <https://www.python.org/downloads/> (дата звернення: 09.06.2024). — Назва з екрана.

26. Quickstart [Электронный ресурс]. — Режим доступа: <https://docs.ultralytics.com/quickstart/> (дата звернення: 09.06.2024). — Назва з екрана.

27. Segment [Электронный ресурс]. — Режим доступа: <https://docs.ultralytics.com/tasks/segment/#models> (дата звернення: 09.06.2024). — Назва з екрана.

28. Workspace Home [Электронный ресурс]. — Режим доступа: <https://app.roboflow.com/> (дата звернення: 09.06.2024). — Назва з екрана.