

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ
СІКОРСЬКОГО»**

Навчально-науковий інститут атомної та теплової енергетики
Кафедра цифрових технологій в енергетиці

"На правах рукопису"
УДК _____

«До захисту допущено»
Завідувач кафедри
_____ Наталія АУШЕВА
“ ____ ” _____ 2022р.

Магістерська дисертація

зі спеціальності - 122 Комп'ютерні науки
за освітньо-професійною програмою магістерської підготовки -
Комп'ютерний моніторинг та геометричне моделювання процесів і систем

на тему «Мобільний додаток для визначення хвороб рослин на основі
методів машинного навчання»

Виконала: студентка 2 курсу, групи ТР-11мп
Смірнова Олександра Дмитрівна
(прізвище, ім'я, по батькові)

(підпис)

Науковий керівник д.т.н., доцент Шушура О.М.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Рецензент д.т.н., професор Гаврилко Є.В.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних
посилань.

Студент _____
(підпис)

Київ - 2022

Національний технічний університет України
“Київський політехнічний інститут ім. Ігоря Сікорського”

Навчально-науковий інститут атомної та теплової енергетики

Кафедра цифрових технологій в енергетиці

Рівень вищої освіти другий, магістерський

За освітньою програмою "Комп'ютерний моніторинг та геометричне моделювання
процесів та систем"

Спеціальності 122 Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри

Наталія АУШЕВА
(підпис)

« » 2022р.

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ ДИСЕРТАЦІЮ СТУДЕНТУ

Смірновій Олександрі Дмитрівні

(прізвище, ім'я, по батькові)

1. Тема дисертації «Мобільний додаток для визначення хвороб рослин на основі методів машинного навчання»

Науковий керівник Шушура Олексій Миколайович, д. т. н., доцент

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

затверджені наказом по університету від “07” листопада 2022 року №4067-с

2. Строк подання студентом дисертації 7 грудня 2022 року

3. Вихідні дані до роботи мобільний додаток для ідентифікації хвороб рослин за фото

4. Перелік питань, які потрібно розробити дослідити методи класифікації зображень; дослідити типи згоркових нейронних мереж; розробити мобільний додаток; налаштувати комунікацію мобільного додатка та сервісу моделі машинного навчання

5. Орієнтований перелік ілюстративного матеріалу приклади зображень з обраного набору даних; графіки метрик тренування та валідації моделей машинного навчання; діаграма класів клієнтської частини; діаграма класів серверної частини додатку; екрани роботи мобільного додатку.

6. Орієнтований перелік публікацій 1 публікація «Мобільний додаток для визначення хвороб рослин на основі методів машинного навчання»

7. Дата видачі завдання «_20_» жовтня 2021р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання магістерської дисертації	строки виконання етапів магістерської дисертації	Примітка
1	Грунтовне ознайомлення з предметною областю, визначення структури магістерської дисертації, вивчення літератури	20 червня 2022 року	
2	Робота над першим розділом магістерської дисертації	1 вересня 2022 року	
3	Проведення наукового дослідження; робота над другим розділом магістерської дисертації	15 вересня 2022 року	
4	Проведення математичних експериментів та систематизація результатів; робота над третім розділом дисертаційної роботи	26 жовтня 2022 року	
4	Робота над четвертим розділом магістерської дисертації; підготовка тез за результатами наукового дослідження; розробка програмного забезпечення	8 листопада 2022 року	
5	Робота над п'ятим розділом магістерської дисертації; завершення роботи над основною частиною магістерської дисертації	15 листопада 2022 року	
6	Оформлення текстової і графічної частин магістерської дисертації	22 листопада 2022 року	

Студент

_____ (підпис)

Смірнова О.Д.

_____ (прізвище та ініціали)

Науковий керівник

_____ (підпис)

Шушура О.М.

_____ (прізвище та ініціали)

РЕФЕРАТ

Магістерська дисертація: 96 ст., 27 рис., 32 табл., 23 джерела, 1 додаток.

Актуальність. Проблема визначення хвороб рослин зустрічається доволі часто, наприклад при домашньому вирощуванні квітів, овочів чи функціонуванні невеликих фермерських господарств. Наразі хвороби рослин здебільшого визначаються лише людиною на основі її власного досвіду після особистого візуального огляду. Такий підхід вимагає постійного самовдосконалення, при появі невідомої хвороби не є швидким та часто приводить до помилок у малодосвідчених спеціалістів. Тому актуальною є автоматизація цього процесу на основі використання легкодоступних інформаційних технологій, наприклад мобільного додатку.

Одним з шляхів автоматизації вирішення задач, в яких присутня візуальна класифікація, є використання комп'ютерного зору та моделей машинного навчання. Саме такий підхід був використаний в даній магістерській дисертації.

Перевагою використання моделей машинного навчання та комп'ютерного зору є швидкість, достатньо висока точність, стабільність, та доступність для користувача в будь який момент часу.

Метою дослідження є автоматизація процесу визначення розповсюджених хвороб рослин шляхом розробки програмного продукту, що складається з мобільного додатку для смартфонів, серверної частини та сервісу, що є обгорткою над моделлю машинного навчання.

Для досягнення зазначеної мети треба виконати **наступні завдання**:

- провести аналіз сучасних підходів та методів машинного навчання для класифікації об'єктів на фотознімках;
- проаналізувати проблему обмеженості набору даних;
- створити модель машинного навчання для вирішення проблеми класифікації хвороб рослин по зображенню хворого листя;
- сформулювати перелік функціональних вимог системи;
- спроектувати архітектуру програмного продукту;

— розробити програмне забезпечення системи і провести його тестування на валідаційній вибірці зображень.

Об’єктом дослідження є процес визначення хвороб рослин, які виражаються на листі та можуть бути ідентифіковані візуально.

Предметом дослідження є методи та інформаційні системи для розпізнавання хвороб рослин на фотознімках на основі машинного навчання.

Методи дослідження: машинне навчання для класифікації об’єктів на фотознімках.

Практичне значення отриманих результатів:

Практичне значення дисертаційної роботи полягає в тому, що було розроблено програмний продукт, що складається з мобільного додатку, серверу та сервісу моделі машинного навчання, який дозволяє в режимі реального часу визначати хвороби рослин за зображенням ураженого листа.

Апробація: результати роботи доповідалися на III Всеукраїнській науково-практичній інтернет-конференції молодих вчених та здобувачів вищої освіти “СУЧАСНА МОЛОДЬ В СВІТІ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ”.

Публікації: результати дослідження класифікаційних моделей були опубліковані в матеріалах III Всеукраїнської науково-практичної інтернет-конференції молодих вчених та здобувачів вищої освіти “СУЧАСНА МОЛОДЬ В СВІТІ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ”.

Ключові слова: КОМП’ЮТЕРНИЙ ЗІР, МАШИННЕ НАВЧАННЯ, НЕЙРОННІ МЕРЕЖІ, КЛАСИФІКАЦІЯ ОБ’ЄКТІВ, МОБІЛЬНИЙ ДОДАТОК, ІДЕНТИФІКАЦІЯ ХВОРОБ РОСЛИН.

ABSTRACT

Topic: “Mobile application for the determination of plant diseases based on machine learning”

Master’s degree thesis: 96 pages, 27 figures, 32 tables, 23 references, 1 attachment.

Topic relevance. The problem of determining plant diseases occurs quite often, for example, when growing flowers and vegetables at home or operating small farms. Currently, plant diseases are mostly determined only by a person based on his own experience after a personal visual inspection. This approach requires constant self-improvement, when an unknown disease appears, it is not fast and often leads to mistakes by inexperienced specialists. Therefore, the automation of this process based on the use of easily accessible information technologies, such as a mobile application, is relevant. One of the ways to automate the solution of problems that involve visual classification is the use of computer vision and machine learning models. This is the approach that was used in this master's thesis. The advantage of using machine learning and computer vision models is speed, sufficiently high accuracy, stability, and availability to the user at any time.

The aim of the research is the automation of the process of identifying common plant diseases by developing a software product consisting of a mobile application for smartphones, a server part and a service that is a wrapper over a machine learning model.

To achieve the specified goal, the **following tasks** must be completed:

- analyze modern approaches and methods of machine learning for the classification of objects in photographs;
- to analyze the problem of limitation of the data set;
- create a machine learning model to solve the problem of classification of plant diseases based on the image of diseased leaves;
- create a list of functional requirements of the system;
- design the architecture of the software product;
- develop the system software and test it on a validation sample of images.

The object of research is the process of identifying plant diseases that are expressed on the leaves and can be identified visually.

The subject of research there are methods and information systems for recognizing plant diseases in photographs based on machine learning.

Methods of research: machine learning for object classification in photographs.

The practical value of the obtained results:

The practical significance of the dissertation is that a software product was developed, consisting of a mobile application, a server, and a machine learning model service, which allows for real-time detection of plant diseases based on the image of affected leaves.

Approbation: the results of the work were reported at the 3rd All-Ukrainian scientific and practical internet conference of young scientists and students of higher education "TODAY'S YOUTH IN THE WORLD OF INFORMATION TECHNOLOGIES".

Publications: the results of the study of classification models were published in the materials of the 3rd All-Ukrainian Scientific and Practical Internet Conference of Young Scientists and Higher Education Graduates "TODAY'S YOUTH IN THE WORLD OF INFORMATION TECHNOLOGIES".

Keywords: COMPUTER VISION, MACHINE LEARNING, NEURAL NETWORKS, OBJECT CLASSIFICATION, MOBILE APPLICATION, PLANT DISEASE IDENTIFICATION.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	11
ВСТУП.....	12
1 ПОСТАНОВКА ЗАДАЧІ АВТОМАТИЗАЦІЇ ВИЗНАЧЕННЯ ХВОРОБ РОСЛИН	14
1.1 Загальне визначення проблематики захворювання рослин	14
1.2 Змістовний опис і аналіз предметної області	16
1.3 Згорткові нейронні мережі	16
1.4 Нейронна мережа EfficientNet	17
1.5 Нейронна мережа MobileNet.....	19
1.6 Аналіз успішних ІТ-проектів	19
1.7 Аналіз існуючих технічних рішень	22
1.8 Аналіз відомих програмних продуктів	23
1.9 Висновки до розділу	24
2 ОГЛЯД ПІДХОДІВ ТА МЕТОДІВ МАШИННОГО НАВЧАННЯ	25
2.1 Машинне навчання	25
2.1.1 Обраний набір даних	25
2.1.2 Алгоритми машинного навчання	27
2.1.3 Метрики оцінки якості.....	28
2.2 Висновки до розділу	30
3 РОЗРОБКА МОДЕЛІ МАШИННОГО НАВЧАННЯ ДЛЯ ВИЗНАЧЕННЯ ХВОРОБ РОСЛИН	31
3.1 Реалізація та тренування згорткових нейронних мереж	31

3.1.1	Розподіл даних.....	31
3.1.2	Аугментації.....	32
3.1.3	Навчання класифікаційних моделей	33
3.1.4	Експериментальні результати.....	34
3.1.5	Результати тренування моделей	39
3.1.6	Аналіз навчених моделей	40
3.2	Висновки до розділу	40
4	РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	41
4.1	Вимоги до програмного забезпечення	41
4.2	Бізнес процеси	49
4.3	Проектування архітектури програмного забезпечення	53
4.3.1	Мобільний додаток	53
4.3.2	Серверна частина	54
4.3.3	База даних	55
4.4	Вибір засобів розробки програмного продукту	56
4.4.1	Бібліотека Numpy	57
4.4.2	Фреймворк Pytorch.....	57
4.4.3	Фреймворк Flask.....	58
4.4.4	Фреймворк React Native.....	59
4.4.5	Мова програмування TypeScript.....	60
4.4.6	Інструмент керування станом MobX.....	61
4.4.7	Мова програмування Python	61
4.5	Функціональність програмного забезпечення	62
4.6	Вимоги до технічного забезпечення	62
4.7	Керівництво користувача	62

4.8	Висновки до розділу	68
5	РОЗРОБКА СТАРТАП ПРОЄКТУ «ЗЕЛЕНИЙ ДОКТОР»	70
5.1	Опис ідеї.....	70
5.2	Технологічний аудит ідеї проєкту	73
5.3	Аналіз ринкових можливостей запуску стартап проєкту	74
5.4	Розробка ринкової стратегії проєкту.....	81
5.5	Розроблення маркетингової програми стартап-проєкту	83
5.6	Висновки до розділу	86
	ВИСНОВКИ	87
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	88
	ДОДАТОК А	91

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

Задача класифікації	Задача, яка має множину об'єктів, кожний з котрих належить до певного класу з детермінованої множини класів. Для вирішення даної задачі необхідно мати можливість визначати клас об'єкта, клас якого не являється невідомим
Аугментації	Видозмінення зображення за допомогою зміни налаштувань зображення (яскравості, сатурації, контрасту тощо) та різних викривлень для отримання додаткових даних які також підходять для навчання моделі машинного навчання
Вхідний шар	Шар нейронної мережі, який отримує дані з датасету для обробки
Вихідний шар	Шар нейронної мережі, що повертає результат роботи нейронної мережі
Прихований шар	Шар, що не являється ні вихідним шаром, ні вхідним шаром
Фреймворк	Платформа, яка детермінує структуру програмного продукту
Авторизація	Надання прав конкретному користувачу на виконання детермінованої кількості дій
Транферування навчання	Процес використання вже готової (натренованої) моделі для донавчання – використання додаткового датасету для розширення можливостей використання моделі та підвищення її точності в потрібному вузькому напрямку.

ВСТУП

Рослини завжди були невід’ємною частиною нашого життя. Для людини представники одноіменного царства є продуктом харчування, джерелом ресурсів для облаштування навколишнього середовища: будування, меблювання і т.д. Також, протягом майже всього періоду існування людства рослини слугували єдиним шляхом лікування та продовжують рятувати життя і до сьогоднішнього дня. З усього вище наведеного можна сказати, що життя людини залежить від рослин.

Питання здоров’я рослин стало ключовим в розвідку людства. Вирішення питання хвороб культивованих видів автоматично анулювало всесвітню проблему голоду та дало неймовірний поштовх у розвитку нашої цивілізації. Зараз рідко трапляються випадки, коли вмирає ціле поле. Для цього мають статись екстраординарно несприятливі події чи скластись незвичайно погані обставини.

Тож проблема хвороб рослин тепер більше стоїть перед власниками рослин “малого масштабу” – хазяями кімнатних рослин, приватного саду чи огорожу. В цьому випадку люди рідко мають можливість звертатись до професіоналів, через брак коштів, чи небажання їх витратити, адже оплата роботи висококваліфікованого агронома має бути відповідною. Чи навіть впевненість в тому, що власних знань буде достатньо (як прояв відомого та розповсюдженого ефекту Даннінга-Крюгера – метокогнітивного викривлення в людей, які мають низьку кваліфікацію, якої виявляється недостатньо для розуміння власної некомпетентності).

Тож метою даної магістерської дисертації стала автоматизація процесу визначення таких хвороб як вірус жовтої курчавості листя, фітофтороз, на таких рослинах як помідори, картопля, полуниця шляхом розробки програмного продукту, що складається з мобільного додатку для смартфонів, серверної частини та сервісу, що є обготкою над моделлю машинного навчання.

Для реалізації поставленої мети було поставлено наступні завдання:

— провести аналіз сучасних підходів та рішень для класифікації об’єктів на фотознімках;

- проаналізувати проблему обмеженості набору даних;
 - сформулювати перелік функціональних вимог;
 - спроектувати архітектуру програмного продукту;
 - імплементувати визначений функціонал в готовому програмному продукті;
 - створити модель машинного навчання для вирішення проблеми класифікації хвороб рослин по зображенню хворого листа;
- надати кінцевому користувачеві можливість взаємодіяти з моделлю машинного навчання.

1 ПОСТАНОВКА ЗАДАЧІ АВТОМАТИЗАЦІЇ ВИЗНАЧЕННЯ ХВОРОБ РОСЛИН

1.1 Загальне визначення проблематики захворювання рослин

Близько 85 відсотків хвороб усіх рослин спричиняються грибковими або грибоподібними організмами. Також за велику частку хвороб харчових культур та кормових культур слід звинувачувати вірусні та бактеріальні організми. Круглі черви, також відомі як нематоди теж можуть псувати життя власникам рослин. І, звичайно, існує такий клас хвороб як абіотичні, тобто причиною поганого стану рослини є не якийсь з перевищених вище подразників, а низька якість необхідних для життя елементів, таких як вода, повітря чи ґрунт [1]. Вони можуть бути токсичними, чи просто в поганій якості, що насправді є великою проблемою, адже наслідки такого ураження проникають в найглибщі клітини рослини, і потім позбутися симптомів може бути вкрай важко.

Почнемо розглядати хвороби, причиною яких є три найрозповсюдженіші патогенні мікроби: віруси, грибки, бактерії [2]. При найменшій підозрі на те, що рослина захворіла, слід ретельно обстежити її візуально, адже типові симптоми цих хвороб виражаються саме у зовнішньому вигляді, і також кожна хвороба має свої характерні ознаки.

Симптомами чи ознаками хвороби рослини є значний вплив захворювання на рослину. Вони можуть включати різні типи зовнішнього прояву, такі як значну зміну кольору, функціонування або форми рослини чи її листя, що є прямою реакцією на патоген.

Нижче наведено декілька прикладів типових ознак та симптомів грибкових, бактеріальних та вірусних захворювань рослин:

- борошниста роса;
- склеротинія (біла цвіль);
- листові іржі (звичайна листові іржі в кукурудзі);
- іржа стебла (іржа стебла пшениці).

Симптоми грибкової хвороби:

- затухання сходів (фітофтора);
- хлороз (пожовтіння листя);
- смуга іржі пустул на листі озимої пшениці є симптомом;
- пташине око на ягодах (антракноз);
- листова пляма (септорія коричнева пляма);
- смугаста іржа.

Симптоми вірусного захворювання:

- пожовкле листя;
- рослини низькі;
- скриплене листя;
- мозаїка листя візерунком.

Ознаки бактеріальних захворювань:

- водяні вогнища;
- потік бактерій у воді з обрізаного стебла;
- бактеріальні сочі.

Симптоми бактеріального захворювання:

- фруктова пляма;
- канкер;
- коронний жовч;
- бактеріальна плямистість листя;
- листова пляма з жовтим ореолом.

Базуючись на даних, наведених вище, можна зробити висновок, що серед ознак бактеріальних, грибкових, та вірусних хвороб існує дуже багато тотожностей, і всі вище перелічені ознаки тим чи іншим чином виражаються та проявляються чи на листі, чи на стеблах, чи на плодах рослини (саме це робить можливим використання комп'ютерного зору для ідентифікації хвороб. Також, коли при візуальному обстеженні виявляються симптоми, що не являються характерними для вищенаведених хвороб, слід включити на розгляд абіотичні хвороби, проблеми

нематод та пошкодження [21]. Ці хвороби є атипічними, їхні ознаки неможливо систематизувати, тому при їх лікуванні спочатку виключають всі інші проблеми.

1.2 Змістовний опис і аналіз предметної області

Машинне навчання – це один з способів функціонування штучного інтелекту, а конкретніше – він надає системі можливість навчатися та покращуватись автоматично на основі досвіду, а не від фактичного кодування. Це стало можливим для втілення у життя тому що на даний момент люди отримали доступ до ніколи не баченої раніше великої кількості даних, завдяки яким тепер стало можливим не програмувати машини а програмувати. Вже, на сьогоднішній день такий «метод використання даних», якщо так назвати машинне навчання, вважається неймовірною технологічною революцією, тому що ця технологія дійсно змінює світ. Кожен з нас вже має в телефоні застосунок, що використовує методи машинного навчання та штучний інтелект в тій чи іншій мірі.

1.3 Згорткові нейронні мережі

Згорткова нейронна мережа – це алгоритм зі сфери глибокого навчання, що може отримувати вхідний файл зображення, виявляти різні деталі на зображенні, тобто відокремлювати їх одне від одного, визначати «на власний розсуд» вагу, чи, якщо перефразувати – важливість різним деталям та частинам на вхідному зображенні.

Попередня обробка, яка є невідомою частиною в згортковій нейронній мережі, значно нижча у порівнянні з такими ж обробками інших відомих алгоритмів класифікації. Якщо в більш простих методах фільтри створюються нативно, то згорткова нейронна мережа знаходиться на відчутно вищому рівні – її функціонал дозволяє досліджувати фільтри та інші параметри, але звичайно за умови наявності потрібного датасету (датасет взагалі є краеугольним каменем кожного проєкту, пов'язаного з машинним навчанням).

При розробці архітектури для нейронної мережі згорткового типу програмісти напевно що надихались тим, як в мозку людини організовані нейрони та як

структурована зорова кора людини. Окремий нейрон буде реагувати на різні подразники тільки в обмеженій зоні зорового поля чи, іншими словами, рецептивного поля. Набір цих полів нашаровується повністю на зорову зону [10].

Вчені в статті Neub Networks Learn Hub [22] показали, що машинне навчання має своєю підмножиною нейронні мережі, які в свою чергу стали базою для алгоритмів глибокого навчання. Якщо описувати структуру нейронної мережі, то це завжди буде один і той самий набір базових айтемів – три базові шари: вхідний шар, деяка кількість прихованих шарів, а саме від одного і більше, і вихідний шар. Кожен шар має в собі окремі вузли, пов'язані між собою. Також кожен з цих вузлів має власну вагу та поріг навчання.

Звичайні нейронні мережі в порівнянні зі згортковими є значно менш ефективними при роботі з вхідними даними, а саме файлами, тобто згорткові нейронні мережі доцільно з точки зору продуктивності використовувати з звуком, текстом чи зображеннями [10]. Структурно ж згорткові нейронні мережі відрізняються наявністю таких шарів:

- об'єднуючий шар;
- згортковий шар;
- повністю з'єднаний шар.

1.4 Нейронна мережа EfficientNet

Коли розробляються згорткові нейронні мережі, це робиться з фіксованою вартістю ресурсів, які потім нарощують для досягнення ліпшої якості, коли наявні ще ресурси. Наприклад ResNet 18 можна поліпшити до моделі ResNet 200, просто збільшивши кількість шарів до вихідної моделі. Взагалі ця техніка масштабування вирішувала проблему кращої точності більшості пакетів даних якісного аналізу.

Тим часом відомі методи масштабування моделі дуже рандомні. Деякі з них масштабуються по глибині, а інші – по ширині. Є такі що роблять картинку з більшою роздільною здатністю, щоб поліпшити результати. Часто призводить до незначного або повного покращення продуктивності така технологія, яка вимагає ручного налаштування та велику кількість людино-годин. Розробники EfficientNet

запропонували збільшити моделі згорткових нейронних мереж, щоб отримати кращу точність і ефективність значно елегантнішим методом.

EfficientNet — це спосіб масштабування та архітектура згорткової нейронної мережі, що рівномірно масштабує всі розміри глибини/ширини/роздільної здатності використовуючі складеного коефіцієнта. Це рівномірно масштабує ширину, глибину та роздільну здатність мережі за допомогою набору фіксованих коефіцієнтів масштабування. Щоб використати в рази більше наявних потужностей, ми можемо просто збільшити глибину мережі, ширину, розмір зображення, з коефіцієнтами, знайденими за маленькою сіткою на невеликій базової моделі. EfficientNet бере коефіцієнт для рівномірного принципового масштабування ширини, глибини та роздільної здатності мережі.

Метод складного масштабування пояснюється, що якщо вхідна картинка більше, то мережа вимагає більше шарів, щоб збільшити поле що обробляється, і більше каналів, щоб захопити частинки зображення з більшою кількістю пікселів на більшій картинці.

Для розробки методу складного масштабування автори систематично вивчали вплив кожного методу масштабування на продуктивність і ефективність моделі. Вони прийшли до висновку, що хоча масштабування окремих вимірів допомагає підвищити продуктивність моделі, збалансування масштабу в усіх трьох вимірах — ширині, глибині та роздільній здатності зображення — враховуючи змінні доступні ресурси, найкраще покращує загальну продуктивність моделі.

Взявши за основу метод масштабування та AutoML, автори виготовили сім моделей різної розмірності, які краще більшості згорткових нейронних мереж і показали сильно ліпшу ефективність.

EfficientNets також добре передає та досягає найсучаснішої точності на CIFAR-100 (91,7%), Flowers (98,8%) та 3 інших наборах даних для навчання передачі з на порядок меншою кількістю параметрів.

Як можемо побачити, результати EfficientNet перевершують усі попередні архітектури згорткових нейронних мереж у більшості наборів даних порівняльного

аналізу. Метод складеного масштабування також можна використовувати для ефективного масштабування інших архітектур згорткових нейронних мереж [17].

1.5 Нейронна мережа MobileNet

MobileNet — це клас CNN, який був відкритий Google, і тому це дає нам чудову відправну точку для навчання наших класифікаторів, які є шалено малими та шалено швидкими. MobileNet використовується для ефективного включення комп'ютерного зору в невеликі пристрої, такі як мобільні телефони та роботи, без значного зниження точності.

Загальна кількість параметрів у стандартній MobileNet становить 4,2 мільйона, що значно менше, ніж у деяких інших архітектурах згорткових нейронних мереж. MobileNet дозволяє користувачам моделей можливість регулювати розмір своєї моделі що впливає на якість, введенням двох нових глобальних гіперпараметрів, які можна налаштувати.

MobileNets базуються на спрощеній архітектурі, яка використовує згортки, що розділяються по глибині, для створення легких глибоких нейронних мереж. Ми представляємо два прості глобальні гіперпараметри, які ефективно компроміс між затримкою та точністю. Ці гіперпараметри дозволяють розробнику моделі вибрати модель потрібного розміру для свого застосування на основі обмежень проблеми.

MobileNets представляє масштабні експерименти щодо балансу ресурсів і точності та демонструє високу продуктивність порівняно з іншими популярними моделями за класифікацією ImageNet. MobileNets показує ефективність у широкому діапазоні додатків і випадків використання, включаючи виявлення об'єктів, дрібну класифікацію, атрибути обличчя та великомасштабну геолокалізацію [19].

1.6 Аналіз успішних ІТ-проектів

Вчені з OpenAI з'ясували, що навчання на парах картинка-текст, навіть сирий текст — це перспективна заміна "звичайним" класифікаторами. Завдання раннього навчання, полягає в тому, щоб для кожної картинки з пари картинка - текст знати наперед, який текстовий опис відповідає саме йому. Мережа, яка може робити

висновки на цю задачу, натренували на 400 мільйонах пар картинка -текст, зібраних з інтернету. Така мережа спроможна розрізняти по метриці схожості картинки і текстові набори. Взагалі таку модель можна буз зусиль використовувати для громадного набору завдань що можна визначити як "навчання без навчання".

Класифікатори картинок до яких ми звикли разом вчать лінійний класифікатор, яким є крайній шар для визначення класу об'єкта і вичленовувач (згорткову мережу до кінцевого рівня). А для визначення коректних пар з набору навчальних зразків картинка-текст CLIP спільно навчає енкодер картинок та декодер текстового опису (трансформер).

Наступне, роблячі векторну представлення картинки на основі класів нового набору даних, який пройшов через кодувальник в текстовій формі в режимі класифікатора на новому наборі даних, попередньо тренований текстовий енкодер автоматично дає лінійний класифікатор в стані "навчання без навчання".

На рисунку 1.1 візуалізація зсуву розподілення для такої речі як «банан». «Банан» є спільним класом для 5 із 7 датасетів звичайного зсуву розподілу. В таблиці показані оцінки точності для зрівняння між найкращою моделлю CLIP – ViT- L / 14 - 336 px та звичайною моделлю, з схожою продуктивністю під час використання ImageNet [4] набору, ResNet101.

Тим часом, задача ідентифікації захворювання рослини за картинкою больного листя вже стоїть перед розробниками та науковцями. Науковці аналізують способи застосування моделей машинного навчання для вирішення цієї задачі, але на даний момент розробки залишаються в кордонах R&D проєктів.

З'ясувалося, що хвороби, які видно на листі, мають унікальні ознаки, людина може їх розпізнати, а тепер, з штучним зором, можливо для розпізнавання програмою.

Результати ж цих досліджень аналізуються лише спеціалістами з машинного навчання або даних. Саме для цього ми інтегруємо нашу модель розпізнавання хвороб машиною у сервіс, щоб отримати зручні інструменти зв'язку з моделлю машинного навчання (наприклад за допомогою REST API).

	Dataset Examples	ImageNet ResNet101	Zero-Shot CLIP	Δ Score
ImageNet		76.2	76.2	0%
ImageNetV2		64.3	70.1	+5.8%
ImageNet-R		37.7	88.9	+51.2%
ObjectNet		32.6	72.3	+39.7%
ImageNet Sketch		25.2	60.2	+35.0%
ImageNet-A		2.7	77.1	+74.4%

Рисунок 1.1 – Модель навчання з нульовим розрядом CLIP краще зміщує розподіл, ніж звичні моделі ImageNet

Застосунки, спрямовані покращення якості спостереження та догляду за рослинами, існують в рамках лише одної мобільної операційної системи, тобто якісь юзері не мають можливості створеного технічного рішення. Така задача пов'язана з фактом, що тепер крос-платформні аплікації розробляються на нових технологіях, і тепер швидко покращуються, але достатньої кількості висококваліфікованих спеціалістів та випадків успішного використання дуже мало, як технології, а також можуть бути гірше у масштабуванні та інших параметрах.

Застосунки роботи з базами даних загальновідомі. Свої додатки мають популярні компанії на кшталт JetBrains та інші. Вони також є на різних розробних платформах. Але, порівнюючі з створеним нами застосунком, вони мають дуже широкий набір функціоналу, що шкідливо для нашого завдання. Враховуючі великий діапазон функцій, застосунки отримують більший об'єм, це можливо критично в різних ситуаціях, а в якихось взагалі неефективно.

1.7 Аналіз існуючих технічних рішень

Група вчених зробили публікацію, де відобразили свою працю пов'язану з виявленням та розпізнаванням захворювань рослин використовуючі картини заболілого листа використовуючі способи машинного навчання та машинного зору. У своїй праці розробники аналізували можливість застосування глибоких згорткових нейронних мереж для цієї задачі, а саме – задачі класифікації, у стандартне розуміння задачі класифікації можна додати задачу виявлення захворювань рослин. Розробники зробили акцент на декількох архітектурах, а саме «AlexNet» та «GoogLeNet» що розроблені для вивчення датасету «ImageNet».

Структура AlexNet ґрунтується на тій самій архітектурі проектування, потім 3 повнозв'язані шари і в вся мережа в кінці отримує шар softMax. Після двох шарів згортки іде кінцевий шар нормалізації та об'єднання, а кінцевий згортковий шар – уніфікованим шаром комбінації. Останній рівень має 38 аутпутів у AlexNet (це збігається з загальною кількістю класів у датасеті для навчання), що видає рівень softMax. Після обробки даних рівнем softMax, з'являється розподіл вірогідностей у для 38 класів, що суммарно дають 1, тобто отримується вірогідність попадання у кожен з 38 класів. Усі AlexNet перші 7 шарів отримали пов'язаний з ними блок ReLu, також мають шар вибуху, пов'язаний із коефіцієнтом відкидання 0,5 перші два зв'язані шари.

Також розробники у своїй праці вказують, архітектура GoogleNet є також цікавою – це архітектура з 22 шарами, але з меншою кількістю параметрів (5 млн) для параметризації мережі, ніж AlexNet в якій для параметризації використано 60 млн параметрів. Важливою відмінністю архітектури GoogleNet є використання архітектури «мережа в мережі» за початковими модулями, які беруть паралельні згортки 1×1 , 3×3 і 5×5 , і шар максупуллінгу, завдяки чому є можливість паралельно фіксувати різні функції.

Для практичності розробки, об'єм обчислень на цьому шарі, відслідковується щоб була збережена ефективність у сенсі цінності і часу, того 1×1 згортки до вищезгаданих 3×3 згортки та 5×5 згортки і після шару найбільшого об'єднання ще приєднано для зменшення об'єму та без ушкоди для якості.

Шар об'єднання фільтра з'єднує результати всіх паралельних шарів. У варіанті архітектури GoogLeNet, що розробники брали у вищевказаних експериментах, задіяно загалом 9 початкових модулів.

Також у своїй праці розробники аналізують ефективність розглянутих архітектур на датасеті PlantVillage, використовуючи навчання з нуля в першому випадку, та пристосовуючі навчені моделі (на датасеті ImageNet) використовуючі трансферування знань.

Важлива різниця між цими методами до навчання (трансферування і навчання з нуля) є у початковому стані вагів для декількох шарів, завдяки чому маємо підхід до трансфування брати багато візуальних знань, які вже містяться в моделях GoogleNet та AlexNet, вже попередньо натреновані на наборі даних ImageNet.

1.8 Аналіз відомих програмних продуктів

Проаналізувавши ринок ми знайшли застосунки зі схожим функціоналом – розпізнавання рослит та їх захворювань.

Тим часом для порівняльного аналізу ми вибрали два найбільш схожих – PlantIn: Plant Identifier та Blossom.

PlantIn: Plant Identifier – це застосунок для мобільних девайсів для догляду за рослинами. Деякі функції цього застосунку:

- ідентифікація рослини - класифікація виду рослини за фото;
- наявність календаря - застосунок дозволяє використовувати особистий календар на базі рекомендованого уходу за конкретним видом з бази знань;
- можливість отримати консультацію онлайн – юзер сплачує за можливість задати питання фахівцю у галузі та отримати додаткову допомогу.

Blossom – це застосунок для мобільних девайсів для догляду за рослинами та їх класифікації і також дозволяє створювати розклад уходу за конкретною рослиною згідно рекомендацій фахівців. Набір фіч та функцій цього застосунку подібні тим що є у мобільного застосунка PlantIn: Plant Identifier.

Також відомо дуже потужний застосунок розроблений компанією Google, у де додатково ще є можливість пошуку за ключовими словами.

Після зрівняльного аналізу застосунок розроблених нами з тими додатками що перелічені вище, можна зробити висновки, що вони не надають відповідь на головні питання, заради яких розроблявся наш застосунок: “Що робити коли виявлено захворювання та як лікувати рослину, де взяти необхідні препарати та які?” моментально, відкрито та безкоштовно.

Google дозволяє зробити пошук аналогів ліків, але це довго і залежить від користувача. PlantIn: Plant Identifier надає спосіб отримати консультації з кваліфікованим спеціалістом, але це за додаткові гроши.

У Blossom перевага в тому що є календар піклування, тож, коли якраз відсутність вчасного уходу за рослиною є причиною захворювання, він допоможе як профілактика, можна з’ясувати різницю між тим, що зроблено з тим, що має бути та зробити висновки.

Таким чином, можна зробити висновок, що розроблений нами застосунок має суттєві переваги перед конкурентами та унікальний функціонал, що дозволить йому стати конкурентноспроможним на ринку.

1.9 Висновки до розділу

В описаному вище розділі було проведено аналіз предметної області, розглянуто різні типи хвороб та отримано розуміння, що створення програмного продукту має під собою теоретичне підґрунтя.

Також в параграфах вище було наведено ґрунтовний аналіз аналогів створюваної системи.

Було з’ясовано, що саме успішних комерційних продуктів з аналогічною «killer feature» – тобто основним функціоналом наразі не представлено на ринку.

В той самий час було розглянуто технічні рішення з аналогічними цілями, та описано їхні недоліки, що робить створення описаного програмного продукту цікавим для кінцевого користувача, адже створене технічне рішення обгорнуто в доступний користувачський інтерфейс.

В даному розділі було також описано можливі технічні інструменти для вирішення поставленої задачі, та описано їхні недоліки та плюси.

2 ОГЛЯД ПІДХОДІВ ТА МЕТОДІВ МАШИННОГО НАВЧАННЯ

2.1 Машинне навчання

Перед початком роботи над кодом будь якого програмного продукту, що включає в себе внутрішнього використання машинного навчання необхідно визначити такі пункти:

- ціль створення моделі, або задачу, яку вона має виконувати;
- підібрати набір даних, наякому модель буде навчатись, набір для тестування та набір для валідації;
- визначити моделі, які будуть навчатись (тобто визначити їх тип та структуру) та навчити їх на обраних даних;
- врахувати неточності та особливості проєкту та обраних даних для більш глибокого розуміння отриманих значень;
- обрати методи оцінки вірності (часто, у випадку моделей машинного навчання – вірність є точністю, адже моделі «відповідають» вірогідностями) отриманих результатів.

2.1.1 Обраний набір даних

В даному проєкті було обрано набір даних NPDD (New Plant Diseases Dataset) що включає в себе картинки хворого та здорового (для уточнення роботи) листя різних культурних рослин.

Всього в датасеті зібрано 87867 картинки. Кожна картинка містить листок з вираженими хворобами, або здорові варіанти. Повний перелік хвороб рослин, описаних в обраному наборі даних показаний на рисунку 2.1.

```
[ 'Peach___Bacterial_spot',
  'Tomato___Late_blight',
  'Cherry_(including_sour)___Powdery_mildew',
  'Pepper,_bell___Bacterial_spot',
  'Potato___healthy',
  'Orange___Haunglongbing_(Citrus_greening)',
  'Corn_(maize)___healthy',
  'Raspberry___healthy',
  'Tomato___Tomato_mosaic_virus',
  'Tomato___Bacterial_spot',
  'Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot',
  'Apple___healthy',
  'Pepper,_bell___healthy',
  'Corn_(maize)___Northern_Leaf_Blight',
  'Potato___Early_blight',
  'Blueberry___healthy',
  'Apple___Apple_scab',
  'Tomato___Spider_mites Two-spotted_spider_mite',
  'Grape___Esca_(Black_Measles)',
  'Apple___Cedar_apple_rust',
  'Grape___Leaf_blight_(Isariopsis_Leaf_Spot)',
  'Grape___Black_rot',
  'Strawberry___Leaf_scorch',
  'Grape___healthy',
  'Tomato___Target_Spot',
  'Apple___Black_rot',
  'Tomato___Tomato_Yellow_Leaf_Curl_Virus',
  'Soybean___healthy',
  'Tomato___healthy',
  'Corn_(maize)___Common_rust_',
  'Tomato___Leaf_Mold',
  'Potato___Late_blight',
  'Squash___Powdery_mildew',
  'Cherry_(including_sour)___healthy',
  'Peach___healthy',
  'Strawberry___healthy',
  'Tomato___Septoria_leaf_spot',
  'Tomato___Early_blight' ]
```

Рисунок 2.1 – Повний перелік всіх представлених в наборі даних хвороб рослин

Нижче на рисунку 2.2 наведено приклади зображень, які представлені в датасеті.

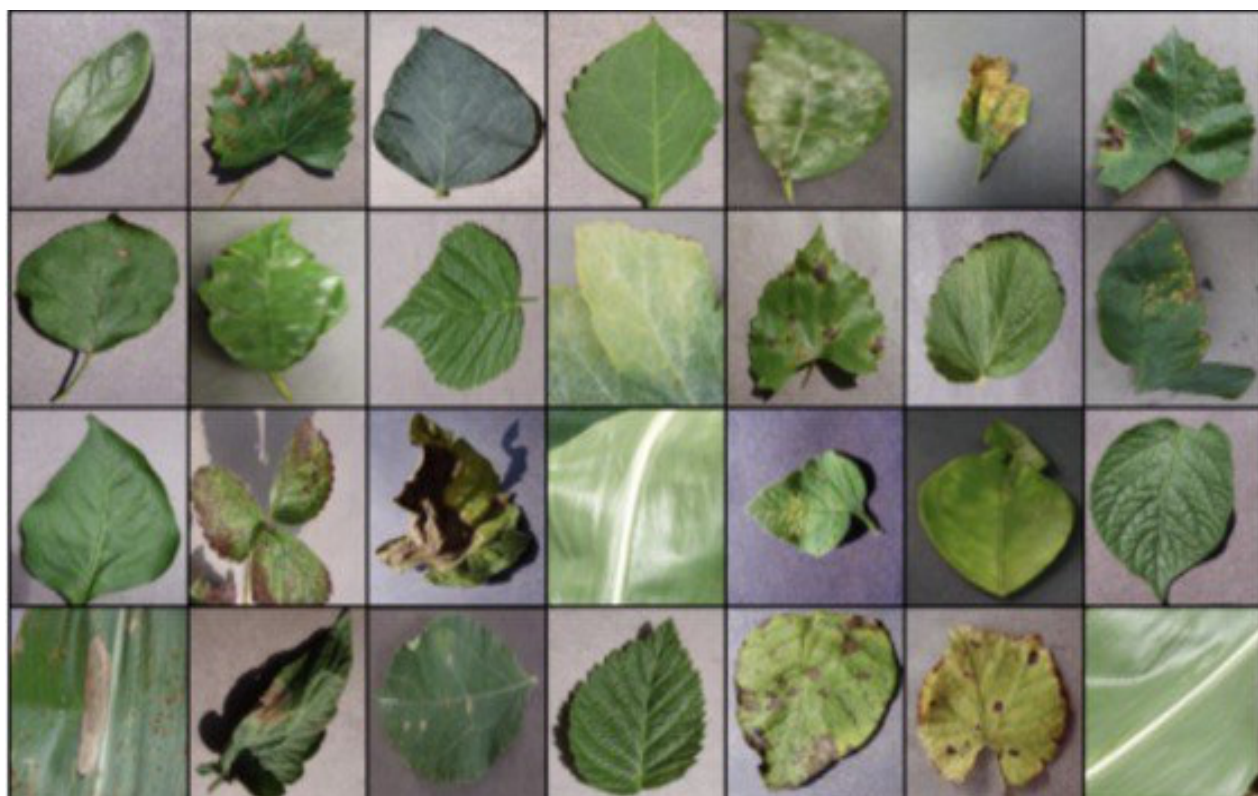


Рисунок 2.2 – Приклад фотографій хворого листя

З прикладів видно, що картини створювались доволі типово, тобто схожий фон, схоже освітлення, схожі контрасти. При таких умовах підвищується ризик перенавчання. Вирішення цієї проблеми описано в третьому розділі.

[illegible]

Рисунок 2.3 – розподіл хвороб та назв типів рослин у відсотках відносно загальної кількості зображень

2.1.2 Алгоритми машинного навчання

Якщо говорити про визначення слова алгоритм, то можна сказати, що це набір послідовних процедур що слугують для вирішення конкретної поставленої задачі. При виборі правильного алгоритму вирішення задачі в даному випадку, а точніше

методу машинного навчання, необхідно врахувати багато факторів зовнішніх і внутрішніх, щоб отримати найоптимальніший результат. Наприклад, в даному програмному продукті можна було використати важку модель з багатьма прихованими рівнями, що мала б більшу точність, але слід також враховувати те, що в даному програмному продукті переважається багато користувачів та важлива швидкість обробки даних. Тож можна зробити висновок, що в даному випадку необхідна кореляція цих двох факторів [20].

Серед сучасних моделей глибинного навчання, що виконують задачу класифікації можна назвати ResNet, EfficientNet та MobileNet.

2.1.3 Метрики оцінки якості

Можна змусити модель машинного навчання працювати, можна знайти великий набір даних та продумано розподілити його на вибірки. Але всі ці пункти стають неважливими, якщо неправильно порахувати точність моделі. Метод оцінки точності є вирішальним в питанні визначення точності моделі.

Також, якщо спуститись на рівень нижче, важливо визначити, як стимулювати модель до розвитку. Як визначити, який результат був ближчий до істини, яку ітерацію стимулювати, а які нейрони роблять систематичні помилки. Адже просто показуючи картинки нейронній мережі неможливо добитись результату, вона має вчитись.

Для підрахунку якості повернутого результату на ітерації навчання глибинної нейронної мережі було обрано функцію крос ентропії [15]. Також в деяких статтях можна зустріти такі назви як функція логарифмічних втрат чи функція лог втрат чи функція логістичних втрат.

В даному методі при отриманні передбаченого результату від моделі він порівнюється з фактичним цільовим результатом 0 або 1, або, якщо інтерполювати на задачу класифікації, чи вгаданий клас (1) чи ні (0). Після чого розраховується «втрата» – тобто різниця цих значень. Вираховане значення штрафує ймовірність, отриману від моделі машинного навчання, базуючись на тому, на скільки велика ця різниця, або на скільки різняться ці значення.

Штраф в даному випадку є логаріфмічним в своїй основі, що є дуже зручним, якщо згадати догаріфмічну функцію, адже всі значення, близькі до 1, отримують великий бал штрафу, в той час як значення що наближені чи прямують до 0 майже не штрафуються. Формула підрахунку штрафу описана під номером 2.1.

$$loss = - \sum_{i=1}^n t_i \log(p_i) \quad (2.1)$$

Де:

n – кількість класів;

t_i – значення дійсні;

p_i – вірогідність, що повертає модель.

Перехресна втрата ентропії, або крос ентропійна втрата, використовується для калібрування вагів моделі під час навчання і плягає в тому, щоб наблизити різницю результату, що повертає модель, і бажаного результату до 0.

Точність для кожної моделі рахувалась за формулою 2.2.

$$\frac{sum(p==l)}{len(p)} \quad (2.2)$$

Де:

p – передбачені значення;

l – цільові значення;

У даному розділі було виконано оцінку вибору датасету та описаний використаний набір даних, описані дані, що зберігає набір даних, проведений аналіз розподілення цих даних. Також було розглянуто підходи у вирішенні питань якості оцінки роботи створеної нейронної мережі, описано можливі метрики та оцінені підходи у вирішенні питання вибору цих метрик.

Також було описано метод підрахунку значень для втрат на кожній ітерації навчання моделі машинного навчання, та описано формулу підрахунку точності моделі.

2.2 Висновки до розділу

В даному розділі було розкрито поняття машинного навчання, описано алгоритм розробки програмного продукту, що має в основі модель машинного навчання.

В даному розділі детально описаний використаний набір даних для навчання і тренування моделей машинного навчання, вказані класи, розподіл зображень по класах та наведені діаграми.

Також в кінці розділу було детально описано підрахунки для процесу тренування, формули підрахунку точності моделі.

3 РОЗРОБКА МОДЕЛІ МАШИННОГО НАВЧАННЯ ДЛЯ ВИЗНАЧЕННЯ ХВОРОБ РОСЛИН

3.1 Реалізація та тренування згорткових нейронних мереж

Для створюваного програмного продукту для навчання були використані згорткові нейронні мережі, адже, на разі, використання саме цього типу нейромереж і в цілому такого методу машинного навчання признаний найбільш ефективним і рекомендується для використання.

3.1.1 Розподіл даних

Всього обраний набір даних NPDD (New Plant Diseases Dataset) містить 87867 зображень листя рослин різних типів. Датасет включає в себе 37 класів, по яким зображення можуть бути розподілені, і кожен описує не тільки хворобу рослини, а ще й її назву, тобто вид рослини. Розподіл зображень по класах описані кількісно на рисунку 3.1.

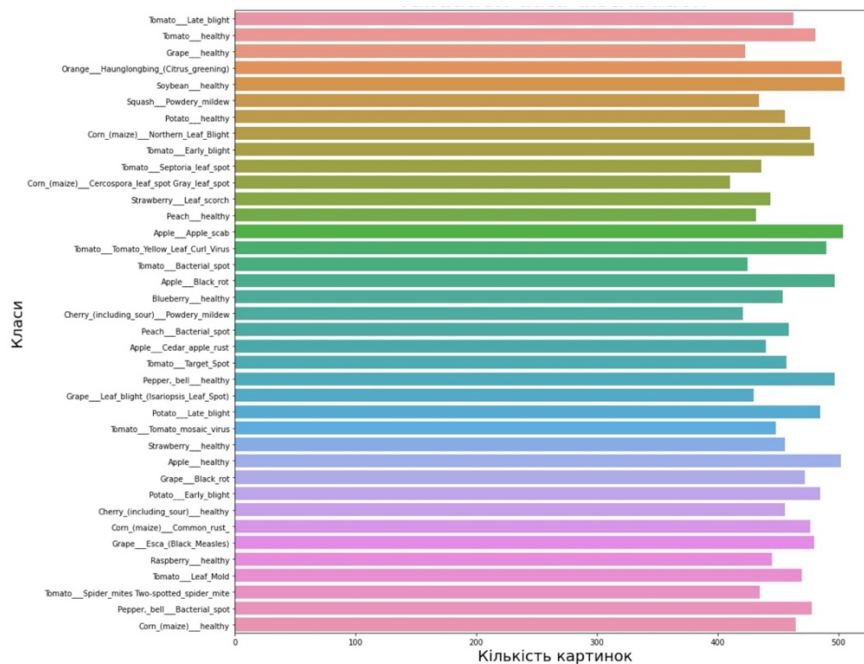


Рисунок 3.1 – Розподіл зображень набору даних по категоріях.

Якщо оглянути рисунок 3.1, то можна помітити, розподіл зображень по класах більш менш рівномірний, що є дуже хорошим показником і допоможе не перенавчатись на конкретному класі. Також видно, що в датасеті доволі велика кількість даних, але через те, що вони однотипні, необхідно використати такий метод як аугментації.

3.1.2 Аугментації

При роботі з картинками одного типу, як у випадку вибраного набору даних, існує ризик того, що використання моделі на реальних даних буде неуспішне, тобто показувати низьку точність, адже кінцеві користувачі можуть фотографувати листя хворої рослини в різних умовах, під різним кутом, в різному освітленні і на різні фотокамери, що дає різницю в балансах білого, кількості пікселів, якості зображення тощо.

Саме для таких випадків був створений метод аугментації зображень. Якщо узагальнити, то це метод розширення набору даних на основі вже існуючих зображень, виконуючі різні перетворення початкових зображень [16].

Загалом існує два типи аугментації:

- позиційна аугментація;
- кольорова аугментація.

Для позиційної аугментації характерні фізичні перетворення, а саме:

- обрізання в центрі: вхідне зображення буде обрізано по центру. Розмір зміни задається користувачем як параметр;
- випадкове обрізання: вхідне зображення буде обрізано у випадковому місці без особливих обмежень, якщо їх не створить користувач;
- вертикальне перевертання: вхідне зображення буде випадковим чином перевернуто чи ні із заданою користувачем ймовірністю;
- горизонтальне перевертання: вхідне зображення буде випадковим чином перевернуто чи ні із заданою користувачем ймовірністю;
- довільне обертання: вхідне зображення буде обернуто на довільну кількість градусів;

— зміна розміру: тобто зміна розміру вхідного зображення на деякий інший розмір;

– випадкове афінне перетворення: перетворення інваріанта центру збереження.

Для кольорової аугментації характерні такі перетворення, як:

— яскравість: дуже корисною і практичною є зміна яскравості зображення. Отримане картинка стає темнішою або більш світлою у порівнянні з вхідним варіантом, і може покривати випадки різного освітлення в кінцевого користувача на зображеннях;

— контрастність: зміна контрастності є теж дуже практичним методом при роботі з різними освітленнями. Зміна контрастності це збільшення або зменшення відстані між значеннями найтемнішої та найсвітлішої зон зображення, тобто розтягнення або звуження цього проміжку для зміни значень в кожній з точок зображення;

— насиченість: зміна насиченості кольорів зображення це заміна кольорів це заміна кольорів на більш або менш насичені варіанти.

3.1.3 Навчання класифікаційних моделей

В бакалаврській роботі, що стала основою даної магістерської дисертації, була натренована ResNet34 модель машинного навчання, яка показала хороші результати в точності, завдяки чому можна було назвати проєкт успішним [23]. Тим не менш, при більш детальному вивченні доменної сфери стало зрозуміло, що ResNet34 не є моделлю з оптимальною архітектурою для даного випадку. В створеному проєкті, коли кінцевих користувачів потенційно велика кількість, і задоволеність кінцевого користувача на пряму залежить від швидкості відображення результату, дуже велику роль відіграє швидкість виконання процедур. Також при клієнт-серверній архітектурі і потенційному розміщенні на орендованих серверах важливим фактором є розмір нейронної мережі, тобто який об'єм оперативної пам'яті необхідний для її зберігання і функціонування.

Саме розглянувши всі фактори вище, було прийнято рішення провести тренування декількох моделей машинного навчання для пошуку оптимального в

випадку створюваного продукту варіанту. Для тренування були обрані MobileNetV1 [19], MobileNetV2 [18], EfficientNet [17] та ResNet.

Розмір картинок, що приходили на вхід для моделей був встановлений як 224 на 224. Кількість зображень на одному циклі тренування складала 128. Повне ж тренування моделей машинного навчання проходило в 5 епох.

В контексті поставленої задачі в якості початкових моделей були обрані навчені на ImageNet [4] датасети. Потім останній шар кожної з моделей були видалені, та додані нові: глобальний пулінг, шар виключення та два повнозв'язних шари.

Шар глобального пулінгу використовується для того, щоб перетворити виходи згоркової нейронної мережі в вектор для подальшого аналізу та класифікації. Шар виключення використовувався, щоб уникнути проблеми перенавчання, за якою модель погано генералізує дані на тренувальній вибірці і тому на тестувальній показує метрики набагато менші. Цей шар з вірогідністю 10 відсотків зануляє нейрони, що не дозволяє конкретному нейрону мати великий вплив на результат.

Перший повнозв'язний шар зважає отримані результати з попередніх шарів та повертає вектор значень розмірністю (512, 1), другий - (N, 1), де N - це кількість класів. Як функція активації в першому шарі була використана *relu*, в останньому використовувалася *softmax* функція, яка повертає оцінку вірогідності для кожного класу. Передтреновані шари взятої моделі були заморожені, тобто вони не змінювалися під час ітерації зворотнього розповсюдження помилки. Додані вручну шари тренувалися та оптимізувалися Adam оптимізатором.

Метрикою оцінювання витрат була обрана функція крос ентропії, що була описана вище у розділі 2.1.3.

3.1.4 Експериментальні результати

В ході тренування обраних для вивчення моделей машинного навчання були побудовані однотипні графіки для аналізу роботи кожної з моделей. Рисунок 3.2 показує графік зміни точності динамічно при різних епохах для моделі MobileNetV1.

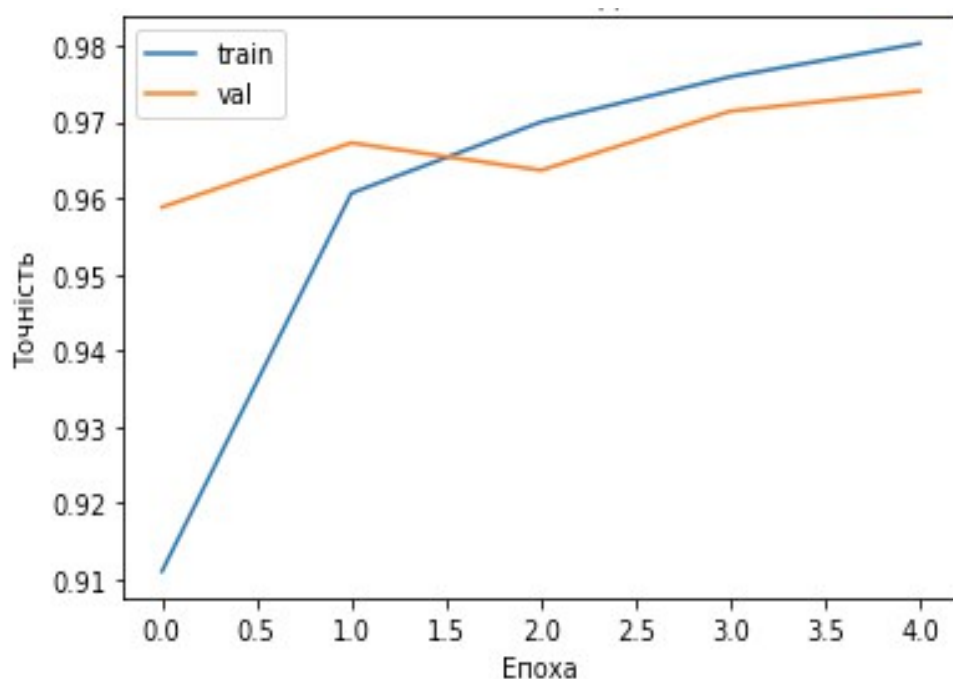


Рисунок 3.2 – Графік зміни точності для MobileNet V1

Також, звичайно необхідно оцінити зміну значень функції втрат для кожної з моделей. Нижче на рисунку 3.3 зображений графік зміни значення функції втрат для кожної епохи.

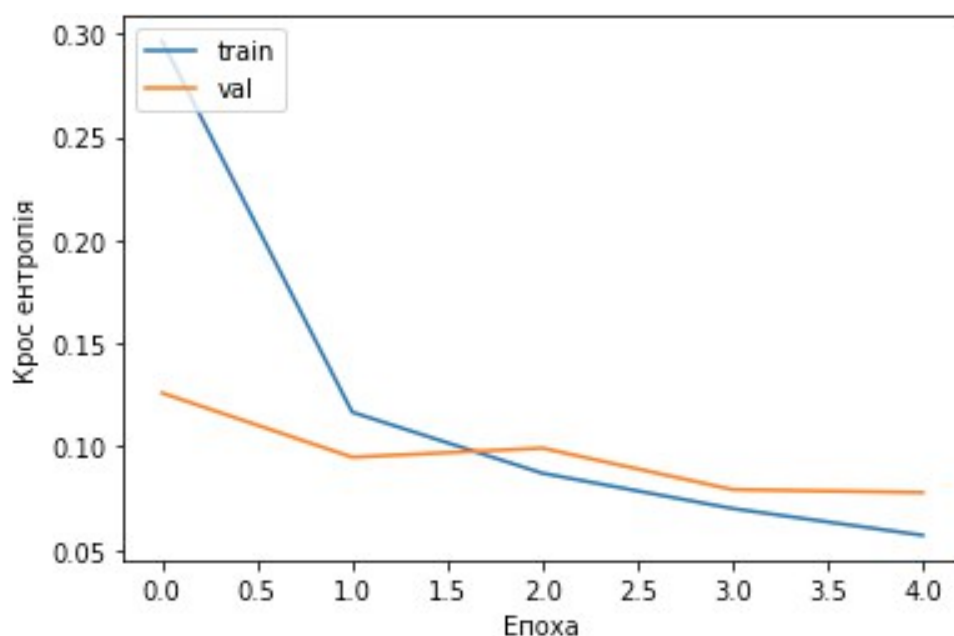


Рисунок 3.3 – Графік зміни значення функції втрат (крос ентропії) для кожної епохи для MobileNetV1

Рисунок 3.4 демонструє графік зміни точності при різних епохах для моделі MobileNet V2.

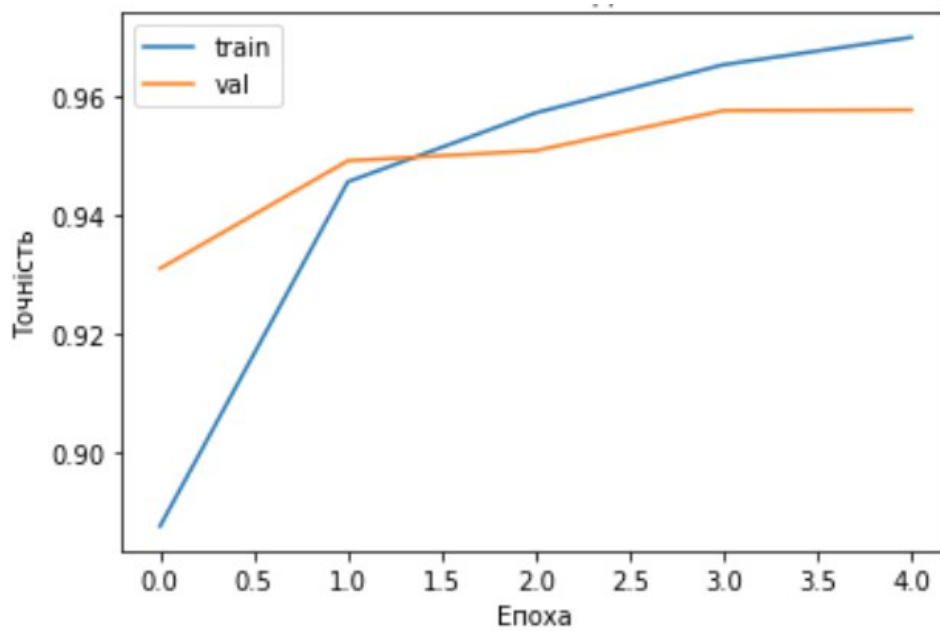


Рисунок 3.4 – Графік точності для MobileNet V2

Також, звичайно необхідно оцінити зміну значень функції втрат для кожної з моделей. Нижче на рисунку 3.5 зображений графік зміни значення функції втрат для кожної епохи.

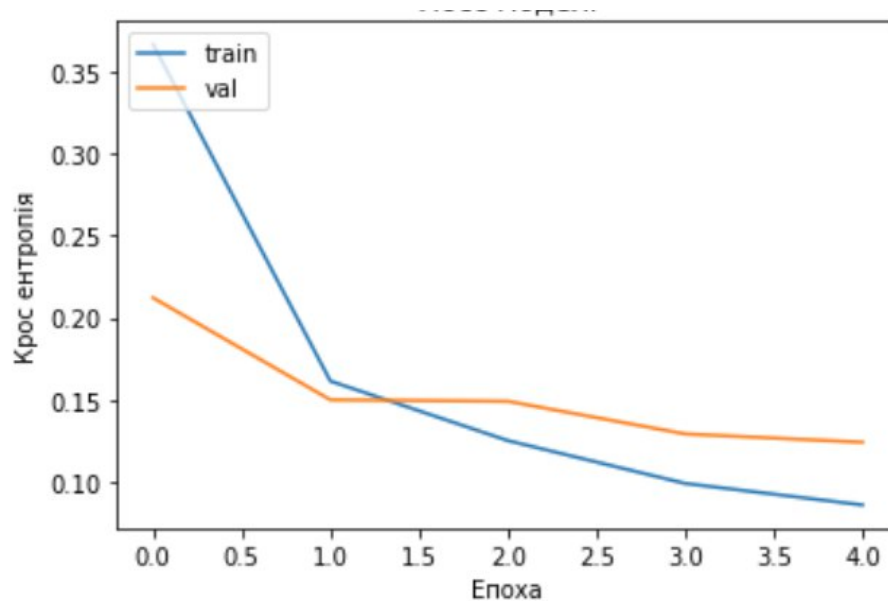


Рисунок 3.5 – Графік зміни значення функції втрат (крос ентропії) для кожної епохи для MobileNetV2

Рисунок 3.6 демонструє графік зміни точності при різних епохах для моделі EfficientNet B1.

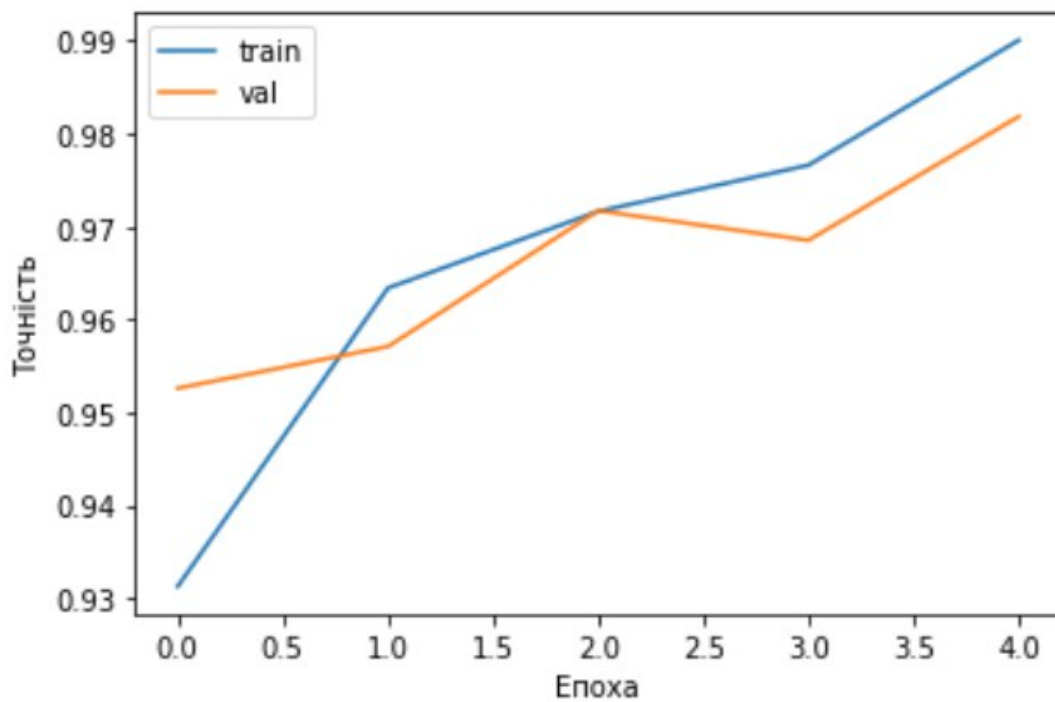


Рисунок 3.6– Графік точності для EfficientNet B1

Нижче на рисунку 3.7 зображений графік зміни значення функції втрат для кожної епохи.

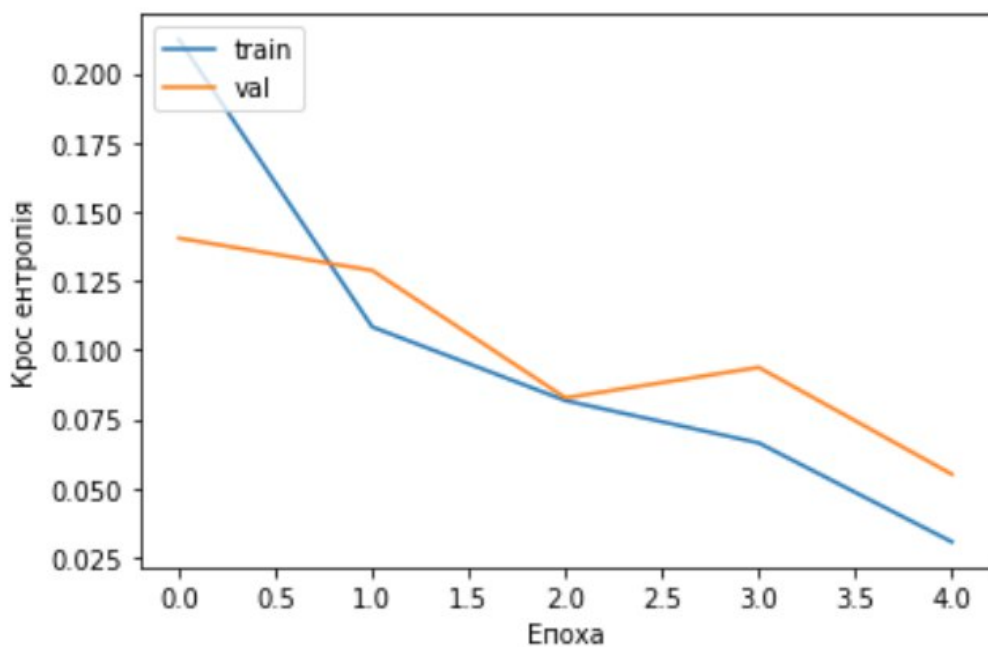


Рисунок 3.7 – Графік зміни значення функції втрат (крос ентропії) для кожної епохи для EfficientNet B1

Рисунок 3.8 демонструє графік зміни точності при різних епохах для моделі ResNet.

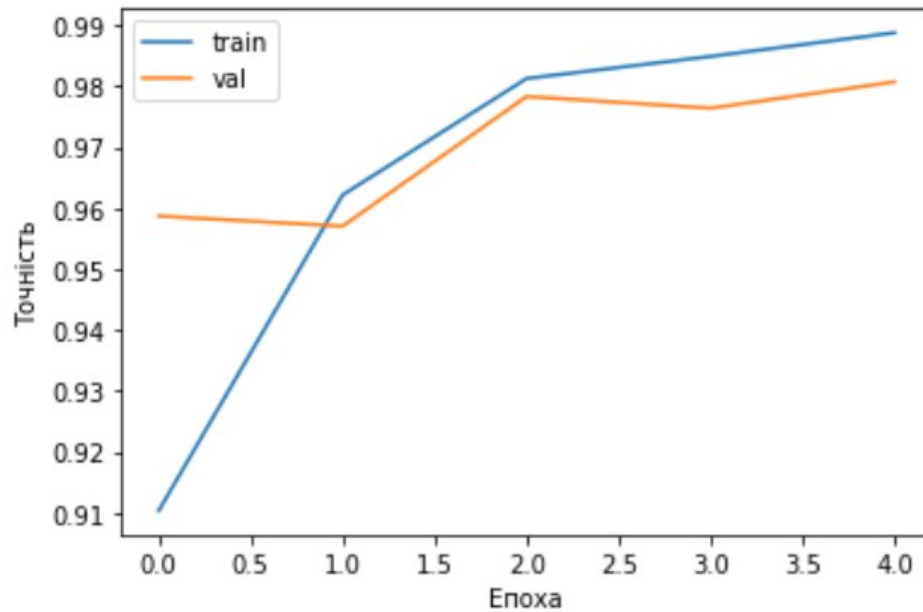


Рисунок 3.8 – Графік точності для ResNet

Нижче на рисунку 3.9 зображений графік зміни значення функції втрат для кожної епохи.

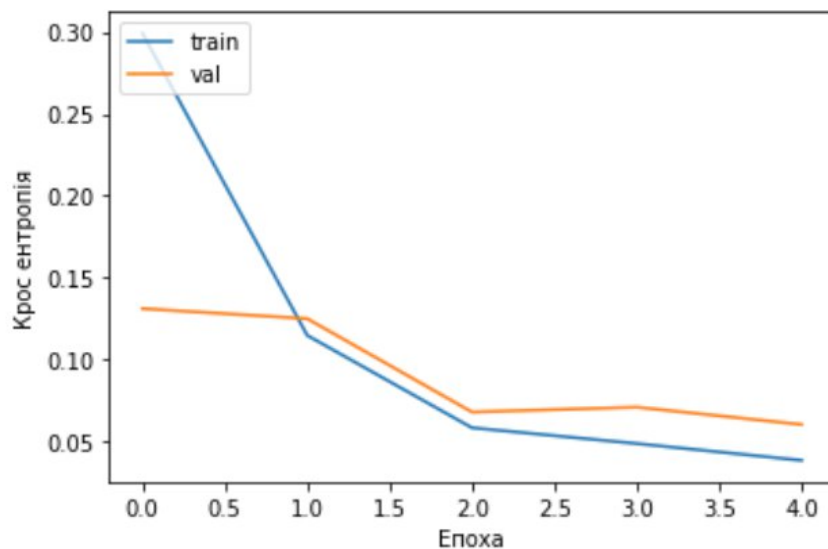


Рисунок 3.9 – Графік зміни значення функції втрат (крос ентропії) для кожної епохи для ResNet

3.1.5 Результати тренування моделей

Після тренування кожної нейронної мережі було проведено аналіз співставлення швидкостей моделей, їхніх точностей та значень функції витрат на останній валідаційній епосі таблиці 3.1:

Таблиця 3.1.

Результати тренування моделей машинного навчання

Нейронна мережа	Точність	Значення функції втрат	Швидкість обробки зображень на CPU	Швидкість обробки зображень на GPU
MobileNet V1	0.9723	0.0813	70 fps	256 fps
MobileNet V2	0.9564	0.1368	87 fps	252 fps
EfficientNet B1	0.9793	0.0550	85 fps	259.7 fps
ResNet	0.9812	0.0658	42 fps	199 fps

В таблиці 3.1 можна побачити, що вказані дві швидкості роботи для кожної моделі машинного навчання. Підрахунків справді було два для кожного типу моделі, тому що юніт графічної обробки та юніт обчислювальної обробки використовуються на різних девайсах по різному. На приклад на смартфоні взагалі не буде використаний юніт графічної обробки.

Через те, що користувацьким інтерфейсом було обрано мобільний додаток, треба врахувати швидкість моделі машинного навчання на юніті обчислювальної обробки, щоб не обмежувати подальший розвиток продукту у можливих напрямках, таких як перехід в офлайн.

Для підрахунку швидкостей для моделей як юніт графічної обробки було обрано NVIDIA T4 Tensor Core, а як юніт обчислювальної обробки – Intel(R) Xeon(R) CPU @ 2.30GHz.

3.1.6 Аналіз навчених моделей

При детальному аналізі видно, що за показником точності найкраще себе показала модель ResNet, що і не дивно, адже вона включає в себе найбільшу кількість шарів і тому є найважчою серед інших обраних для тренування моделей.

За показником значення функції втрат найкращі результати отримано від нейронної мережі EfficientNet B1, а саме 0.0550, що є доволі хорошим показником в цьому випадку.

При вимірюванні швидкості на юніті графічної обробки найшвидшу обробку зображень показала EfficientNet B1, а найгірші – ResNet зі значним відривом від інших моделей.

При вимірюванні швидкості на юніті обчислювальної обробки найшвидшу обробку зображень показала MobileNet V2, а найгіршу – ResNet зі значним відривом від інших моделей, що не є дивним враховуючи ті самі причини, за яких модель стала найточнішою.

Аналізуючи все вище наведене можна зробити висновок, що найефективнішою для даного проєкту з асумою показників є EfficientNet B1.

3.2 Висновки до розділу

У даному розділі був описаний розподіл зображень по класах, що дало можливість вивчити чи призведе розподіл зображень до перенавчання та оцінити якість обраного набору даних.

Був описаний такий метод розширення набору даних як аугментація, наведені приклади та деталі використання цього методу.

Також було розкрито алгоритм обраний для навчання моделей машинного навчання, описано кроки, кількість картинок на ітерацію та кількість епох.

Було наведено графіки показників протягом навчання кожної моделі, проаналізовано фінальні результати на валідаційній вибірці та на основі цих результатів обрано модель машинного навчання, а точніше нейронну мережу EfficientNet B1 для продовження роботи над даною магістерською роботою.

4 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Вимоги до програмного забезпечення

Створений програмний продукт має відповідати наступним нефункціональним вимогам:

- мова інтерфейсу – англійська;
- операційна система смартфона має бути: IOS 14 або вище чи Android 4.4 або вище.

При створенні мобільного додатку для смартфона було виділено шляхи використання, описані на рисунку 4.1:

- ідентифікація захворювання рослини та шляхів її лікування за фото з галереї. Так як цей шлях включає в себе визначення хвороби рослини, цей шлях є одним з найважливіших у списку. Але, враховуючи те, що необхідно надавати користувачеві можливість переглядати історію ідентифікацій хвороб, важливо щоб користувач отримував доступ до цього шляху використання лише після автентифікації в мобільному додатку та системі загалом. Цей сценарій детально описаний в таблиці 4.1;

- огляд попередніх результатів передбачень для конкретного користувача. Саме цей шлях використання робить користування додатком більш комфортним у перспективі, адже дає можливість аналізувати власні патерни догляду за рослинами, наприклад. Звичайно, для того, щоб користувач мав доступ до своєї історії, програмний продукт має мати змогу ідентифікувати конкретного користувача, а отже, обов'язковою умовою використання вказаного сценарію є автентифікація користувача. І звичайно, необхідно, щоб в користувача було хоча б одне попереднє сканування. Даний шлях використання детально описаний в таблиці 4.2;

- ідентифікація захворювання рослини та шляхів її лікування за фото з камери. Так як цей шлях включає в себе визначення хвороби рослини, так само як і сценарій, описаний в таблиці 4.1, цей шлях є одним з найважливіших у списку. Так само, як і для шляху використання, описаного в таблиці 4.1, важливо щоб користувач

отримував доступ до цього шляху використання лише після автентифікації в мобільному додатку. Вказаний сценарій детально описаний в таблиці 4.3;

— реєстрація користувача в системі. Саме реєстрація користувача в системі робить можливим повноцінне використання шляхів використання, описаних в таблиці 4.1, таблиці 4.2 та таблиці 4.3, адже саме завдяки цьому сценарію стає можлива автентифікація користувача в системі, тобто ідентифікація конкретного користувача в системі як такого. Саме цей сценарій покроково і детально описано в таблиці 4.4;

— автентифікація користувача в системі. Автентифікація користувача в системі робить можливим використання сценаріїв, описаних в таблиці 4.1, таблиці 4.2 та таблиці 4.3, але вона не є можливою без реєстрації користувача, тобто шляху використання, описаного в таблиці 4.4. Саме після авторизації користувач отримує токен, який перевіряється при надсиланні запитів на сервер. Цей шлях використання описано в таблиці 4.5;

Також, не менш важливими при створенні програмного продукту є функціональні вимоги. Після аналізу, в даній системі було виділено такі функціональні вимоги:

— додаток має отримувати зображення з камери чи галереї користувача, відправляти отримане зображення на бек енд сервіс та демонструвати отриману від бек енд сервісу відповіді у читабельному для користувача вигляді. Якщо скоротити, то користувач має надавати системі зображення листя хворої рослини, а система має повертати користувачеві текстову інформацію про назву рослини, її хворобу, якщо вона хвора, або інформацію, що рослина здорова, а також, у разі, якщо рослина все ж має якусь хворобу, надавати розгорнутий текст про лікування цієї рослини. Дана вимога описана в таблиці 4.6;

— Мобільний застосунок, а точніше його функціонал, такий як ідентифікація хвороби рослини, має бути доступний користувачу, автентифікованому за логіном(електроною поштою) та паролем. Дана функціональна вимога детально описана в таблиці 4.8;

— додаток має надавати історію сканувань (тобто зображення) користувача. По натисканню на картинку користувач має отримати результат історичного сканування моментально. Під моментально мається на увазі без повторного сканування, а лише за зверненням в базу даних через сервер. Дана вимога описана в таблиці 4.7.

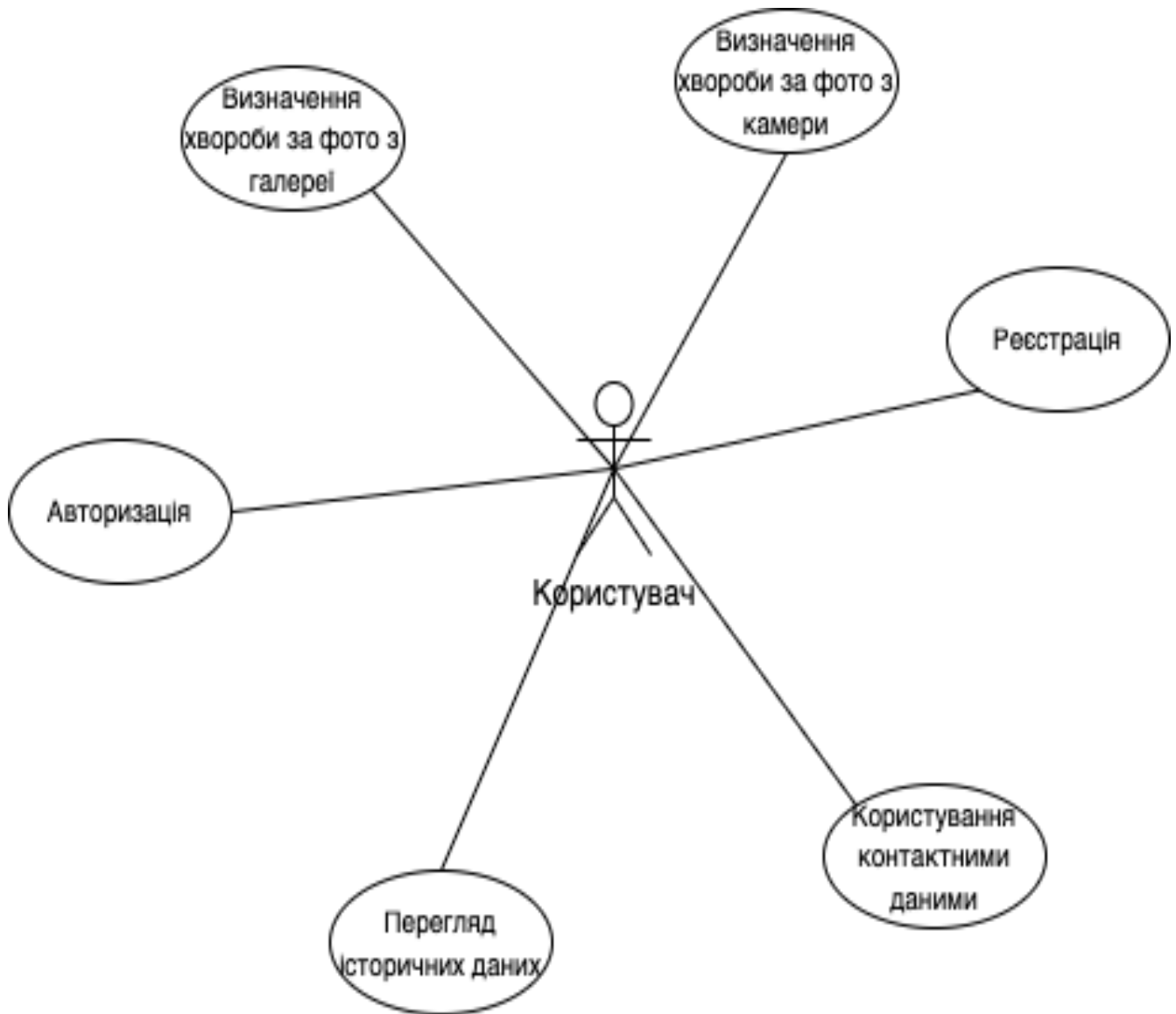


Рисунок 4.1 – Схема варіантів використання

Таблиця 4.1.

Шлях використання UC001

Назва	Ідентифікація захворювання рослини та шляхів її лікування за фото з галереї
-------	---

Продовження таблиці 4.1.

Опис	Проводиться класифікація наданого зображення до підходящого захворювання та співставлення захворювання з налаштованими шляхами лікування
Учасники	Користувач мобільного додатку
Передумови	Наявна рослина. Користувач автентифікований в системі.
Післяумови	Користувач має назву хвороби рослини із зображення та шлях її лікування
Основний сценарій	1) Користувач запускає мобільний додаток; 2) Користувач натискає на Choose photo; 3) Користувач натискає на фото в галереї; 1) Жме кнопку predict; 4) Програма виконує обробку наданої картинки, класифікує, співставляє з хворобою та відображає результати.

Таблиця 4.2.

Шлях використання UC002

Назва	Огляд попередніх результатів передбачень для конкретного користувача
Опис	Передбачення хвороби мають бути збережені з прив'язкою до конкретного користувача та, користувач може отримати ці історичні дані за запитом в додатку
Учасники	Користувач мобільного додатку

Продовження таблиці 4.2.

Передумови	Користувач автентифікований в системі. Користувач один чи більше разів використовував UC001 з акаунту в якому він на разі автентифікований.
Післяумови	Користувач бачить історію попередніх ідентифікацій рослин як сітку зображень та може побачити деталі по кожній історичній ідентифікації.
Основний сценарій	1) Користувач відкриває додаток; 2) Переходить на сторінку історії; 3) Натискає на одне з зображень; 4) Додаток показує отримані раніше результати користувачеві.

Таблиця 4.3.

Шлях використання UC003

Назва	Ідентифікація захворювання рослини та шляхів її лікування за фото з камери
Опис	Проводиться класифікація наданого зображення до підходящого захворювання та співставлення захворювання з налаштованими попередньо шляхами лікування для рослини
Учасники	Користувач мобільного додатку
Передумови	Наявна рослина. Користувач автентифікований в системі.
Післяумови	Користувач має назву хвороби рослини із зображення та шлях її лікування

Продовження таблиці 4.3.

Основний сценарій	1) Користувач запускає мобільний додаток; 2) Користувач натискає на Create photo; 3) Користувач робить фото на вбудовану камеру додатку; 4) Жме кнопку predict; 5) Програма виконує обробку наданої картинки, класифікує, співставляє з хворобою та відображає результати.
-------------------	---

Таблиця 4.4.

Шлях використання UC004

Назва	Реєстрація користувача в системі
Опис	Користувач вказує особисті дані та отримує доступ до системи
Учасники	Користувач мобільного додатку
Передумови	В користувача є електронна пошта
Післяумови	Користувач може користуватися системою та автентифікуватися з даними, введеними при реєстрації – а саме – електронною поштою та паролем.
Основний сценарій	1) Користувач запускає додаток; 2) Користувач вводить запитану інформацію у відповідних полях форми для реєстрації 3) Жме – «Register».

Таблиця 4.5.

Шлях використання UC005

Назва	Автентифікація користувача.
Опис	Користувач вказує дані, попередньо введені при реєстрації, а саме електронну пошту та пароль, та отримує доступ до системи.
Учасники	Користувач мобільного додатку.
Передумови	Користувач пройшов UC004 – тобто реєстрацію в мобільному додатку.
Післяумови	Користувач має доступ до функціоналу ідентифікації хвороб та історії ідентифікації.
Основний сценарій	1) Користувач запускає додаток; 2) Користувач вказує дані, попередньо введені при реєстрації в полях для авторизації; 3) Жме кнопку Login

На основі описаних вище шляхів використання додатку необхідно створити готовий програмний продукт, який буде відповідати наступним функціональним вимогам:

Таблиця 4.6.

Функціональна вимога REQ001

Номер	REQ001
Назва	Аналіз вхідних даних

Продовження таблиці 4.6.

Опис	Додаток має отримувати зображення з камери чи галереї користувача, відправляти отримане зображення на бек енд сервіс та демонструвати отриману від бек енд сервісу відповіді у читабельному для користувача вигляді.
------	--

Таблиця 4.7.

Функціональна вимога REQ002

Номер	REQ002
Назва	Демонстрація історичних даних
Опис	Додаток має надавати історію сканувань (тобто зображення) користувача. По натисканню на картинку користувач має отримати результат історичного сканування (тобто назву рослини, хворобу та шлях її лікування) миттєво.

Таблиця 4.8.

Функціональна вимога REQ003

Номер	REQ003
Назва	Авторизація
Опис	Користувач має мати змогу авторизуватись в мобільному додатку.

Таблиця 4.9.

Функціональна вимога REQ004

Номер	REQ004
Назва	Доступ до функціоналу мобільного застосунка

Продовження таблиці 4.9.

Опис	Мобільний застосунок, а точніше його функціонал, такий як ідентифікація хвороби рослини, має бути доступний користувачу, автентифікованому за логіном(електроною поштою) та паролем
------	---

4.2 Бізнес процеси

На основі проведеного аналізу та всіх описаних вище вимог було виділено перелік таких бізнес вимог:

— реєстрація в програмному продукті для збору даних, збереження історії ідентифікацій та доступу до них проілюстрована на рисунку 4.2. Також цей сценарій робить можливим збір даних для донавчання моделі [6], що робить можливим розвиток проєкту на різні сфери і ринки, а також робить користувача більш задоволеним завдяки тому що при донавчанні на реальних даних значно підвищується точність;

— користувач має мати можливість після створення фотографії ураженого листа через вбудовану камеру у додатку, або завантаживши зображення з галереї пристрою, на якому встановлено мобільний додаток та відправивши обране зображення на обробку серверній частині програмного продукту і отримати назву рослини, її захворювання та методи лікування даного захворювання, що ретельно показано на рисунку 4.3;

— автентифікованому користувачеві, яких попередньо виконував ідентифікації хвороб, має бути доступна історія всіх ідентифікацій, включно із зображенням, хворобою рослини та методу її лікування без додаткової роботи моделі машинного навчання (щоб запобігти показанню користувачеві різних результатів, хоч ймовірність і невелика). Весь цей описний вище бізнеспроцес описаний на рисунку 4.4.

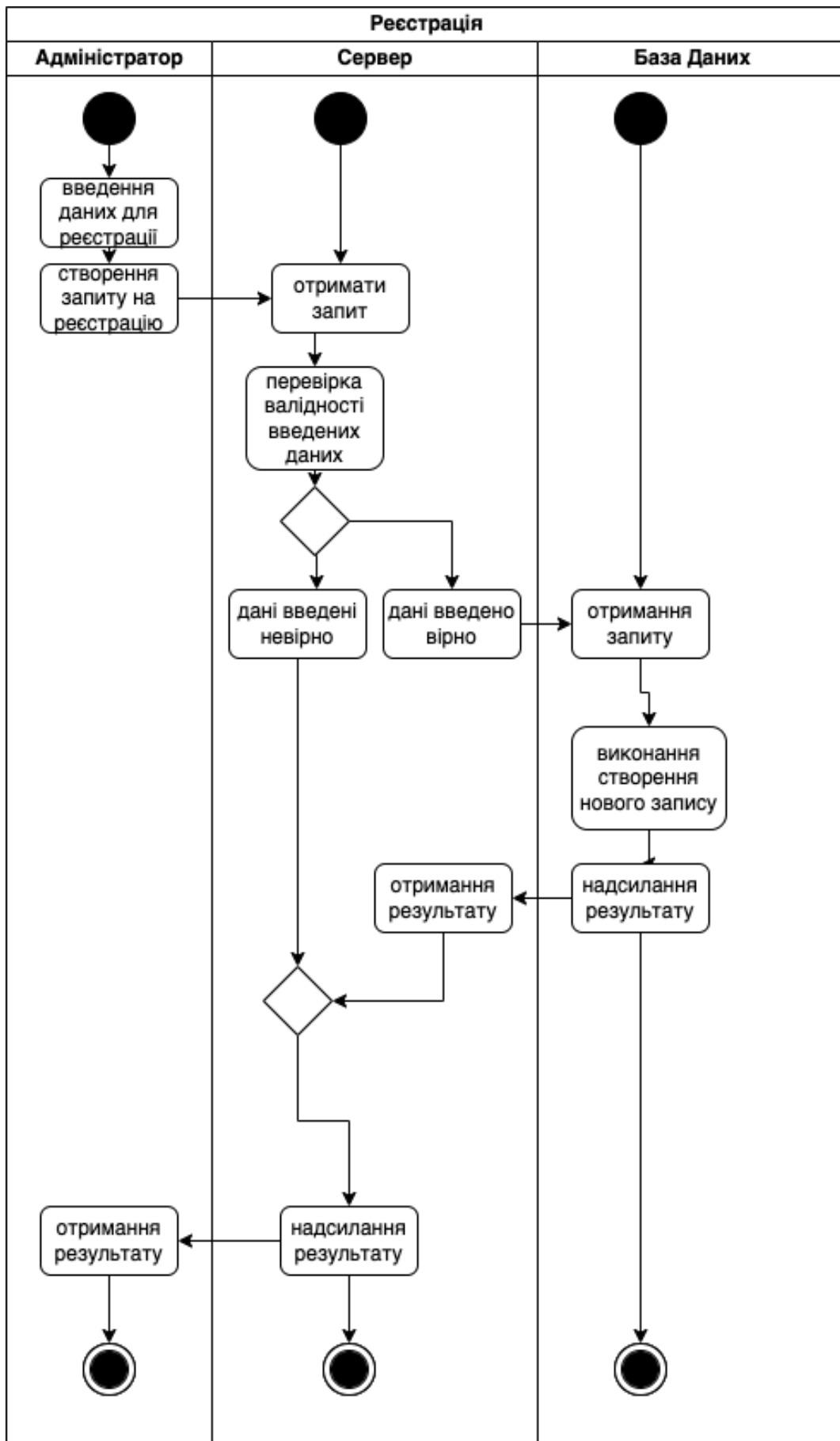


Рисунок 4.2 – Діаграма бізнес процесу реєстрації

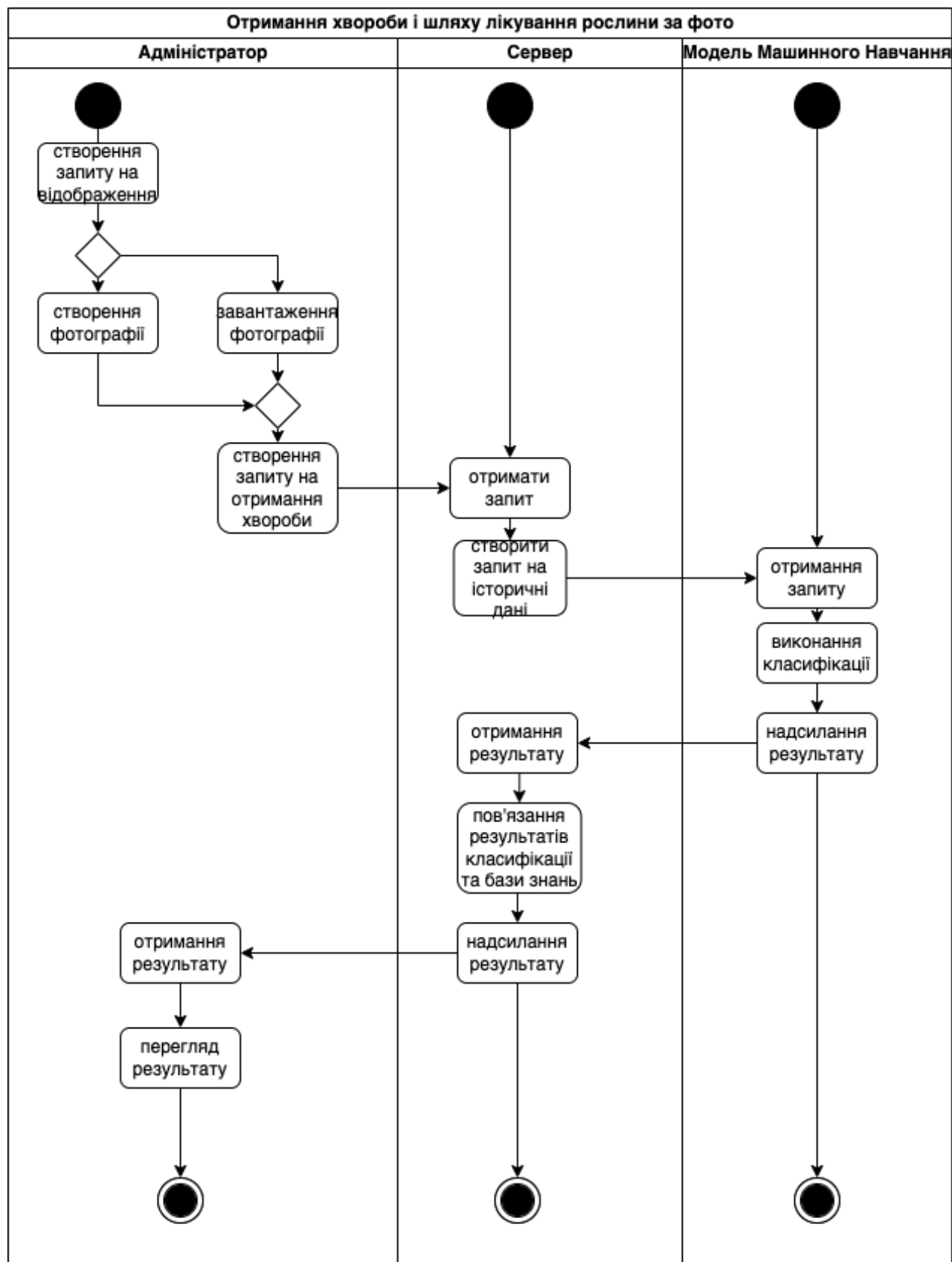


Рисунок 4.3 – Діаграма бізнес процесу ідентифікації захворювання рослини за фотографією листа

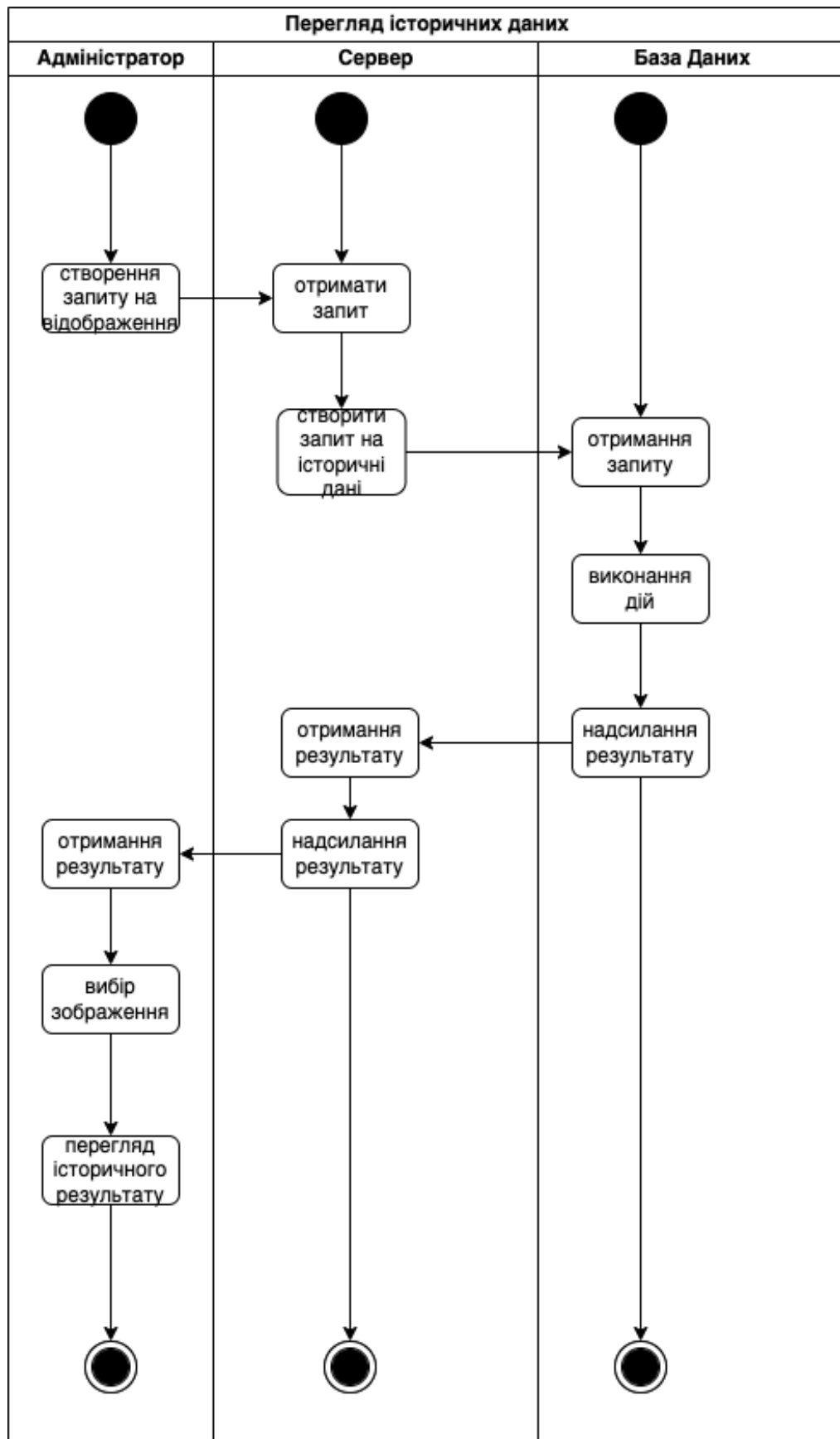


Рисунок 4.4 – Діаграма бізнес процесу ознайомлення користувача із історією сканувань

4.3 Проектування архітектури програмного забезпечення

Створений програмний продукт має за основу клієнт-серверну архітектуру, а саме складається з:

- мобільний застосунок, з яким на пряму взаємодіє користувач;
- серверу обробки який отримує запити від мобільного застосунку, та взаємодіє з сервісом моделі машинного навчання [5, 9];
- сервіс моделі машинного навчання;
- база даних для збереження даних користувача.

Архітектура створеної системи відображена на рисунку 4.5:

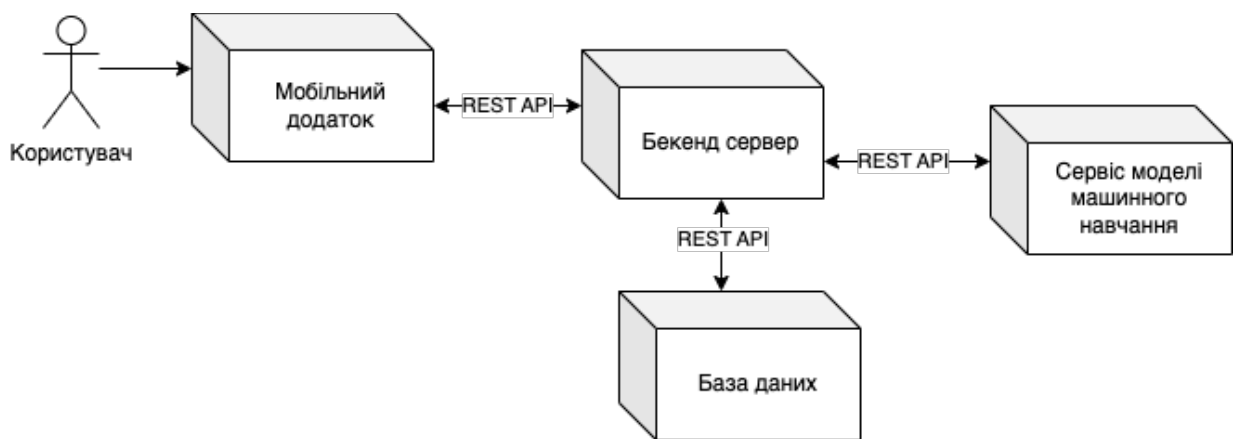


Рисунок 4.5 – Архітектура створеної системи

4.3.1 Мобільний додаток

В якості клієнтської сторони було створено мобільний застосунок, який надає можливість завантажувати фотознімок з галереї чи створювати фотознімок через вбудовану фотокамеру та відправки на обробку на серверну частину та подальшого відображення отриманих результатів з серверної частини.

Архітектура створеного мобільного застосунку показана на рисунку 4.6.

На діаграмі з рисунку 4.6 показано класи, що демонструють більшість функціоналу мобільного додатку [14]. Усі класи поєднані між собою, так як для створення та продовження роботи додатку було обрано для реалізації бібліотеку MobX, яка передумовлює пов'язаність всіх класів, що мають наслідування від основної бібліотеки.

Усі класи мобільного додатку мають обмежений набір методів, жоден з яких не включає в себе обчислень.

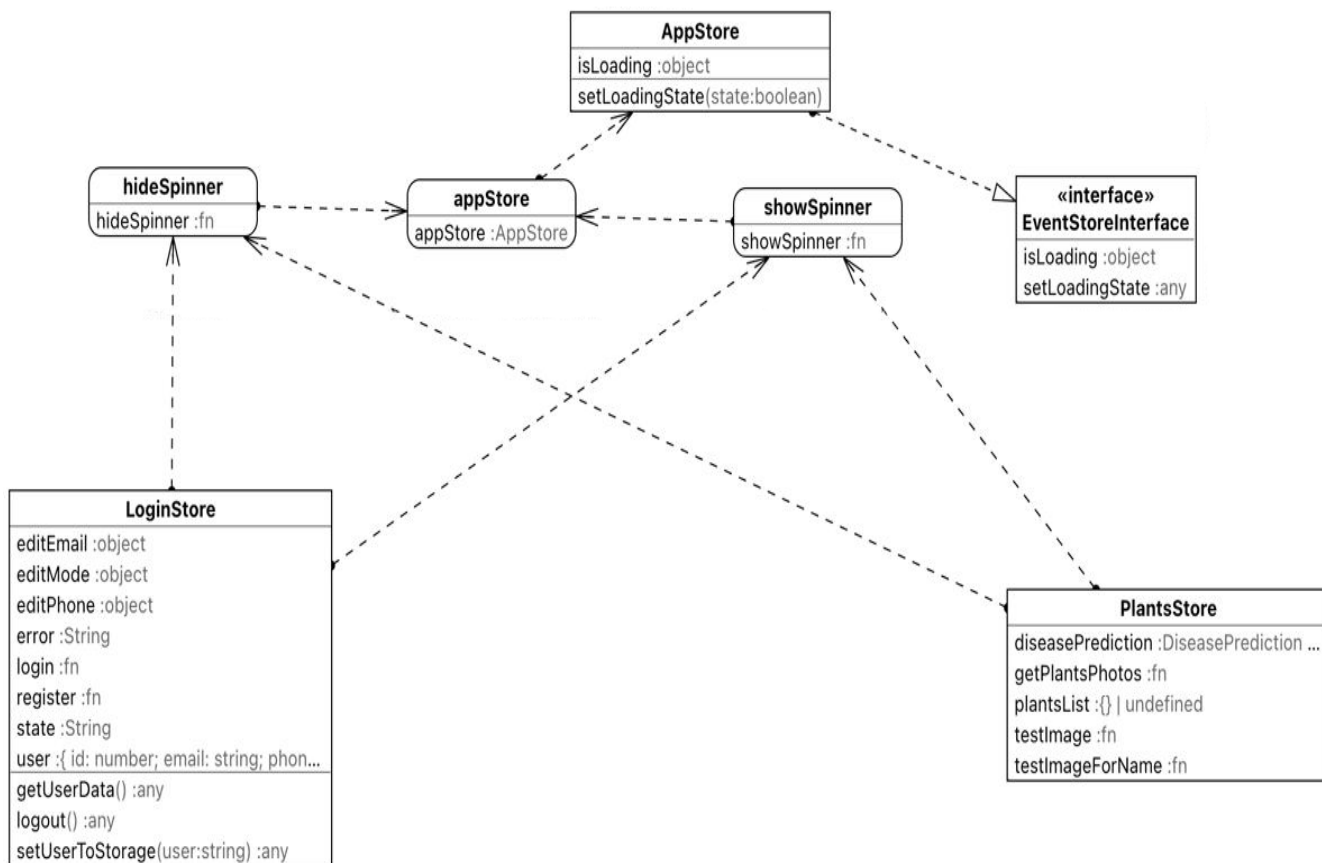


Рисунок 4.6– Діаграма ієрархії класів мобільного застосунку

4.3.2 Серверна частина

З діаграми на рисунку 4.7 зображена ієрархія класів бек енд частини програмного продукту. Найбільше зв'язків видно через те, що в системі наявна авторизація і автентифікація [11], через яку необхідно додатково захистити всі компоненти від стороннього доступу.

Також, по стандартам доменної сфери, на сторону сервера перенесені всі обчислення, адже обчислення на стороні клієнта є неефективними через використання виключно центральних юнітів обробки, в той час як на стороні сервера таких обмежень немає. Також Важливим є те, що обчислення на стороні клієнта є небезпечними.

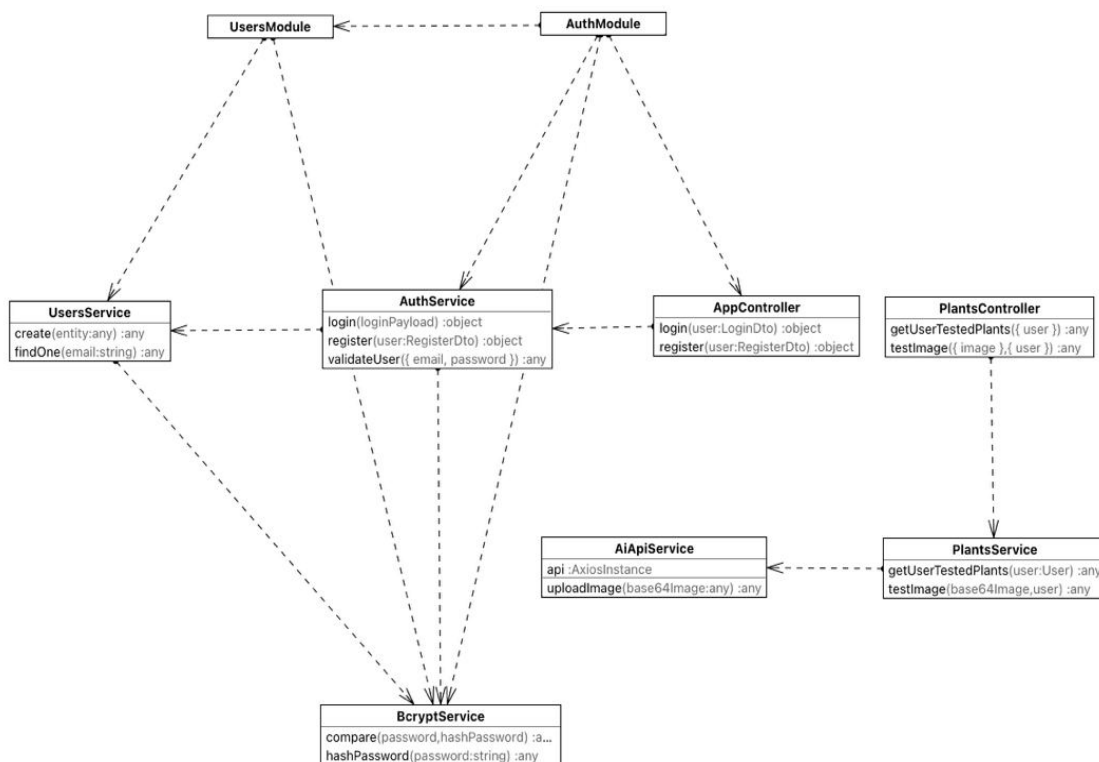


Рисунок 4.7 – Діаграма ієрархії класів бекенд сервісу

4.3.3 База даних

При створенні архітектури бази даних перше що необхідно зробити – це обрати систему управління базами даних. Для даного проєкту було обрано PostgreSQL

PostgreSQL - це система управління об'єктно-реляційними базами даних з відкритим кодом, ORDBMS, яка не належить або контролюється однією компанією чи особою. Оскільки програмне забезпечення postgresSQL є відкритим кодом, ним керують здебільшого за допомогою скоординованих онлайн-спроб активним світовим співтовариством розробників, ентузіастів та інших добровольців. Вперше випущений в середині 1990-х, postgresSQL написано на С. Основними конкурентами є Oracle DB, SQL Server та MySQL[3].

Плюси використання Postges в тому, що система є безкоштовною, працює з найпоширенішою мовою запитів в базах даних SQL та надає користувачеві такі методи керування:

- складні запити;
- зовнішні ключі;

- тригери;
- змінювані уявлення;
- транзакційна цілісність;
- багатOVERсійність.

А завдяки вільній ліцензії, PostgreSQL дозволяється безкоштовно використовувати, змінювати і поширювати всім і для будь-яких цілей - особистих, комерційних чи навчальних [12].

4.4 Вибір засобів розробки програмного продукту

Засобом девелопменту та виконання навчання моделей та виконання девелопменту ПЗ для даної праці обрано Python, який побачив світ у 1991 році і наразі є дуже популярною мовою програмування.

Python — це мова програмування високого рівня загального призначення. Його філософія дизайну наголошує на читабельності коду з використанням значних відступів [7].

Використано мови та фреймворки:

- Python;
- TypeScript;
- Flask;
- MobX;
- Expo.

У той же час тренування моделей машинного навчання, процесінг даних, вивчення даних здійснено за допомогою наступних технологій, засобів та фреймворків:

- pytorch;
- numpy.

Мобільний застосунок було розроблено за допомогою наступних мов та фреймворків:

- Flask;
- React Native.

Далі детальний розгляд описаних мов, фреймворків, засобів, аналіз їх плюсів та мінусів.

4.4.1 Бібліотека NumPy

NumPy - це бібліотека, загально-доступний модуль мови програмування Python код якої відкритий. NumPy застосовується для роботи з масивами, NumPy має здатність для роботи в сфері перетворення Фур'є та матриць, лінійної алгебри.

NumPy (Numeric Python) дозволяє базові методи для процесінгу з великими матрицями і масивами [8].

NumPy дозволяє використовувати і числові і математичні перетворення у вигляді прескомпільованих, швидких функцій. Функції надалі роблять високорівневі пакети, які забезпечують схожий функціонал з функціоналом MatLab.

Недоліки NumPy:

Що стосується недоліків, то, якщо порівняти масиви вбудовані в мову програмування Python і масиви, вбудовані у NumPy, другі є строго типізованими, тобто в масиві не може бути елементів різного типу. Ще один недолік це необхідність що використання numpy тільки з іноді PyPy CPython.

Переваги NumPy:

— стрімка обробка даних у вигляді векторів та матриць. Стрімка обробка даних досягнута за рахунок того, що більшість функціоналу NumPy розроблено на C++;

— при розробці векторів, матриць, та інших об'єктів реалізована оптимізація пам'яті;

— Cython – для кодування власного коду за допомогою numpy для векторизації довільної операції.

4.4.2 Фреймворк Pytorch

PyTorch — це платформа машинного навчання на основі бібліотеки Torch яка використовується для таких програм, як комп'ютерне бачення та обробка природної мови, спочатку розроблена Meta AI, а тепер є частиною парасольки Linux Foundation.

PyTorch це науково-обчислювальний пакет на основі мови програмування Python, який працює на базі графічних процесорів. PyTorch є представником найкращих дослідницьких платформ машинного навчання.

Це безкоштовне програмне забезпечення з відкритим кодом. PyTorch розповсюджується за модифікованою ліцензією BSD. PyTorch має інтерфейс C++. Хоча інтерфейс Python є більш якісним і основним напрямком розробки.

На основі PyTorch створено ряд програм для глибокого навчання, зокрема Tesla Autopilot, Uber's Pyro, Hugging Face's Transformers, PyTorch Lightning і Catalyst.

Переваги Pytorch:

- Pytorch надає архітектуру за допомогою якої розробник може змінювати під час виконання динамічні обчислювальні операції;
- динамічні обчислювальні операції;
- зручний інтерфейс;
- Pytorch плавно інтегрується зі усім що пов'язано знанням про дані Python. З цього можна зробити висновок, що Pytorch може використовувати всі функції, які розроблені для мови програмування Python.

4.4.3 Фреймворк Flask

Flask — це мікро-веб-фреймворк, написаний на Python, він був розроблений для девелопменту веб-застосунків.

У Flask відсутній рівень абстракції бази даних, аналіз форми або будь-яких інших компонентів, бібліотеки сторонніх розробників додають потрібні функції.

Існує багато розширень до Flask, розширення додають нові можливості девелоперу. Одні з найважливіших є розширення для:

- перевірки форм;
- контролю процесу завантаження;
- маніпуляцій об'єктно-реляційних зв'язків;
- підтримки відкритих технологій аутентифікації.

Переваги Flask : легкість в розробці програмного забезпечення, робота з cookie на стороні веб-браузера, система менеджменту управління запитами REST, робота з

шаблонами Jinja2, чітка документація. Існує багато розширень розроблених іншими розробниками, можливість працювати з функціоналом інших мов та сумісність з Google App Engine.

Flask дає можливість використовувати швидкість розробки на фреймворку, сумісний з RESTful API, сумісний з Unicode.

Недоліки Flask: важкість організації архітектури веб-застосунків та розробки складних багаторівневих веб-застосунків.

4.4.4 Фреймворк React Native

React Native — це фреймворк програмного забезпечення з відкритим вихідним кодом, створений Meta Platforms, Inc. Він використовується для розробки програм для різних платформах, від смартфонів до смарт ТВ, дозволяючи розробникам використовувати фреймворк React разом із власні можливості платформи. Він також використовується для розробки програм віртуальної реальності в Oculus.

Увесь код компілюється у нативний код для обраної платформи, що робить його достатньо швидким та популярним серед розробників. Він створює інтерфейс юзера з використанням бібліотеки JavaScript.

Переваги React Native:

- використання Javascript та ReactJS, це робить спрощення пошуку розробників через схожість платформ;
- функція гарячого перезавантаження;
- покращений інтерфейс користувача та виняткова продуктивність;
- активна спільнота розробників;
- міжплатформовий підхід;
- перезавантаження. Ця функція економить багато часу та зусиль;
- модульний дизайн.

Недоліки:

- проблеми дебагу та сумісності. Дебагінг досить непростий. Він додає ще один рівень;
- при розробці під різні платформи, майже 30 процентів коду переробляється;

— невисокий стан управління пам'яттю.

4.4.5 Мова програмування TypeScript

TypeScript — це об'єктно-орієнтована мова програмування з відкритим вихідним кодом і строго типізована мова. Любий існуючий код JavaScript є дійсним кодом TypeScript. Це означає, що все, що написано в JavaScript, працює за допомогою TypeScript разом із додаванням розширених функцій. Код TypeScript легше інтегрувати в проекти JavaScript.

Для великого проекту його використання може призвести до більш надійного програмного забезпечення, яке разом можна розгортати і там, де буде працювати будь-яка програма JavaScript.

Спочатку JavaScript був розроблений як легка, проста у вивченні мова, головним чином зосереджена на простих маніпуляціях з DOM, але стандарти змінювалися з часом, і саме тут з'явився TypeScript, який додає покращені функції до JavaScript.

Підтримка класів і об'єктів також є причиною його зростаючої популярності, це полегшує розуміння та реалізацію концепцій OOPS що забезпечується рідним JavaScript. Використання TypeScript у JavaScript Frameworks, на кшталт Angular, допомогло TypeScript. Typescript застосовує базові будівельні блоки вашої програми з JavaScript. Увесь код TypeScript перероблюється на JavaScript для застосування.

TypeScript надає переваги: статична типізація, перевірка коректності коду на етапі компіляції, швидка розробка застосунків, уникнення помилок пов'язаних з некоректними типами параметрів, краще сприйняття коду, можливість покращення створеного коду з більшою швидкістю, більш зрозуміла робота коду.

Недоліками Typescript назвемо: вища складність для початківців, імовірність використання разом та Javascript, необхідність використання інших бібліотек для real-time компіляції та старту коду.

4.4.6 Інструмент керування станом MobX

MobX є одним із багатьох інструментів керування станом, доступних розробникам React. MobX - це бібліотека що доказала свою ефективність, яка дозволяє керувати станом додатків простим та масштабованим з зрозумілим застосуванням функціональної реактивної розробки [13].

Переваги MobX:

- мінімалістичний, шаблонний код, який відповідає вашим намірам, асинхронне оновлення даних, під час оновлення даних в асинхронному процесі не потрібні спеціальні інструменти;
- MobX дозволяє контролювати стан вашої програми поза будь-якої схеми інтерфейсу юзера. Програма виходить з незв'язним, портативним кодом, який також і простий для тестування.

4.4.7 Мова програмування Python

Python — це мова розробки високого рівня і використовується для загального призначення. Його філософія дизайну - читабельність коду з використанням відступів. Python використовує динамічну типізацію а також вбудований колектор сміття. Він підтримує кілька парадигм розробки, структуроване функціональне програмування, процедурне, об'єктно-орієнтоване програмування.

Гвідо ван Россум почав працювати над Python наприкінці 1980-х як наступником мови програмування ABC і вперше випустив її в 1991 році. Синтаксис ядра мови мінімалістичний.

Python використовує легкий і доступний синтаксис та дозволяє швидко програмування будь-яких застосунків. Наразі Python застосовують для розробки веб-додатків, бекенду, кодування обчислень та кодування системних скриптів. Недоліком Python є повільне виконання на відміну від компільованих мов програмування, таких як C++ або C.

4.5 Функціональність програмного забезпечення

Розроблений мобільний додаток має наступний функціонал:

- створення фото або його завантаження з галереї мобільного девайсу;
- відправка отриманого зображення на обробку;
- відображення отриманого знімку разом з інформацією про хворобу рослини та шлях її лікування;
- реєстрація користувача в системі;
- збереження історії ідентифікацій.

4.6 Вимоги до технічного забезпечення

Для його розгортання створеного мобільного додатку необхідно:

- оперативна пам'ять 2 ГБ і більше;
- мобільний пристрій з платформою IOS чи Android;
- вільна пам'ять на девайсі 30 МБ.

4.7 Керівництво користувача

Після активації мобільного застосунку користувач переходить на початковий екран, що зображений на рисунку 4.8.

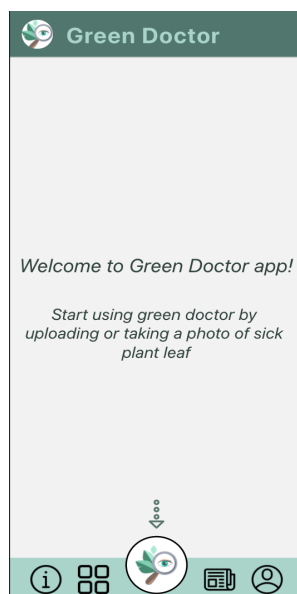


Рисунок 4.8 – Початковий екран

Автентифікувавшись в мобільному додатку, користувач бачить головну сторінку (рисунок 4.9).



Рисунок 4.9 – Головна сторінка

На рисунку 4.9 користувач обирає, як йому зручніше додати зображення в систему: чи зробивши фотознімок в реальному часі, чи завантажити вже готове фото, Після чого він переходить на екран з рисунку 4.10.

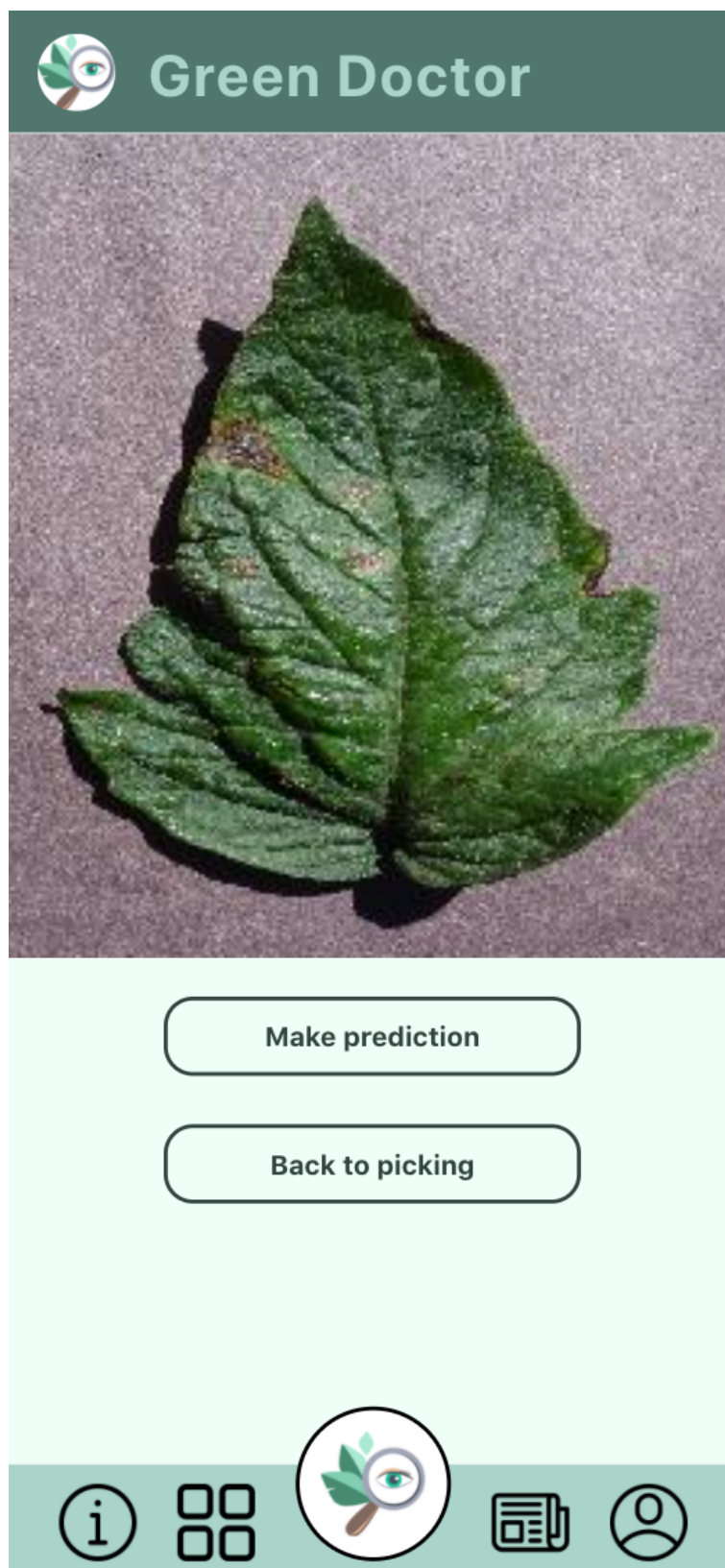


Рисунок 4.10 – Сторінка з кнопкою продовження процедури ідентифікації

Після того, як користувач забажає дізнатись хворобу та натисне відповідну кнопку, буде запущена процедура ідентифікації, кінцевою точкою якої буде на рисунку 4.11.



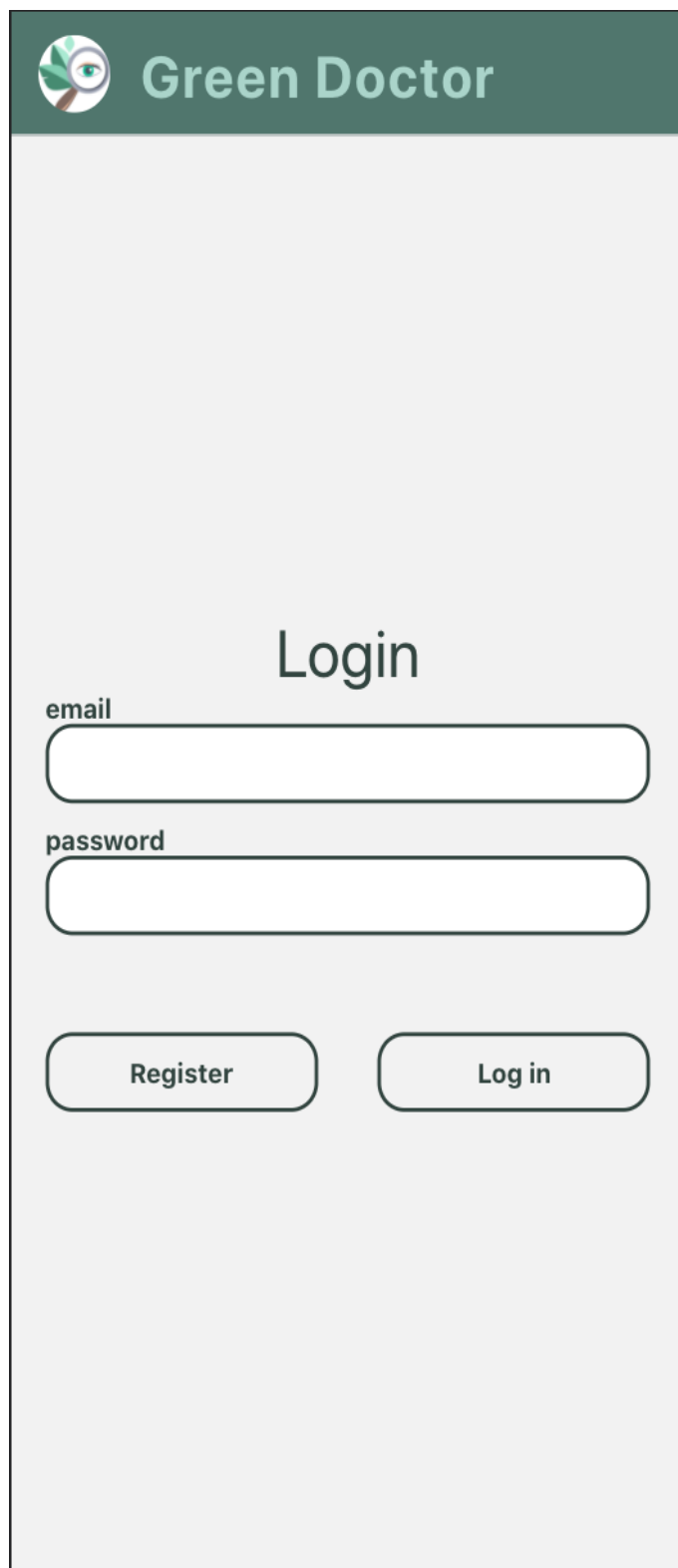
Рисунок 4.11 – Сторінка з захворюванням та шляхом лікування

При переході в галерею (рисунок 4.12) користувач бачить всі свої попередні сканування, а при натиску на зображення користувач перейде на детальну інформацію про сканування, відповідну рисунку 4.12.



Рисунок 4.12 – Галерея історичних сканувань

При деавторизації користувача, чи відкритті додатку перший раз, користувач побачить форму логіну з рисунку 4.13, з якої, в разі бажання, користувач може потрапити на сторінку реєстрації, тобто створення користувача, що показана на рисунку 4.14.



The image shows a mobile application login screen. At the top, there is a dark green header bar containing a circular logo with a stylized green leaf and a white eye-like shape on the left, and the text "Green Doctor" in a white sans-serif font on the right. Below the header, the main area has a light gray background. In the center, the word "Login" is displayed in a large, dark gray sans-serif font. Underneath "Login", there are two input fields. The first is labeled "email" in a small, dark gray font to its left, and the second is labeled "password" in a small, dark gray font to its left. Both labels are positioned above their respective input fields. At the bottom of the form, there are two rounded rectangular buttons. The left button is labeled "Register" and the right button is labeled "Log in", both in a dark gray sans-serif font.

Рисунок 4.13 – Сторінка логіну



Green Doctor

Registration

First name

Last name

Email

Phone

Password

Repeat password

Log in

Register

Рисунок 4.14 – Сторінка реєстрації

4.8 Висновки до розділу

В даному розділі детально описано технічну реалізацію програмного продукту, наведені діаграми взаємодії архітектурних елементів системи з ієрархічними уточненнями, описано ієрархічні структури фронт енд та бек енд частин клієнт-серверного архітектурного рішення.

Також в розділі вище було проаналізовано можливі сценарії використання програмного продукту кінцевим користувачем, тобто розписано сценарії, які надалі

можна використовувати для рекламування продукту, та розуміння вигідних сторін додатку.

Розділ вище також описує інструкцію користування мобільним додатком для користувача та наводить можливі екрани користування додатком для кінцевого користувача з описом вмісту кожного екрану, з кроками переходів між описаними екранами.

5 РОЗРОБКА СТАРТАП ПРОЄКТУ «ЗЕЛЕНИЙ ДОКТОР»

5.1 Опис ідеї

В попередніх розділах було розглянуто теоретичні аспекти та деталі імплементації системи що дозволяє ідентифікувати хворобу за зображенням ураженого листа. В даному розділі буде розглянуто стартап проєкт мобільного додатку для визначення хвороби за фотографією хворого листа в реальному часі.

Загальною ідеєю проєкту є створення мобільного застосунку, за допомогою якого можна сфотографувати, надіслати на обробку та отримати назву хвороби рослини та шляхи її лікування, застосовуючи алгоритми та методи, що були описані в розділах вище.

Структурована загальна ідея в таблиці 5.1.

Таблиця 5.1.

Опис ідеї стартап проєкту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Створення мобільного застосунку, за допомогою якого можна сфотографувати, надіслати на обробку та отримати назву хвороби рослини та шляхи її лікування	Визначення хвороби рослини за фото листа	Отримання напрямку для пошуку шляхів лікування
	Визначення шляху лікування рослини	Отримання здорової рослини
	Перевірка, чи є стан рослини хворобливим	Розуміння, чи потрібно приймати міри на ранніх етапах захворювання
	Збереження історії хвороб рослини	Вивчення власних методик утримання рослин

Отже, користувачеві надається новий автоматизований спосіб ідентифікації хвороби рослини, що не є прив'язаний до робочих годин спеціаліста, та робить з необхідності комунікації з біологами та агрономами майже рудиментом.

Розглянемо техніко-економічні переваги такого проекту відносно готових продуктів потенційних конкурентів:

- визначаємо перелік техніко-економічних властивостей та характеристик ідеї;
- визначаємо попереднє коло конкурентів (проектів-конкурентів) або товарів-замінників чи товарів-аналогів, що вже існують на ринку;
- проводимо збір інформації щодо значень техніко-економічних показників для ідеї власного проекту та проектів-конкурентів відповідно до визначеного вище переліку;
- проводимо порівняльний аналіз показників: для власної ідеї визначено показники, що мають а) гірші значення (W, слабкі); б) аналогічні (N, нейтральні) значення; в) кращі значення (S, сильні) (табл. 5.2).

Таблиця 5.2.

Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Техніко- економічні характерис- тики ідеї	(потенційні) товари/концепції конкурентів			W	N	S
		Мій проект	PlantInn Plant identifier	Blossom			
1.	Здатність працювати в режимі реального часу	+	+	–			+
2.	Автоматизоване надання інформації про хворобу рослини	+	–	–			+

Продовження таблиці 5.2

№ п/п	Техніко- економічні характерис- тики ідеї	(потенційні) товари/концепції конкурентів			W	N	S
		Мій проект					
3.	Збереження історії ідентифікації хвороб	+	–	–			+
4.	Здатність працювати на Мобільному пристрої	+	+	+		+	
5.	Створення особистого акаунту користувача	+	+	+		+	
6.	Створення календаря по догляду за росл инами	–	+	+	+		

Проаналізувавши два відомі аналогічні продукти, мною було виявлено, що мій додаток має переваги в здатності працювати в режимі реального часу, в тому, що в додатку є можливість ідентифікування хвороби та отримання шляхів лікування. Крім того моя система може зберігати історію ідентифікацій.

Серед недоліків моєї системи було виявлено те, що користувач не може створити в моєму додатку календар по догляду за рослинами з нагадуваннями про час поливу, прикорму, чи чогось іншого.

5.2 Технологічний аудит ідеї проєкту

Технологічна здійсненність проєкту описана в таблиці 5.3:

Таблиця 5.3.

Технологічна здійсненність ідеї проєкту

№ п/п	Ідея проєкту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Ідентифікація хвороб рослин за фото	Модель машинного навчання	Потребує розробки	Доступна за рахунок використання засобів з відкритим вихідним кодом
2	Авторизація користувача	Засоби розробки інформаційної системи	Потребує розробки	Доступна за рахунок використання засобів з відкритим вихідним кодом
3	Зберігання зображень, надісланих користувачем	Засоби розробки інформаційної системи	Потребує розробки	Доступна за рахунок використання засобів з відкритим вихідним кодом
Обрана технологія реалізації проєкту: Клієнт-серверна архітектура та використання моделі машинного навчання обгорнутої в сервіс				

5.3 Аналіз ринкових можливостей запуску стартап проекту

Визначимо ринкові можливості, які можна використати під час ринкового впровадження проекту, та ринкові загрози, які можуть перешкодити його реалізації. Саме аналіз ринку є важливою складовою процесу при створенні стартап проекту, адже саме поведінка ринку в загальному розумінні детермінує майбутнє створеного продукту.

Наприклад, при створенні продукту без аналогів необхідно включити в свою стратегію пояснення користувачеві процесів та продукту загалом, і саме такий вид стартап проєктів є найважчим для створення і рідко досягає успіху, навіть в порівнянні із загально дуже низьким відсотком успіху стартапів.

Це дозволяє розрахувати складність впровадження проекту на масовий ринок.

Спочатку проведемо аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку та структурувати дані в таблиці 5.4.

Таблиця 5.4.

Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	2
2	Загальний обсяг продаж, грн/ум.од	500
3	Динаміка ринку (якісна оцінка)	Стагнує
4	Наявність обмежень для входу (вказати характер обмежень)	Мала кількість даних для навчання.
5	Специфічні вимоги до стандартизації та сертифікації	Відсутні
6	Середня норма рентабельності в галузі (або по ринку), %	30%

Надалі визначаємо потенційні групи клієнтів, їх характеристики, та формуємо орієнтовний перелік вимог до товару для кожної групи клієнтів, що описано в таблиці 5.5.

Саме клієнт визначає долю стартап проекту, і саме розуміння клієнтської аудиторії є ключем до успіху, адже саме факт того, чи зацікавиться користувач продуктом, робить проект прибутковим чи ні.

Таблиця 5.5.

Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Потреба в лікуванні рослини у власності	Приватні особи	немає	Можливість отримання назви хвороби та шляку лікування в реальному часі
2	Потреба в оцінці методів догляду за рослиною на історичних даних	Приватні особи	немає	Можливість отримання історій ідентифікації з усіма теталями

Стартап має одну можливість для монетизації – продаж програмного забезпечення комунальним службам або приватним, публічним підприємствам.

Проаналізовані фактори потенційних загроз для майбутнього стартап-проекту. Результати наведені у таблиці 5.6.

Таблиця 5.6.

Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Невідомість	Клієнти надаватимуть перевагу особистому досвіду	Вкладення коштів у рекламу та маркетинг
2	Консерватизм	Не бажання клієнтів переходити на автоматизовану технологію	Вкладення коштів у рекламу та маркетинг

Основні фактори можливостей стартап-проекту, описані в таблиці 5.7:

Таблиця 5.7.

Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Залучення професійних агрономів для кращого надання рекомендації по лікуванню	Збільшення довіри до сервісу з точки зору користувачів	Виділення коштів на оплату роботи агрономів
2	Партнерство з агромагазинами	Покращення досвіду користувача	Залучення партнерів в проєкт

Ступеневий аналіз конкуренції на ринку описаний у таблиці 5.8:

Таблиця 5.8.

Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентноспроможною)
1. Вказати тип конкуренції - монополія/олігополія/ монополістична/чиста	Олігополія, так як на ринку існує декілька ключових гравців (PlantIn, Blossom)	Орієнтація на міжнародний ринок.
2. За рівнем конкурентної боротьби - локальний/національний/...	Інтернаціональний, так як основні гравці на ринку – інтернаціональні компанії	Вихід на міжнародний ринок, після отримання чіткого становлення на ринку України
Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентноспроможною)
3. За галузевою ознакою - міжгалузева/ внутрішньогалузева	Внутрішньогалузева, так як діяльність систем зосереджена на ідентифікації рослин	Дотримання основних концепцій та принципів функціонування сервісів в даній галузі

Продовження таблиці 5.8

4. За характером конкурентних переваг - цінова / нецінова	Цінова, так як сервіс є платним для використання користувачів	Підтримання високої конкуренції за рахунок унікальних переваг для користувача
---	---	---

Аналіз конкуренції в галузі за М. Портером описаний у таблиці 5.9:

Таблиця 5.9.

Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари замітники
	1. PlantIn 2. Blossom	Відсутні	Доступ до системи постачається за рахунок мережі Інтернет	Визначити фактори сили споживачів	Рекомендації та системи компаній-конкурентів
Висновки	Інтенсивність конкурентної боротьби зі сторони прямих конкурентів є низькою	Потенційні конкуренти відсутні	Відсутні постачальники, що диктують умови роботи на ринку	Відсутні клієнти, що диктують умови роботи на ринку	Робота системи обмежується чинним законодавством країни, у яку постачаються послуги

Обґрунтування факторів конкурентноспроможності описано в таблиці 5.10:

Таблиця 5.10.

Обґрунтування факторів конкурентноспроможності

№ п/п	Фактор конкурентноспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Ідентифікація хвороб рослини за фото ураженого листа	Завдяки цьому користувач зможе дізнатися хвороби рослини без залучення спеціаліста
2	Визначення шляхів лікування хвороби рослини	Завдяки цьому користувач зможе почати процес лікування

Порівняльний аналіз сильних та слабких сторін системи детально описано в таблиці 5.11:

Таблиця 5.11.

Порівняльний аналіз сильних та слабких сторін системи

№ п/п	Фактор конкурентноспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з системою ідентифікації хвороб рослин						
			-3	-2	-1	0	+1	+2	+3
1	Ідентифікація хвороб рослини за фото ураженого листа	20				+			
2	Визначення шляхів лікування хвороби рослини	20				+			

SWOT- аналіз стартап-проєкту описано в таблиці 5.12:

Таблиця 5.12.

SWOT- аналіз стартап-проект

Сильні сторони: – ідентифікація хвороб рослин; – надання шляхів лікування рослини; – збереження історії ідентифікації.	Слабкі сторони: – обмежений список модолвих хвороб для ідентифікації; – одна мова локалізації.
Можливості: – ідентифікація хвороб в реальному часі; – висока точність аналізу хвороби; – перегляд історії ідентифікації.	Загрози: – низька поширеність ринку; – обмеження по спроможності користувачів робити зображення.

Альтернативи ринкового впровадження описано в таблиці 5.13:

Таблиця 5.13.

Альтернативи ринкового впровадження

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Випуск MVP-версії сервісу в публічний доступ	Ресурси будуть надані інвесторами	2 місяці
2	Покращення візуальної складової сервісу	Відсутня	1 місяць

5.4 Розробка ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів.

При виборі цільової групи в першу чергу було керування розумінням, що господарства, які заробляють на агрокультурах, і для яких догляд за рослинами є критичною частиною циклу отримання прибутку, адже такі підприємства заробляють тільки коли в них купують вирощений врожай, не можуть ризикувати навіть сотими долями точності результату класифікації хвороби, і вже точно мають у штаті працівника, що відповідає за здоров'я рослин, їх лікування у випадку хвороб та створення умов, при яких рослина не буде хворіти взагалі, отримає за це зарплатню і має профільну освіту.

Саме тому користувачем даного продукту було обрано приватних осіб, що часто не мають необхідного досвіду для визначення хвороби рослини, але мають рослини у власності. Цей тип користувача і його характеристики детально описані в таблиці 5.14.

Таблиця 5.14.

Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Приватні особи	Середня	Є	Середня	Середньо

За результатами аналізу потенційних груп споживачів ми обрали цільові групи приватні або публічні підприємства різних розмірів, які можуть використовувати засоби моніторингу засмічень задля покращення стану своєї території.

Для роботи в обраному сегменті ринку необхідно сформулювати базову стратегію розвитку що описано в таблиці 5.15.

Таблиця 5.15.

Визначення базової стратегії розвитку

№ п/ п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Створення легкого для використання мобільного додатку для ідентифікації хвороб	Диференційован ий маркетинг	Визначення хвороби рослини в режимі реального часу; Надання історичної інформації.	Стратегія спеціаліза ції

Визначення базової стратегії конкурентної поведінки описано детально в таблиці 5.16:

Таблиця 5.16.

Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
1	Ні	Компанія буде шукати нових споживачів	Ні	Стратегія заняття конкретної ніші

Визначення стратегії позиціонування стартап-проекту з описом вимогами до готового продукту від цільової аудиторії детально описано в таблиці 4.17:

Таблиця 5.17.

Визначення базової стратегії конкурентної поведінки

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1	Точність визначення хвороби рослини	Стратегія спеціалізації	Підвищення точності ідентифікації хвороб	Рослини, хвороби, лікування
2	Збереження історії ідентифікації	Стратегія спеціалізації	Відображення історії ідентифікації користувача	Доцільність, адекватність, вичерпність.

Результатом даного підрозділу є детальна система рішень щодо ринкової поведінки компанії, вона визначає в якому напрямі буде працювати компанія на ринку.

5.5 Розроблення маркетингової програми стартап-проекту

Під час розроблення маркетингової програми першим кроком є розробка маркетингової концепції товару, який отримає споживач.

У таблиці 5.18 підсумовуємо результати аналізу конкурентоспроможності створеного товару в рамках даного проекту на загальному ринку споживання. Опис трьох рівнів моделі товару здійснений у вигляді таблиці 5.19

Таблиця 5.18.

Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Потреба в лікуванні власних рослин	Автоматичне рішення для визначення хвороби рослини та шляху її лікування	Можливість роботи в режимі реального часу; надання історичної інформації, надання даних про хворобу рослини автоматизовано.

Таблиця 5.19.

Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Задоволення потреби визначення хвороб рослин та шляхів їх лікування		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх/Тл/Е/Ор
	Дизайн	Нм	Вр/Е
	Зручність використання	Нм	Вр/Е/Тл
	Релевантність рекомендацій	М	Тх/Тл
	Безпека використання	Нм	Вр/Тх/Тл
	Якість: відповідає міжнародним стандартам інформаційної безпеки BS 7799-1: 2005, ISO/IEC 17799: 2005.		
II. Товар із підкріпленням	Інформацію про товар, та його переваги користувач може отримати за допомогою контекстної реклами.		
За рахунок чого потенційний товар буде захищено від копіювання: за рахунок захисту ідеї товару.			

Наступним кроком є визначення цінових меж, якими необхідно керуватися при встановленні ціни на потенційний товар, це передбачає аналіз цін товарів конкурентів, та доходів споживачів продукту що виконано в таблиці 5.20, таблиці 5.21 та таблиці 5.22.

Таблиця 5.21.

Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Купівля в приватних цілях	Робота з документами магазинів	1	Магазини мобільних додатків

Таблиця 5.22.

Концепція маркетингових комунікацій

№ п/ п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Розроблен ня додатку для ідентифіка ції хвороб	Інтернет	1.Робота в режимі реального часу	1. Донести до споживача назву 2. Переконати у зручності	Демонстрац ія роботи системи

Визначення меж встановлення ціни

№ п/п	Рівень цін на товари- замінники	Рівень цін на товари- аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	1000 грн \ міс.	70 грн \ міс.	Не важливо	700 - 350 грн.

5.6 Висновки до розділу

На даний момент, реалізацію системи автоматичного моніторингу та оцінки ступені засмічень на основі фотокамер можна віднести потенційно вигідного стартап-проекту, оскільки він має комерційні можливості для монетизації. Програмне забезпечення підходить для використання приватним особам, що мають хатні рослини, а таких людей дуже багато. В даному розділі було визначено ключові переваги та недоліки проекту, було порівняно його з потенційними продуктами конкурентів.

Було також проаналізовано та розроблено ринкову стратегію продукту, до якої входять вибір цільових груп потенційних споживачів та визначення базової стратегії розвитку.

В кінці розділу було розроблено маркетингову програму стартап-проекту, а саме:

- визначення ключових переваг концепції потенційного товару;
- визначення меж встановлення ціни;
- формування системи збуту;
- створення концепції маркетингових комунікацій.

ВИСНОВКИ

В даній дисертаційній роботі було автоматизовано процес визначення розповсюджених хвороб рослин шляхом розробки програмного продукту, що складається з мобільного додатку для смартфонів, серверної частини та сервісу, що є обгорткою над моделлю машинного навчання.

Було проведено огляд існуючих методів машинного навчання для класифікації зображень за результатом якого було визначено типи нейронних мереж для виконання подальших досліджень.

Було створено модель машинного навчання на основі згорткової нейронної мережі EfficientNet B1. Створена модель машинного навчання показала точність 97,93% при швидкості 85 фреймів на секунду на центральному юніті обробки, і 259,7 фреймів на секунду на графічному юніті обробки.

Для використання створеної моделі машинного навчання було визначено вимоги до програмного продукту та інформаційні моделі UML системи, спроектовано архітектуру програмного продукту, розроблено мобільний додаток як інтерфейс для взаємодії з кінцевим користувачем та бекенд сервіс для організації роботи розробленого мобільного застосунку та сервісу, в якому була реалізована модель машинного навчання. Для забезпечення можливості донавчання створеної моделі машинного навчання реалізовано зберігання завантажених зображень.

Сворений програмний продукт надає можливість кінцевому користувачеві в режимі реального часу визначити яку хворобу має рослина за фотографією ураженого листя та отримати інструкції щодо лікування цієї хвороби.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Марков І. Неінфекційні хвороби рослин: причини виникнення та діагностика [Електронний ресурс] / Іван Марков. – 2022. – Режим доступу до ресурсу: <https://agronomy.com.ua/statti/1083-neinfektsiini-khvoroby-roslyn-prychyny-vynyknennia-ta-diahnostyka.html>.
2. Черлінка В. Хвороби Рослин: Види, Лікування Та Профілактика [Електронний ресурс] / Василь Черлінка // EOS Data Analytics. – 2022. – Режим доступу до ресурсу: <https://eos.com/uk/blog/hvoroby-roslyn/>.
3. Що таке postgresql? [Електронний ресурс] // uk.theastrologypage.com. – 2022. – Режим доступу до ресурсу: <https://uk.theastrologypage.com/postgresql>.
4. Krizhevsky A. ImageNet Classification with Deep Convolutional Neural Networks [Електронний ресурс] / A. Krizhevsky, I. Sutskever, G. Hinton – Режим доступу до ресурсу: <https://papers.nips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf>.
5. Architectural Services [Електронний ресурс] – Режим доступу до ресурсу: <https://www.bc-architecture.com/architectural-services/#:~:text=The%20services%20that%20an%20architect,architectural%20programming%20and%20project%20management>.
6. Brownlee J. A Gentle Introduction to Transfer Learning for Deep Learning [Електронний ресурс] / Jason Brownlee. – 2017. – Режим доступу до ресурсу: <https://machinelearningmastery.com/transfer-learning-for-deep-learning/#:~:text=Transfer%20learning%20is%20the%20improvement,that%20has%20already%20been%20learned.&text=Transfer%20learning%20is%20related%20to,of%20study%20for%20deep%20learning>.
7. Python. [Електронний ресурс]. - 2020 - : <https://www.python.org/about/>
8. Numpy. [Електронний ресурс]. - 2020 - : <https://numpy.org/>

9. Microservices architecture style [Электронный ресурс] // Microsoft Documentation. – 2019. – Режим доступа до ресурсу: <https://docs.microsoft.com/en-us/azure/architecture/guide/architecture-styles/microservices>
10. Abade A. Plant Diseases recognition on images using Convolutional Neural Networks: A Systematic Review [Электронный ресурс] / A. Abade, P. Ferreira, F. Vidal. – 2020. – Режим доступа до ресурсу: <https://arxiv.org/pdf/2009.04365>.
11. Understanding Authentication, Authorization, and Encryption [Электронный ресурс] – Режим доступа до ресурсу: <https://www.bu.edu/tech/about/security-resources/bestpractice/auth/>.
12. Yıldırım S. Practical Introduction to PostgreSQL [Электронный ресурс] / Soner Yıldırım. – 2020. – Режим доступа до ресурсу: <https://towardsdatascience.com/practical-introduction-to-postgresql-5f73d3d394e>.
13. Maldonado L. The Guide to MobX [Электронный ресурс] / Leonardo Maldonado. – 2021. – Режим доступа до ресурсу: <https://www.telerik.com/blogs/guide-to-mobx>.
14. The Ultimate Guide to Mobile Application Architecture [Электронный ресурс]. – 2020. – Режим доступа до ресурсу: <https://scand.com/company/blog/the-ultimate-guide-to-mobile-application-architecture/>.
15. Kiprono Elijah Koech. Cross-Entropy Loss Function [Электронный ресурс] / Kiprono Elijah Koech. – 2020. – Режим доступа до ресурсу: <https://towardsdatascience.com/cross-entropy-loss-function-f38c4ec8643e>.
16. Deval Shah. The Essential Guide to Data Augmentation in Deep Learning [Электронный ресурс] / Deval Shah. – 2022. – Режим доступа до ресурсу: <https://www.v7labs.com/blog/data-augmentation-guide>.
17. Mingxing Tan, Quoc V. Le. “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks”. arXiv:1905.11946, 2019.
18. Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, Liang-Chieh Chen. “MobileNetV2: Inverted Residuals and Linear Bottlenecks”. arXiv:1801.04381, 2018.

19. Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam. “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications” - arXiv:1704.04861.
20. Deep Residual Learning for Image Recognition [Електронний ресурс] / К.Не, Х. Zhang, S. Ren, J. Sun // arxiv. – 2015. – Режим доступу до ресурсу: <https://arxiv.org/pdf/1512.03385.pdf>.
21. Загальна фітопатологія [Електронний ресурс] / Н. В.Пінчук, М. П. Вергелес, Т. М. Коваленко, С. Є. Окрушко // ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ АГРАРНИЙ УНІВЕРСИТЕТ. – 2018. – Режим доступу до ресурсу: <http://repository.vsau.org/getfile.php/21042.pdf>.
22. Deep Residual Learning for Image Recognition [Електронний ресурс] / К.Не, Х. Zhang, S. Ren, J. Sun // arxiv. – 2015. – Режим доступу до ресурсу: <https://arxiv.org/pdf/1512.03385.pdf>.
23. Смірнова О. Д. МОБІЛЬНИЙ ДОДАТОК ДЛЯ ВИЗНАЧЕННЯ ХВОРОБ РОСЛИН НА ОСНОВІ МЕТОДІВ МАШИННОГО НАВЧАННЯ / О. Д. Смірнова, О. М. Шушура. // «Сучасна молодь в світі інформаційних технологій». – 2022. – С. 98–100.

ДОДАТОК А

Мобільний додаток для визначення хвороб рослин на основі методів машинного навчання

Апробації

УКР.НТУУ «КПІ ім. Ігоря Сікорського»_ТР-1106мп_22М

Аркушів 5

2022

Міністерство освіти і науки України
Херсонський державний аграрно-економічний університет
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Вінницький національний медичний університет
ім. М. І. Пирогова
Кременчуцький національний технічний університет
ім. Михайла Остроградського
Вінницький національний технічний університет
Херсонський національний технічний університет
Сумський державний університет
Херсонська державна морська академія

Матеріали
III Всеукраїнської науково-практичної
інтернет-конференції
молодих вчених
та здобувачів вищої освіти
«СУЧАСНА МОЛОДЬ В СВІТІ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ»

присвячена Дню науки

16 травня 2022р.
Херсон-Кропивницький

УДК 004.7+004.05]:005.5](06)

С 91

- С 91 **«Сучасна молодь в світі інформаційних технологій»:** матеріали ІІІ Всеукраїнської науково-практичної інтернет-конференції молодих вчених та здобувачів вищої освіти присвяченої Дню науки (16 травня 2022 р., м. Херсон, м. Кропивницький) / за ред. Н.В. Кириченко, Г.О. Димової та ін. – Херсон-Кропивницький: Книжкове видавництво ФОП Вишемирський В.С., 2022. – 163 с.

ISBN 978-617-7941-79-7 (електронне видання)

Конференція «Сучасна молодь в світі інформаційних технологій» присвячується Дню науки. Метою конференції є висвітлення розробок, результатів досліджень та досягнень молодих вчених України та здобувачів вищої освіти при розробці, використанні та впровадженні інформаційних технологій в різних галузях науки.

Тези наукової конференції містять результати наступних досліджень: менеджмент інформаційних технологій; прогнозування соціально-економічних процесів за умов невизначеності та ризику; управління проектами на підприємствах агропромислового комплексу; сучасні тенденції розвитку інформаційних технологій; впровадження інновацій та сучасних технологій; інформаційні технології в науці, освіті, економіці, логістиці, туристичній сфері, транспорті; математичні методи, моделі, інформаційні системи і технології в економіці; моделювання та оптимізація інформаційних систем; інвестиційне проектування в різних сферах суспільного життя; інформаційно-аналітичні та інформаційно-керуючі системи; системи відображення інформації і комп'ютерні технології; використання нових інформаційних технологій в медичній галузі; новітні технології в енергетичних системах та в галузі енергозбереження.

Роботи друкуються в авторській редакції, в збірці максимально зменшено втручання в обсяг та структуру відібраних до друку матеріалів. Редакційна колегія не несе відповідальності за достовірність інформації, що надано в рукописах, та залишає за собою право не розподіляти поглядів деяких авторів на ті чи інші питання.

АДРЕСА ОРГКОМІТЕТУ

73006, Україна, м. Кропивницький, Університетський проспект, 5/2
Херсонський державний аграрно-економічний університет, економічний факультет
кафедра менеджменту та інформаційних технологій
e-mail: conference.mywit@gmail.com, matematika_ek2017@ukr.net

УДК 004.7+004.05]:005.5](06)

ISBN 978-617-7941-79-7 (електронне видання)

- © Херсонський державний аграрно-економічний університет, 2022
- © Видавництво ФОП Вишемирський В.С., 2022

ЗМІСТ

СЕКЦІЯ «МЕНЕДЖМЕНТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ»

Жосан Г.В., Хорошко А.Ф.	
Молодіжне підприємництво – досвід, реалії, перспективи	9
Лобода О.М., Григорюк О.І.	
Використання інформаційного менеджменту в сервісній діяльності	12
Узун А.С., Шушура О.М.	
Резервне копіювання в системах зовнішнього документообігу на прикладу сервісу Signy	14

СЕКЦІЯ «ПРОГНОЗУВАННЯ СОЦІАЛЬНО-ЕКОНОМІЧНИХ ПРОЦЕСІВ ЗА УМОВ НЕВИЗНАЧЕНОСТІ ТА РИЗИКУ»

Дебела І.М., Дедікова В.О.	
Статистичний прогноз тенденції динаміки об'ємів роздрібної торгівлі в Херсонській області на 2022 рік	17
Романчук А.О., Дашко І.М.	
Атестація персоналу: сутність та види	20
Толоконнікова А.В., Дашко І.М.	
Організація та цілі навчання персоналу на підприємстві	23

СЕКЦІЯ «СУЧАСНІ ТЕНДЕНЦІЇ РОЗВИТКУ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ»

Бісікало О.В., Юрчук М.С.	
Класифікація чат-ботів для електронної комерції	27
Федік Л.Ю., Черняк С.О.	
Проектування заземлення електроустановок систем автоматизації	30

СЕКЦІЯ «ВПРОВАДЖЕННЯ ІННОВАЦІЙ ТА СУЧАСНИХ ТЕХНОЛОГІЙ»

Варнавська І.В.	
Чинники ефективного впровадження інтерактивних технологій на уроках економіки... ..	33
Воєділо В.А.	
Обумовлення оптимальних виробничих потужностей для післядрукарської обробки поліграфічного замовлення	36
Дашко І.М.	
Роль інновацій в сучасній економіці	39
Кучевський С.М., Ратайчук П.Є.	
State-менеджери в контексті веб-розробки	41
Лазарєв О.В., Лазарєв В.О.	
Інформаційна система забезпечення технічного обслуговування та відновлювальних робіт у залізничній автоматизації	43
Розгон О.В.	
Етапи життєвого циклу інновацій і технологій: порівняльний аспект	45
Слободянюк К.О., Ключковський О.В.	
Інновації як фактор підвищення конкурентоспроможності продукції вітчизняного машинобудування	50

Смірнова О.Д., Шушура О.М. Мобільний додаток для визначення хвороб рослин на основі методів машинного навчання	98
--	----

СЕКЦІЯ «ІНВЕСТИЦІЙНЕ ПРОЄКТУВАННЯ В РІЗНИХ СФЕРАХ СУСПІЛЬНОГО ЖИТТЯ»

Берідзе Т.М. Інвестиційні ризики підприємства	102
Огренич Ю.О. Напрямки покращення фінансової та інвестиційної діяльності промислових підприємств України в сучасних економічних умовах	106
Федірко В.К. Визначення тенденцій впровадження інновацій у видавничо-поліграфічному комплексі	109

СЕКЦІЯ «ІНФОРМАЦІЙНО-АНАЛІТИЧНІ ТА ІНФОРМАЦІЙНО-КЕРУЮЧІ СИСТЕМИ»

Вдовиченко О.А. Вибір моделі організації корпоративної бази даних малої друкарської фірми	112
Дзівідзінська М.І., Очеретяний О.К. Організація робочого простору в IT-компаніях у період пандемії	114
Димова Г.О., Швидченко І.А. Організація інформаційного аналізу великих даних на базі інтелектуальної платформи.	117
Лазарєва Н.М. Застосування бази знань Сугено в моделі нечіткого керування складними об'єктами ...	120
Мирилко А.І., Комісаров О.С. Дослідження засобів підвищення ефективності та економічності числового програмного управління за допомогою Arduino	122
Федотова М.О., Дідик О.К., Скриннік І.О. Встановлення спектрального опису сигналів багатовимірних об'єктів з розподіленими параметрами за даними експерименту	124

СЕКЦІЯ «СИСТЕМИ ВІДОБРАЖЕННЯ ІНФОРМАЦІЇ І КОМП'ЮТЕРНІ ТЕХНОЛОГІЇ»

Валуєва В.В., Огісва О.Є. Дослідження можливостей Starlink в умовах гібридної війни	128
Головіна Н.В., Ляшенко О.М. Дослідження методів відновлення та реконструкції зображень	130
Кіцун А.В., Шахновський А.М., Квітка О.О. Автоматизоване налаштування параметрів регулятора в схемі виробництва циклогексанону	133
Радюк П.М., Калита О.Д., Яцунь І.С. Модель групування сумісних рухів м'язових структур людського обличчя	135
Теслюк С.В., Данилець Є.В. Створення менеджера паролів з відкритим кодом	139

УДК 004.8

О.Д. Смірнова, О.М. Шушура

Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського"
delievaalexandra@gmail.com

МОБІЛЬНИЙ ДОДАТОК ДЛЯ ВИЗНАЧЕННЯ ХВОРОБ РОСЛИН НА ОСНОВІ МЕТОДІВ МАШИННОГО НАВЧАННЯ

Більшість хвороб рослин – близько 85 відсотків – спричинені грибковими або грибоподібними організмами. Однак інші серйозні захворювання харчових та кормових культур викликають вірусні та бактеріальні організми. Нематоди (круглі черви) також викликають захворювання рослин. Деякі хвороби рослин класифікуються як "абіотичні" або неінфекційні хвороби, що включають пошкодження внаслідок забруднення атмосферного повітря, дефіциту поживних речовин або токсичності, і ростуть за менш ніж оптимальних умов.

Наразі, питання щодо ідентифікації хвороби рослини за зображенням ураженого листа вже стоїть перед розробниками та науковцями даних. Люди вивчають способи використання моделей машинного навчання для вирішення цієї проблеми, але наразі дані здобуття не вийшли за рамки R&D роботи.

Такі дослідження стали можливі завдяки тому, що хвороби, які виражаються на листі, мають унікальні ознаки, які може ідентифікувати людське око, а тепер, з розвитком машинного зору, можливо для детекції програмою.

Метою даної роботи є автоматизація процесу визначення таких хвороб як вірус жовтої курчавості листа, фітофтороз, на таких рослинах як помідори, картопля, полуниця.

Для виконання поставленої мети були виконані такі задачі: перевірити можливість класифікації хвороб рослин за зображенням хворого листа, аналіз кількох можливих технічних рішень для класифікації зображень, створення мобільного додатку для кінцевого користувача.

Задача класифікації зображень - це формалізована задача, яка містить множину об'єктів (ситуацій), поділених певним чином на класи. Задана скінченна множина об'єктів, для яких відомо, до яких класів вони належать. Ця множина називається вибіркою. До якого класу належать інші об'єкти невідомо.

Для розв'язання задачі було обрано відкритий набір даних щодо хвороб рослин, що містить 71500 зображень у форматі JPG. В датасеті містяться фотографії листа рослин. Серед категорій хвороб є вірус жовтої курчавості листа, фітофтороз та інші, серед видів рослин - картопля: полуниця: кукурудза та інші. В наборі даних присутнє як хворе, так і здорове листя кожної з рослин.

На рисунку 1 представлено приклади зображень з набору даних.

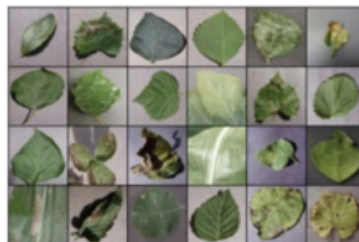


Рисунок 1 – Приклад зображень з набору даних

На вхід у модель машинного навчання надходять зображення розміром 128x128x3 (де 3 – це кольори у RGB).

Для вирішення задачі аналізу технічних рішень за модель машинного навчання була обрана нейронна мережа MobileNet [1]. Ця модель була обрана не випадково, адже