

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет інформаційних технологій
Кафедра інтелектуальних технологій**

**ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
БАКАЛАВРА
НА ТЕМУ**

Інтелектуальна система аналізу зображень квітів рослин

Галузь знань **12 «Інформаційні технології»**

Спеціальність **122 «Комп'ютерні науки»**

Освітня програма **«Аналітика даних»**

Освітній рівень: бакалавр

Виконав: студент 4 курсу, групи АнД-41

Українець А. О.

(прізвище та ініціали)



Керівник

Білан С. М.

(прізвище та ініціали)

ДОЦЕНТ

(науковий ступінь, звання)



Випускна кваліфікаційна робота бакалавра допущена до захисту
рішенням кафедри *інтелектуальних технологій*
Протокол № 13 від 05.06.2023 р.
зав. кафедри _____ доц. Іларіонов О.Є.

Київ – 2023

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА
ШЕВЧЕНКА**

Факультет інформаційних технологій

Кафедра інтелектуальних технологій

Спеціальність 122 «Комп'ютерні науки»

ЗАТВЕРДЖУЮ
Завідувач кафедри
інтелектуальних технологій
Іларіонов О.Є.

_____ 2023 р.
“ ___ ” _____

ЗАВДАННЯ

НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Українцю Андрію Олександровичу

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи)

Інтелектуальна система аналізу зображень квітів рослин

затверджена протоколом засідання кафедри від « 11 » листопада 2022 р. протоколом № 4

2. Термін здачі студентом закінченого проекту (роботи) травня 2023 року

3. Вихідні дані до проекту (роботи)

Нейронна мережа, яка класифікує та видає назву квітки та рекомендацію по догляду за нею

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їй належить розробити)

1) Аналітичний огляд методів розпізнавання образів та їх застосування

2) Інструменти та методи реалізації нейронної мережі для класифікації

3) Розробка програмного забезпечення інтелектуальної системи аналізу зображень квітів рослин

5. Перелік презентаційного матеріалу (з точним зазначенням обов'язкових презентацій)

Актуальність теми (1 слайд), мета дослідження дипломної роботи (1 слайд), вибір методу розпізнавання образів (2 слайди), опис навчального набору даних (1 слайд), процес навчання штучної нейронної мережі (2 слайди), схема архітектури застосунка (1 слайд), інтерфейс та оцінка результатів роботи застосунка (2 слайди), висновки (1 слайд).

6. Консультанти з випускної кваліфікаційної роботи із зазначенням розділів випускної кваліфікаційної роботи, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1			
2			
3			

7. Дата видачі завдання 15 лютого 2023

Керівник .



/ Білан С. М. /
(ПБ)

Завдання прийняв до виконання .



(підпис)

/ Українець А. О. /
(ПБ)

КАЛЕНДАРНИЙ ПЛАН

Пор. №	Назва етапів випускної кваліфікаційної роботи	Термін виконання етапів випускної кваліфікаційної роботи	Примітка
1	Обговорення з керівником постановки завдання та змісту пояснювальної записки	15.02.2023 – 22.02.2023	
2	Аналіз постановки задачі, формалізація задачі, аналіз літературних джерел та написання 1 розділу роботи	23.02.2023 – 08.03.2023	
3	Проектно-технологічна реалізація штучної нейронної мережі для класифікації зображень квітів рослин	09.03.2023 – 10.04.2023	
4	Розробка та тестування веб-застосунку для штучної нейронної мережі	11.04.2023 – 13.05.2023	
5	Оформлення пояснювальної записки, підготовка презентації	14.05.2023 - 29.05.2023	

Студент-дипломник .



/ Українець А. О. /
(ПБ)

Керівник випускної кваліфікаційної роботи .



/ Білан С. М. /
(ПБ)

Анотація

Українець Андрій Олександрович виконав випускню кваліфікаційну роботу на тему «Інтелектуальна система аналізу зображень квітів рослин» за спеціальністю 122 - «Комп'ютерні науки», освітня програма «Аналітика даних»

У випускній кваліфікаційній роботі проведено аналіз сучасних методів розпізнавання образів, розглянуто різні види штучних нейронних мереж, розроблено згортова штучну нейронну мережу, яка класифікує зображення квітів та інтерфейс, який транслює отримані результати класифікації.

Ключові слова: рослина, класифікація, конволюційна штучна нейронна мережа, застосунок.

Summary

The degree project: «Intelligent system of image analysis of plant flowers» has completed by Andrii Ukrainets specialty 122 – «Computer Science», educational program «Data Analysis».

In the graduation work, an analysis of modern pattern recognition methods was carried out, various types of artificial neural networks were considered, a convolutional artificial neural network was developed that classifies images of flowers and an interface that broadcasts the obtained classification results.

Key words: plant, classification, convolutional artificial neural network, application.

Зміст

Анотація.....	4
Summary.....	4
Зміст.....	5
Вступ.....	7
Розділ 1. Аналіз предметної галузі та постановка задачі.....	9
1.1 Опис предметної області.....	9
1.2 Сучасні методи розпізнавання образів.....	10
1.2.1 Статистичні методи.....	10
1.2.2 Структурні методи.....	11
1.2.3 Нейронні мережі.....	13
1.2.4 Зіставлення шаблонів.....	14
1.2.5 Нечіткі методи.....	15
1.2.6 Гібридні методи.....	16
1.3 Приклади застосування методів для класифікації.....	17
1.4 Постановка задачі.....	19
1.5 Опис вимог.....	21
1.5.1 Функціональні вимоги.....	21
1.5.2 Нефункціональні вимоги.....	21
1.6 Вибір мови програмування.....	22
1.7 Висновок до розділу.....	22
Розділ 2. Інструменти та методи реалізації нейронної мережі для класифікації.....	23
2.1 Види штучних нейронних мереж.....	23
2.1.1 Персептрон.....	23
2.1.2 Штучні нейронні мережі прямого розповсюдження.....	23
2.1.3 Багатошаровий персептрон.....	24
2.1.4 Конволюційна ШНМ.....	25
2.1.5 Радіальна базова функція.....	25

2.2	Опис інструментів для реалізації програмної частини.....	26
2.3	Процес навчання штучних нейронних мереж.....	27
2.4	Розробка програмної системи.....	28
2.4.1	Підготовка даних до класифікації.....	28
2.4.2	Розробка штучної нейронної мережі.....	31
2.5	Висновки результатів роботи ШНМ.....	39
2.6	Архітектура застосунку.....	39
2.7	Висновок до розділу.....	43
	Розділ 3. Розробка програмного забезпечення інтелектуальної системи аналізу зображень квітів рослин.....	45
3.1	Структура застосунку	45
3.2	Опис даних.....	46
3.3	Опис інтерфейсу.....	47
3.3.1	Опис основних сторінок.....	47
3.3.2	Опис переходів між сторінками.....	50
3.4	Опис тест-кейсів.....	51
3.5	Висновок до розділу.....	58
	Висновки.....	59
	Список використаних джерел.....	60
	Додатки.....	61
	Додаток А.....	61
	Додаток Б.....	62

Вступ

Штучна нейронна мережа – це програмна або апаратна реалізація структури та парадигми, яка імітує поведінку нашого мозку і складається з великої кількості штучних нейронів, які є органічними перемикачами. Мережі можуть бути навчені за допомогою вчителя та самостійно.

Проблемі класифікації рослин на теперішній час приділяється багато уваги спеціалістів. Автоматизація процесу розпізнавання квітів за їх зображенням дозволяє спростити процес догляду за рослинами різної природи. Дана тема актуальна в сьогоденні тому, що багато людей купляють собі рослини і можуть не знати, як вони називаються або можуть сплутати назву і доглядати за нею будуть не правильно.

Для класифікації квітів в роботі використовуються зображення різних квітів. Було обрано 299 різних квітів, які навіть між собою можуть бути схожі за кольором та розміром.

Дипломна робота буде корисна для всіх людей, які хочуть придбати собі рослину. Ця класифікація квітів допоможе людині переконатись в тому, що вона купляє правильну рослину, яка і написана у магазині. Також якщо ви будете десь у подорожі і побачите у когось гарну рослину і вона вам сподобається ви також зможете дізнатися інформацію про неї.

В даній дипломній роботі розроблено метод класифікації зображень за допомогою технологій штучного інтелекту. Для цього необхідно провести дослідження факторів впливу на кінцевий результат класифікації. Також провести порівняльні дослідження різних методів класифікації. Визначити найефективнішу модель, яка найкраще працює з вибраними даними.

Класифікація квітів на основі зображень є непростим завданням. Бувають різні квіти, які між собою схожі за формою чи кольором, а відрізнятися тільки пелюстками або довжиною стебла. Таким чином, знаходження відмінностей серед квітів, може інколи бути важким для людини. Однак за допомогою нейронної мережі, зможемо навчитися класифікувати зображення.

У дипломній роботі розглянуто теоретичні та практичні основи штучних нейронних мереж. Для вирішення задачі було вирішено використовувати поширені бібліотеки для швидкого будування штучних нейронних мереж різних типів.

Розділ 1. Аналіз предметної галузі та постановка задачі

1.1 Опис предметної області

Поява машинного навчання покращила життя багатьох людей. Це один з методів, що реалізує одну з функцій штучного інтелекту, який використовується для створення алгоритмів виявлення закономірностей між даними в великих масивах, а також подальшого використання для самонавчання. Головна особливість машинного навчання – досягти вирішення не одної конкретної задачі, а й навчитися виконувати подібні завдання[1].

Сьогодні використання машинного навчання, можна побачити у багатьох різних сферах. Сучасні інструменти машинного навчання дають можливість вирішувати багато різних задач, таких як: класифікація, виявлення спаму, рекомендаційні системи, проведення аналізу та діагностики, відтворення зображень та багато інших.

Розглянемо одну із задач, яку дозволяє рішати застосування машинного навчання – класифікацію.

Задача класифікації – це задача розбиття множини об'єктів або спостережень на задані класи (групи). В кожному класі, об'єкти вважаються між собою схожими та мають приблизні властивості та ознаки. Знаходження результату відбувається за допомогою ознак.

Існує чотири типи класифікації:

1. Бінарна класифікація.
2. Багатокласова класифікація.
3. Класифікація з кількома мітками.
4. Незбалансована класифікація.

Бінарна класифікація – це задачі, які мають лише дві мітки класів. Наприклад: відноситься лист до спаму чи ні, здорова людина чи ні та інші.

Багатокласова класифікація на відміну від бінарної має декілька міток класів. Наприклад: категоризація обличь, класифікація видів рослин та інші.

Класифікація з кількома мітками полягає в тому, що один об'єкт може відноситись до однієї або більше міток класу. Наприклад: передбачення декількох предметів на фотографії.

Незбалансована класифікація – це задачі, які включають більшість екземплярів, які відносяться до нормального класу і меншу кількість екземплярів, які відносяться до ненормального класу. Наприклад: розслідування шахрайству, клінічні діагностичні процедури, тощо.

За допомогою машинного навчання задача класифікації дозволяє реалізувати прогнозування, де для кожного об'єкту буде передбачено мітку класу. Для моделі подається навчальний набір даних, на яких вона вчиться відносити об'єкти до класів, які більш підходять за ознаками. Чим більше навчальна вибірка, тим краща точність класифікації моделі, але більшу часу витрачається на навчання.

Розглянемо процедуру розпізнавання:

1. Сприйняття образу. На цьому етапі отримуємо інформацію про об'єкт, а саме характеристику, властивості за допомогою (вимірювання, фотографування, зчитування звуку).

2. Попередня обробка. Підготовка даних до класифікації: видалення шумів, обрізання зображення та інші фактори.

3. Виділення характеристичних ознак та визначення їх кількісних характеристик. Вимірюються кількісні значення характеристичних ознак об'єкта.

4. Класифікація. Знаходження підходящого класу для вхідного об'єкту.

1.2 Сучасні методи розпізнавання образів

Розглянемо декілька популярних груп методів для розпізнавання образів.

1.2.1 Статистичні методи

Цей метод застосовується для побудови статистичної моделі [2]. Це випадкова величина з статистичними характеристиками, об'єкти якої описані за допомогою ознак класу. За допомогою методів математичної статистики можна будувати алгоритм розпізнавання, оптимальний за критерієм якості. Модель може передбачити ймовірнісний характер закономірностей. Вибрані ознаки використовуються для формування кластерів. Розподіл ймовірностей об'єкту аналізується, і система відповідно адаптується. Потім модель застосовує об'єкти тестування для класифікації нових об'єктів.

Розпізнавання образів даними методами застосовують у тих випадках, коли ознаки характеризуються числовими значеннями та ці ж значення об'єктів в одному класі можуть мати різні значення.

При навчанні моделі задається навчальна вибірка, що складається з об'єктів, які мають певні ознаки, а також назву класу. Відповідно до цих даних на скільки вони інформаційні й містять статистичні характеристики моделі, будуються статистичні оцінки параметрів (ознак об'єктів). Отримані оцінки підставляють у рівняння для розпізнавання.

1.2.2 Структурні методи

Структурні методи застосовуються до задач розпізнавання образів, в яких опис об'єкта (образу) здійснюється за допомогою підобразів [2]. Основна задача методів полягає у знаходженні загальних ознак (структур) об'єкта, які відомі шуканому класу. Структурний підхід ґрунтується на тому, що образи, які відносяться до одного класу, мають ряд загальних властивостей, які показують подібність таких образів. Потім ці властивості частково вносяться в пам'ять системи розпізнавання, для подальшого зіставлення з наступними образами, які можемо подати на вхід.

Розглянемо декілька образів, які треба описати структурою (рис. 1.1).

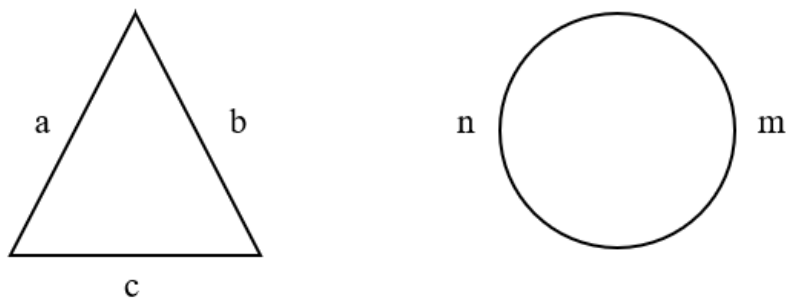


Рисунок 1.1 – вхідні дані для методу

На рисунку 1.1 зображено дві фігури: трикутник і коло. Аналіз першої фігури показує, що вона має три сторони, а друга фігура має дві дуги. За допомогою цих відрізків можна побудувати ієрархічну структуру зображень для класифікації інших образів (рис. 1.2).

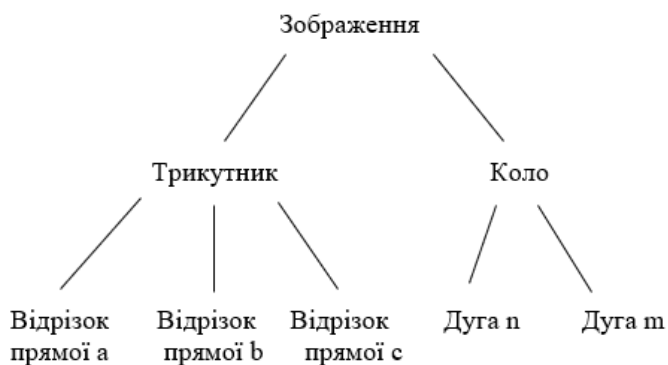


Рисунок 1.2 – Ієрархічна структура

Також даний метод часто називають лінгвістичним тому, що розпізнавання включає в себе етап синтаксичного аналізу (граматичний розбір), що описує вхідний образ. Таким чином вхідний образ можливо розбити на більш менші деталі для класифікації інших.

Структурні методи стали популярними для розпізнавання текстів, символів, фігур.

Головна перевага методів – можливість подати велику кількість рішень у вигляді множини непохідних елементів і граматичних правил.

Недоліки:

- Обмеженість у використанні через те, що використовуємо не весь образ, а лише його частину.
- Метод не має прямих вирішальних правил.

1.2.3 Нейронні мережі

Штучна нейронна мережа – це система, яка імітує людський мозок, що ґрунтується на сукупності з'єднаних штучних нейронах[3]. Вони між собою взаємодіють, тим що обробляють інформацію і передають іншому сигнал для наступної обробки. Кожен штучний нейрон знаходиться в своєму шарі і робить свою обробку, кожний шар також може виконувати різні види перетворень вхідних даних[4] (рис. 1.3).

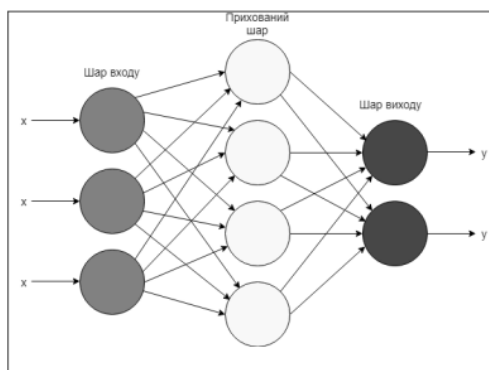


Рисунок 1.3 – проста штучна нейронна мережа

На рисунку 1.3 відображається значення входу “х”, які передаються до шару входу, де штучні нейрони даного шару роблять операції перетворення або обчислення. Далі з кожного штучного нейрону попереднього шару передається інформація до кожного штучного нейрону наступного шару і також робляться, якісь дії. Переходи відбуваються доки дані не пройдуть всі сховані шари штучної нейронної мережі і не потраплять до вихідного шару. Після того, як інформація вийшла з вихідного шару, отримуємо рішення в залежності від поставленого завдання.

Однією із найважливіших властивостей штучної нейронної мережі є здатність до навчання та засвоєння знань. Натренована мережа на заданих даних,

може узагальнювати за допомогою знайдених закономірностей і потім показувати гарні результати на даних, які не брали участі у тренуванні.

Штучні нейронні мережі мають два способи тренування:

З вчителем. Означає те, що наша мережа отримує, якісь вхідні дані та вихідні. Наприклад: параметри об'єкту та назву його. На основі цих даних, штучна нейромережа відносить новий вхідний масив до одного з заданих класів, який більше підходить під задані параметри.

Без вчителя. Більш складний процес за попередній, тим що на вході мережа не має початково заданих класів. Вона повинна сама класифікувати вхідні дані між собою за якоюсь схожістю.

Також нейромережа має перелік своїх недоліків:

1. Не можуть дати сто відсоткову однозначну відповідь, так як роблять прогнозування з якоюсь ймовірністю.
2. Потребують великої кількості даних для навчання.
3. Складність реалізації для кожного окремого випадку.

1.2.4 Зіставлення шаблонів

Зіставлення шаблонів або евристичні методи за основу, яких взято досвід людини, а саме перерахування членів класу і загальні властивості [2].

В більшості випадків зображення одного класу є майже незмінним (розмір та інші характеристики). Тому в таких ситуаціях краще використовувати розпізнавання за допомогою порівняння зображення з шаблонами (зразками).

В методі зіставлення шаблонів, деяка множина об'єктів, має одного представника кожного класу, який знаходиться в пам'яті системи для використання, як шаблон. Подається на вхід некласифікований образ і він порівнюється з шаблоном кожного класу. Заздалегідь вибирається критерій класифікації. Таким чином, якщо вхідний об'єкт краще підходить до і-го шаблону, то він й належить цьому класу образів. Частіше за все використовують такий підхід для читання символів (рис. 1.4).

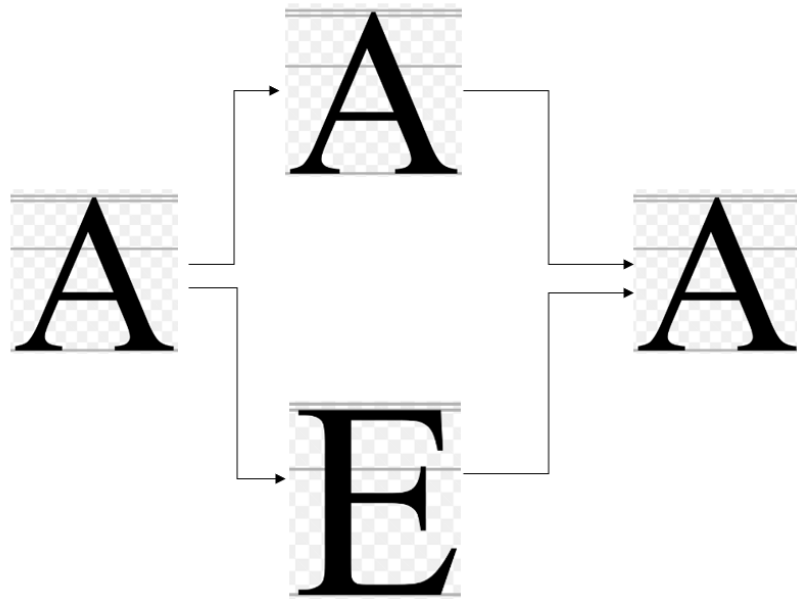


Рисунок 1.4 – зіставлення шаблонів

На рисунку 1.4 показано зіставлення шаблонів для класифікації літери.

Головним недоліком цього методу, є те, що важко підібрати вірний шаблон для вхідного образу, особливо, якщо ще й присутні, якісь спотворення або шуми. Гарним прикладом буде розпізнавання рукописних літер. У кожної людини свій почерк і тому одна літера або символ можуть виглядати по різному.

1.2.5 Нечіткі методи

Дані методи на основі нечіткості застосовують концепцію нечіткої логіки, для групування об'єктів, що мають однакові характеристики, у нечітку множину, яка використовує значення істинності від 0 до 1 [2]. У нечіткій моделі можна застосувати деякі правила, щоб узгодити заданий вхід із відповідним виходом. Ця модель дає хороші результати, оскільки вона підходить для невизначених областей.

В нечітких методах є головним визначення області для вхідного та вихідного простору та інтервал розділення на n областей, що перекриваються за допомогою трикутних функцій, як показано на рисунку 1.5. Далі кожен об'єкт відноситься до якоїсь області. Після цього утворюють правило і обчислюють його за допомогою добутку ступенів належності нечітких множин.

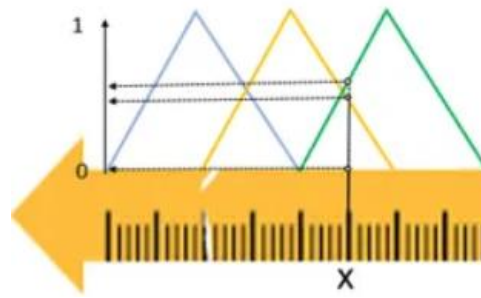


Рисунок 1.5 – розбиття області

Найпростішим нечітким класифікатором на основі правил є нечітка система “if – then”

IF x_1 is small AND x_2 is medium THEN class is 1

IF x_1 is small AND x_2 is large THEN class is 2

IF x_1 is large AND x_2 is small THEN class is 2

1.2.6 Гібридні методи:

Різні методи машинного навчання мають свої переваги і недоліки. Часто багато з цих методів можуть здаватися обмежувачими, якщо вони використовуються ізольовано, особливо якщо для вирішення проблеми доступно кілька джерел даних.

Гібридні методи використовуються для створення гібридної моделі, яка використовує кілька класифікаторів для розпізнавання об’єктів [2]. Тому що по одинці методи машинного навчання можуть не справитись з поставленою задачею, але, якщо об’єднати сили двох методів, то разом вони доповнять або посилять один одного. Кожен конкретний класифікатор проходить навчання на основі просторів ознак. Для отримання висновку використовується набір об’єднувачів і класифікаторів. Функція прийняття рішень використовується для визначення точності класифікаторів.

Для даного методу є дуже багато способів поєднати декілька методів в один, але це все потрібно відштовхуватись від поставленої задачі, бо ці методи, як правило, мають високу обчислювальну складність і вимагають великої кількості даних.

1.3 Приклади застосування методів для класифікації

Однією із задач машинного навчання є реалізація класифікації. Це процес віднесення об'єкта до одного із заданих класів. На перший погляд, може показатися, що людина і сама може розпізнавати все навколо, але вона приходить до цього з роками, а різні системи зроблені за допомогою методів розпізнавання образів це роблять за декілька годин може і хвилин, все залежить від складності задачі. Ускладнювати процес класифікації можуть: шуми, розмитість, освітлення на зображеннях та інші фактори.

Системи поки навчаються, можуть виділяти для себе багато різних закономірностей в образах: колір предмету, розмір, розміщення та інші.

Останнім часом зростає інтерес до використання систем для класифікації у різних галузях людського життя.

Прикладом успішного використання штучних нейронних мереж у фінансовій сфері, а саме у деяких банках США. Використовують цю мережу для того, щоб класифікувати людину, чи спроможна вона виплатити у подальшому кредит банку чи ні[5].

Також популярне використання й у медицині, але це дуже відповідальне рішення, бо від цього може залежати життя людини. Застосовують для діагностики онкологічних захворювань. Вчені з університету Дюка (США) створили штучну нейронну мережу, яка розпізнає злоякісні тканини і успішно застосовується для діагностики раку молочної залози. Роблять і карді діагностику, такі системи використовуються у госпіталях Англії для попередження серцево-судинних захворювань та їх рівню[6].

У галузі охорони людини, штучні нейронні мережі: ідентифікують особу, розпізнають голос, ідентифікують загрозу в натовпі, розпізнавання автомобілів та їх номерних знаків, виявлення підробок. Розпізнавання автомобілів використовуються на дорогах, для відстеження їх порушення. Ідентифікацію особи почали використовувати в розумних будинках для входу, також є дана функція і в телефоні, для розблокування[7].

Один з гарних прикладів класифікації є пошукова система Google Lens. Створена вона для зручного пошуку предмета, який ви можете сфотографувати або закинути його фото, також це може бути текст і система буде намагатися знайти подібні фотографії до тієї, яку ви надали системі[8].

На рисунку 1.6 представлено принцип роботи системи, де вона виділяє межі пошуку і намагається знайти подібні малюнки. Також система сама пропонує, якісь основні точки на зображенні від, яких далі можна буде відштовхуватись. Після того, як вона видає результат користувач може подивитись і якщо воно правильно знайшло, то користувач може залишити відгук коректності пошуку і надалі, якщо хтось буде шукати, щось подібне, то система буде відштовхуватись і від цих відгуків.

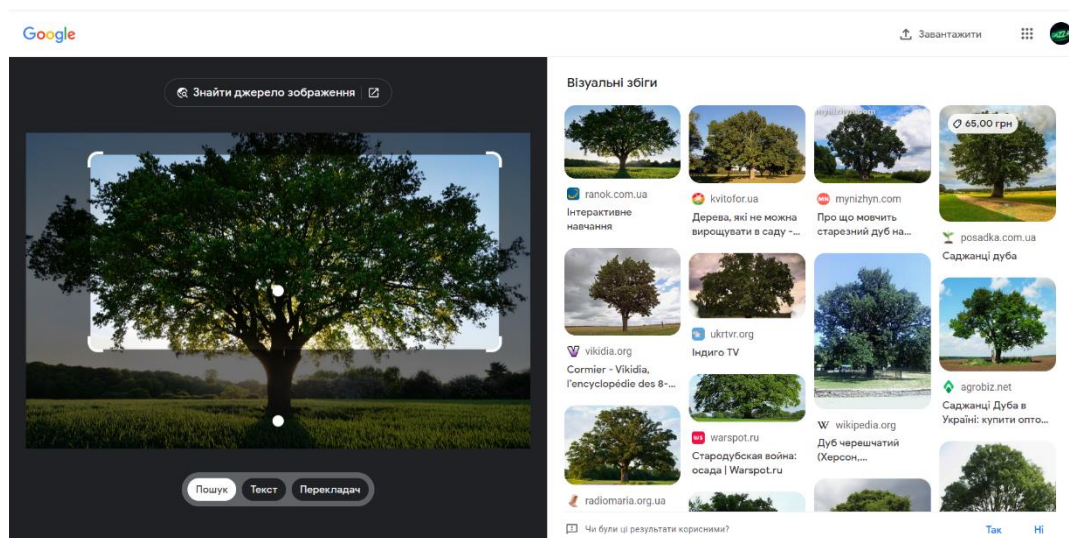


Рисунок 1.6 – робота Google Lens

Кожен день людина зіштовхується з елементами де зустрічаються літери, символи, чи якісь позначки. Розпізнавання літер найчастіше використовують в сканерах для введення до комп'ютера друкованих зображень текстів. Ще розпізнавання текстів використовується для перекладу на інші мови. Також розпізнавання штрих-кодів, які містять в собі числа або зображення у вигляді штрихів – це все використовують для товарів в магазині або для того, щоб зробити швидкий перехід за посиланням на сайт.

1.4 Постановка задачі

Метою дипломної роботи є створення штучної нейронної мережі та програмного застосунку для класифікації зображень рослин та надання рекомендації по догляду за ними.

Завдання дипломної роботи є проведення аналізу предметної області; описати вимоги до ПЗ; вибрати мову програмування та ШНМ; розробити програмний застосунок у вигляді сайту та створити датасет.

Об'єктом дослідження є методи застосовані для дослідження обраної бази даних (dataset). Проаналізувати на скільки точно можна буде класифікувати зображення.

Предметом дослідження є фотографії квітів, за інформацією яких відбувається класифікація назви рослини з використанням штучної нейронної мережі.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- провести аналіз та знайти підходящий dataset з квітами для класифікації.
- провести огляд та аналіз штучних нейронних мереж, їх видів та до яких задач їх можна застосувати.
- розробити штучну нейронну мережу для класифікації фотографій квітів.
- перевірити ефективність роботи мережі на тестових даних.

Під час створення моделі нейромережі необхідно дослідити вплив параметрів шарів на кінцеву ефективність роботи мережі. Для тренування та тестування потрібно використати зібрані зображення різних квітів. Щоб отримати найкращі результати роботи мережі потрібно буде зробити попередню обробку зображень і розбити дані рівномірно для тестувальної та тренувальної вибірки.

У результаті виконання роботи необхідно отримати таку мережу, яка буде найкраще справлятися з нашою задачею, а саме: час виконання, точність результату та простота моделі штучної нейронної мережі.

На рисунку 1.7 зображено діаграму IDEF0. Вхідними даними, є зображення рослини. Управління такою системою виконують алгоритм класифікації

зображення та користувач, адже він завантажує зображення. Механізмом є правила обробки зображення та вимоги до завантаження файлу. Результатом є назва квітки, яку класифікувала наша система.

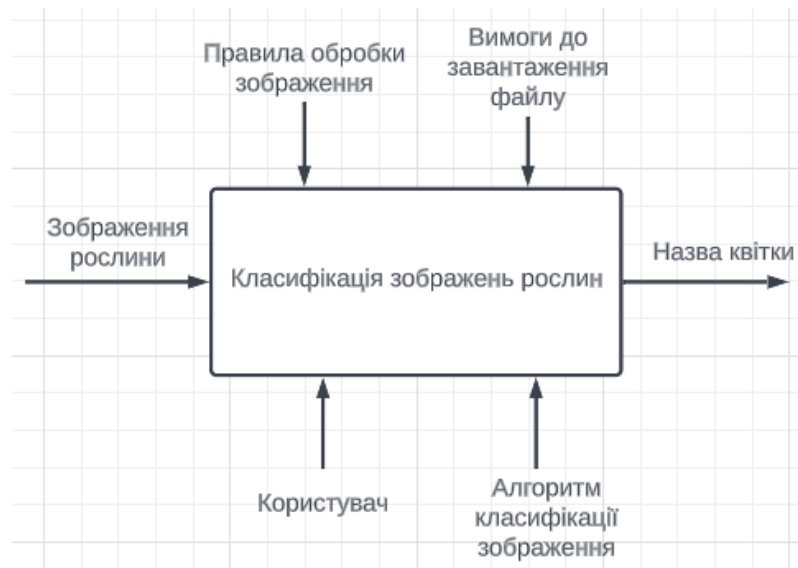


Рисунок 1.7 – схема IDEF0 “Класифікація зображень рослин”

На рисунку 1.8 наведено дерево функцій інтелектуальної системи класифікації зображення квітів рослин. Основними функціями є отримання зображення, його обробка та формування результатів. Отримання зображення можна представити таким чином, що користувачу необхідно завантажити зображення заданого формату, після чого йде перевірка на правильність формату. Обробка вхідного зображення поділяється на такі підпроцеси, як форматування та нормалізація зображення. Останнім етапом є формування результатів, яке складається з трьох функцій: подання обробленого зображення на вхід, застосування алгоритму класифікації зображення та отримання назви квітки та рекомендації по догляду за нею.

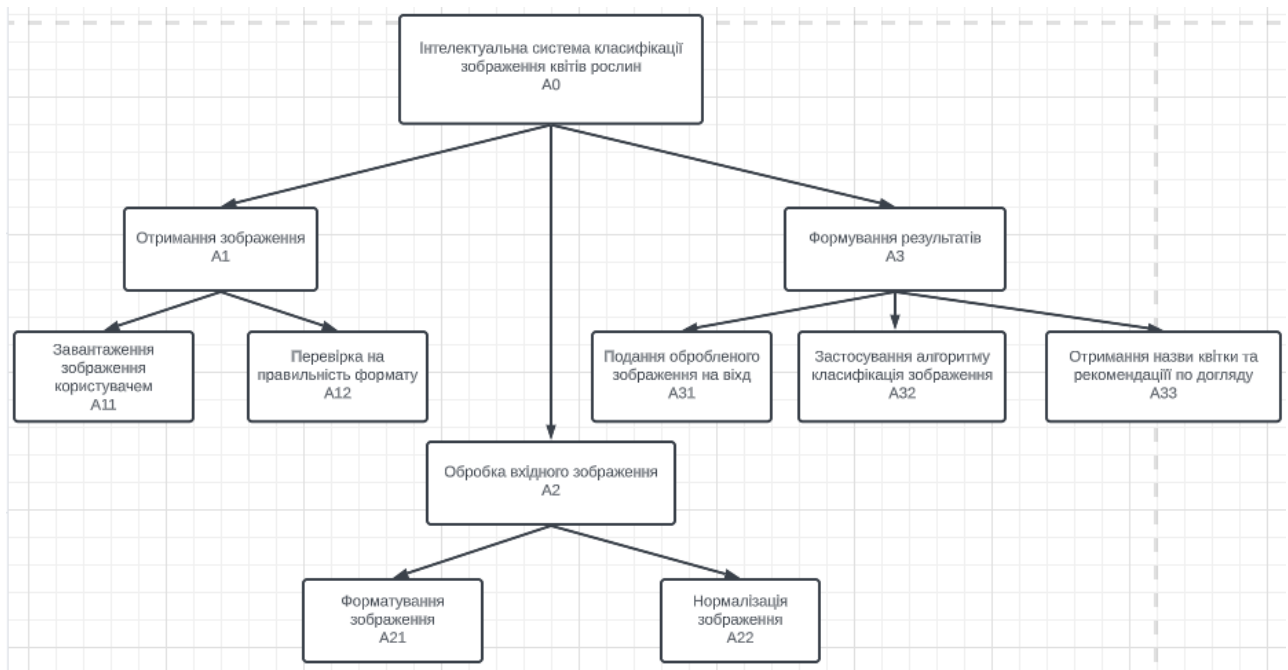


Рисунок 1.8 – Дерево функцій інтелектуальної системи класифікації зображення квітів рослин

1.5 Опис вимог

1.5.1 Функціональні вимоги

Функціональні вимоги до сайту:

- Зручний та зрозумілий інтерфейс, який буде реалізовано українською мовою;
- Класифікація завантаженого зображення;
- Створення бібліотеки рослин;
- Перегляд інформації про рослину, а саме, як доглядати за нею.

1.5.2 Нефункціональні вимоги

До нефункціональних вимог відносяться наступні:

- Надійність: якщо користувач робить некоректні дії, то система буде реагувати на це без збоїв.
- Коректна відповідь системи на будь-які дії користувачів.
- Система буде в робочому стані та матиме зв'язок з таблицею.

1.6 Вибір мови програмування

Для виконання дипломної роботи була обрана мова програмування Python. Ця мова останнім часом все частіше використовується для створення штучних нейронних мереж та аналізу даних. Використання цієї мови не призведе до труднощів тому, що за допомогою невеликих частинок коду можна будувати наглядні штучні нейронні мережі, виконувати складні математичні операції. Також в Python є велика кількість різноманітних відкритих бібліотек.

1.7 Висновок до розділу

В даному розділі було детально розглянуто тему машинного навчання, а саме одну із задач – розпізнавання образів. Описано різні типи методів для вирішення даної проблеми, їх особливості, переваги та недоліки. Також наведено приклади їх застосування в різних сферах діяльності: мова, захист, медицина, пошук, фінанси та інші. Проаналізувавши дані методи, було прийнято рішення використовувати штучні нейронні мережі для поставленої задачі – класифікація зображень рослин.

Окрім того, в даному розділі були визначені задачі, предмет та об'єкт дослідження, а також вибрано мову програмування за допомогою, якої буде реалізовано саму систему.

Розділ 2. Інструменти та методи реалізації нейронної мережі для класифікації

2.1 Види штучних нейронних мереж

Розглянемо найпопулярніші види штучних нейронних мереж (ШНМ) для класифікації та їх конструкції. Як вже було зазначено вище, що штучна нейромережа складається з вхідного шару, прихованого та вихідного. В кожному шарі є своя кількість штучних нейронів. Нейрон отримує на вхід інформацію, за допомогою функції активації він працює і на виході дає ваги.

2.1.1 Персептрон

Персептрон один із самих простих алгоритмів навчання з вчителем, який класифікує дані тільки за двома категоріями. Він приймає на вхід дані та застосовує функцію активації, щоб отримати результат[9] (рис. 2.1).

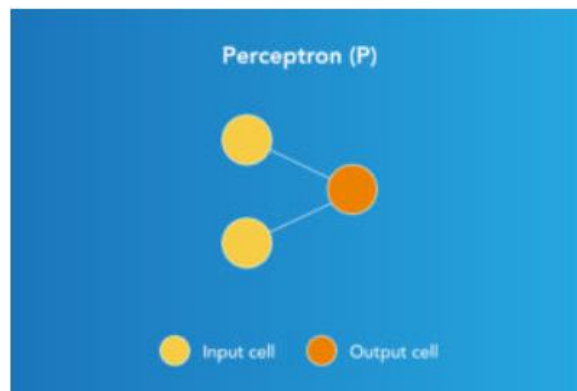


Рисунок 2.1 – персептрон

Переваги: найпростіший класифікатор.

Недоліки: можуть вивчати лише лінійні задачі з двома класами.

2.1.2 Штучні нейронні мережі прямого розповсюдження

Також найпростіша форма нейромережі, де вхідні дані рухаються лише в одному напрямку і не мають зворотного поширення (рис. 2.2). Означає, що кількість нейронів на кожному шарі рівна. Використовують для простої класифікації, розпізнавання обличчя[9].

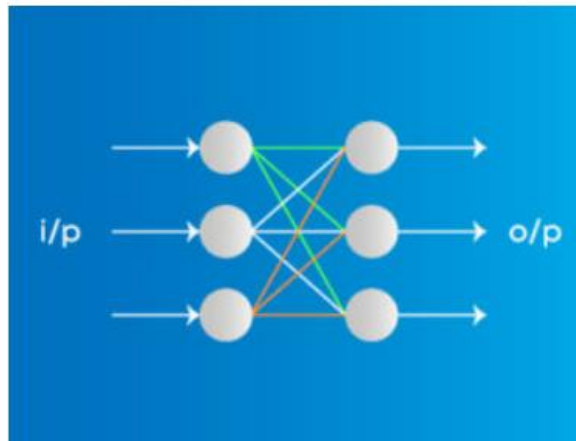


Рисунок 2.2 – мережа прямого зв'язку

Переваги: простий у проектуванні, швидкий у виконанні.

Недоліки: не можна використовувати для глибинного навчання.

2.1.3 Багатошаровий перцептрон

Більш складна мережа від попередніх. Вхідні дані проходять через різні шари штучних нейронів (рис.2.3). Кожен вузол з'єднаний з усіма наступними штучними нейронами наступного шару. Складається принаймні з трьох шарів, а то можуть бути і більше. Має двонаправлене поширення. Використовується для: класифікації, розпізнавання мови, машинного перекладу[9].

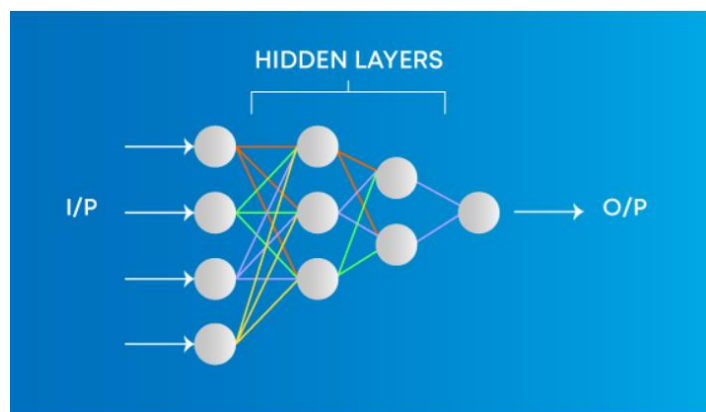


Рисунок 2.3 – багатошаровий перцептрон

Переваги: можна використати в глибинному навчанні.

Недоліки: залежить від кількості прихованих шарів.

2.1.4 Конволюційна ШНМ

Найпопулярніший метод для класифікації зображень. Перший шар штучної нейронної мережі називається згортковим (рис.2.4). Кожен штучний нейрон даного шару обробляє частково інформацію зображення зменшуючи кількість пікселів. Обробка передбачає зменшення зображення, а саме перехід від шкали RGB до HIS. Використовують для: обробки зображень, комп'ютерного зору, розпізнавання мови[9].

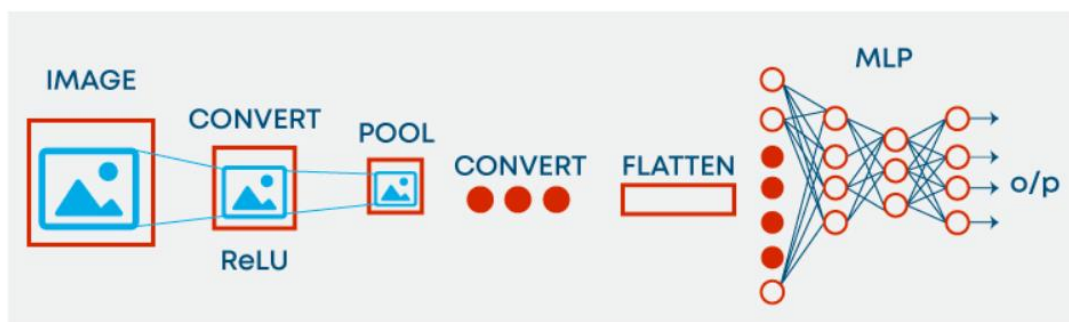


Рисунок 2.4 – конволюційна мережа

Переваги: використовується для глибинного навчання, менше параметрів для вивчення.

Недоліки: складна у проектуванні та обслуговуванні, залежить від кількості прихованих шарів.

2.1.5 Радіальна базова функція

Мережа в якій класифікація виконується шляхом вимірювання подібності вхідних даних до точок даних із навчального набору (рис. 2.5). Коли новий вектор вхідних даних потрапляє до шару, то нейрон обчислює евклідову відстань між вхідним сигналом і його прототипом. На виході отримуємо клас, до якого наш вхідний вектор більш всього підходить[9].

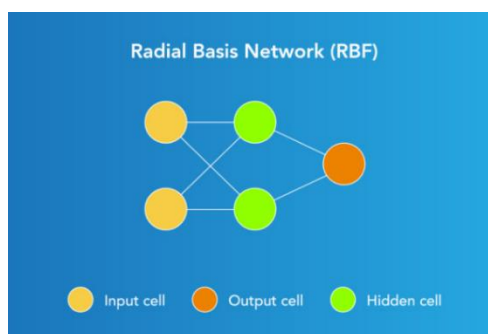


Рисунок 2.5 – радіальна базова функція

Проаналізувавши види штучних нейронних мереж за допомогою, яких класифікують дані, можна вибрати собі для подальшої роботи. Для поставленої задачі більше всього підходить конволюційна мережа тому, що існує необхідність працювати з фотографіями. Саме ця модель справиться краще за всіх з цією задачею[10].

2.2 Опис інструментів для реалізації програмної частини

Для розробки штучної нейронної мережі було вибрано мову програмування Python, як це вже зазначалося вище. В роботі було використано такі бібліотеки для побудови штучної нейромережі та роботи з даними.

1. NumPy – це бібліотека, в якій реалізовано різні математичні функції для обробки масивів великих розмірів. NumPy був створений у 2005 році. Дана бібліотека пропонує багато різних функцій для операцій з масивами, які мають швидке виконання[11].

2. Pandas – одна з найпопулярніших бібліотек Python в роботі з даними та їх аналізу. За допомогою цієї бібліотеки, можемо швидко створити об'єкт DataFrame з інтегрованим індексуванням. Можливість зчитування та запис у різні види файлів. Гнучка зміна розмірності та типів даних[12].

3. Os – для роботи з папками на комп'ютері або гугл диску. За допомогою цієї бібліотеки можна зчитувати і записувати інформацію у масиви або датафрейми[13].

4. TensorFlow – популярна бібліотека для побудови штучних нейронних мереж. Має багато модулів за допомогою, яких можемо реалізовувати різні види шарів штучної нейронної мережі. Також можемо тренувати нашу модель і після цього прогнозувати значення[14].

5. Для розробки сайту було використано фреймворк Flask [15]. Він класифікується, як мікрофреймворк, оскільки він не залежить від зовнішніх бібліотек для виконання певних завдань. Flask має свої інструменти, технології та бібліотеки для підтримки функцій розробки веб-додатків. Оскільки цей фреймворк більш незалежний та гнучкий, вважається, що краще починати саме з нього. Також його краще обирати, якщо це маленький проект, а не здоровий, тому саме його і було обрано для розробки інтерфейсу.

2.3 Процес навчання штучних нейронних мереж

Навчання штучної нейронної мережі є основним процесом її побудови. Властивість штучної нейронної мережі навчатися на прикладах дозволить нам спростити розв'язок конкретної задачі, в порівнянні з системами, які створили люди, в яких все залежить від заданих правил[16].

В дипломній роботі навчання штучної нейромережі буде проходити за допомогою вчителя. Штучна нейронна мережа буде отримувати на вхід дані, потім пройдёт етап обробки всередині мережі. Потім штучна нейронна мережа порівнює отримані результати зі значеннями функції. Далі мережа обчислює помилку і після чого змінює вагові коефіцієнти.

Залежно від задачі будуть змінюватись типи даних навчальної вибірки. Розглянемо на прикладі класифікації зображень. В наявності фотографія предмету і його назва. Зображення потрібно нормалізувати – перевести кожний піксель в цифрове значення. Для того, щоб навчити штучну нейромережу часто використовують великі об'єми даних, тому для цього потрібні великі об'єми оперативної пам'яті комп'ютера.

Також для навчання штучної нейронної мережі вхідні дані ділять на дві вибірки: тренувальну і тестуючу. Якщо в тестуючій вибірці будуть нові класи, які не були при навчанні, то вони будуть класифіковані не правильно або будуть просто віднесені до схожого наявного класу.

Після того, як було створено модель штучної нейронної мережі потрібно запустити процес навчання, до якого було передано такі критерії: оптимізатор, функція похибки і критерій зупинки.

Самий популярний оптимізатор для навчання – Adam(). Це один з методів навчання, застосовується в багатьох моделях і оснований на стохастичному градієнтному спуску, який базується на адаптивній оцінці моментів першого і другого порядку. Сутність методу полягає в пошуку мінімального значення функції втрати для визначення оптимальних параметрів.

Одна із функцій похибки, яка використовується під час вирішення поставленої задачі є розріджена категорична крос-ентропія. Вона обчислює втрату крос-ентропії між мітками та прогнозами. Використовується ця функція коли наші мітки задані, як цілі числа.

Критерій зупинки в основному може бути двома типами: задана кількість епох тренування моделі або мінімальне значення функції похибки.

Після того, як були задані параметри навчання, модель починає тренуватися на даних з кожною епохою покращуючи точність. Головне при навчанні – не перевчити або недовчити модель, що використовується.

По закінченні існує можливість переходу до тестування моделі на тестуючій вибірці.

2.4 Розробка програмної системи

2.4.1 Підготовка даних до класифікації

Після того, як було знайдено зображення рослин, вони були завантажені на гугл диск, звідки вони і зчитувались. Всі зображення знаходяться у форматі JPG у папках з назвою квітки.

За допомогою бібліотек Python використане середовище було під'єднане до гугл диску для того, щоб зчитувати інформацію з папок і обробляти зображення і додавати в масив.

Першим кроком стягувались назви папок для того, щоб класифікувати, всього класифікувалось 100 рослин тому, що недостатньо обчислювальної потужності для того, щоб запустити штучну нейронну мережу для класифікації більшості зображень (рис. 2.6).

```
['Abutilon', 'Acacia', 'Aconite', 'AfricanDaisy', 'Agapanthus', 'Ageratum', 'Alchemilla', 'Allium',
```

Рисунок 2.6 – назви перших рослин масиву

Далі кожній назві присвоювався номер, таким чином змінювались стрінгові значення на числові для того, щоб наша модель змогла навчитися класифікувати.

Наступним кроком треба зробити саму матрицю, в якій будуть знаходитися всі зображення. Для обробки зображення необхідно також замінити пікселі фотографії на число. Першим кроком треба задати розмір зображення. Так як вхідні зображення спочатку знаходяться в різних розмірах між собою, то їх потрібно привести до однакових. Також зменшення розміру зображення допоможе уникнути великого навантаження на оперативну пам'ять ноутбуку, але це може призвести до втрати, якихось важливих точок зображення за допомогою, яких би мережа змогла краще класифікувати. Розмір фотографії задається, як 50 на 50 пікселів. Також задавались параметри, що зображення будуть кольоровими це хоч і дасть навантаження на оперативну пам'ять, але поставлена задача полягає в класифікації різних видів квітів і колір гратиме важливу роль у даній задачі. Останнє це посилання на гугл диск де знаходяться зображення рослин.

Наступним кроком перетворювали зображення у матрицю чисел (2.1).

$$f = \frac{p-min}{-min} \quad (2.1)$$

Вище подано формулу нормалізації даних де,

p – значення пікселя від 0 до 255.

\min – мінімальне значення пікселя 0.

\max – максимальне значення пікселя 255.

Зробивши цю операцію з кожним пікселем фотографії отримується матриця значень для того, щоб існувала можливість надати інформацію штучній нейронній мережі для подальшої класифікації.

Всього вийшло 40178 зображень різних квітів.

І останній крок підготовки даних до класифікації – розділити наші дані на тренуючу та тестуючу вибірку (табл. 2.1). Для цього використовується функція, що вбудована в бібліотеку `sklearn`. В неї передаються вхідні дані: матриця значень зображень і вихідні – назви рослин, які також перетворюються у цифри (рис. 2.7 та 2.8). Тестуюча вибірка буде зіставляти 20% даних від усіх, які є в наявності. Також ця функція перемішує існуючі дані рівномірно, щоб була змога нормально перевіряти точність класифікації моделі.

Таблиця 2.1- співвідношення тестових та тренувальних даних

Тренуюча	Тестуюча	Всього
32142	8036	40178

```
[[[0.9647059 , 0.9647059 , 0.9647059 ],
 [0.9647059 , 0.9647059 , 0.9647059 ],
 [0.9647059 , 0.9647059 , 0.9647059 ],
 ...,
 [0.9647059 , 0.9647059 , 0.9647059 ],
 [0.9647059 , 0.9647059 , 0.9647059 ],
 [0.9647059 , 0.9647059 , 0.9647059 ]]]
```

```
[[[0.5372549 , 0.5686275 , 0.5803922 ],
 [0.6392157 , 0.6627451 , 0.6627451 ],
 [0.7137255 , 0.7176471 , 0.7372549 ],
 ...,
 [0.627451 , 0.59607846, 0.7882353 ],
 [0.5882353 , 0.5568628 , 0.7411765 ],
 [0.5647059 , 0.54901963, 0.6784314 ]],
```

Рисунок 2.7 – вигляд матриці зі значеннями пікселей

```
array([92, 91, 71, ..., 62, 8, 32])
```

Рисунок 2.8 – вигляд матриці з переведеними назвами рослин у значення

Далі здійснюється перехід до побудови моделі для навчання штучної нейронної мережі.

2.4.2 Розробка штучної нейронної мережі

Побудуємо першу модель штучної нейронної мережі на основі моделі CNN, як подано на рисунку 2.4. Створена модель складається з чотирьох шарів згортки, які перемножують значення фільтрів на вихідні значення пікселів зображення, після чого всі добутки сумуються і таким чином виходить згортка. Наприклад якщо на вхід подалося зображення 50 на 50 і згортка відбувається блоком 3 на 3, то на виході буде отримано 48 на 48. Також використовується три шари максимального пулінгу, які зменшують розмірність матриці вдвічі, для того щоб зменшити навантаження на оперативну пам'ять, бо попередні шари вже для себе відмітили нові ознаки зображення і нового можуть нічого не знайти. Також було використано чотири шари викидання, ці шари працюють шляхом випадкового зменшення кількості взаємопов'язаних штучних нейронів для того, щоб зменшити шанс перенавчання штучної нейронної мережі. Один з важливих шарів згладжування, він виконує зведення вхідних даних, якщо йому на вхід подати дані розмірністю 2 на 2, то на виході отримаємо просто масив з 4 значеннями. Останній шар – повнозв'язний, який виводить вектор для визначення класів. Модель подана на (рис 2.9)[17].

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 48, 48, 64)	1792
conv2d_1 (Conv2D)	(None, 46, 46, 128)	73856
dropout (Dropout)	(None, 46, 46, 128)	0
max_pooling2d (MaxPooling2D)	(None, 23, 23, 128)	0
conv2d_2 (Conv2D)	(None, 21, 21, 256)	295168
dropout_1 (Dropout)	(None, 21, 21, 256)	0
max_pooling2d_1 (MaxPooling2D)	(None, 10, 10, 256)	0
conv2d_3 (Conv2D)	(None, 8, 8, 512)	1180160
dropout_2 (Dropout)	(None, 8, 8, 512)	0
max_pooling2d_2 (MaxPooling2D)	(None, 4, 4, 512)	0
flatten (Flatten)	(None, 8192)	0
dense (Dense)	(None, 512)	4194816
dropout_3 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 100)	51300

=====
Total params: 5,797,092
Trainable params: 5,797,092

Рисунок 2.9 – структура першої моделі

Далі відбувається запуск тренування моделі з такими параметрами: оптимізатор Adam(), функція розрахунку похибки - розріджена категорична крос-ентропія і критерій зупинки – 25 епох.

```
Epoch 21/25
1005/1005 [=====] - 24s 24ms/step - loss: 0.4500 - accuracy: 0.8675 - val_loss: 1.6458 - val_accuracy: 0.6850
Epoch 22/25
1005/1005 [=====] - 25s 25ms/step - loss: 0.4395 - accuracy: 0.8725 - val_loss: 1.6550 - val_accuracy: 0.6939
Epoch 23/25
1005/1005 [=====] - 25s 25ms/step - loss: 0.4029 - accuracy: 0.8800 - val_loss: 1.7414 - val_accuracy: 0.6834
Epoch 24/25
1005/1005 [=====] - 25s 25ms/step - loss: 0.3904 - accuracy: 0.8857 - val_loss: 1.6916 - val_accuracy: 0.6862
Epoch 25/25
1005/1005 [=====] - 24s 24ms/step - loss: 0.4098 - accuracy: 0.8816 - val_loss: 1.7347 - val_accuracy: 0.6909
```

Рисунок 2.10 – останні епохи тренування

Після тренування моделі (рис. 2.10) проводиться відображення на графіках точності і похибки моделі на кожній епосі.

На рисунку 2.11 відображено точність моделі на тренувальних даних – червона лінія і значення майже 90%, а синьою позначена точність тестуючих даних і воно має 69%.

На рисунку 2.12 видно значення похибки моделі на тренувальних даних – червона лінія, значення нижче 0.5, а синьою позначено значення функції похибки тестуючих даних, воно гірше і становить 1.6.

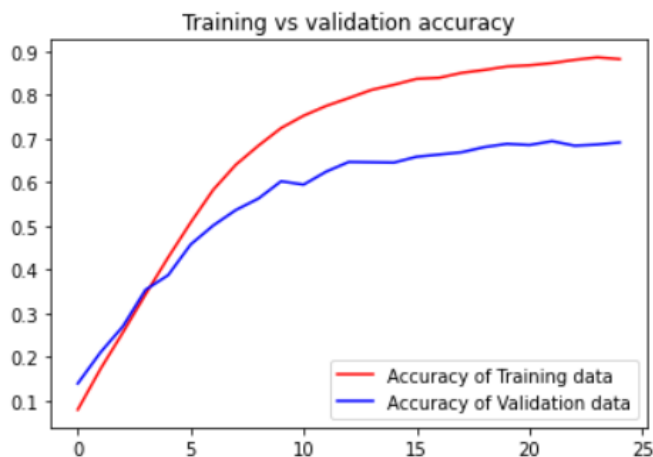


Рисунок 2.11 – графік точності класифікації першої моделі



Рисунок 2.12 – графік значень похибки класифікації першої моделі

Розглянемо рисунок 2.13, на якому зображено результат класифікації першої моделі на 24 зображеннях. Результати вийшли непогані – шість зображень тільки було невірно класифіковано. Навіть ті які неправильно класифікувало, віддана перевага з маленькою вірогідністю. Також можна помітити, що колір квітів, які класифікувало невірно схожий на інші рослини, тому запропонована модель і видала такі результати.

Можна сказати, що результат для класифікації ста класів непоганий, але можна й краще, якщо була би потужність ноутбуку краща.



Рисунок 2.13 – результат роботи першої моделі

Побудуємо ще одну модель штучної нейронної мережі для порівняння. Видалимо один шар пулінгу у третьому блоці згортки. Додаємо у другий блок один згортковий шар, в якому змінимо параметр згортки. Згортка відбувалася блоком 3 на 3, а тепер 4 на 4. Також змінився і в першому блоці один такий шар (рис 2.14).

Layer (type)	Output Shape	Param #
conv2d_19 (Conv2D)	(None, 48, 48, 64)	1792
conv2d_20 (Conv2D)	(None, 45, 45, 128)	131200
dropout_21 (Dropout)	(None, 45, 45, 128)	0
max_pooling2d_12 (MaxPoolin g2D)	(None, 22, 22, 128)	0
conv2d_21 (Conv2D)	(None, 20, 20, 128)	147584
conv2d_22 (Conv2D)	(None, 17, 17, 128)	262272
dropout_22 (Dropout)	(None, 17, 17, 128)	0
max_pooling2d_13 (MaxPoolin g2D)	(None, 8, 8, 128)	0
conv2d_23 (Conv2D)	(None, 6, 6, 256)	295168
dropout_23 (Dropout)	(None, 6, 6, 256)	0
flatten_6 (Flatten)	(None, 9216)	0
dense_12 (Dense)	(None, 512)	4719104
dropout_24 (Dropout)	(None, 512)	0
dense_13 (Dense)	(None, 100)	51300

=====
 Total params: 5,608,420
 Trainable params: 5,608,420

Рисунок 2.14 – друга модель ШНМ

Модель на рисунку 2.14 тренується на таких же параметрах, що й попередня: оптимізатор Adam(), функція похибки - розріджена категорична крос-ентропія і 25 епох.

Проведемо аналіз двох графіків на рисунках 2.15 і 2.16. Модель почала навчатися дуже добре і видавала гарні результати, але після епохи 14 вона перестала ефективно навчатися і темп дуже сильно знизився. В результаті точність дійшла до 66%.

Проводився аналіз графіків з похибкою. Тут теж саме, навіть гірше, бо після епохи 14 значення похибки почало збільшуватися – це означає, що модель вже просто почала запам'ятовувати, а не вчитися.

Дана модель була ефективна тим, що навчилася швидше ніж перша, але в результаті дала меншу точність і більшу похибку.

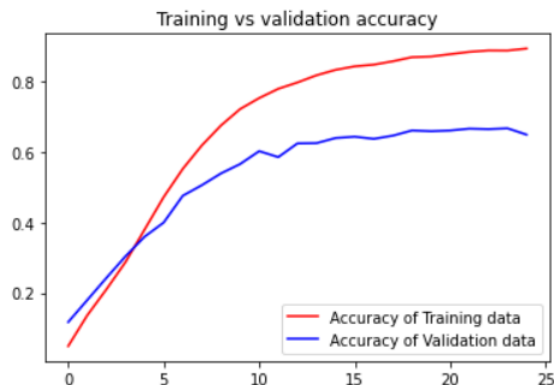


Рисунок 2.15 – графік точності класифікації другої моделі

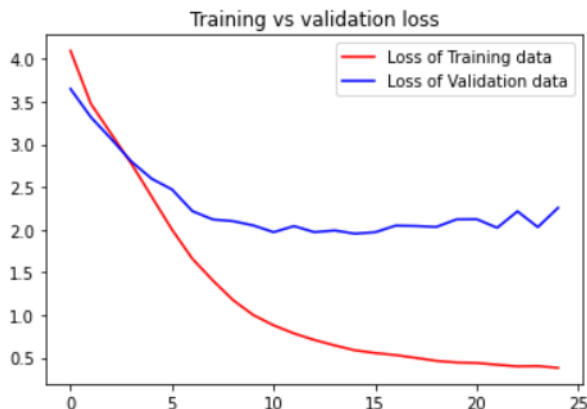


Рисунок 2.16 – графік значень похибки класифікації другої моделі

Проведемо аналіз зображень на рисунку 2.17 з класифікованими квітами. З 24 зображень класифіковано невірно тільки 4. Тепер подивимося на зображення квітів, які класифіковані вірно. Аналіз показує, що точність класифікації не є 100%, отже створена штучна нейронна мережа сумнівається в своєму виборі і не може точно сказати, що на зображенні дана квітка.

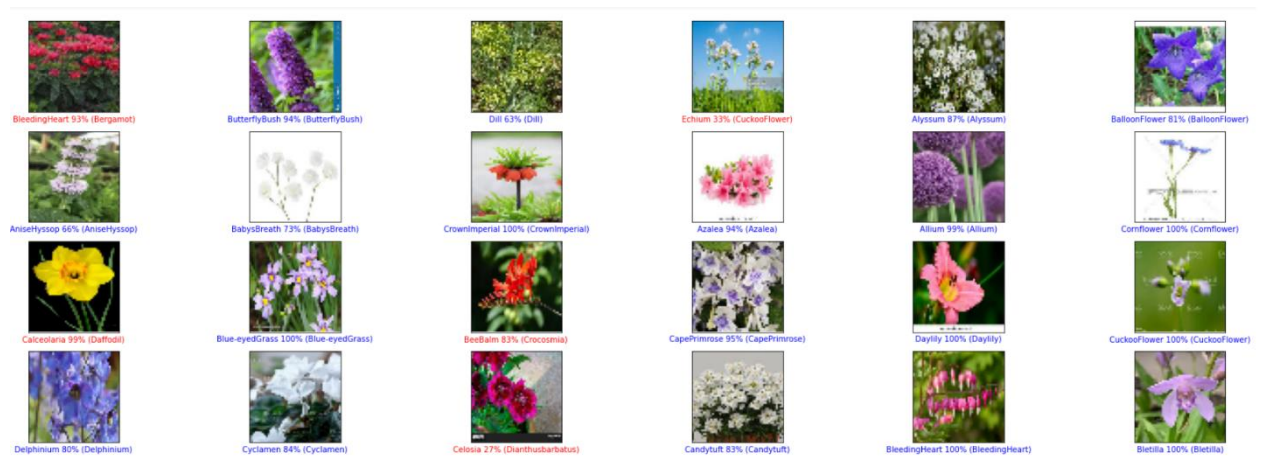


Рисунок 2.17 – результати роботи другої моделі

Для повного аналізу була обрана ще одна модель, в якій додано більше штучних нейронів в шарах моделі і також додано декілька згорткових шарів і змінено параметри в існуючих (рис. 2.18)

Layer (type)	Output Shape	Param #
conv2d_8 (Conv2D)	(None, 46, 46, 256)	19456
max_pooling2d_6 (MaxPooling 2D)	(None, 23, 23, 256)	0
dropout_8 (Dropout)	(None, 23, 23, 256)	0
conv2d_9 (Conv2D)	(None, 20, 20, 256)	1048832
max_pooling2d_7 (MaxPooling 2D)	(None, 10, 10, 256)	0
dropout_9 (Dropout)	(None, 10, 10, 256)	0
conv2d_10 (Conv2D)	(None, 6, 6, 512)	3277312
max_pooling2d_8 (MaxPooling 2D)	(None, 3, 3, 512)	0
conv2d_11 (Conv2D)	(None, 1, 1, 512)	2359808
flatten_2 (Flatten)	(None, 512)	0
dense_4 (Dense)	(None, 512)	262656
dropout_10 (Dropout)	(None, 512)	0
dense_5 (Dense)	(None, 100)	51300

=====
Total params: 7,019,364
Trainable params: 7,019,364

Рисунок 2.18 – структура третьої моделі ШНМ

Модель на рисунку 2.18 запускається на тих же параметрах, що й попередні лише змінюється кількість епох на 40 і кількість зображень, які проходять один цикл обробки це 60.

На рисунках 2.19 і 2.20 зображено точність і похибка третьої моделі, в якій збільшено кількість штучних нейронів та шарів. Точність класифікації становить 64%, але значення похибки дуже погане 2.44. Це все трапилось із-за того, що було додано більше шарів, а зображення, яке класифікується залишилось в тих же розмірах. Тому потрібно збільшувати розмір зображення, але потужність ноутбука цього не дозволяє.

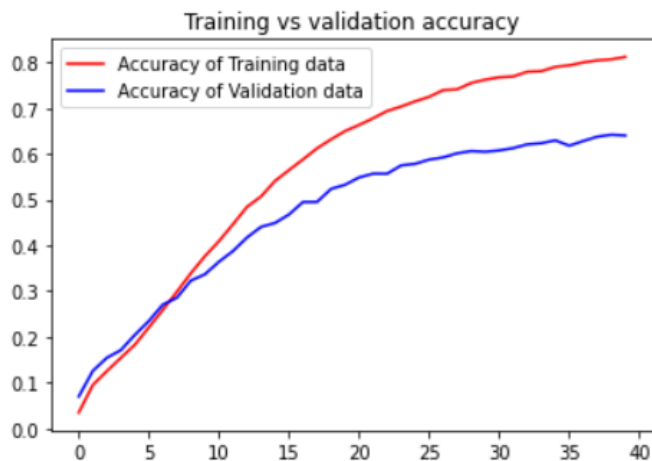


Рисунок 2.19 – графік точності класифікації третьої моделі

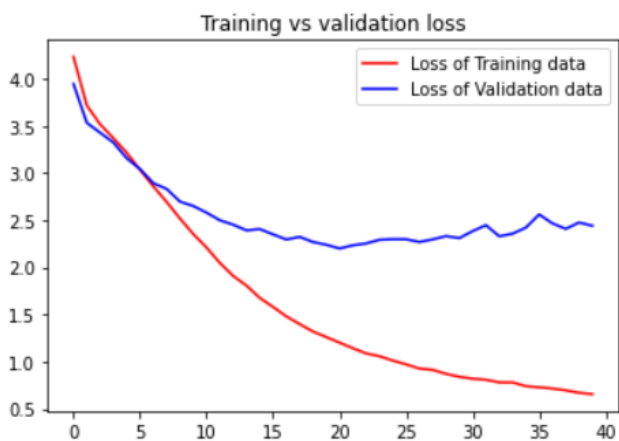


Рисунок 2.20 – графік значень похибки класифікації третьої моделі

На рисунку 2.21 класифікація зображень за допомогою третьої моделі. Нейронна мережа класифікувала неправильно 6 зображень. Серед цих шести фотографій, також є квіти, які класифікувала модель не з високою вірогідністю.



Рисунок 2.21 – результати роботи третьої моделі

2.5 Висновки результатів роботи ШНМ

Порівнявши результати роботи трьох моделей штучних нейронних мереж, можна зробити такі висновки. Найточнішою моделлю була перша з точністю класифікації у 69%. Дана точність залежить від побудови штучної нейронної мережі і вхідних даних. Чим більше зображення подається на вхід тим більше потрібно використовувати згорткових шарів для моделі. Також в модель потрібно додавати такі шари, як максимального пулінгу для того, щоб скорочувати розмірність зображення і не навантажувати оперативну пам'ять і шари втрат для того, щоб штучні нейрони втрачали між собою зв'язок і не було перенавчання штучної нейромережі.

2.6 Архітектура застосунку

Веб-застосунок матиме просту клієнт-серверну структуру і складатися з чотирьох вузлів: серверний, клієнтський навчання ШНМ та даних (рис. 2.22).

Серверний вузол складається з наступних компонентів:

- Підготовка зображення – форматує зображення та нормалізує його;
- Передання зображення в модель;
- Класифікація – надання назви квітки зображеної на зображенні;
- Підготовка даних – формування масиву з назвою рослини, її відсотку та догляду з нею;
- Серверна мережа.

Серверна частина буде обробляти дані системи і за запитом користувача виконувати функцію класифікації за допомогою вбудованої штучної нейронної мережі. Всі результати будуть виводитись користувачу в інтерфейсі застосунку.

Клієнтський вузол складається з наступних компонентів:

- Інтерфейс користувача;
- Отримання зображення користувача;
- Клієнтська мережа.

Клієнтська частина буде відповідати за надання доступу користувачу до використання основного функціоналу застосунку та відображення й візуалізацію даних системи. Всі файли, які завантажує користувач відправляються на сервер для подальшої обробки.

Вузол побудови ШНМ складається з наступних компонентів:

- Підготовка зображень для навчання;
- Побудова моделі;
- Навчання;
- Створена модель.

Даний вузол створює штучну нейронну мережу, яка потім зберігається та використовується в самому застосунку.

Вузол даних складається з наступних компонентів:

- Таблиця з даними;
- Створення матриці даних.

Даний вузол містить файл з даними про догляд за рослинами і в ньому робиться матриця даних, до якої буде звертатися система для виводу інформації для відповідної квітки.

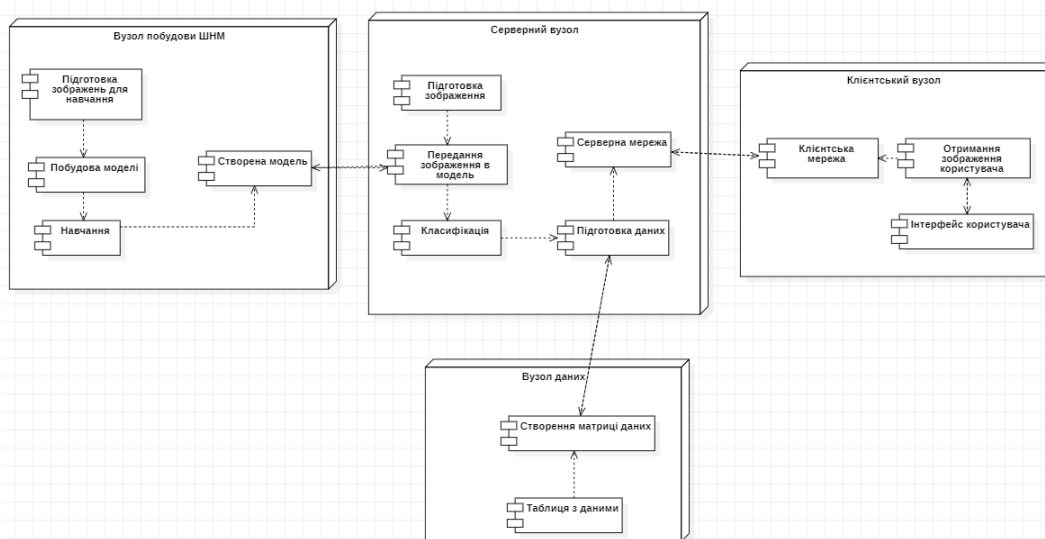


Рисунок 2.22 – Компонентна архітектура застосунку

Для розуміння послідовності процесів веб-застосунку було розроблено UML діаграму послідовності (рис. 2.23). З поданої діаграми бачимо, що користувач завантажує зображення, яке надсилається в програмний модуль і там воно обробляється під норми штучної нейронної мережі. Після чого ШНМ видає нам назву квітки зображеної на малюнку. В результаті користувач у себе на екрані отримає назву квітки та як за нею доглядати.

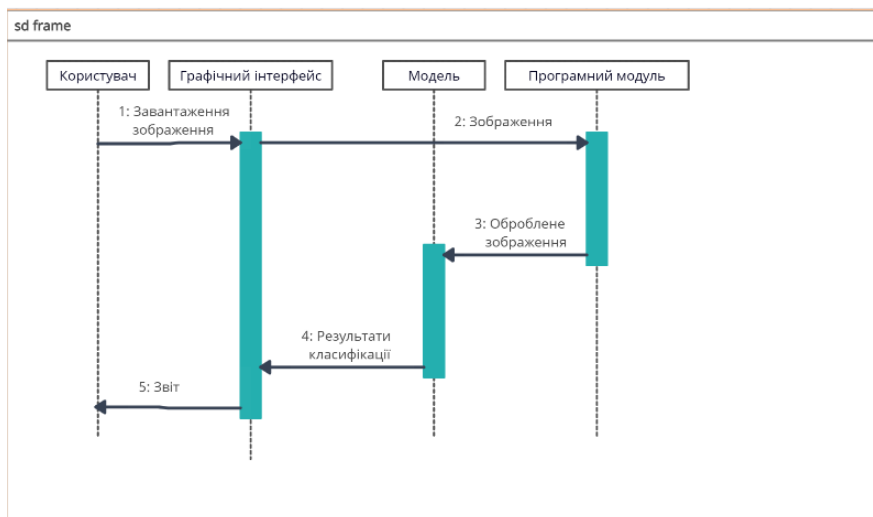


Рисунок 2.23 – UML діаграма послідовності

Також було розроблено UML діаграму прецедентів (рис. 2.24), яка демонструє, відношення між елементами та прецедентами в системі.

Завантаження даних. Користувачу надається можливість завантажити своє зображення квітки для класифікації.

Нормалізація зображення. Застосунок приймає зображення та приводить його до належного виду, щоб штучна нейронна мережа могла класифікувати квітку.

Передача зображення до моделі. Нормалізоване зображення потрапляє до моделі для класифікації.

Класифікація. Під час класифікації ми отримуємо назву квітки та як за нею доглядати.

Отримання результатів. Результат, який виводиться на екран користувачу після класифікації.



Рисунок 2.24 – UML діаграма прецедентів

Було розроблено UML діаграму діяльності (рисунок 2.25), яка демонструє, що може робити користувач в системі. Спочатку користувач потрапляє у веб-застосунок звідки він може перейти на три різні сторінки в залежності, що йому потрібно. Якщо він залишиться на початковій сторінці, то він може класифікувати завантажене зображення і після цього перейти на сторінку з доглядом за цією квіткою. Також він може відвідати сторінку з бібліотекою рослин або як працює сайт.

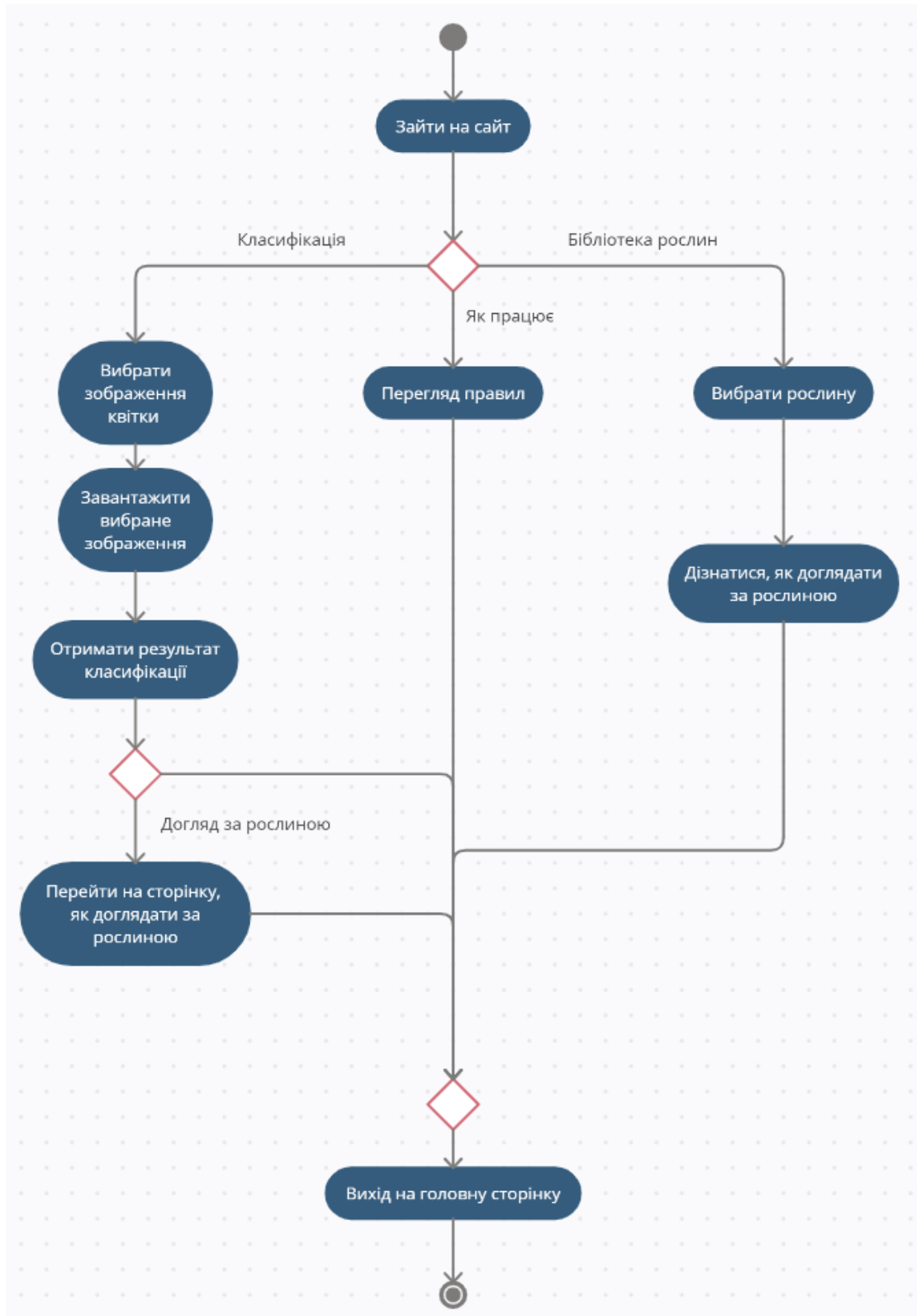


Рисунок 2.25 – Діаграма діяльності

2.7 Висновок до розділу

У даному розділі детально розглянуто різні види штучних нейронних мереж та які задачі вони виконують краще. З цієї інформації прийнято рішення використовувати конволюційну штучну нейронну мережу для вирішення поставленої задачі.

Окрім того, було розглянуто різні інструменти для реалізації системи розпізнавання зображення.

Також, у цьому розділі було розглянуто детально процес навчання ШНМ, а також сама реалізація системи класифікації, а саме: підготовка даних для класифікації, побудова моделі штучної нейронної мережі. Було створено три моделі, які між собою порівняно і обрано найкращу.

Розділ 3. Розробка програмного забезпечення інтелектуальної системи аналізу зображень квітів рослин

3.1 Структура застосунку

Застосунок складається із наступних об'єктів (рис. 3.1):

- Папки
 - «templates» - папка з шаблонами, які використовуються для відтворення сторінок веб-сайту.
 - «static» - папка зі статичними файлами. В цій папці зазвичай розташовують різні файли: css, зображення та інші файли. За допомогою цього організується структура проекту і всі файли, які там зберігаються мають один шлях. Таким чином Flask з легкістю буде знаходити всі потрібні файли для проекту.
- Python скрипт
 - «main.py» - головний файл додатку, в якому будуються функції та маршрути для оптимальної роботи програмного застосунку, щоб він міг відповідати на всі запити користувача. Також до нього підключається створена штучна нейронна мережа та датасет з інформацією про догляд за рослинами.

Розглядаючи рисунок 3.1 бачимо, що в папці «static» є такі папки та файли «css» - папка, якій розписані стилі сторінок, «Flowers» - папка, яка містить зображення рослин, які розміщені в бібліотеці, «img» - папка, в якій зберігаються фотографії, які завантажує користувач сайту. Далі йдуть такі файли «Flower.xlsx» - ексель таблиця, в якій зберігається інформація про догляд за рослинами, «my_model.h5» - сама модель, яка класифікує зображення та останній файл зображення фону сайту.

В папці «templates» містяться всі HTML сторінки веб-застосунку.

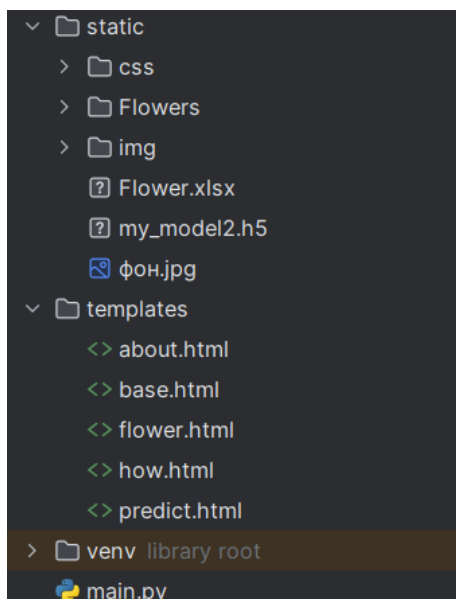


Рисунок 3.1 – структура застосунку

3.2 Опис даних

Для дипломної роботи було використано два датасети: перший містить зображення 300 видів квітів по 300-400 зображень кожного виду, а другий містить інформацію за доглядом цих квітів.

Розглянемо перший датасет, для прикладу візьмемо декілька зображень різних квітів (рис. 3.2).



Рисунок 3.2 – зображення квітів

На рисунку 3.2 можна побачити, що кожна квітка характерно відрізняється за кольором та формою, саме на цьому й вчилася штучна нейронна мережа, яка використовувалась у роботі.

Наступний датасет (рис. 3.3) містить такі поля:

- Name – назва рослини;
- Desc – короткий опис про рослину;
- PlantType – тип рослини;
- Color – колір квітки;
- HardinessZones – рівень морозостійкість рослини;
- BloomsIn – пора року коли розквітає рослина;
- Height – висота рослини;
- SoilNeeds – потреба ґрунту;
- SunNeeds – потреба сонця;
- WaterNeeds – потреба води;
- Maintenance – потреба росли у догляді.

Name	Desc	PlantType	Color	HardinessZones	BloomsIn	Height	SoilNeeds	SunNeeds	WaterNeeds	Maintenance
Aconite	Aconitum, також відомий як Aconite, — це рід понад 300 видів квіткових рослин із родини	Багаторічники	Квіти бувають блакитні	3-8	Осінь, Літо	Аконіт — це висока рослина	Аконіт найкраще розвивається	Багато сонця, Ч	середній	середній
Agapanthus	Agapanthus — це рід, що складається з близько 10 видів багаторічних рослин родини Amaryllidaceae.	Цибулини, Багаторічники	Ці квіти зазвичай білі	6-11	Осінь, Літо	Висота агаяпантуса варіюється	Агаяпантус найкраще розвивається	Багато сонця, Ч	середній	низький
Ageratum	Ageratum — це рід близько 50 видів однорічних рослин, що належать до родини Asteraceae, що	Однорічники	Квіти бувають відтінки	2-11	Осінь, Літо	Ці рослини можуть вирости	Агератум віддає перевагу	Багато сонця, Ч	середній	низький
Allium	Allium — це рід понад 800 видів цибулинних багаторічних рослин родини Amaryllidaceae, які	Цибулини, Багаторічники	Більшість Alliums має	3-10	Весна, Літо	Аліуми мають різні розміри	Аліуми не вимагають дос	Багато сонця, Ч	середній	низький
Alstroemeria	Alstroemeria, також відома як перуанська лілія, — це рід близько 60 видів багаторічних рослин	Цибулини, Багаторічники	Квіти бувають різних	5-11	Осінь, Літо	Різновиди варіюються від	Рослина найкраще росте	Багато сонця, Ч	середній	низький
Amaryllis	Amaryllis — рід двох видів квіткових цибулин — Amaryllis belladonna і Amaryllis paradisicola. Ці два	Цибулини, Багаторічники	Забарвлення квітки з	7-10	Осінь, Літо	Амаріліс має стебло, яке	Амаріліс найкраще росте	Багато сонця, Ч	середній	низький
Anemone	Anemone — рід близько 120 видів багаторічних квіткових рослин родини Ranunculaceae, що	Багаторічники	Квіти можна знайти в	4-11	Весна, Осінь, Літо	Анемона може вирости	Рослина любить вологі	Багато сонця, Ч	середній	низький

Рисунок 3.3 – датасет про догляд за рослинами

3.3 Опис інтерфейсу

3.3.1 Опис основних сторінок

Сторінка, на яку потрапляє користувач вперше – це головна сторінка (рис. 3.4), на якій є верхнє меню, тіло сайту та нижнє меню. За допомогою верхнього та нижнього меню, яке використано на всіх сторінках сайту, користувач може переходити на інші сторінки. В тілі сайту знаходиться поле для додавання зображення та можливість завантажити його до сайту. Після того, як користувач завантажив фото до сайту, йому виводить зображення, яке він завантажив, а також назву рослини зображену на рисунку з відсотком класифікації, окрім цього

виводиться ще три можливі назви рослини зображеної на малюнку. Після того, як ШНМ класифікувала рослини, користувач може перейти за посиланням, щоб дізнатися, як доглядати за цією рослиною.

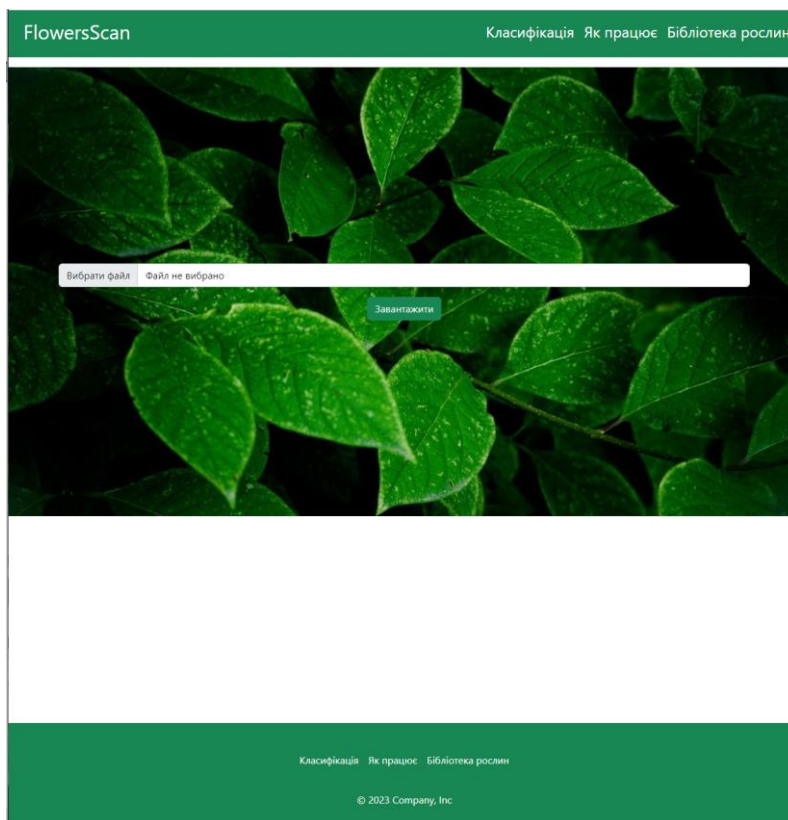


Рисунок 3.4 – головна сторінка

Далі користувач може відвідати сторінку, на якій по пунктах розписано, як користуватися сайтом (рис. 3.5), також можна відразу перейти на відповідні сторінки, які зазначені в пунктах.

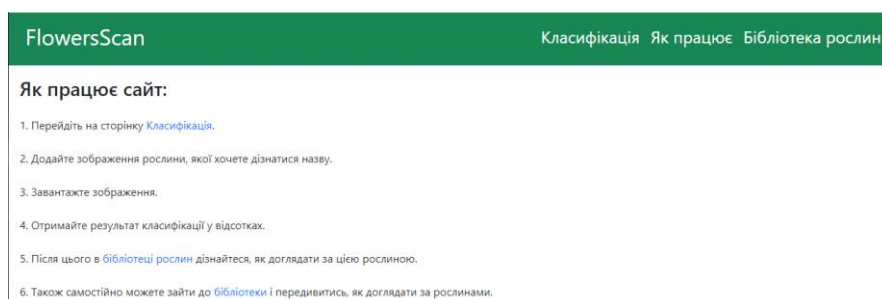


Рисунок 3.5 – сторінка з правилами користування

Наступна сторінка – бібліотека рослин (рис. 3.6). На цій сторінці подані назви рослин, про яких користувач може отримати інформацію, а саме, як доглядати за ними.

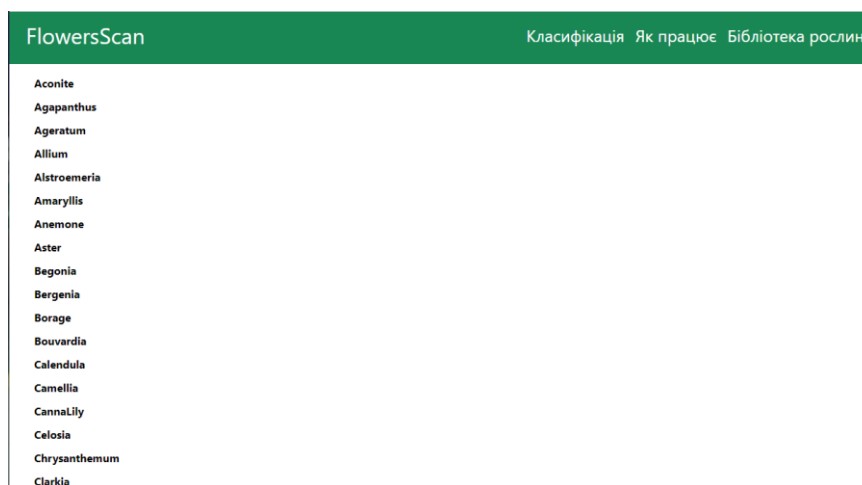


Рисунок 3.6 – сторінка з бібліотекою рослин

Ще одна сторінка сайту – інформація про рослину (рис. 3.7). На цій сторінці виводяться три зображення квітки та інформація, про те як доглядати за рослиною, яка розміщена в таблиці (рис. 3.3). Також є кнопка назад, яка повертає користувача на одну сторінку назад.

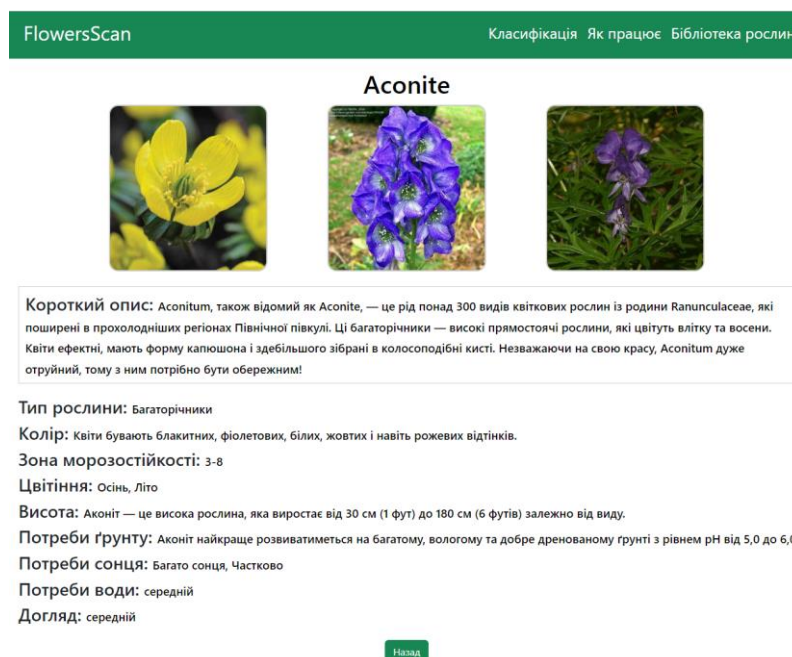


Рисунок 3.7 – сторінка з інформацією про рослину

3.3.2 Опис переходів між сторінками

На рисунку 3.8 нижче показано можливі переходи між сторінками.

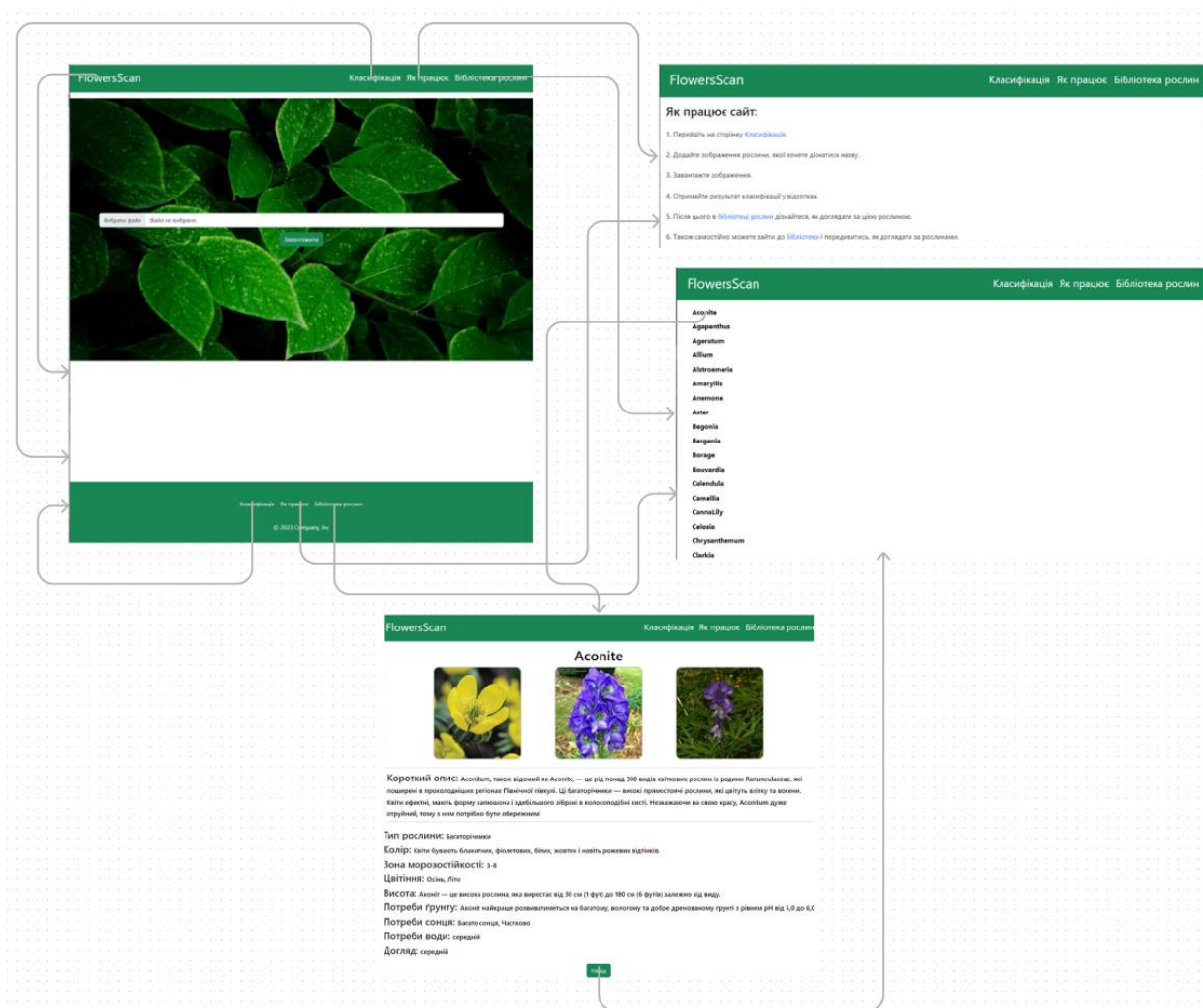


Рисунок 3.8 – макет можливих переходів між сторінками в Figma

Як бачимо, з головної сторінки, куди заходять всі користувачі, ми можемо перейти або на цю ж сторінку, або на сторінку, щоб дізнатися, як працює сайт, або в бібліотеку рослин, для того, щоб переглянути, які квіти є.

Зі сторінки, про те як користуватися сайтом, ми можемо перейти на сторінку з класифікацією або потрапити у бібліотеку рослин.

На сторінці бібліотека рослин, клацнувши на будь-яку назву рослини, можна потрапити на іншу сторінку про те, як потрібно доглядати за цією рослиною.

Зі сторінки про рослину можна повернутися назад до бібліотеки за допомогою кнопки “назад”.

3.4 Опис тест-кейсів

Однією з функціональних вимог до системи є зручний та зрозумілий інтерфейс. Якщо користувач заїде на сайт, то він одразу потрапить на сторінку з прогнозуванням куди може відразу завантажити своє зображення. Також після класифікації зображення він може подивитися, як доглядати за рослиною. Якщо користувачу все таки буде, щось не зрозуміле, то він може перейти на сторінку з правилами користування сайтом і там вже все доступно розписано, що потрібно робити користувачу.

Наступною вимогою було – створити сам класифікатор. Для демонстрації роботи візьмемо п’ять зображень різних рослин (рис. 3.9) та перевіримо на скільки коректно працює штучна нейронна мережа.



Рисунок 3.9 – тестові рослини

На рисунку 3.9 зображені такі рослини: Ageratum, Aster, Camellia, Clarkia, Dahlia.

Після того, як користувач завантажить зображення на сайті (рис. 3.10), вони пройдуть обробку. Спочатку зображення зменшиться до розмірів 50 на 50 пікселів, потім кожен піксель переведеться у значення і сформується матриця значень пікселів. Оброблене вхідне зображення отримує штучна нейронна мережа, яка класифікує його за допомогою коефіцієнтів, які вона знайшла після навчання. Коли штучна нейронна мережа закінчує класифікувати на цій же сторінці виводиться назва рослини та з якою ймовірністю мережа класифікувала вхідне зображення (рис. 3.11). Також виводиться ще три можливі назви на які може бути схожа ця рослина.

Завантажимо перше зображення на сайт та перевіримо чи правильно класифікує штучна нейронна мережа.

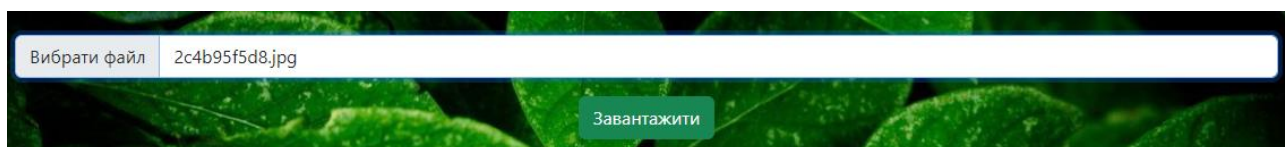


Рисунок 3.10 – завантажене зображення

Ваше зображення:



Результат прогнозу: *Ageratum*, 99.17%

Також можлива така назва: Bellflower, 0.66%

Також можлива така назва: EnglishBluebell, 0.09%

Також можлива така назва: Chrysanthemum, 0.07%

Рисунок 3.11 – класифіковане зображення *Ageratum*

На рисунку 3.11 видно, що штучна нейронна мережа класифікувала перше зображення правильно та ще й з високою ймовірністю в 99.17%. Також їй ця

рослина схожа на Bellflower, EnglishBluebell та Chrysanthemum, але з маленькою ймовірністю.

Перевіримо наступні зображення.

На рисунку 3.12 видно, що зображення класифіковані з ймовірністю майже 90%. Також й віддало 8.5% на те, що це може бути Bluebonnets (рис. 3.13). Скоріш за все мережа так подумала бо ці дві квітки мають майже однакову забарвленість.

Ваше зображення:



Результат прогнозу: Aster, 89.51%

Також можлива така назва: Bluebonnets, 8.54%

Також можлива така назва: Crocus, 1.17%

Також можлива така назва: Brassica, 0.69%

Рисунок 3.12 – класифіковане зображення Aster



Рисунок 3.13 – квітка Bluebonnets

На рисунку 3.15 видно, що класифікація відбулась вже не так впевнено, але все одно більше половини і становить 61%, а також 23% віддало, що це може бути Carnation (рис. 3.14). Порівнюючи ці квітки видно, що вони між собою трохи схожі, але ШНМ все таки обрала правильну назву для завантаженого зображення.



Рисунок 3.14 – квітка Carnation

Ваше зображення:



Результат прогнозу: Camellia, 61.12%

Також можлива така назва: Carnation, 23.88%

Також можлива така назва: Diascia, 5.23%

Також можлива така назва: Delphinium, 3.02%

Рисунок 3.15 – класифіковане зображення Camellia

Рисунок 3.16 показує, що штучна нейронна мережа класифікувала квітку з ймовірністю у 65%, а також з ймовірністю 15.5% думає, що це Candytuft (рис. 3.17). Квітки на рисунках 3.16 і 3.17 схожі трохи за формою але Clarkia більша за розміром від Candytuft. На рисунку 3.18 зображену квітку, яку ШНМ класифікувала з ймовірністю у 69% та з незначною кількістю відсотків (6%) дала назву CannaLily.

Ваше зображення:



Результат прогнозу: Clarkia, 64.83%

Також можлива така назва: Candytuft, 15.47%

Також можлива така назва: DesertRose, 5.6%

Також можлива така назва: BalloonFlower, 3.47%

Рисунок 3.16 – класифіковане зображення Clarkia



Рисунок 3.17 – квітка Candytuft

Ваше зображення:



Результат прогнозу: Dahlia, 68.47%

Також можлива така назва: Cannalily, 5.78%

Також можлива така назва: Clarkia, 3.4%




Також можлива така назва: Chionodoxa, 2.73%

Рисунок 3.18 – класифіковане зображення Dahlia

Аналіз тестових зображень показав, що ШНМ класифікує правильно хоч і не завжди з високою точністю, але все таки правильно. Тому створену систему на основі штучної нейронної мережі можна використовувати для, того, щоб класифікувати квіти за допомогою їх зображення.

Наступною функціональною вимогою було створення бібліотеки квітів, як показано на рисунку 3.6. Останньою вимогою було створення сторінки для кожної квітки, як правильно доглядати за нею. Це також було зроблено. На цю сторінку можемо потрапити з бібліотеки рослин, а також після того, як ми завантажили зображення. На сайті з'являється посилання на сторінку з цією квіткою. Візьмемо приклад рисунок 3.18. На ньому є назва квітки виділена зеленим кольором, при натисканні на назву здійснюється перехід на відповідну сторінку (рис. 3.19). На сторінці відображається ще різні види цих рослин, а також можна докладно прочитати про цю квітку: її опис та, що їй потрібно для того, щоб вона у вас цвіла і була в гарному стані.

Dahlia

Короткий опис: Dahlia — це рід бульбоподібних трав родини складноцвітих, які походять із Центральної та Південної Америки. Існує 42 види цієї багаторічної рослини, і кожен з них має різний колір, розмір і форму. Жоржина вважається одним із найпопулярніших садових квітів через її зубчасте зелене листя та привабливі барвисті квіти.

Тип рослини: Цибулини, Багаторічники

Колір: Цвітуть жоржини всіма кольорами, крім синього.

Зона морозостійкості: 3-11

Цвітіння: Осінь, Літо

Висота: Висота видів різна: від 40 см (1 фут) до 2 метрів (7 футів).

Потреби ґрунту: Жоржина найкраще розвивається на багатому, вологому та добре дренованому ґрунті злегка кислотою.

Потреби сонця: Багато сонця

Потреби води: середній

Догляд: середній

[Назад](#)

Рисунок 3.19 – перехід на сторінку догляду за рослиною після класифікації

Наступні вимоги, які необхідно протестувати – нефункціональні. І першою такою вимогою було стійкість сайту. Це означає, що якщо користувач, щось робить не коректно, то наш сайт не ламається і все одно продовжує працювати.

Перше, що було зроблено це надання можливості завантажувати тільки зображення формату .jpg, всі інші формати воно не бачить (рис. 3.20). Це було зроблено для того, щоб система не ламалась, бо на вході штучна нейронна мережа отримувала не зображення, а якийсь інший файл і вона не знала б, що з ним робити і була би помилка із-за якої сайт не працював.



Рисунок 3.20 – видимість тільки формату .jpg

Також, якщо користувач ще не вибрав фото і захоче завантажити пусте поле, то сайт просто нічого не буде робити і перенаправить його на цю ж сторінку заново. Також було зроблено скрите поле інформації про класифіковане зображення доки не буде завантажено зображення.

Всі зображення, які користувач завантажує, зберігаються в папці сайту, таким чином можна розширяти дані для навчання штучної нейронної мережі.

Також розроблений сайт зв'язаний з таблицею, в якій записана вся інформація про догляд рослин і воно завжди звертається до неї коли відкривається сторінка відповідної рослини, тому сайт завжди в робочому стані і має коректний зв'язок з даними.

3.5 Висновок до розділу

Отже, у даному розділі було детально розглянуто розробку інтерфейсу: які сторінки було розроблено, який функціонал доступний на кожній з сторінок. Також наведено схему переходів між сторінками та якими кнопками вони пов'язані між собою. Більш того, проведено тестування системи на коректність різних дій користувача, а також проаналізовано правильність роботи штучної нейронної мережі на сайті за допомогою п'яти зображень, які були вибрані в якості тестуючої вибірки.

Також було описано дані, які використовуються для тренування штучної нейронної мережі та представлення даних на сайті.

Висновки

У ході виконання дипломної роботи було розроблено конволюційну штучну нейронну мережу, яка найбільше підходила до поставленої задачі, а саме класифікація зображень квітів. Дана штучна нейронна мережа працює таким чином, що задається ядро згортки при якому вхідне зображення поступово зменшується для знаходження оптимальних коефіцієнтів. Також використовується шар для відрізання зображення, щоб мережа могла знайти інші залежності на зображенні.

Також у роботі розглянуто та детально описано й інші методи класифікації та різновиди нейронних мереж, серед, яких і було обрано мережу, яка була використана для дипломної роботи.

В роботі описано, як правильно будувати штучні нейронні мережі, та як правильно підготовляти зображення для них. Підготовка полягає в створенні тренуючої та тестуючої вибірки і перетворення зображення в матрицю значень пікселів. Детально розписано, що робити після того, як побудували мережу та підготували дані, а саме, як її навчити на даних, щоб вона надалі правильно класифікувала зображення. ШНМ не повинна перенавчитися і повинна отримати достатню кількість даних для навчання. Представлений детальний опис дає можливість сформулювати практичну настанову програмного застосування при роботі ШНМ для розпізнавання квітів.

Для презентації роботи штучної нейронної мережі розроблено сайт, на якому є сторінка з полем для вводу зображення та кнопкою завантажити зображення для класифікації. В результаті користувач отримує класифіковане зображення з відсотком, який указує наближення до класу квітки. Також є сторінка з правилами користуванням, бібліотекою рослин та описом, як доглядати за кожною рослиною.

Список використаних джерел

1. Machine Learning Algorithms – A Review, Batta Mahesh:
[https://www.researchgate.net/publication/344717762_Machine_Learning_Algorithms_-A_Review](https://www.researchgate.net/publication/344717762_Machine_Learning_Algorithms_-_A_Review)
2. Priya Pedamkar Pattern Recognition Algorithms:
<https://www.educba.com/pattern-recognition-algorithms/>
3. Applied Soft Computing Volume 38 January 2016 pages 788-804
4. Oludare Isaac Abiodun , Aman Jantan, Abiodun Esther Omolara, Kemi Victoria Dada, Nachaat AbdElatif Mohamed, Humaira Arshad, State-of-the-art in artificial neural network applications: A survey:
<https://www.sciencedirect.com/science/article/pii/S2405844018332067>
5. Yajiao Tang, Junkai Ji, Shangce Gao, Hongwei Dai, Yang Yu, Yuki Todo A Pruning Network Model in Credit Classification Analysis:
<https://www.hindawi.com/journals/cin/2018/9390410/>
6. Journal of Applied Biomedicine Volume 11 Issue 2 2013
7. Smriti Tikoo, Nitin Malik, Detection, Segmentation and Recognition of Face and its Features Using Neural Network: <https://arxiv.org/abs/1701.08259>
8. <https://lens.google/intl/uk/howlensworks/>
9. by Great Learning Team, Types of Neural Networks and Definition of Neural Network: <https://www.mygreatlearning.com/blog/types-of-neural-networks/>
10. Oliver Knocklein Classification Using Neural Networks:
<https://towardsdatascience.com/classification-using-neural-networks-b8e98f3a904f>
11. <https://numpy.org/>
12. <https://pandas.pydata.org/>
13. <https://docs.python.org/3/library/os.html>
14. <https://www.tensorflow.org/>
15. <https://flask.palletsprojects.com/en/2.3.x/>
16. How to Train Neural Networks Genevieve B. Orr and Klaus-Robert Müller
17. https://www.tensorflow.org/api_docs/python/tf/keras/layers

Додатки

Додаток А

Лістинг програми, де реалізовано навчання ШНМ.

```
model=tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(64,(3,3),activation='relu',input_shape=(50, 50, 3)),
    tf.keras.layers.Conv2D(128,(3,3),activation='relu'),
    tf.keras.layers.Dropout(0.25),
    tf.keras.layers.MaxPooling2D((2,2)),

    tf.keras.layers.Conv2D(256,(3,3),activation='relu'),
    tf.keras.layers.Dropout(0.25),
    tf.keras.layers.MaxPooling2D((2,2)),

    tf.keras.layers.Conv2D(512,(3,3),activation='relu'),
    tf.keras.layers.Dropout(0.25),
    tf.keras.layers.MaxPooling2D((2,2)),

    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(512,activation='relu'),
    tf.keras.layers.Dropout(0.3),
    tf.keras.layers.Dense(100,activation='softmax')
])

model.compile(optimizer='adam',loss='sparse_categorical_crossentropy',metrics=['accuracy'])
h = model.fit(X_train,y_train,epochs=25,validation_data=(X_test,y_test), batch_size=30)
```

Додаток Б

Лістинг програми, де реалізовану класифікацію зображення на сайті

```
def upload():
    if request.method == 'POST':
        if 'photo' not in request.files:
            return render_template("predict.html")
        photo = request.files['photo']
        if photo.filename == '':
            return render_template("predict.html")
        photo.save(os.path.join(app.root_path, 'static/img',
photo.filename))
        image = load_img('static/img/' + photo.filename,
grayscale=False, color_mode='rgb', target_size=(50, 50))
        image_ = img_to_array(image)
        image_ = image_ / 255.0
        image_ = np.array([image_])
        result = model.predict(image_)
        result = result.flatten()
        perc = []
        ind = []

        for i in range(4):
            ind.append((np.where(result == max(result))[0][0])+i)
            perc.append(round(result[ind[i]-i]*100, 2))
            result = np.delete(result, ind[i]-i)
            label_flower = []

        for i in range(4):
            label_flower.append(Label[ind[i]])

        return render_template("predict.html", label=label_flower,
photo='img/'+photo.filename, percent=perc)
    else:
        return render_template("predict.html")
```