МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
УМАНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
Факультет економіки, підприємництва та інформаційних технологій
Кафедра інформаційних технологій

Допущено до захисту
"_____" _____ 2026 р.
Зав. кафедри, к.т.н., доцент
_____ **Ліщук Р.І.**

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

РОЗРОБКА ДОДАТКУ ДЛЯ РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ ФРУКТІВ НА БАЗІ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ

Виконав – студент 4 курсу, групи 41-з-кн
освітньої програми "Комп’ютерні науки"
спеціальності 122 Комп’ютерні науки
Малінов Михайло Віталійович

Науковий керівник – к.т.н., доцент
Ліщук Роман Ігорович

Умань – 2026

Зміст

ВСТУП 3

РОЗДІЛ 1 АНАЛІЗ ПРОГРАМ ДЛЯ РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ ФРУКТІВ 5

1.1. Аналіз предметної області 5

1.2. Порівняльний аналіз програм для розпізнавання зображень фруктів 6

1.3. Постановка задачі 10

РОЗДІЛ 2 ПРОЕКТУВАННЯ ДОДАТКУ РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ ФРУКТІВ 12

2.1. Обґрунтування вибору засобів для вирішення поставлених задач 12

2.2. Проектування структури та поведінки системи 23

2.3. Алгоритм навчання згорткової нейронної мережі 26

2.4. Алгоритм розпізнавання зображень 28

РОЗДІЛ 3 РЕАЛІЗАЦІЯ ДОДАТКУ ДЛЯ РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ ФРУКТІВ НА БАЗІ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ 30

3.1. Розробка підсистеми навчання згорткової нейронної мережі 30

3.2. Розробка підсистеми розпізнавання зображень 45

3.3. Розробка веб-інтерфейсу додатку 48

3.4. Тестування розробленого додатку 51

ВИСНОВКИ 62

СПИСОК ЛІТЕРАТУРИ 64

ДОДАТКИ 67

ВСТУП

Задача ідентифікації та класифікації фруктів на цифрових зображеннях є одним із пріоритетних напрямів розвитку систем комп'ютерного зору. Практична цінність таких рішень охоплює сфери роздрібної торгівлі, агропромислового комплексу, логістики та автоматизованої обробки візуальних баз даних [1-3].

Складність машинного аналізу візуальних даних зумовлена високою варіативністю вхідних параметрів, серед яких інтенсивність освітлення, кут зйомки, фокусна відстань та наявність перекриттів. Використання класичних алгоритмів розпізнавання, зокрема шаблонного чи структурного підходів, виявляється малоефективним через неможливість вичерпної алгоритмізації всіх можливих просторових та оптичних спотворень.

Водночас сучасні дослідження підтверджують високу ефективність застосування апарату штучних нейронних мереж [4-8] для вирішення подібних задач. Нейромережеві моделі нівелюють вищезазначені недоліки завдяки здатності до виявлення закономірностей шляхом індуктивного навчання на репрезентативних масивах даних. Використання достатньо об'ємної навчальної вибірки забезпечує прийнятну точність класифікації без необхідності жорсткого детермінованого програмування ознак.

На сьогодні проблема повної автоматизації процесів візуальної ідентифікації фруктів залишається відкритою. Більшість існуючих програмних систем є пропрієтарними та мають вузьку спеціалізацію, яка суворо обмежується специфікою застосованих тренувальних наборів даних. З огляду на це, розробка власного адаптивного програмного комплексу для розпізнавання відповідних об'єктів становить значний науковий та практичний інтерес.

Обробка візуальних даних високої роздільної здатності за допомогою повнозв'язних штучних нейронних мереж, зокрема архітектури багатошарового персептрона, є обчислювально неефективною через надмірну кількість синаптичних ваг.

Натомість архітектура згорткових нейронних мереж спеціально оптимізована для роботи з просторовою топологією зображень [9-13]. Тому в даному дослідженні обґрунтовано застосування згорткових нейромережевих моделей, адаптованих до специфіки вхідних візуальних даних. Такий підхід забезпечує оптимізацію співвідношення між точністю класифікації та обчислювальною швидкодією, що має вагоме науково-практичне значення.

Метою роботи є створення програмного забезпечення для розпізнавання зображень фруктів на базі методів штучних нейронних мереж.

Для досягнення поставленої мети сформульовано такі задачі:

- проаналізувати поточний стан та специфіку предметної області;
- дослідити існуючі програмні аналоги та формалізувати постановку задачі;
- сформувати специфікацію функціональних вимог до системи;
- розробити логічну та фізичну архітектуру програмного комплексу;
- програмно реалізувати алгоритми класифікації зображень фруктів;
- виконати верифікацію та тестування розробленого програмного забезпечення.

Об’єктом розробки є інформаційна система для розпізнавання візуальних образів фруктів.

Предмет дослідження – алгоритмічні та програмні засоби класифікації зображень фруктів із застосуванням штучних нейронних мереж.

Теоретико-методологічною основою роботи є інструментарій галузі штучного інтелекту, зокрема методи глибинного навчання та архітектури згорткових нейронних мереж.

Практичне значення одержаних результатів полягає у створенні функціонального програмного продукту, здатного здійснювати автоматичну класифікацію фруктів на цифрових зображеннях. Впровадження розробленого рішення дозволяє мінімізувати часові витрати та суттєво підвищити точність ідентифікації об'єктів, що є перспективним для застосування на підприємствах агропромислового сектору, у сфері роздрібної торгівлі, а також для систем автоматизованої каталогізації візуального контенту.

РОЗДІЛ 1 АНАЛІЗ ПРОГРАМ ДЛЯ РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ ФРУКТІВ

1.1. Аналіз предметної області

Предметною областю кваліфікаційної роботи є автоматизована ідентифікація фруктів на основі аналізу їхніх цифрових зображень. Практична цінність цього завдання полягає у забезпеченні точної класифікації видів плодів. Визначення таксономічної приналежності є базою для отримання супутніх інформаційних даних, зокрема комерційної вартості товару, агротехнічних вимог до вирощування та оптимальних умов зберігання.

Процес мануальної класифікації об'єктів характеризується високою ресурсомісткістю та низькою оперативністю. За відсутності належного експертного досвіду суттєво зростає ймовірність хибної класифікації під час візуального розпізнавання. З огляду на це, актуальним є застосування спеціалізованих програмних систем для автоматизації даного процесу. Джерелом вхідних візуальних даних для таких систем переважно виступають оптичні сенсори мобільних пристроїв або стаціонарні камери відеоспостереження торговельних комплексів.

Водночас наявні програмні рішення, зокрема аналоги Leafsnap [1] та Plantnet [2], поширюються із закритим вихідним кодом, що суттєво обмежує їхнє застосування у навчальному процесі та наукових дослідженнях. Окрім того, існуючі застосунки відрізняються складністю експлуатації та розраховані на ідентифікацію лімітованої кількості видів фруктів.

Зважаючи на вищевикладене, у межах кваліфікаційної роботи обґрунтовано необхідність розробки власного програмного забезпечення для оперативного розпізнавання зображень фруктів, що відзначається ергономічністю користувацької взаємодії та не вимагає від оператора спеціалізованих знань.

1.2. Порівняльний аналіз програм для розпізнавання зображень фруктів

На сучасному етапі розвитку інформаційних технологій існує широкий спектр програмних засобів для ідентифікації фруктів за їхніми цифровими зображеннями. Одним із відомих рішень є система «Leafsnap» [1], створена науковими установами США з метою візуального розпізнавання рослин та їхніх структурних елементів. Зазначений програмний продукт поширюється на безоплатній основі та розроблений для функціонування на мобільних пристроях з операційною системою iOS.

Однак цільовим призначенням даної розробки є насамперед загальна класифікація флори, поширеної на території США та Канади, а не вузькоспеціалізоване розпізнавання виключно плодових культур. Додатковим обмежувальним фактором, що звужує сферу практичного використання системи, є відсутність багатомовної підтримки, оскільки користувацький інтерфейс застосунку реалізовано виключно англійською мовою.

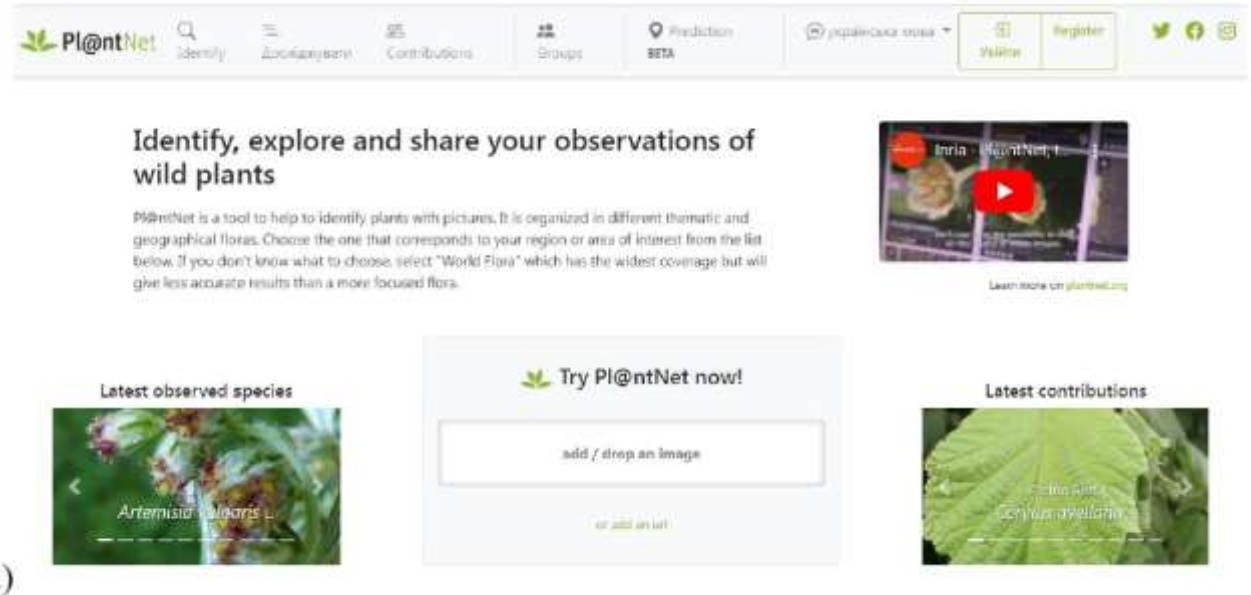


Рисунок 1.1 – Фрагмент головної сторінки офіційного вебресурсу програмної системи «Leafsnap» [1]

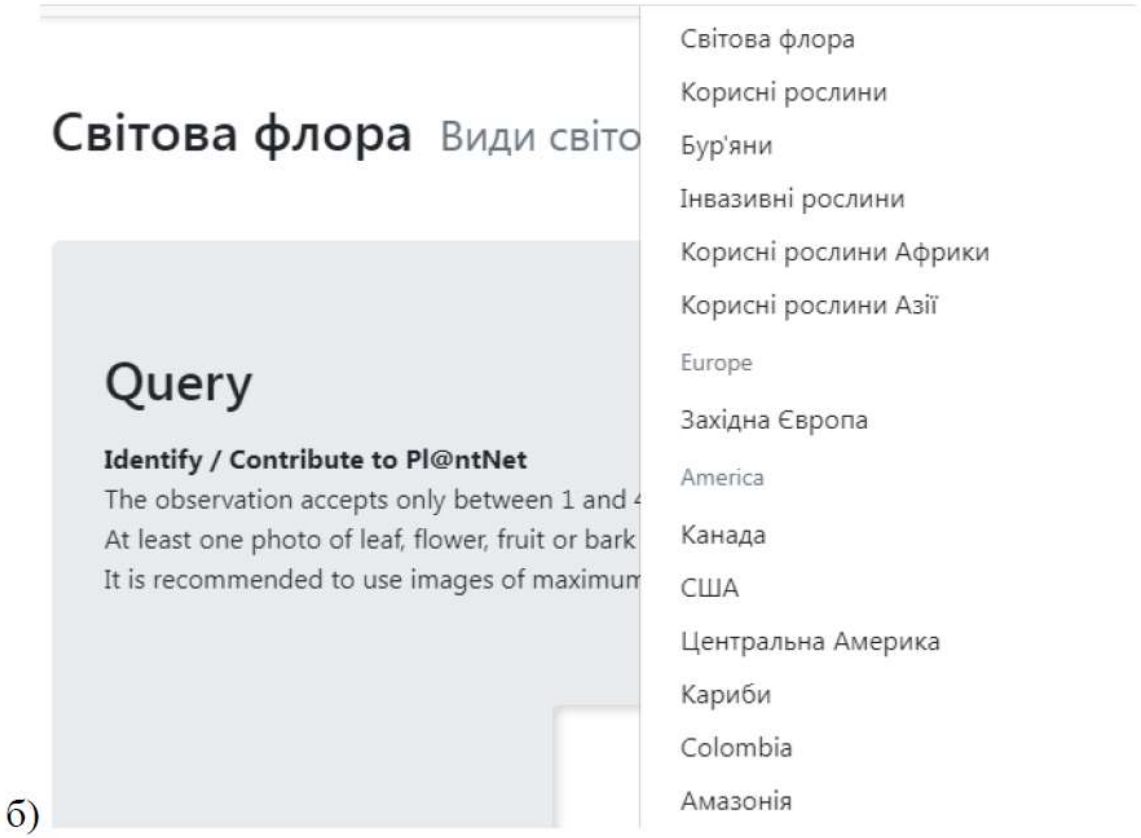
Таблиця 1.1 – Порівняльний аналіз програмних засобів для розпізнавання зображень

№ п/п	Назва програми	Апаратна платформа	Операційна система	Мова інтерфейсу	Точність розпізнавання	Умови використання
1	Leafsnap	iPhone, iPad	iOS	англійська	Висока	Безкоштовна
2	Plantnet	smartphone, PC	iOS, Android	англійська, українська, …	Висока	Безкоштовна
3	Google Lens	smartphone, Apple	iOS, Android	Англійська, українська,..	Висока	Безкоштовна

Програмний комплекс «Plantnet» [2], інтерфейс якого проілюстровано на рисунку 1.2, призначений для розпізнавання об'єктів рослинного світу, включно з плодовими та овочевими культурами, на основі аналізу їхніх фотографій. Отримання вхідних візуальних даних здійснюється шляхом безпосереднього захоплення зображень із відеокамери. Зазначена система забезпечує ідентифікацію видової приналежності рослини, а також надає супутню довідкову інформацію щодо її біологічних характеристик та необхідних умов агротехнічного догляду.



а)



б)

Рисунок 1.2 – Програма «Plantnet»: а) фрагмент головної сторінки сайту програми; б) вибір регіону поширення рослини [2]

Перевагою застосунку є наявність україномовної локалізації користувацького інтерфейсу. Водночас експлуатація даного програмного забезпечення вимагає обов'язкової попередньої реєстрації користувача, а сам продукт поширюється із закритим вихідним кодом, що відображено в результатах порівняльного аналізу в таблиці 1.1.

Програмна система Google Lens [3] базується на технологіях машинного розпізнавання зображень, розроблених корпорацією Google. Даний інструмент забезпечує ідентифікацію об'єктів шляхом глибокого візуального аналізу, що ґрунтується на апараті штучних нейронних мереж. Вперше інформацію про створення цього продукту було оприлюднено у 2017 році. На початкових етапах життєвого циклу рішення позиціонувалося як автономний застосунок, однак згодом його було інтегровано до базового програмного забезпечення камер мобільних пристроїв під управлінням операційної системи Android.

У процесі фокусування оптичного сенсора на фізичному об'єкті система ініціює алгоритм автоматичної ідентифікації, що дозволяє, зокрема, здійснювати класифікацію фруктів. Слід зауважити, що платформа має універсальне призначення для аналізу широкого спектра візуальних образів і не володіє вузькою спеціалізацією виключно щодо плодових культур. Суттєвою перевагою системи є розширена мультимовна підтримка користувацького інтерфейсу, яка синхронізована з лінгвістичною базою сервісу перекладу від Google та містить повноцінну українську локалізацію.

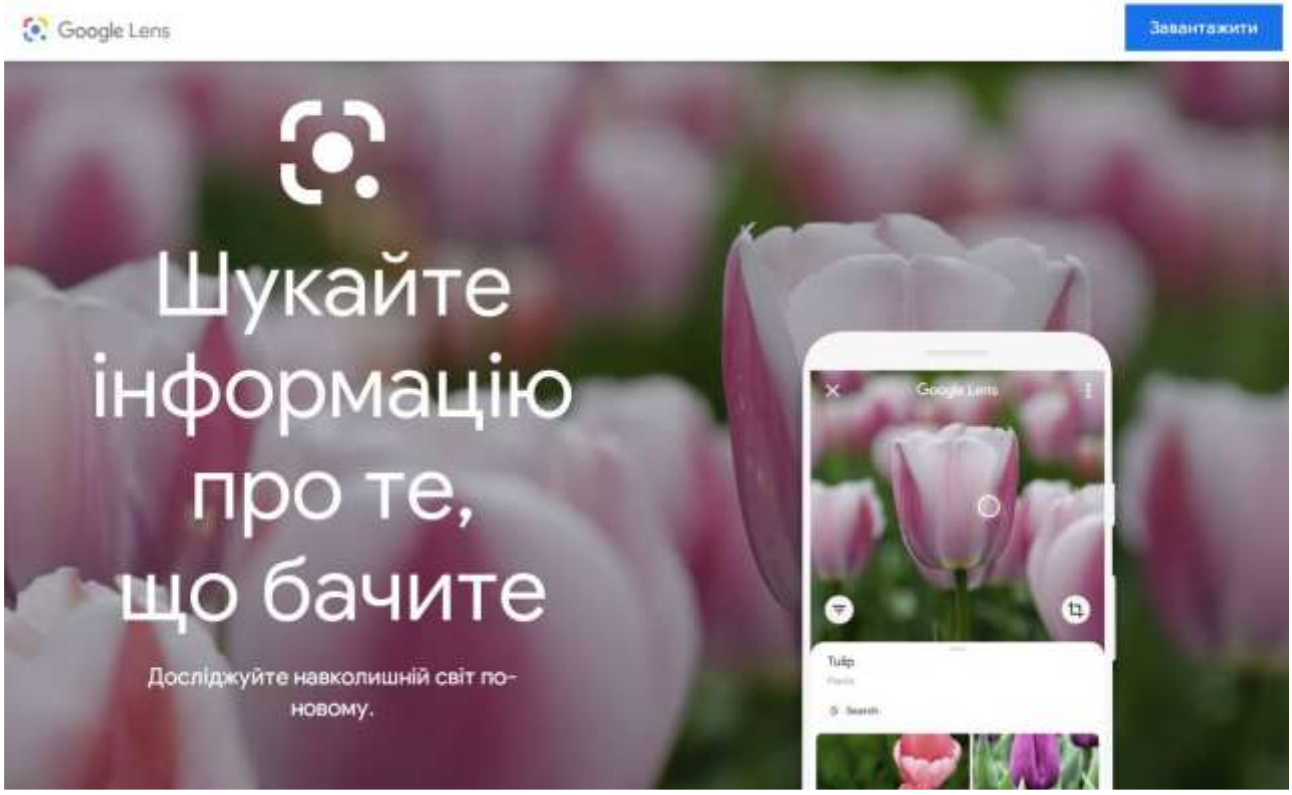


Рисунок 1.3 – Фрагмент головної сторінки офіційного веб-ресурсу програмної системи «Google Lens» [3]

Підсумовуючи результати порівняльного аналізу, наведені у таблиці 1.1, варто зазначити, що розглянуті програмні системи розпізнавання візуальних даних є пропрієтарними рішеннями із закритим вихідним кодом. Водночас вони характеризуються широким спектром застосування і не мають вузької спеціалізації щодо ідентифікації виключно плодових культур. Процес класифікації об'єктів у зазначених комплексах переважно реалізується на базі математичного апарату штучних нейронних мереж, через що ефективність та

межі практичного використання таких інтелектуальних систем суворо детермінуються репрезентативністю та обсягами залучених тренувальних наборів даних.

Окрім того, більшість наявних продуктів архітектурно адаптовано для функціонування в середовищі мобільних операційних систем iOS та Android, тоді як існує об'єктивна необхідність у наявності десктопного програмного забезпечення для роботи на платформі MS Windows. Додатковим обмежувальним фактором для низки досліджених застосунків є відсутність україномовної локалізації користувацького інтерфейсу.

Зважаючи на виявлені функціональні та технічні обмеження існуючих аналогів, виникає обґрунтована науково-практична потреба у проектуванні та розробці власного спеціалізованого програмного забезпечення для автоматизованого розпізнавання цифрових зображень фруктів.

1.3. Постановка задачі

Спираючись на результати дослідження програмних аналогів, наведені у таблиці 1.1, здійснено наукове обґрунтування вибору об'єкта та предмета дослідження. Об'єктом розробки визначено інформаційну систему автоматизованої ідентифікації візуальних образів фруктів. Предметом дослідження виступає алгоритмічне та програмне забезпечення процесу класифікації цифрових зображень фруктів на базі математичного апарату штучних нейронних мереж.

Головна мета кваліфікаційної роботи полягає у проектуванні та практичній реалізації програмного забезпечення для розпізнавання зображень фруктів із застосуванням технологій нейромережевого моделювання.

Відповідно до поставленої мети сформульовано наступний комплекс дослідницьких та практичних завдань:

- здійснити системний аналіз предметної області щодо специфіки автоматизованої ідентифікації плодових культур за їхніми цифровими поданнями;

- критично оцінити функціональні можливості існуючих інформаційних систем візуального розпізнавання, серед яких програмні рішення Leafsnap, Plantnet та Google Lens, із подальшою формалізацією постановки задачі на розробку спеціалізованого нейромережевого застосунку для класифікації фруктів;
- сформувати специфікацію функціональних вимог до проектованої системи, що передбачає реалізацію модулів захоплення відеопотоку та імпорту графічних файлів, попередньої розмірної трансформації візуальних даних, а також імплементацію алгоритмів тренування та валідації згорткової нейронної мережі на відповідних вибірках із подальшою експлуатацією навченої моделі для розпізнавання нових об'єктів;
- розробити логічну та фізичну архітектуру програмно-апаратного комплексу, що включає декомпозицію логічної структури на підсистеми машинного навчання та безпосередньої класифікації образів, а також інтеграцію апаратних компонентів у вигляді пристрою відеозахоплення та центрального обчислювального вузла;
- здійснити програмну імплементацію розроблених алгоритмів мовою програмування Python із залученням обчислювальних потужностей хмарного середовища Google Colaboratory;
- виконати комплексну верифікацію та тестування створеної інформаційної системи з використанням ізольованого тестового набору візуальних даних.

Окреслене коло завдань формує методологічний та практичний базис для виконання наступних етапів проектування додатку.

РОЗДІЛ 2 ПРОЕКТУВАННЯ ДОДАТКУ РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ ФРУКТІВ

2.1. Обґрунтування вибору засобів для вирішення поставлених задач

Задачу класифікації та ідентифікації візуальних об'єктів найбільш доцільно вирішувати із застосуванням архітектури згорткових нейронних мереж. Відповідно, інструментарієм штучного інтелекту для автоматизованого аналізу цифрових зображень фруктів у даному дослідженні обрано саме цей клас математичних моделей. Процес їх експлуатації передбачає попередній етап індуктивного навчання на спеціалізованих масивах даних із подальшим виконанням безпосереднього розпізнавання нових візуальних образів.

Згорткові нейронні мережі є фундаментальною складовою технологій глибинного навчання, які базуються на принципах багатошарової обробки сигналів із використанням значних обсягів навчальних прикладів [9-13]. Базову архітектуру сучасних згорткових нейромереж було розроблено Яном Лекуном у 1988 році. Зазначений клас моделей спеціально оптимізовано для аналізу візуальної інформації, оскільки їхня структурна організація імітує нейрофізіологічні механізми зорової системи живих організмів. Головною математичною операцією під час обробки просторових даних у таких системах виступає двовимірна згортка, що й зумовило відповідну назву цього класу штучних нейронних мереж.

Стрімке поширення даного підходу відбулося у 2012 році після успішної участі згорткової архітектури у змаганні ImageNet, де вдалося досягти безпрецедентного зниження частки хибних розпізнавань із двадцяти шести до п'ятнадцяти відсотків. На сучасному етапі згорткові мережі глибоко інтегровані в інфраструктуру провідних технологічних корпорацій. Зокрема, розробки компанії Facebook застосовують їх для автоматизованої генерації метаданих контенту, сервіси Google – для контекстного пошуку графічної інформації, а платформи Amazon – у рекомендаційних алгоритмах електронної комерції. Тим не менш, домінуючим напрямом їхньої експлуатації залишається задача багатокласової ідентифікації різноманітних об'єктів на цифрових зображеннях,

включаючи біометричний аналіз, розпізнавання сільськогосподарських культур, оптичне розпізнавання символів та детектування транспортних засобів.

З точки зору топологічної організації, архітектура згорткових штучних нейронних мереж формується шляхом чергування двох основних типів обчислювальних блоків: шарів згортки та шарів субдискретизації, які також ідентифікуються як шари підвибірки, що схематично відображено на рисунку 2.1.

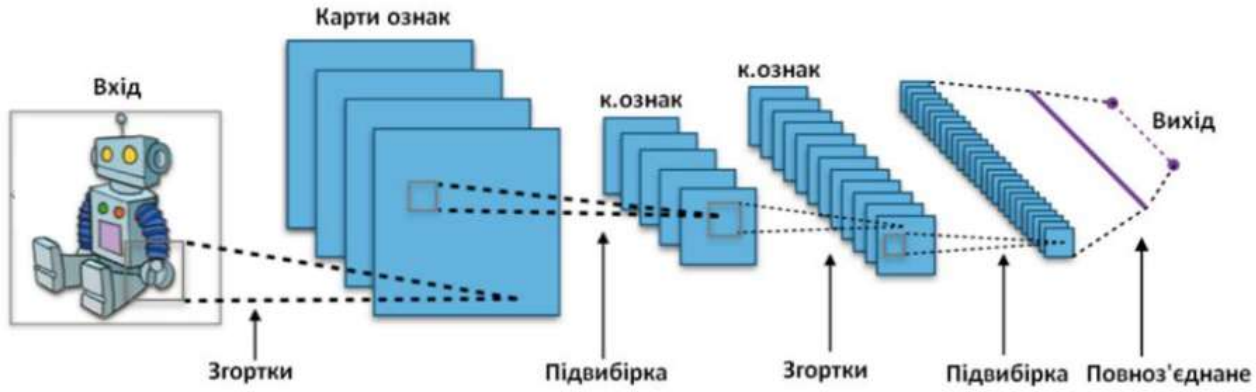


Рисунок 2.1 – Схематичне подання архітектури згорткової нейронної мережі

У структурній організації згорткових нейронних мереж відбувається циклічне чергування шарів згортки та субдискретизації. Архітектура таких моделей є принципово багатошаровою та односпрямованою. Оптимізація параметрів мережі переважно здійснюється за допомогою алгоритму зворотного поширення помилки, механізм якого базується на обчисленні розбіжностей між фактичними та еталонними значеннями на вихідному шарі для подальшого коригування синаптичних ваг.

Функціональне призначення згорткових шарів полягає у виконанні математичної операції згортки вхідних сигналів відповідного рівня з матрицею ядра фільтра. Специфічною рисою даної архітектури є обов'язкова наявність згорткового шару як первинного етапу обробки даних. Вхідними даними для нейромережевої моделі виступають цифрові зображення у монохромному або поліхромному форматах.

Джерелом отримання візуальної інформації слугують растрові графічні файли або пристрої прямого відеозахоплення. Під час програмної обробки

монохромне цифрове зображення формалізується у вигляді прямокутної матриці $f = \big(f(i,k)\big)$:

$$f = \begin{bmatrix} f(0,0) & \dots & f(0,k) & \dots & f(0,N-1) \\ \dots & \dots & \dots & \dots & \dots \\ f(i,0) & \dots & f(i,k) & \dots & f(i,N-1) \\ \dots & \dots & \dots & \dots & \dots \\ f(M-1,0) & \dots & f(M-1,k) & \dots & f(M-1,N-1) \end{bmatrix}, \quad (2.1)$$

де $f(i, k)$ – елемент матриці, що відповідає значенню інтенсивності пікселя із просторовими координатами iта k;

i – індекс рядка;

k – індекс стовпця;

$i = 0, ..., M-1;$

$k = 0, ..., N-1$;

M – вертикальна роздільна здатність зображення у пікселях;

N – горизонтальна роздільна здатність зображення у пікселях.

У контексті автоматизованої ідентифікації фруктів колористичні характеристики об'єктів формують критично важливий простір ознак. З огляду на це, визначальною умовою забезпечення високої точності класифікації є обробка саме повноколірних цифрових зображень, що проілюстровано на рисунку 2.2.

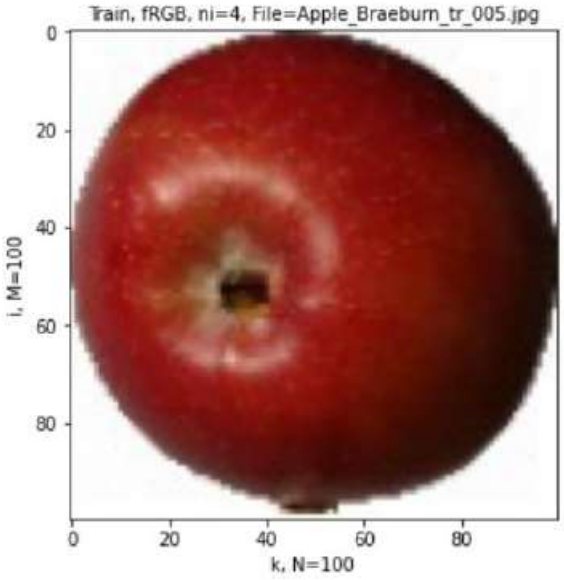


Рисунок 2.2 – Цифрове кольорове зображення в системі просторових координат i та k

Під час комп'ютерної обробки поліхромних зображень у колірному просторі RGB для кожної спектральної компоненти, а саме червоної, зеленої та синьої, ініціалізується окрема двовимірна матриця просторовою розмірністю $M \times N$елементів. Абсолютні значення інтенсивності пікселів переважно підлягають нормалізації у діапазоні дійсних чисел від 0 до 1.

Математична операція згортки в архітектурі згорткових нейронних мереж виконується у просторовій області над масивом вхідних даних f_1 розмірністю $M_1 \times N_1$елементів із залученням матриці ваг ядра фільтра wрозмірністю $M_w \times N_w$ елементів. Здебільшого на практиці застосовуються ядра фільтрів квадратної форми, для яких справджується рівність $M_w = N_w$. Наслідком виконання згорткового перетворення є обчислення вихідної матриці ознак g_1 просторовою розмірністю $M_2 \times N_2$ елементів відповідно до такого аналітичного виразу:

$$g_1(i,k) = \sum_{m=0}^{Mw-1} \sum_{n=0}^{Nw-1} f_1(i-m+M_{w2}, k-n+N_{w2}) \cdot w(m,n), \qquad (2.2)$$

де $i = 0, \ldots, M_1 - 1$, $k = 0, \ldots, N_1 - 1$;

$M_w = 2M_{w2} + 1$ – вертикальний розмір ядра фільтра;

$N_w = 2N_{w2} + 1$ – горизонтальний розмір ядра фільтра;

M_{w2} та N_{w2} – цілочисельні значення половини відповідних просторових вимірів ядра фільтра.

У скороченому вигляді математична операція згортки масиву f_1 із ядром w формалізується наступним чином:

$$g_1 = f_1 * w. \qquad (2.3)$$

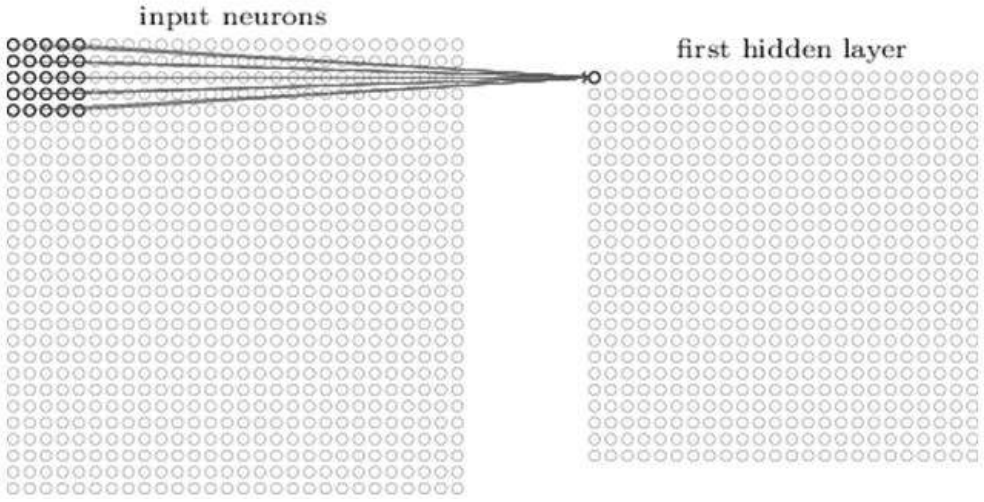
Отже, формування окремого елемента результуючої матриці g_1 здійснюється на основі $M_w \times N_w$ компонентів первинного масиву f_1, які перекриваються ядром фільтра. У теорії згорткових нейронних мереж відповідна локальна ділянка вхідного зображення термінологічно визначається як рецептивне поле або поле сприйняття. Матриця w, котра параметризує ядро

фільтра, також ідентифікується як матриця вагових коефіцієнтів. При цьому кожний згортковий шар нейромережевої моделі оперує власним унікальним набором синаптичних ваг.

Принциповою особливістю виконання просторової згортки є виникнення крайових ефектів уздовж периметра оброблюваного зображення. З метою нівелювання зазначених просторових артефактів результуюча матриця g_1 формується зі зменшеною розмірністю, а саме на $2M_{w2}$ елементів меншою за вертикаллю та на $2N_{w2}$ елементів меншою за горизонталлю відносно початкових габаритів масиву f_1. З огляду на це, просторові параметри матриці після згорткового перетворення становлять:

$$M_C = M_1 - 2 \cdot M_{w2}, \; N_C = N_1 - 2 \cdot N_{w2}. \tag{2.4}$$

Зменшення просторової розмірності вихідних матриць у процесі згортки забезпечує просторову локалізацію ядра фільтра виключно в межах визначених кордонів первинного зображення. Такий математичний підхід унеможливлює виникнення крайових артефактів та структурних спотворень візуальних даних, що наочно продемонстровано на рисунку 2.3.



а) б)

Рисунок 2.3 – Візуалізація математичного перетворення згортки: а – вхідний масив даних f_1розмірністю 28×28елементів, що обробляється ядром фільтра розміром 5×5елементів; б – сформована вихідна матриця g_1, яка виступає картою ознак просторовою розмірністю 24×24елементи

Математична операція згортки візуальних даних відповідно до виразу 2.2 застосовується в архітектурі згорткових нейронних мереж з метою локалізації та екстракції специфічних характеристик досліджуваного об'єкта. Зокрема, дане перетворення довело свою обчислювальну ефективність під час ідентифікації базових геометричних примітивів, серед яких прямолінійні відрізки, кола, трикутники та квадрати.

Зважаючи на високу топологічну та структурну складність реальних оптичних образів, для їхнього достовірного детектування засобами згорткових нейромережевих моделей залучається широка номенклатура спеціалізованих ядер фільтрів, що наочно проілюстровано на рисунку 2.4.

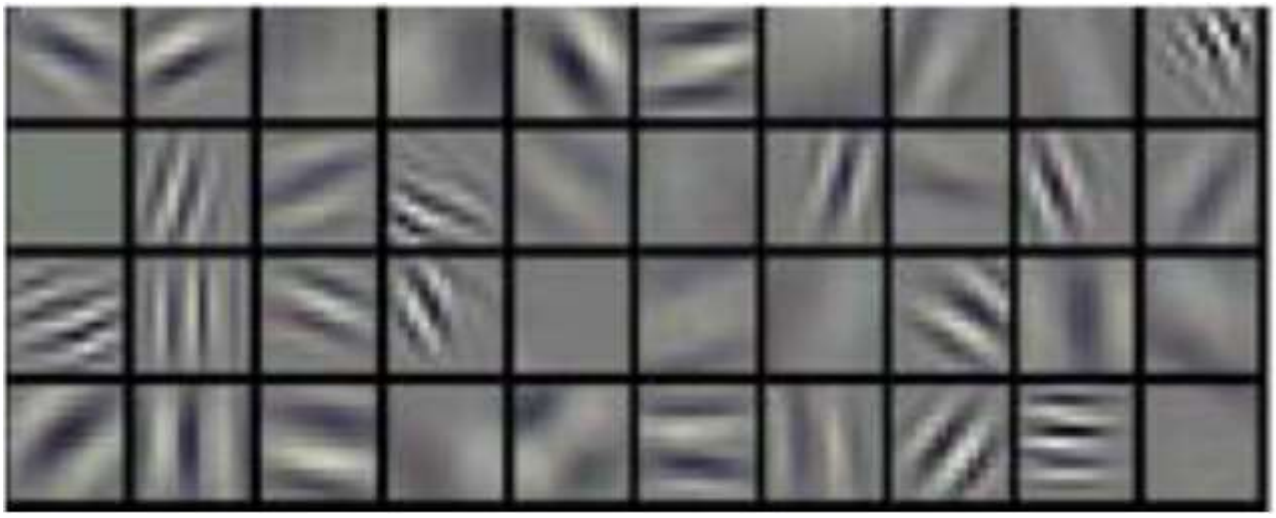


Рисунок 2.4 – Зразки ядер фільтрів w, що застосовуються в архітектурі згорткових нейронних мереж

У класичних моделях згорткових нейромереж відбувається багатократне дублювання згорткових шарів. Відповідний алгоритм послідовної обробки сигналів ілюструється схемою на рисунку 2.5. У запропонованій структурній декомпозиції вхідний візуальний масив позначається терміном Input, згорткові обчислювальні блоки – Conv, активаційні функції нейронів на базі лінійної ректифікації – ReLU, шари субдискретизації або підвибірки – Pool, а повнозв'язний шар – Fully Connected.

У межах даної архітектури активаційна функція ReLU реалізується у вигляді вузла лінійної ректифікації, який здійснює нелінійне математичне перетворення $\max(x, 0)$ над вхідним сигналом x. Застосування активаційних функцій гарантує масштабування та утримання вихідних сигналів штучних нейронів у межах припустимих діапазонів значень. З огляду на те, що обчислення функції лінійної ректифікації відрізняється мінімальною алгоритмічною складністю, її інтеграція сприяє суттєвому підвищенню загальної швидкодії процесу індуктивного навчання згорткової нейронної мережі.



Рисунок 2.5 – Послідовність обробки сигналів в архітектурі згорткової нейронної мережі

Процес ідентифікації візуальних образів фруктів із застосуванням архітектури згорткових нейронних мереж реалізується за визначеним ієрархічним алгоритмом. На початковому етапі перший згортковий шар здійснює детектування базових низькорівневих ознак, серед яких фрагменти контурів різної просторової орієнтації, а також локальні ділянки з відмінними колористичними характеристиками та градієнтами яскравості, що визначають границі плоду. Подальша обробка на наступних згорткових шарах забезпечує виділення високорівневих комплексних ознак, які формуються шляхом просторової композиції раніше виявлених базових елементів, утворюючи загальні абриси досліджуваного об'єкта. Зі збільшенням глибини нейромережевої моделі закономірно зростає семантична складність розпізнаваних патернів. Відповідно, на глибоких етапах обробки, зокрема на передостанніх шарах архітектури, стає можливим цілісне розпізнавання фрукта як єдиного об'єкта.

Функціональне призначення шарів субдискретизації у структурі згорткових нейромереж полягає у цілеспрямованому зменшенні просторової

розмірності масивів даних, що здебільшого супроводжується двократним стисненням сигналів за вертикаллю та горизонталлю. Алгоритмічно шар субдискретизації переважно реалізує операцію заміщення локальної прямокутної ділянки карти ознак, сформованої на попередньому згортковому шарі, її максимальним значенням. Зазначений механізм обробки наочно проілюстровано на рисунку 2.6. Інтеграція шарів субдискретизації дозволяє не лише мінімізувати обчислювальну складність та скоротити часові витрати на обробку зображень, але й суттєво підвищити загальну інваріантність нейромережевої моделі щодо просторових трансформацій, зокрема до зміни масштабу та трансляційного зсуву об'єктів у полі зору.

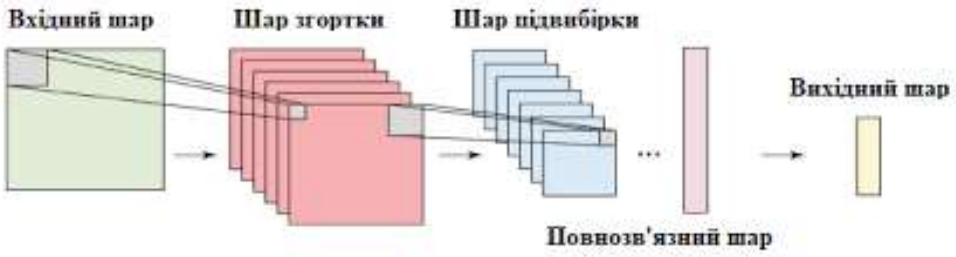


Рисунок 2.6 – Принцип функціонування шару субдискретизації згорткової нейронної мережі, заснований на екстракції максимального значення з локальних областей розмірністю 2×2 елементи

Завершальним структурним компонентом згорткової нейронної мережі виступає повнозв'язний шар, вихідні сигнали якого відповідають числовим ідентифікаторам класів розпізнаваних образів, тобто ідентифікаторам конкретних видів фруктів. На основі отриманого числового коду здійснюється детермінація точної номенклатурної назви певного плоду. Оскільки значення вихідних сигналів нейромережевої моделі підлягають процедурі математичної нормалізації в інтервалі від 0 до 1, ці показники інтерпретуються як імовірнісна міра належності аналізованого вхідного зображення до відповідної класифікаційної категорії. Як ілюстрація, формування вихідного вектора значень [0, 0.2, 0.7, 0.1] свідчить про те, що досліджуване зображення з

імовірністю 20% класифікується як об'єкт другого класу, а з імовірністю 70% належить до третього класу.

Загальну топологічну організацію згорткової нейронної мережі, орієнтованої на вирішення задач візуальної ідентифікації, наведено на рисунку 2.7. До класичних та найбільш поширених еталонних архітектур даного типу математичних моделей належать, зокрема, конфігурації LeNet-5 та AlexNet, структурну специфіку яких відображено на рисунку 2.8 [11, 13].

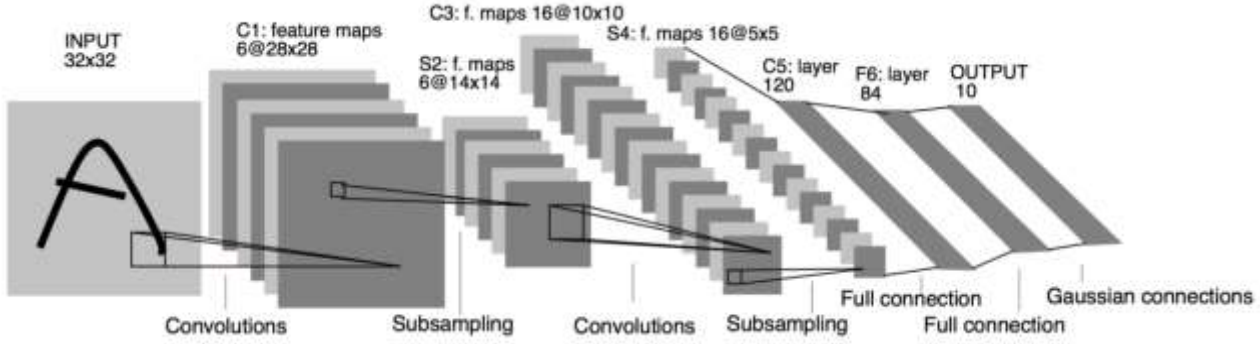


Рисунок 2.7 – Структурна організація згорткової нейронної мережі, орієнтованої на вирішення задач розпізнавання візуальних образів

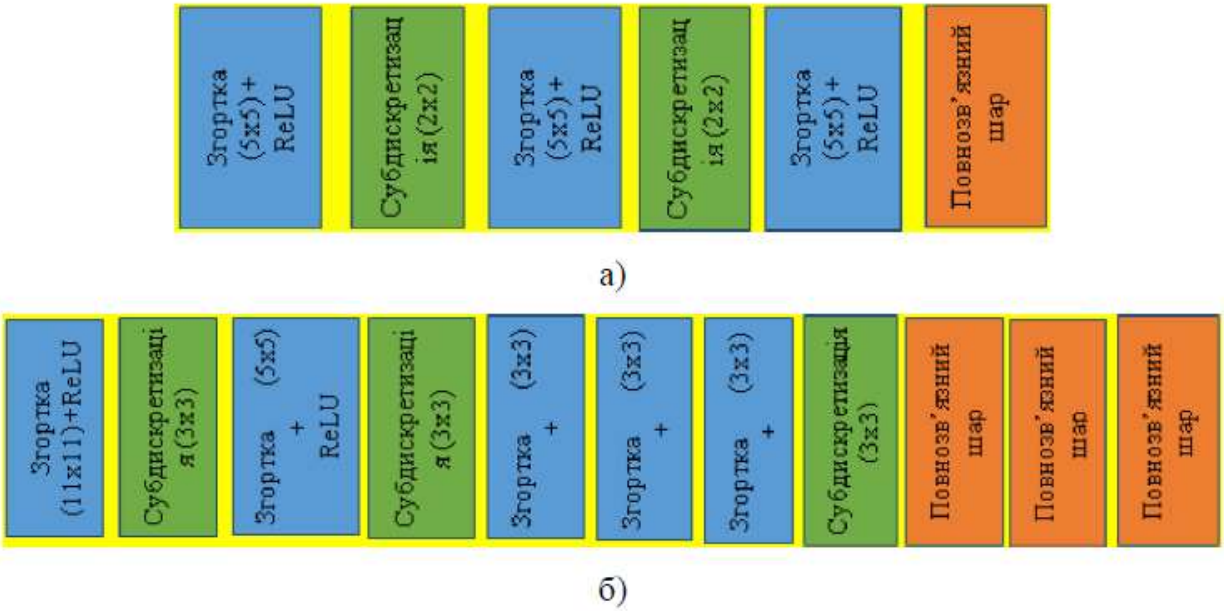


Рисунок 2.8 – Базові архітектури згорткових нейронних мереж: а – LeNet-5; б – AlexNet [11]

Архітектура LeNet-5 є класичним різновидом згорткових нейромереж, розробленим Яном Лекуном. Вона довела свою обчислювальну ефективність під час вирішення задач класифікації візуальних образів різної природи, зокрема при розпізнаванні рукописних символів.

Відмінними характеристиками конфігурації LeNet-5 є наступні:

- циклічне чергування згорткових шарів та шарів субдискретизації;
- обмеження значень вихідних сигналів після операції згортки за допомогою функцій активації, зокрема алгоритму лінійної ректифікації;
- застосування математичного апарату згортки з метою екстракції просторових ознак аналізованих об'єктів;
- використання матриці вагових коефіцієнтів ядра фільтра просторовою розмірністю 5×5 елементів;
- наявність повнозв'язного шару як термінального елемента архітектури.

Первинна версія досліджуваної топології містила три згорткові шари та була оптимізована для обробки вхідних матриць розміром 28×28 пікселів, що відповідає стандарту еталонного набору даних рукописних цифр.

Модель AlexNet, створена колективом дослідників на чолі з Алексом Крижевським, Іллею Суцкевером та Джеффрі Хінтоном, концептуально наслідує базові принципи LeNet-5, проте відрізняється значно вищим рівнем структурної складності. Це виражається у масштабуванні кількості згорткових та повнозв'язних обчислювальних блоків. Специфічною рисою даної архітектури є безпосереднє каскадне сполучення кількох згорткових шарів поспіль без проміжних етапів субдискретизації.

Застосування механізму просторової згортки суттєво мінімізує загальну кількість параметрів, що підлягають оптимізації у процесі машинного навчання. Це досягається завдяки принципу розподілу ваг, згідно з яким ідентичне ядро фільтра застосовується до всієї площі оброблюваного масиву даних у межах одного шару.

Інформаційний масив даних для тренування нейромережевої моделі структурно розділяється на три функціональні множини:

1. Тренувальна вибірка – призначена для ітеративного коригування синаптичних ваг системи.
2. Валідаційна вибірка – застосовується для моніторингу процесу з метою уникнення ефекту перенавчання моделі.
3. Екзаменаційна вибірка – використовується для об'єктивного оцінювання здатності навченої мережі до генералізації на нових даних.

Алгоритм вирішення задачі класифікації із залученням нейромережевих технологій передбачає послідовне виконання таких етапів:

1. Вибір базового класу математичної моделі, зокрема згорткової архітектури.
2. Проектування структурної топології системи, що включає детермінацію кількості обчислювальних рівнів, числа нейронів у кожному вузлі та конфігурації міжнейронних зв'язків.
3. Обґрунтування методу оптимізації, зокрема алгоритму зворотного поширення помилки, із відповідним налаштуванням гіперпараметрів процесу навчання.

Програмну імплементацію досліджуваної моделі та розробку цільового застосунку для ідентифікації фруктів доцільно здійснювати мовою програмування Python [14] на базі хмарних обчислювальних потужностей середовища Google Colaboratory [15] через інтерактивний веб-інтерфейс Jupyter Notebook. До ключових переваг обраного інструментарію належать відкритий доступ, відсутність необхідності локального розгортання робочого середовища, а також можливість інтеграції спеціалізованих фреймворків машинного навчання TensorFlow та Keras. Окрім того, дане середовище забезпечує підтримку широкого спектра допоміжних математичних та графічних пакетів, серед яких NumPy, Matplotlib та SciPy, детальний перелік яких наведено у таблиці 2.1.

Таблиця 2.1 – Специфікація програмних бібліотек мови Python, залучених до розробки застосунку

№	Назва	Призначення
1	Keras	Нейромережевий програмний інтерфейс, що функціонує як високорівнева надбудова над обчислювальною платформою Tensorflow;
2	MatPlotLib	Інструментальний пакет для генерації графіків та здійснення комплексної візуалізації інформаційних масивів;
3	NumPy	Програмна бібліотека, оптимізована для виконання складних математичних обчислень;
4	OpenCV	Спеціалізоване програмне забезпечення для вирішення прикладних задач комп'ютерного зору;
5	Pandas	Програмний інструментарій для структурування, маніпулювання та аналітичної обробки фреймів даних;
6	Scipy	Програмні модулі для здійснення цифрової обробки сигналів та комп'ютерного аналізу зображень;
7	Tensorflow	Базова програмна інфраструктура з відкритим вихідним кодом, орієнтована на проектування та тренування моделей машинного навчання.

2.2. Проектування структури та поведінки системи

Архітектура інформаційної системи автоматизованої ідентифікації фруктів на базі нейромережевих технологій, окрім безпосередньо базової згорткової моделі, структурно включає спеціалізований модуль тренування штучної нейронної мережі та функціональний блок класифікації візуальних образів, що схематично проілюстровано на рисунку 2.9.

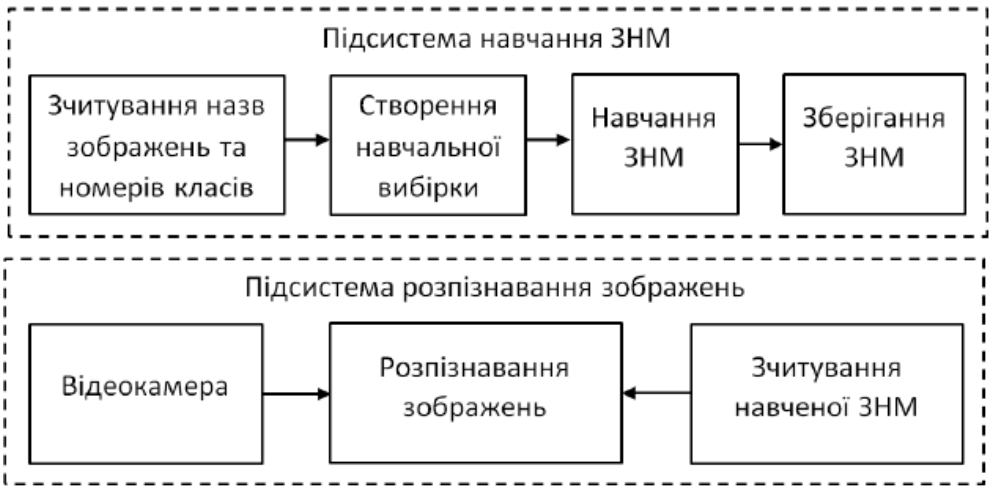


Рисунок 2.9 – Структурна організація інформаційної системи розпізнавання зображень фруктів

Модуль тренування згорткової нейронної мережі охоплює функціональні блоки імпорту ідентифікаторів графічних файлів із відповідними номенклатурними індексами класів, формування тренувального набору даних, що складається з масивів вхідних та еталонних вихідних векторів, а також алгоритмічний блок оптимізації вагових коефіцієнтів методом зворотного поширення помилки з подальшим експортом параметрів навченої моделі до зовнішнього сховища.

У межах підсистеми класифікації візуальних образів конфігурація попередньо навченої моделі ініціалізується у відповідному блоці завантаження. Наступним етапом є безпосередня класифікація поліхромного відеопотоку, отриманого з пристрою захоплення зображень, яка виконується модулем розпізнавання.

Концептуальне моделювання інформаційної системи ідентифікації плодів формалізує ключові функціональні можливості, доступні кінцевому користувачеві. За допомогою діаграми прецедентів уніфікованої мови моделювання [16-17] розроблену концептуальну модель проілюстровано на рисунку 2.10. Дана архітектура визначає такі базові прецеденти:

- Імпорт ідентифікаторів та індексів класів – автоматизоване завантаження з текстового документа назв графічних файлів із закріпленими за ними класифікаційними номерами.
- Завантаження зображень із файлової системи – імпорт візуальних даних із растрових файлів.
- Отримання відеопотоку з камери – захоплення цифрового зображення безпосередньо з оптичного сенсора.
- Розмірна нормалізація зображень – уніфікація просторової роздільної здатності всіх вхідних графічних даних відповідно до архітектурних вимог рецептивного поля нейронної мережі.

- Формування тренувального набору – генерація вибірки, де вхідні вектори містять числові еквіваленти колірних компонентів зображень, а вихідні вектори репрезентують відповідні ідентифікатори класів.
- Тренування нейромережевої моделі – оптимізація синаптичних ваг згорткової мережі за алгоритмом зворотного поширення помилки.
- Збереження параметрів моделі – запис конфігурації та вагових коефіцієнтів навченої штучної нейронної мережі у вигляді файлів.
- Ініціалізація параметрів мережі – завантаження конфігурації та вагових коефіцієнтів попередньо навченої згорткової нейромережевої моделі з файлової системи.
- Класифікація візуального образу – ідентифікація об'єкта на цифровому зображенні із подальшим встановленням його числового індексу та відповідної номенклатурної назви класу.

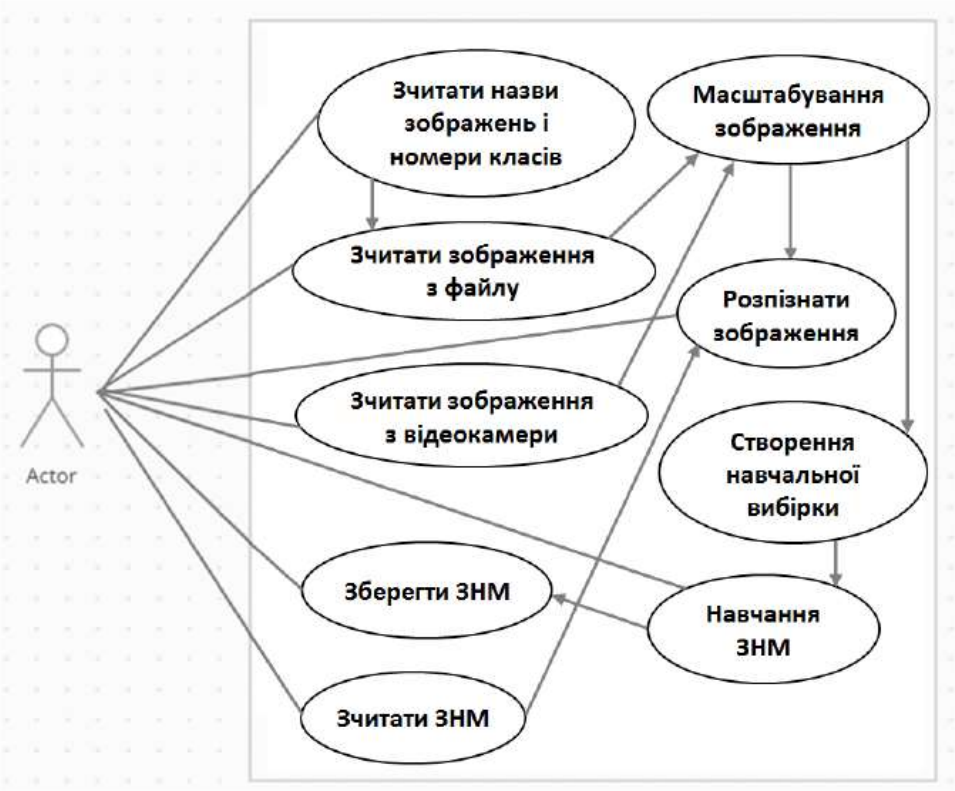


Рисунок 2.10 – Концептуальна модель інформаційної системи розпізнавання візуальних образів фруктів

2.3. Алгоритм навчання згорткової нейронної мережі

Спираючись на результати аналізу архітектурних особливостей та механізмів функціонування згорткових нейронних мереж, викладені у другому розділі, було спроєктовано алгоритм індуктивного навчання розроблюваної моделі, структурно-логічну схему якого проілюстровано на рисунку 2.11.

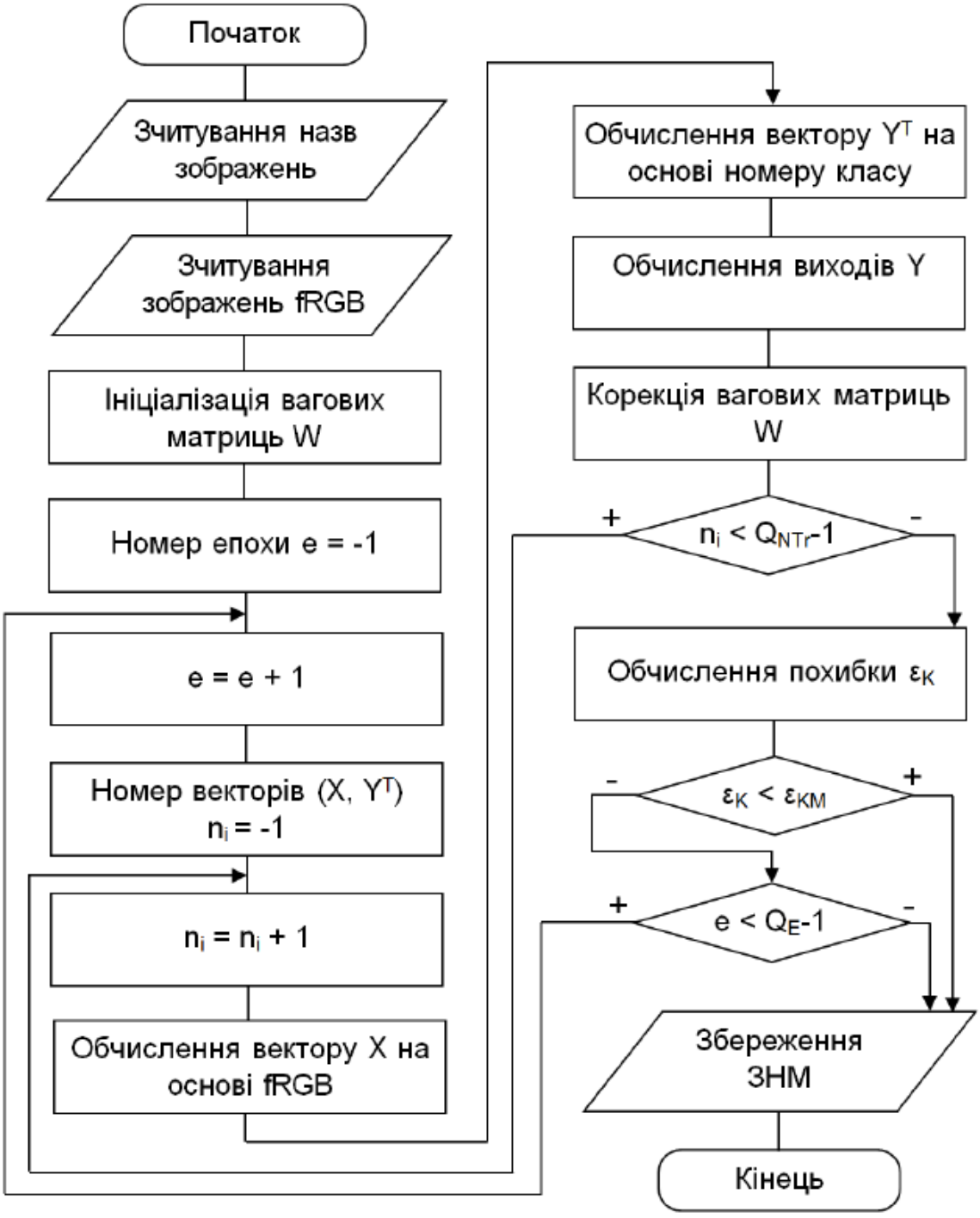


Рисунок 2.11 – Структурно-алгоритмічна схема процесу індуктивного навчання згорткової нейронної мережі

На початковому етапі алгоритму здійснюється імпорт ідентифікаторів зображень, що формують тренувальну вибірку, разом із відповідними індексами їхніх класифікаційних категорій. Після цього з растрових графічних файлів завантажуються поліхромні масиви візуальних даних навчальної множини f_{RGB}.

Відбувається первинна ініціалізація матриць вагових коефіцієнтів W для всіх обчислювальних шарів нейромережевої моделі шляхом присвоєння їм псевдовипадкових числових значень. Наступним кроком ініціюється ітеративний процес машинного навчання, де кожний цикл оптимізації позначається індексом епохи e.

У межах кожної епохи опрацьовується повний масив зображень тренувальної вибірки з індексами n_i, загальний обсяг якої становить Q_{NTr} елементів. Базуючись на масивах візуальних даних f_{RGB}, що попередньо пройшли процедуру просторової нормалізації до єдиного масштабу, формуються вхідні вектори X для подачі на рецептивне поле нейромережі.

Відповідно до класифікаційного індексу конкретного зображення генерується еталонний вектор цільового вихідного сигналу моделі Y_T. У результаті прямого поширення вхідного вектора X обчислюються фактичні вихідні значення мережі Y.

Шляхом обчислення розбіжності між еталонним Y_T та фактичним Y векторами визначається поточна похибка навчання моделі ϵ_K, зокрема із застосуванням метрики середньоквадратичного відхилення, після чого реалізується алгоритмічна корекція вагових коефіцієнтів W.

Процес оптимізації параметрів системи детерміновано завершується у разі досягнення похибкою ϵ_K значень, нижчих за встановлений допустимий поріг, або при вичерпанні ліміту максимальної кількості епох Q_E. У разі залучення валідаційної вибірки застосовується механізм ранньої зупинки, який перериває процес навчання за умови стабільного зростання валідаційної похибки впродовж визначеної кількості ітерацій.

На фінальному етапі конфігурація та параметри навченої моделі експортуються до файлової системи для забезпечення можливості її подальшої експлуатації.

2.4. Алгоритм розпізнавання зображень

Беручи до уваги архітектурні особливості та принципи функціонування згорткових нейронних мереж, було розроблено алгоритм автоматизованого розпізнавання зображень, структурно-логічну схему якого наведено на рисунку 2.12.

На початковому етапі алгоритму відбувається ініціалізація параметрів попередньо навченої нейромережевої моделі шляхом їх завантаження з відповідних файлів. Після цього здійснюється імпорт первинних поліхромних зображень f_{RGB} та виконується їх просторова нормалізація до уніфікованої роздільної здатності. На базі відмасштабованих масивів візуальних даних обчислюється та генерується вхідний вектор X.

У результаті нейромережевої обробки вхідних векторів X обчислюються значення вихідних сигналівY моделі, де порядковий індекс вихідного вузла є еквівалентним числовому ідентифікатору класифікаційної категорії зображення. Результуючий індекс класу ідентифікованого об'єкта cl_{max}детермінується як індекс елемента вихідного вектора Y, що характеризується максимальним числовим значенням.

На завершальному етапі, спираючись на визначений числовий ідентифікатор cl_{max}, остаточно встановлюється відповідна номенклатурна назва конкретного виду фрукта.



Рисунок 2.12 – Схема алгоритму розпізнавання зображень

РОЗДІЛ 3 РЕАЛІЗАЦІЯ ДОДАТКУ ДЛЯ РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ ФРУКТІВ НА БАЗІ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ

3.1. Розробка підсистеми навчання згорткової нейронної мережі

Практичну реалізацію програмного забезпечення для класифікації зображень фруктів здійснено мовою програмування Python із залученням обчислювальних ресурсів хмарної платформи Google Colaboratory. Фундаментальним компонентом створеного застосунку є підсистема індуктивного навчання штучної нейромережевої моделі. Вихідний код логічно структуровано за відповідними комірками інтерактивного середовища Jupyter Notebook, повний лістинг якого винесено у додатки до кваліфікаційної роботи.

Процедура завантаження масивів даних до хмарного середовища реалізована через стандартні модулі взаємодії з файловою системою:

```
from google.colab import files
uploaded = files.upload()
for fn in uploaded.keys():
    print('User uploaded file "{name}" with length {length} bytes'.format(name=fn, length=len(uploaded[fn])))
```

Після успішного імпорту файлів відбувається ініціалізація та підключення необхідних спеціалізованих бібліотек для виконання математичних обчислень та візуалізації даних.

Користувацький інтерфейс програмного комплексу побудовано на базі інтерактивних елементів управління, зовнішній вигляд яких наведено на відповідному рисунку. Зазначені контент-модулі інтегруються безпосередньо у веб-середовище та забезпечують ергономічну взаємодію оператора з функціоналом системи через графічні оболонки, надаючи швидкий доступ до керованих процесів (рис. 3.1.).

```
# Ініціалізація елементів графічного інтерфейсу
from google.colab import widgets
import ipywidgets
from IPython.display import display, clear_output

output = ipywidgets.widgets.Output()

button_name = []
button_name.append({'name':'Read_DataSet'})
```

```
    button_name.append({'name':'Train_CNN'})
    button_name.append({'name':'Save_CNN'})
    button_name.append({'name':'Read_CNN'})
    button_name.append({'name':'Recogn_Image_File'})
    button_name.append({'name':'Recogn_Image_Cam'})

    # Обробник подій взаємодії з елементами управління
    def on_button_clicked(b):
        with output:
            clear_output()
            sign_name = str(b.description)

            if (sign_name == 'Read_DataSet') or (sign_name ==
'Train_CNN'):
                # Імпорт ідентифікаторів класів та їх загальної
кількості
                QClass, mClass_Name = Read_Name_Class()
                QNTr, mClass_NumTr, mFile_NameTr = Read_Name_Train()

                # Формування тренувального набору даних
                x_train,      yt_train,      yt_trainC,      QY1      =
Create_Train_Set(QClass, QNTr, Ms, Ns, mClass_NumTr, mFile_NameTr)

            if sign_name == 'Train_CNN':
                Qepochs = tQepochs.value
                model  =  CNN_Train(QNTr,  x_train,  yt_trainC,  QNV,
x_valid, yt_validC, Ms, Ns, QY, Qepochs)

            if sign_name == 'Save_CNN':
                model.save('model_CNN_3') # Експорт параметрів навченої
моделі

            if sign_name == 'Read_CNN':
                loaded_model = tf.keras.models.load_model('model_CNN_3')
# Ініціалізація існуючої моделі

            if sign_name == 'Recogn_Image_File':
                print('\n', 'Command: Recogn_Image_File', '\n')
                Mode_Camera = 0 # Режим роботи з графічними файлами
                file_name2 = dFile_Name.label
                Recogn(Mode_Camera, file_name2)

            if sign_name == 'Recogn_Image_Cam':
                print('\n', 'Command: Recogn_Image_Cam', '\n')
                Mode_Camera  =  1  #  Режим  роботи  з  пристроєм
відеозахоплення
                Recogn(Mode_Camera, file_name2)

            print('\n', 'Button: ', str(b.description), '\n')
            print('\nSelect button')
```

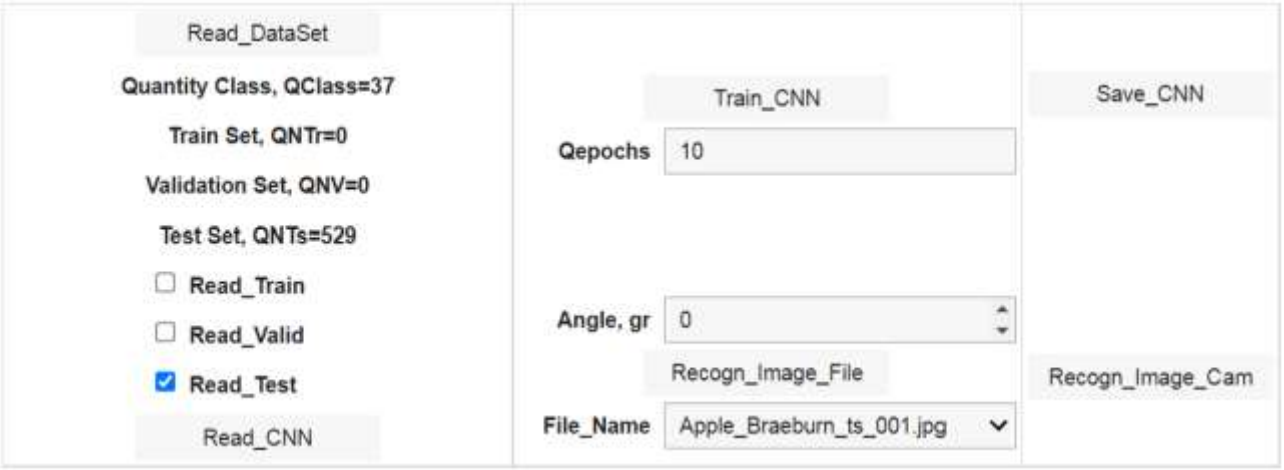


Рисунок 3.1 – Графічний інтерфейс програмного забезпечення

Керування базовим функціоналом у розробленому користувацькому інтерфейсі здійснюється за допомогою відповідних елементів управління:

- Read_DataSet – ініціалізація процедури імпорту номенклатури класів із визначенням їхньої загальної кількості QClass, а також завантаження тренувального, валідаційного та екзаменаційного наборів даних.
- Train_CNN – запуск алгоритму індуктивного навчання згорткової нейронної мережі, де гранична кількість епох Qepochs конфігурується через відповідний інтерактивний елемент.
- Save_CNN – експорт вагових коефіцієнтів та архітектури налаштованої нейромережевої моделі до файлової системи.
- Read_CNN – імпорт параметрів попередньо навченої згорткової нейронної мережі зі збережених файлів.
- Recogn_Image_File – активація модуля класифікації візуального образу з імпортованого растрового документа, при цьому ідентифікатор цільового файлу FileName визначається користувачем через спеціалізоване випадне меню.
- Recogn_Image_Cam – ініціалізація процесу ідентифікації об'єкта на основі візуальних даних, отриманих із підключеного пристрою відеозахоплення.

Процедура завантаження масивів візуальних даних та відповідних класифікаційних індексів для формування тренувальної множини реалізується за допомогою спеціалізованої програмної функції Create_Train_Set. За

аналогічним принципом забезпечується імпорт елементів валідаційної та екзаменаційної вибірок.

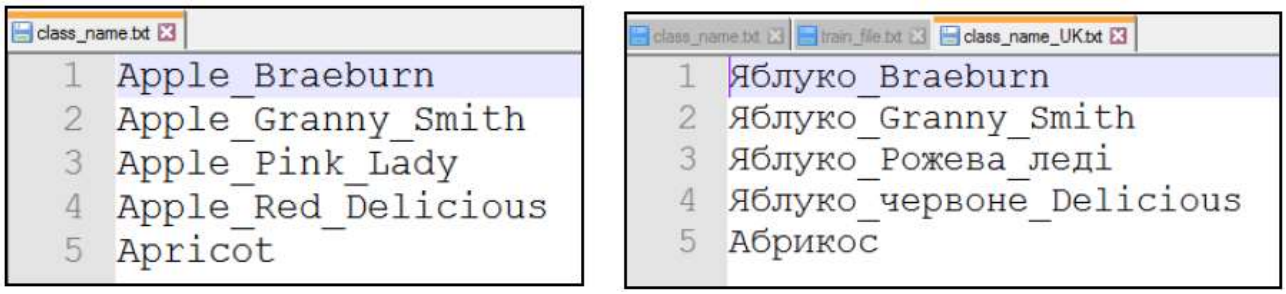


Рисунок 3.2 – Структура текстового файлу «class_name.txt» із номенклатурою класів фруктів англійською та українською мовами

Програмна функція Read_Name_Class() забезпечує імпорт ідентифікаторів класів mClass_Name та визначення їхньої загальної кількості QClass із зовнішнього текстового документа «class_name.txt», структуру якого наведено на рисунку 3.2.

Завантаження інформації з файлу реалізується за допомогою методу read_csv, оптимізованого для обробки даних із роздільниками, із подальшим формуванням структурованого фрейму даних dataI0. Масив найменувань класів mClass_Name генерується на основі першого стовпця сформованого фрейму з урахуванням специфіки нульової індексації базової мови програмування, при цьому загальний обсяг класів QClass відповідає кількості лінійних записів qRowI у вихідному файлі.

Підпрограма Read_Name_Train() виконує екстракцію параметрів тренувальної вибірки, зокрема загальної кількості зображень QNTr, масиву імен файлів mFile_NameTr та відповідних класифікаційних індексів mClass_NumTr із документа «train_file.txt», що проілюстровано на рисунку 3.3. У межах поточного розділу алгоритмічні аспекти функціонування програмного забезпечення розглядаються на базі обмежених за обсягом тренувальної, валідаційної та екзаменаційної вибірок. Натомість для комплексного тестування інформаційної системи у наступному розділі задіяно репрезентативні масиви даних значно більшої розмірності.



Рисунок 3.3 – Структура текстового документа «train_file.txt» із зазначенням ідентифікаторів графічних файлів та асоційованих класифікаційних індексів першого варіанта тренувальної вибірки

Структурування даних із файлу «train_file.txt» також здійснюється шляхом їх імпорту до об'єкта dataI0 за допомогою виклику функції read_csv. Розмірність тренувальної множини QNTr детермінується загальною кількістю рядків qRowI. Класифікаційні індекси mClass_NumTr вилучаються з першого стовпця згенерованого фрейму, тоді як ідентифікатори графічних файлів mFile_NameTr імпортуються з другого стовпця.

```
    # Формування тренувальної вибірки
    # Імпорт загальної кількості зображень QNTr, ідентифікаторів файлів mFile_NameTr
    # та класифікаційних індексів mClass_NumTr
    def Read_Name_Train():
        dataI0    =    pd.read_csv('train_file.txt',    delimiter='    ', header=None)
        qRowI, qColumnI = dataI0.shape
        QNTr = qRowI
        print('Розмір навчальної вибірки, QNTr =', QNTr)

        mClass_NumTr = np.zeros(shape=(QNTr), dtype=np.int64)
        for ni in range(0, QNTr):
            mClass_NumTr[ni] = (dataI0.values[ni, 0:1])
        print('Класифікаційні індекси тренувальної вибірки, mClass_NumTr =', mClass_NumTr)
```

```
        mFile_NameTr = []
        mFile_NameTr0 = dataI0.values[0:QNTr, 1:2]
        for ni in range(0, QNTr):
            st1 = str(mFile_NameTr0[ni])
            len1 = len(st1)
            mFile_NameTr.append(st1[2:len1-2])
        print('Ідентифікатори файлів тренувальної вибірки, mFile_NameTr
=', mFile_NameTr)

        return QNTr, mClass_NumTr, mFile_NameTr
```

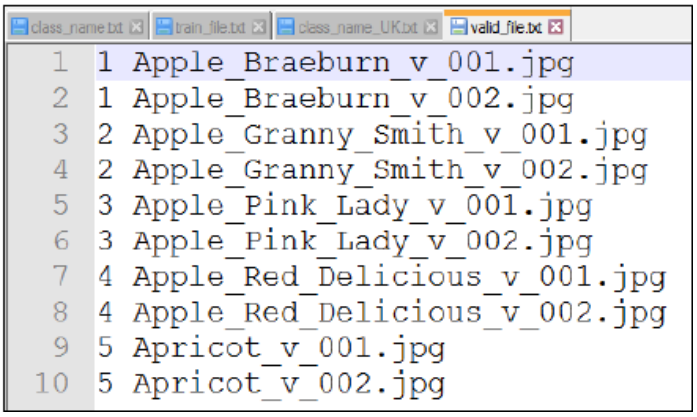


Рисунок 3.4 – Структура текстового документа «valid_file.txt» із зазначенням ідентифікаторів графічних файлів та асоційованих класифікаційних індексів першого варіанта валідаційної вибірки

Програмна функція Read_Name_Valid() забезпечує імпорт параметрів валідаційної вибірки, зокрема загальної кількості зображень QNV, масиву ідентифікаторів файлів mFile_NameV та класифікаційних індексів mClass_NumV із зовнішнього текстового документа «valid_file.txt», що проілюстровано на рисунку 3.4. Завантаження інформації з файлу реалізується за допомогою функції read_csv із подальшим формуванням фрейму даних dataI0. Розмірність валідаційної множини QNV детермінується загальною кількістю рядків qRowI у зазначеному файлі.

За аналогічним принципом підпрограма Read_Name_Test() виконує екстракцію характеристик екзаменаційної вибірки, а саме кількості об'єктів QNTs, найменувань графічних файлів mFile_NameTs та відповідних номерів класів mClass_NumTs із документа «test_file.txt», як показано на рисунку 3.5. Імпорт даних із цього текстового файлу також здійснюється за допомогою методу read_csv до структури dataI0. Обсяг екзаменаційної вибірки QNTs еквівалентний сумарній кількості лінійних записів qRowI.

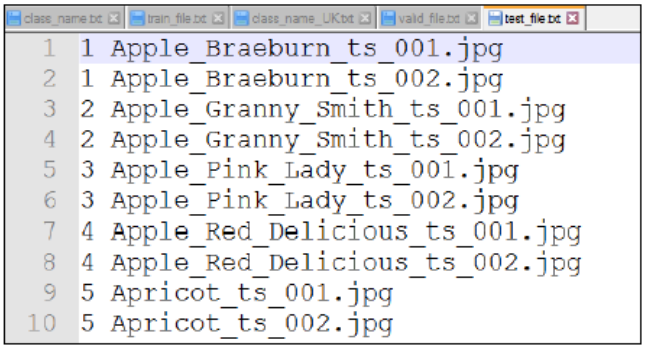
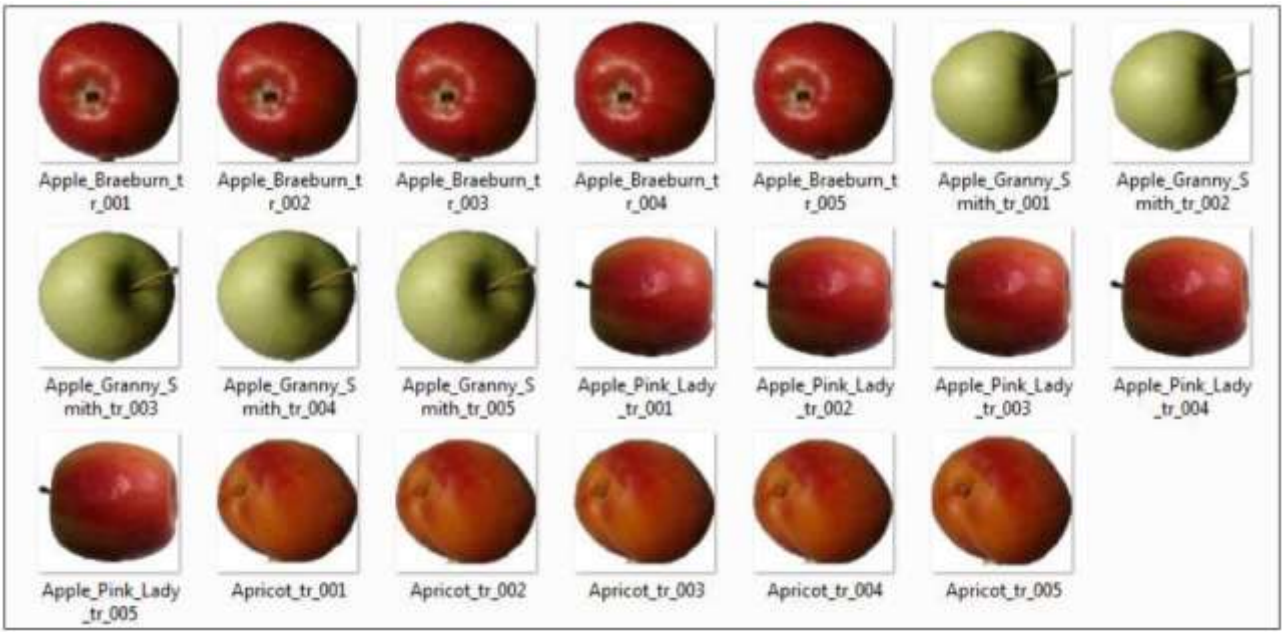


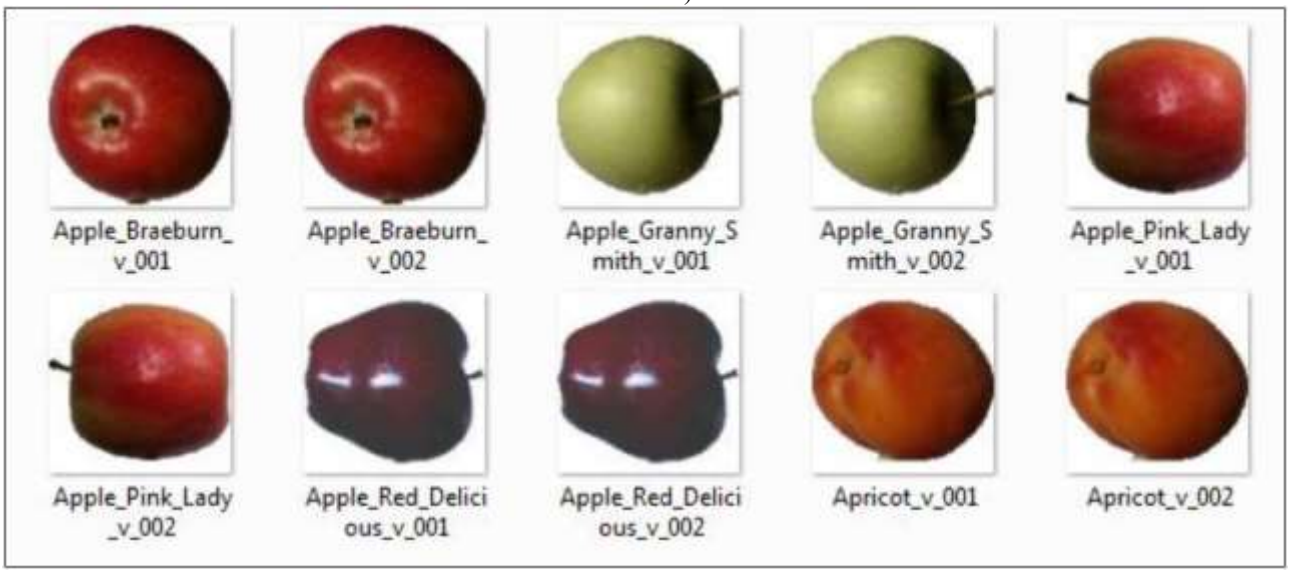
Рисунок 3.5 – Структура текстового документа «test_file.txt» із зазначенням ідентифікаторів та індексів першого варіанта екзаменаційної вибірки

Алгоритмічна процедура формування тренувального, валідаційного та екзаменаційного наборів даних на базі попередньо імпортованих візуальних масивів наочно проілюстрована відповідними структурними схемами, наведеними на рисунках 3.6, а, 3.6, б та 3.6, в. Програмну імплементацію модулів генерації зазначених інформаційних вибірок реалізовано у вигляді наведеного лістингу:

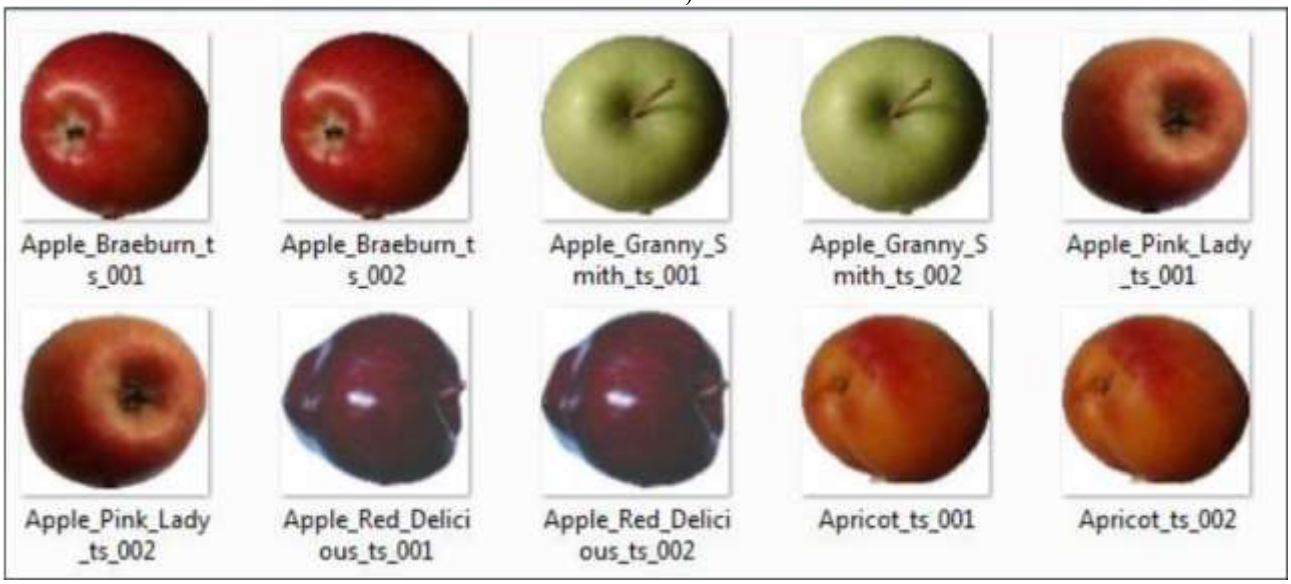
```
# Формування тренувальної вибірки
def Create_Train_Set(QClass, QNTr, Ms, Ns, mClass_NumTr,
mFile_NameTr):
    set1 = 'Train'
    # Ініціалізація вхідного тензора тренувальної вибірки обсягом
QNTr зображень
    x_train = np.zeros(shape=(QNTr, Ms, Ns, 3), dtype=np.float64)
    QY = QClass # Детермінація кількості вихідних вузлів моделі
відповідно до кількості класів
    yt_trainC = np.zeros(shape=(QNTr), dtype=np.int64)
    yt_trainC = yt_trainC + (mClass_NumTr - 1)
    for ni in range(0, QNTr):
        File_Name1 = mFile_NameTr[ni]
        M, N, f, fRGB0, fRGB = Read_image(File_Name1)
        # Просторова нормалізація зображення до уніфікованої
розмірності Ms x Ns
        fRGBs = Calc_fRGBs(fRGB0, fRGB, M, N, Ms, Ns)
        # Формування масиву вхідних векторів тренувальної вибірки
        x_train[ni, :, :, :] = x_train[ni, :, :, :] + fRGBs
        fRGB_show(fRGB, ni, File_Name1, M, N, set1)
        fRGBs_show(fRGBs, ni, File_Name1, M, N, set1)
    return x_train, yt_train, yt_trainC, QY
# Формування валідаційної вибірки
def Create_Valid_Set(QNV, Ms, Ns, mClass_NumV, mFile_NameV, QY):
    set1 = 'Valid'
    x_valid = np.zeros(shape=(QNV, Ms, Ns, 3), dtype=np.float64)
    yt_validC = np.zeros(shape=(QNV), dtype=np.int64)
    fRGB_show(fRGB, ni, File_Name1, M, N, set1)
    fRGBs_show(fRGBs, ni, File_Name1, M, N, set1)
    return x_valid, yt_valid, yt_validC
```



а)



б)



в)

Рисунок 3.6 – Приклади зображень: а – першого варіанта тренувальної вибірки; б – першого варіанта валідаційної вибірки; в – першого варіанта екзаменаційної вибірки [18]

Програмна функція Read_image(File_Name1) забезпечує завантаження поліхромного зображення f_{RGB} у колірному просторі RGB просторовою розмірністю $M \times N$ елементів із файлу, ідентифікатор якого визначається параметром File_Name1. Значення інтенсивності пікселів за кожним спектральним каналом підлягають процедурі нормалізації у діапазоні дійсних чисел від 0 до 1.

Підпрограма fRGB_show(fRGB, ni, File_Name1, M, N, set1) виконує графічну візуалізацію вихідного кольорового масиву даних f_{RGB} із виведенням його порядкового індексу у відповідній вибірці n_i, найменування графічного файлу File_Name1, а також просторових параметрів висоти M та ширини N у пікселях, що наведено на рисунку 3.6, а. За аналогічним алгоритмом функція fRGBs_show(fRGBs, ni, File_Name1, M, N, set1) реалізує відображення просторово відмасштабованого зображення f_{RGBs} із зазначенням ідентичних атрибутів, зокрема індексу n_i, імені файлу File_Name1 та початкових габаритів M і N, як проілюстровано на рисунку 3.7, в.

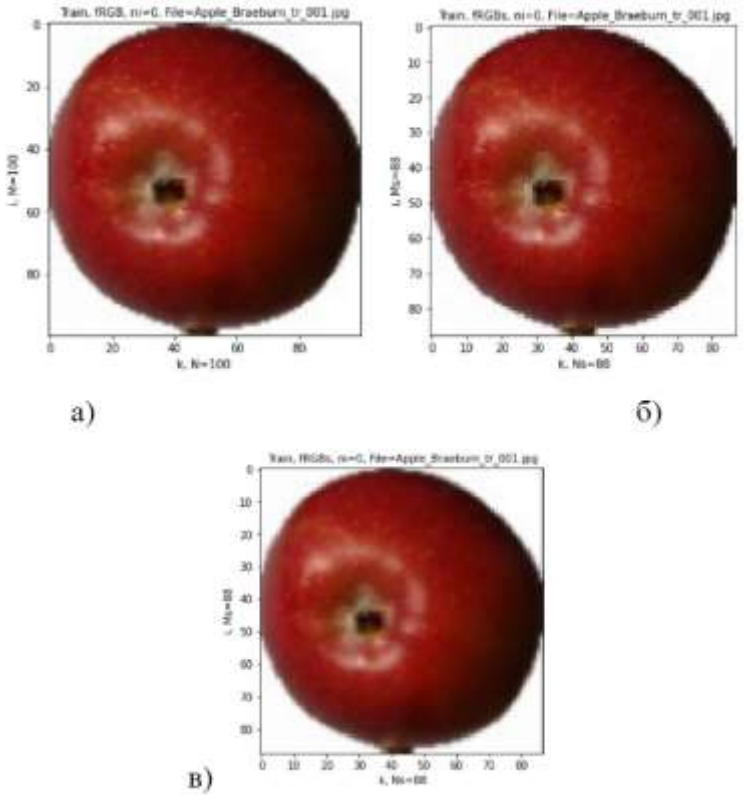


Рисунок 3.7 – Візуалізація імпорту зображення тренувальної вибірки з індексом n_i = 0$: а – первинний візуальний масив f_{RGB}; б – масив f_{RGBs} після процедури просторового масштабування; в – масив f_{RGBs} після просторового масштабування та фільтрації за алгоритмом Гауса із середньоквадратичним відхиленням $Sigma = 0.5$

Програмна функція Calc_fRGBs(fRGB0, fRGB, M, N, Ms, Ns) здійснює обчислення відмасштабованого зображення f_{RGBs} із просторовою розмірністю $Ms \times Ns$ елементів, зокрема 88×88 пікселів, на базі первинного масиву f_{RGB} розмірністю $M \times N$. Процедура редукції просторової роздільної здатності застосовується з метою інтенсифікації процесу машинного навчання нейромережевої моделі. Алгоритмічно це реалізується шляхом визначення коефіцієнта вертикального масштабування$Scale1 = M/Ms$ та розрахунку відповідного горизонтального розміру $Ns2 = \text{round}(N/Scale1)$. Операція зміни масштабу виконується ітеративно для кожної з трьох спектральних компонент червоної, зеленої та синьої. Безпосереднє масштабування спектральних площин f_R забезпечується викликом методу resize із параметрами бікубічної інтерполяції з бібліотеки комп'ютерного зору OpenCV.

У випадках, коли розрахована ширина $Ns2$ перевищує цільове значення Ns, виконується відсікання надлишкових крайових пікселів. Натомість за умови меншого значення $Ns2$ порівняно з Ns, утворені порожні вертикальні зони з обох боків заповнюються пікселями нульової інтенсивності. Зважаючи на те, що процедура даунсемплінгу провокує виникнення артефактів пікселізації, як показано на рисунку 3.7, б, для мінімізації структурних спотворень застосовується згладжування сформованого масиву f_{RGBs} за допомогою фільтра Гауса. Для цього використовується метод ndimage.gaussian_filter із параметром середньоквадратичного відхилення ядра $sigma_C = 0.5$, результати роботи якого наведено на рисунку 3.7, в.

З метою забезпечення інваріантності згорткової нейронної мережі до афінних просторових перетворень, зокрема ротаційних, на етапі тренування перед процедурою масштабування ініціюється випадковий поворот вхідних зображень на кут $rAngle$ у межах від 0 до 359 градусів. Дана геометрична трансформація імплементується за допомогою функції rotate_fon_white(fRGB0, rAngle) на базі інструментарію OpenCV.

Алгоритмічний модуль Create_Train_Set(QClass, QNTr, Ms, Ns) відповідає за генерацію тренувальної множини даних. Багатовимірний масив

x_{train} акумулює вхідні тензори навчальної вибірки загальним обсягом $QNTr$ зображень, кожне з яких характеризується розмірністю $Ms \times Ns$ пікселів та трьома колірними каналами. Кількість вихідних вузлів моделі QY детермінується загальною кількістю класифікаційних категорій $QClass$. Вектор yt_{trainC} призначений для зберігання еталонних вихідних сигналів обсягом $QNTr$ елементів, де кожному вхідному візуальному патерну ставиться у відповідність ідентифікатор цільового класу $ncl = mClass_NumTr[ni]$. Наповнення тензора x_{train} здійснюється шляхом ітеративного імпорту зображень за їхніми системними іменами $File_Name1 = mFile_NameTr[ni]$ із використанням функції Read_image, їх подальшої просторової нормалізації функцією *Calc_fRGBs* та безпосереднього запису згенерованих масивів f_{RGBs} до структури x_{train}.

Формування валідаційної множини даних забезпечується функцією Create_Valid_Set(QNV, Ms, Ns, mClass_NumV, mFile_NameV, QY). За аналогічним принципом ініціалізується масив x_{valid} для зберігання QNV нормалізованих вхідних зображень, а також вектор yt_{validC} для фіксації відповідних еталонних ідентифікаторів класів $ncl = mClass_NumV[ni]$. Наповнення зазначених структур даних реалізується через послідовне виконання процедур читання та просторового масштабування графічних файлів із наступним збереженням отриманих масивів f_{RGBs} до x_{valid}.

Генерування екзаменаційної вибірки покладено на функцію Create_Test_Set(QNTs, Ms, Ns, mClass_NumTs, mFile_NameTs, QY). Відповідно до описаного алгоритму, формується вхідний тензор $xtest$ ємністю $QNTs$ зображень та вектор цільових значень yt_{testC}, де кожний елемент містить еталонний індекс $ncl = mClass_NumTs[ni]$. Заповнення багатовимірного масиву x_{test} відбувається шляхом завантаження візуальних даних, їх подальшого масштабування до цільової розмірності та інтеграції до результуючої структури.

Як базова математична модель використовується еталонна архітектура згорткової нейронної мережі LeNet-5 із певною топологічною модифікацією,

яку проілюстровано на рисунку 3.8. Ключова архітектурна адаптація полягає у встановленні розмірності рецептивного поля на рівні $Ms = Ns = 88$ пікселів. Таке значення є компромісним оптимумом, оскільки подальше збільшення просторової роздільної здатності призводить до нераціонального зростання обчислювальних витрат на тренування, тоді як її зменшення критично знижує загальну точність класифікації.

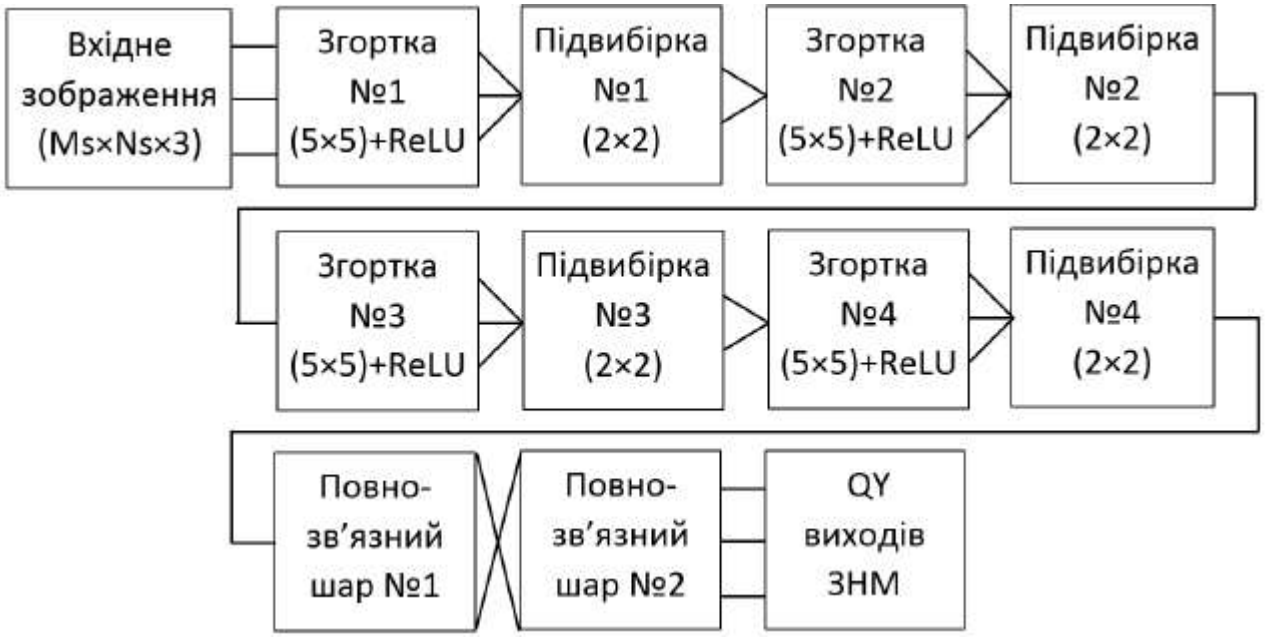


Рисунок 3.8 – Структурна організація згорткової нейронної мережі на базі архітектури LeNet-5

З огляду на задану просторову розмірність вхідного масиву візуальних даних $Ms \times Ns$, архітектуру розроблюваної нейромережевої моделі було масштабовано до чотирьох згорткових шарів та аналогічної кількості шарів субдискретизації. Це є суттєвою модифікацією порівняно з класичною базовою топологією LeNet-5, котра оперує лише трьома згортковими рівнями. Окрім того, з метою максимізації метрик точності класифікації об'єктів, термінальний блок моделі доповнено двома повнозв'язними шарами замість одного стандартного.

Наступним алгоритмічним етапом ініціюється процедура індуктивного навчання згорткової нейронної мережі. Базові класи математичних моделей та функціональні обчислювальні шари імпортуються за допомогою інструментарію програмної бібліотеки Keras.

```
model = models.Sequential()
model.add(layers.Conv2D(16, (5, 5), activation='relu', input_shape=(Ms,
Ns, 3)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(32, (5, 5), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (5, 5), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(128, (5, 5), activation='relu'))
model.add(layers.Flatten())
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(QY))
model.summary()
```

Проектування загальної топології моделі здійснюється шляхом послідовної суперпозиції згорткових та повнозв'язних обчислювальних вузлів. Формування згорткових рівнів реалізується викликом методу Conv2D. Його параметрична конфігурація суворо визначає кількість генеруваних карт ознак, просторові розміри ядра фільтра, зокрема його вертикальний вимір M_w та горизонтальний вимір N_w, а також тип нелінійної функції активації штучних нейронів.

Інтеграція шарів субдискретизації забезпечується оператором MaxPooling2D із заданим просторовим вікном сканування розміром 2 на 2 елементи. Даний метод імплементує алгоритм двократного зменшення просторової роздільної здатності оброблюваної карти ознак одночасно за обома координатними осями.

Завершальні рівні нейромережевої архітектури виконують функцію класифікатора та формуються як повнозв'язні шари за допомогою методу Dense. Його конфігурація передбачає чітку специфікацію кількості залучених штучних нейронів та застосовуваного активаційного алгоритму.

Унаслідок виконання наведеного програмного коду синтезується цільова згорткова нейромережева модель, ієрархічну структурну специфікацію якої відображено на рисунку 3.9.

Layer (type)	Output Shape	Param #
conv2d_44 (Conv2D)	(None, 84, 84, 16)	1216
max_pooling2d_33 (MaxPooling2D)	(None, 42, 42, 16)	0
conv2d_45 (Conv2D)	(None, 38, 38, 32)	12832
max_pooling2d_34 (MaxPooling2D)	(None, 19, 19, 32)	0
conv2d_46 (Conv2D)	(None, 15, 15, 64)	51264
max_pooling2d_35 (MaxPooling2D)	(None, 7, 7, 64)	0
conv2d_47 (Conv2D)	(None, 3, 3, 128)	204928
flatten_11 (Flatten)	(None, 1152)	0
dense_22 (Dense)	(None, 32)	36896
dense_23 (Dense)	(None, 4)	132

Рисунок 3.9 – Структурна організація розробленої згорткової нейронної мережі, імплементована засобами програмної бібліотеки Keras

Процес індуктивного навчання, який полягає в ітеративному коригуванні вагових коефіцієнтів моделі, реалізується за допомогою алгоритму адаптивної оцінки моментів Adam. Як критерій оцінювання якості функціонування класифікатора застосовується метрика точності accuracy. Цей показник формалізується як відношення кількості безпомилково ідентифікованих об'єктів до загального обсягу аналізованої множини даних. Для кількісного оцінювання похибки оптимізації функцією втрат обрано алгоритм розрідженої категоріальної крос-ентропії sparse_categorical_crossentropy.

За умови залучення валідаційної множини даних імплементується механізм ранньої зупинки процесу тренування. Алгоритм перериває оптимізацію синаптичних ваг, якщо значення функції втрат на валідаційній вибірці val_loss не демонструє тенденції до зниження впродовж визначеної

кількості епох, зокрема дванадцяти ітерацій. Відповідна логіка зупинки конфігурується за допомогою параметра callbacks1.

Безпосередня ініціалізація процесу навчання згорткової нейромережі здійснюється шляхом виклику програмного методу fit за таким синтаксисом:

```
model.fit(x_train, yt_trainCN, epochs=Qepochs, callbacks=callbacks1, validation_data=(x_valid, yt_validCN))
```

У наведеному програмному виразі змінна $yt_trainCN$ визначає масив еталонних вихідних сигналів моделі для тренувальної вибірки з дотриманням специфікацій викликаного методу. Параметр $Qepochs$ детермінує гранично допустиму кількість ітерацій навчання, тоді як $yt_validCN$ репрезентує еталонні класифікаційні індекси для валідаційної множини даних.Із системного об'єкта історії навчання history виконується екстракція значень похибки Loss та метрики точності *mAccuracy* для кожної завершеної епохи. За умови активації режиму валідації, що визначається значенням керуючої змінної $Mode_Valid = 1$, забезпечується паралельне обчислення метрик похибки як для тренувальної, так і для контрольної вибірок, динаміку яких відображено на рисунку 3.10.

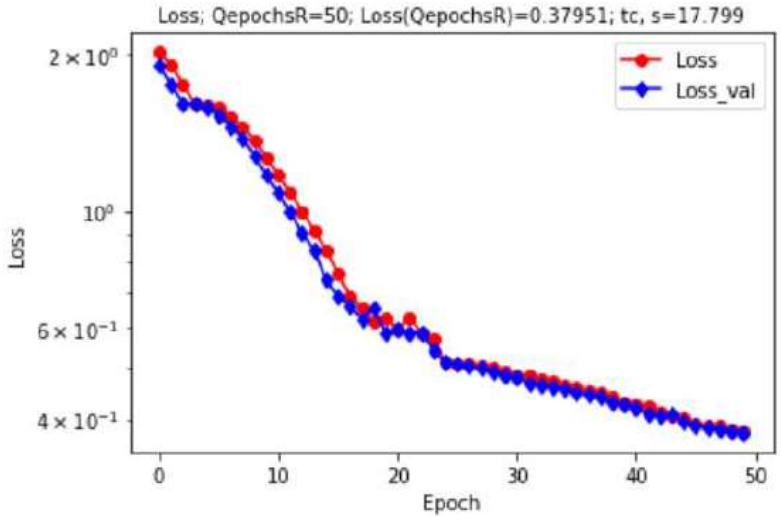


Рисунок 3.10 – Динаміка зміни функції втрат згорткової нейронної мережі залежно від ітерації навчання: $Loss$ – для тренувальної вибірки, $Loss_val$ – для валідаційної вибірки

Програмна функція $mLoss_show$ забезпечує графічну візуалізацію значень похибки навчання $mLoss$ для кожної ітерації e в межах загальної кількості епох $QepochsR$, а також відображає сумарні часові витрати на процес тренування моделі t_c. Процедура експорту конфігурації та оптимізованих вагових коефіцієнтів навченої штучної нейронної мережі до зовнішньої файлової системи реалізується шляхом ініціалізації програмного методу $model.save$.

3.2. Розробка підсистеми розпізнавання зображень

Підсистема ідентифікації візуальних образів на базі згорткових нейронних мереж імплементована мовою програмування Python із залученням обчислювальних ресурсів хмарного середовища Google Colaboratory. Процес класифікації зображень здійснюється попередньо навченою нейромережевою моделлю, ініціалізація якої реалізується за допомогою програмної функції load_model.

```
def Recogn(Mode_Camera, file_name2, rAngle):
    if Mode_Camera == 0:
        fRGB0, file_name = Read_ImageCNN(file_name2)
        File_Name1 = file_name2

    if Mode_Camera == 1:
        try:
            fRGB0, filename = Read_Camera()
            file_name = filename
            File_Name1 = file_name
            print('Saved to {}'.format(filename))
        except Exception as err:
            print(str(err))
    fRGB0_size = fRGB0.shape
    M = fRGB0_size[0]
    N = fRGB0_size[1]
    fRGB = np.zeros(shape=(M, N, 3), dtype=np.float64)
    fRGB = fRGB + fRGB0
    fRGB = fRGB / 255
    fRGB_showR(M, N, fRGB, File_Name1)
    start_time = time.time()
    fRGBs = Calc_fRGBsR(fRGB0, fRGB, M, N, Ms, Ns, rAngle)
    x_train1 = np.zeros((1, Ms, Ns, 3))
    x_train1[0, :, :, :] = x_train1[0, :, :, :] + fRGBs
    y1 = model.predict(x_train1)
    y2 = np.zeros(shape=(QY), dtype=np.float64)
    y2 = y2 + y1
    y1_min = np.min(y1)
    y1_max = np.max(y1)
    y2 = (y2 - y1_min) * 1.0 / (y1_max - y1_min)
    y1 = y2
```

```
    cl_max = -1
    y_max = -1
    for j in range(0, QY):
        if (y_max < y1[0, j]):
            y_max = y1[0, j]
            cl_max = j

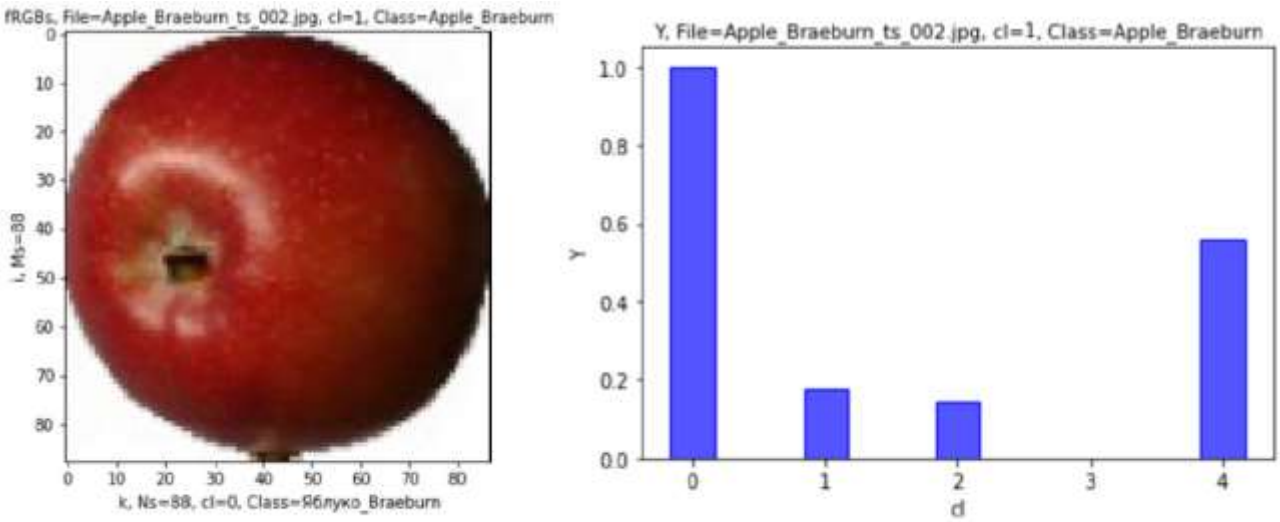
    stop_time = time.time()
    tr = stop_time - start_time
    fRGBs_showR(fRGBs, Ms, Ns, File_Name1, cl_max, mClass_Name,
mClass_Name_UK)
    mY_show(y1, cl_max, QY, mClass_Name, File_Name1)
    mY_bar(y1, cl_max, QY, mClass_Name, File_Name1)
```

Залежно від стану керуючої змінної $Mode_Camera$, джерелом візуальних даних виступає файлова система за умови нульового значення параметра, або ж безпосередньо пристрій відеозахоплення у разі встановлення одиничного значення.

Алгоритмічний модуль Read_ImageCNN забезпечує імпорт графічних даних до багатовимірного масиву f_{RGB0} із зазначеного файлу відповідно до його системного ідентифікатора $file_name$. Натомість підпрограма Read_Camera реалізує захоплення цифрового зображення безпосередньо з оптичного сенсора відеокамери.

Просторово нормалізований масив даних f_{RGBs} трансформується у вхідний тензор x_{train1}, що надходить на рецептивне поле згорткової нейронної мережі. Базуючись на значеннях вхідного вектора x_{train1}, за допомогою методу predict обчислюються вихідні сигнали моделі y_1. Отримані значення підлягають математичній нормалізації у діапазоні дійсних чисел від 0 до 1.

Серед компонентів вихідного вектора y_1 визначається елемент із максимальним числовим значенням, індекс якого cl_{max} є еквівалентним числовому ідентифікатору класу розпізнаного об'єкта. На завершальному етапі за встановленим класифікаційним індексом остаточно детермінується відповідна номенклатурна назва візуального образу, що наочно проілюстровано на рисунку 3.11.



а) б)

Рисунок 3.11 – Результати класифікації об'єкта екзаменаційної вибірки «Apple_Braeburn_ts_002»: а – просторово нормалізований візуальний масив; б – фактичні вихідні сигнали нейромережевої моделі $y_1 = [1.0,0.176,0.144,0.558]$ порівняно з еталонними значеннями $y_t = [1,0,0,0]$при визначеному ідентифікаторі класу $cl_{max} = 0$

Під час обробки переважної більшості графічних даних першого варіанта екзаменаційної вибірки було зафіксовано коректні результати ідентифікації. Однак, зважаючи на статистичну обмеженість обсягу початкової тренувальної множини, для забезпечення об'єктивної та комплексної верифікації розробленого програмного забезпечення було згенеровано другий варіант інформаційних вибірок із суттєво розширеним масивом візуальних даних.

По завершенню процедури оптимізації синаптичних ваг конфігурація згорткової нейронної мережі експортується до директорії хмарного середовища Google Colaboratory під відповідним ідентифікатором, зокрема «model_CNN_2». З метою локального збереження згенерованої директорії ініціюється процедура стиснення даних за допомогою системної команди !zip -r /content/model_CNN_2.zip /content/model_CNN_2. Надалі сформований архівний файл завантажується на фізичний накопичувач викликом програмного методу files.download('model_CNN_2.zip').

Процедура зворотного імпорту архіву з параметрами навченої моделі до робочого простору Google Colaboratory вимагає попереднього розпакування даних, що алгоритмічно реалізується через виконання команди !unzip "/content/model_CNN_2.zip" -d "/content/model_CNN_2/".

Унаслідок успішного виконання зазначених операцій та ініціалізації параметрів згорткова нейромережева модель стає повністю придатною для подальшої експлуатації в рамках вирішення прикладних задач із розпізнавання візуальних образів фруктів.

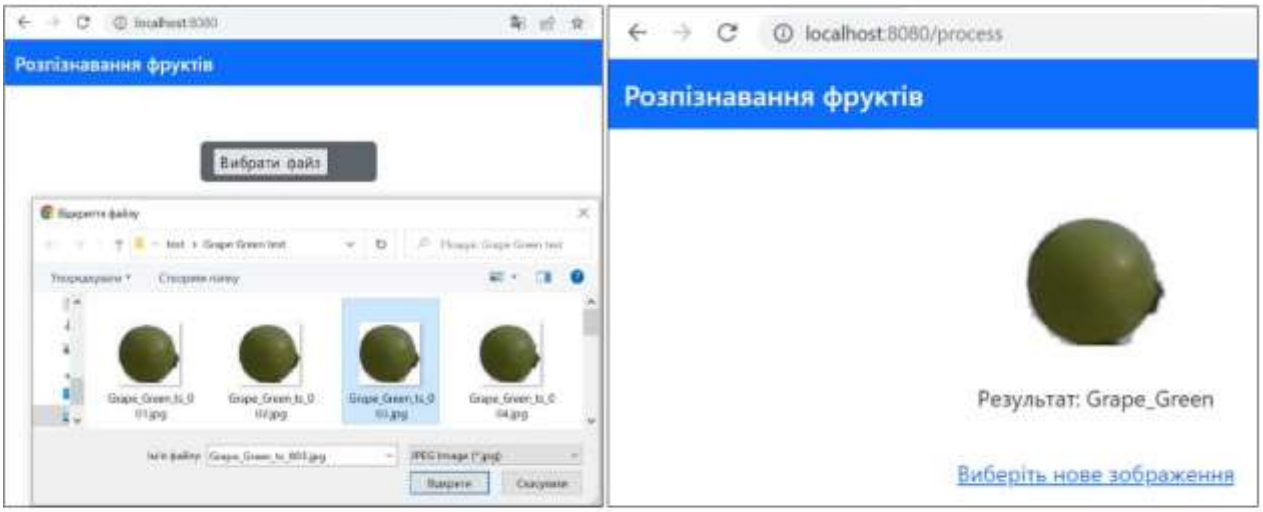
3.3. Розробка веб-інтерфейсу додатку

З метою оптимізації процесу ідентифікації візуальних образів та підвищення загальної ергономічності системи було спроектовано спеціалізований веб-інтерфейс програмного комплексу. Програмну імплементацію зазначеного графікального модуля здійснено на базі інтерпретатора мови Python третьої версії, при цьому основний виконуваний алгоритм інкапсульовано у головному системному файлі «main.py». Забезпечення клієнт-серверної інформаційної взаємодії реалізується через спеціально розроблену структуру об'єктно-орієнтованого класу «ServerHandler».

Функціональна архітектура застосунку включає низку спеціалізованих підпрограм. Безпосередній імпорт графічних масивів даних забезпечується методом «Read_ImageCNN», тоді як завантаження номенклатури класифікаційних категорій покладено на алгоритмічний блок «Read_Name_Class». Процедура просторової нормалізації та масштабування візуальних даних виконується функцією «Calc_fRGBsR», а остаточна класифікація цільових об'єктів ініціюється розрахунковим модулем «Recogn».

Активація системного файлу «main.py» через інтерфейс командного рядка ініціює розгортання локального сервера. Подальший доступ до інтерактивної оболонки користувача здійснюється за допомогою

стандартного браузера за мережевою адресою http://localhost:8080. Візуальне представлення розробленого веб-інтерфейсу наведено на рисунках 3.12 та 3.13.



а) б)

Рисунок 3.12 – Процес ідентифікації образу з використанням веб-інтерфейсу: а – етап завантаження вхідного файлу; б – кінцевий результат

За результатами практичної експлуатації розробленого веб-орієнтованого програмного забезпечення було підтверджено абсолютну коректність ідентифікації всіх проаналізованих візуальних образів фруктів.

Підсумовуючи вищевикладене, було спроектовано алгоритмічне забезпечення для індуктивного навчання згорткової нейронної мережі та подальшої класифікації візуальних даних. На базі розроблених алгоритмів реалізовано спеціалізований програмний комплекс для розпізнавання зображень фруктів, написаний мовою програмування Python із залученням обчислювальних потужностей хмарного середовища Google Colaboratory. Архітектурно вихідний код структуровано за окремими робочими комірками інтерактивного середовища розробки Jupyter Notebook. Структурна організація розробленого застосунку базується на двох ключових компонентах: модулі тренування нейромережевої моделі та підсистемі класифікації візуальних образів. Взаємодія з користувачем забезпечується через інтегровані графічні віджети та спеціально розроблений веб-інтерфейс.

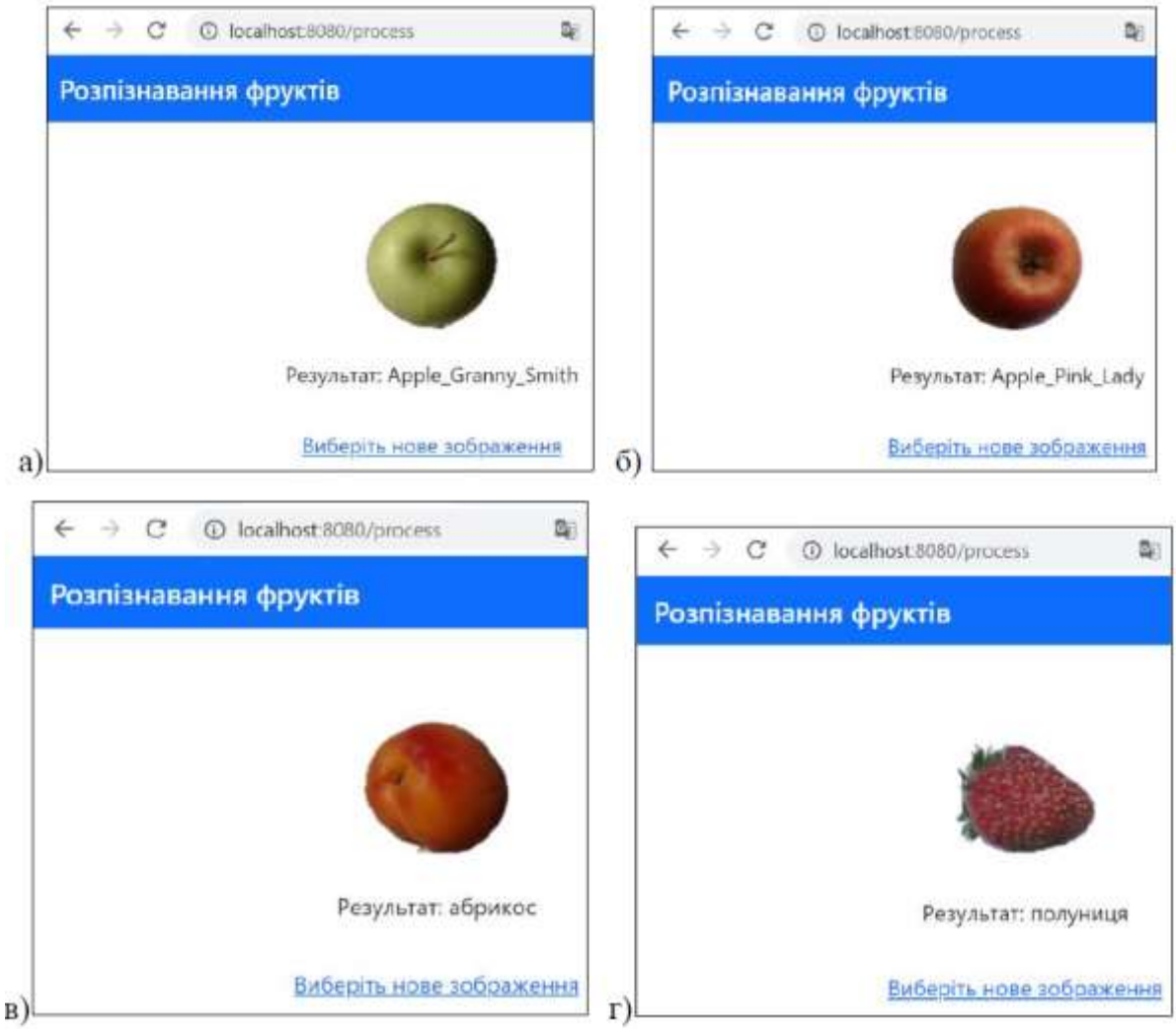


Рисунок 3.13 – Результати автоматизованої ідентифікації плодів екзаменаційної вибірки: а – яблуко сорту «Apple_Granny_Smith»; б – яблуко сорту «Apple_Pink_Lady»; в – абрикос; г – полуниця

Функціонал модуля тренування охоплює процедури імпорту ідентифікаторів зображень та відповідних класифікаційних індексів для формування тренувальної, валідаційної та екзаменаційної множин даних. Наступним етапом здійснюється генерування зазначених вибірок на основі масивів візуальних даних, що попередньо пройшли процедуру просторової нормалізації до єдиного масштабу. З метою нівелювання артефактів пікселізації, які неминуче виникають у процесі редукції просторової роздільної здатності, застосовується алгоритм згладжування за допомогою фільтра Гауса. Для забезпечення інваріантності класифікатора до геометричних

трансформацій об'єктів на етапі оптимізації вагових коефіцієнтів реалізовано механізм випадкової ротації вхідних зображень.

За основу математичної моделі обрано еталонну архітектуру LeNet-5, яку було модифіковано до конфігурації з чотирьох згорткових рівнів, чотирьох шарів субдискретизації та двох повнозв'язних блоків. У згорткових шарах застосовано рецептивні поля просторовою розмірністю 5 на 5 елементів. Процес ініціалізації та навчання моделі реалізовано за допомогою функціоналу програмних бібліотек TensorFlow та Keras. Оптимізація синаптичних ваг мережі здійснюється за алгоритмом адаптивної оцінки моментів Adam. Якість функціонування класифікатора оцінюється за метрикою точності accuracy, що відображає відношення кількості безпомилково ідентифікованих зразків до загального обсягу аналізованих даних. Кількісним критерієм похибки навчання виступає функція крос-ентропії crossentropy. Динаміка функції втрат систематично моніториться як на тренувальній, так і на валідаційній вибірках. Механізм регуляризації процесу тренування передбачає його примусове переривання у разі стабільного зростання валідаційної похибки впродовж визначеного ліміту епох. Оптимізована конфігурація навченої моделі остаточно експортується до файлової системи.

Модуль класифікації візуальних образів також імплементовано мовою Python у середовищі Google Colaboratory. Зазначена підсистема забезпечує завантаження параметрів попередньо навченої нейромережевої моделі, захоплення цифрових зображень із локальних графічних файлів або безпосередньо з пристроїв відеозахоплення, із подальшим виконанням алгоритмів автоматизованого розпізнавання досліджуваних об'єктів.

3.4. Тестування розробленого додатку

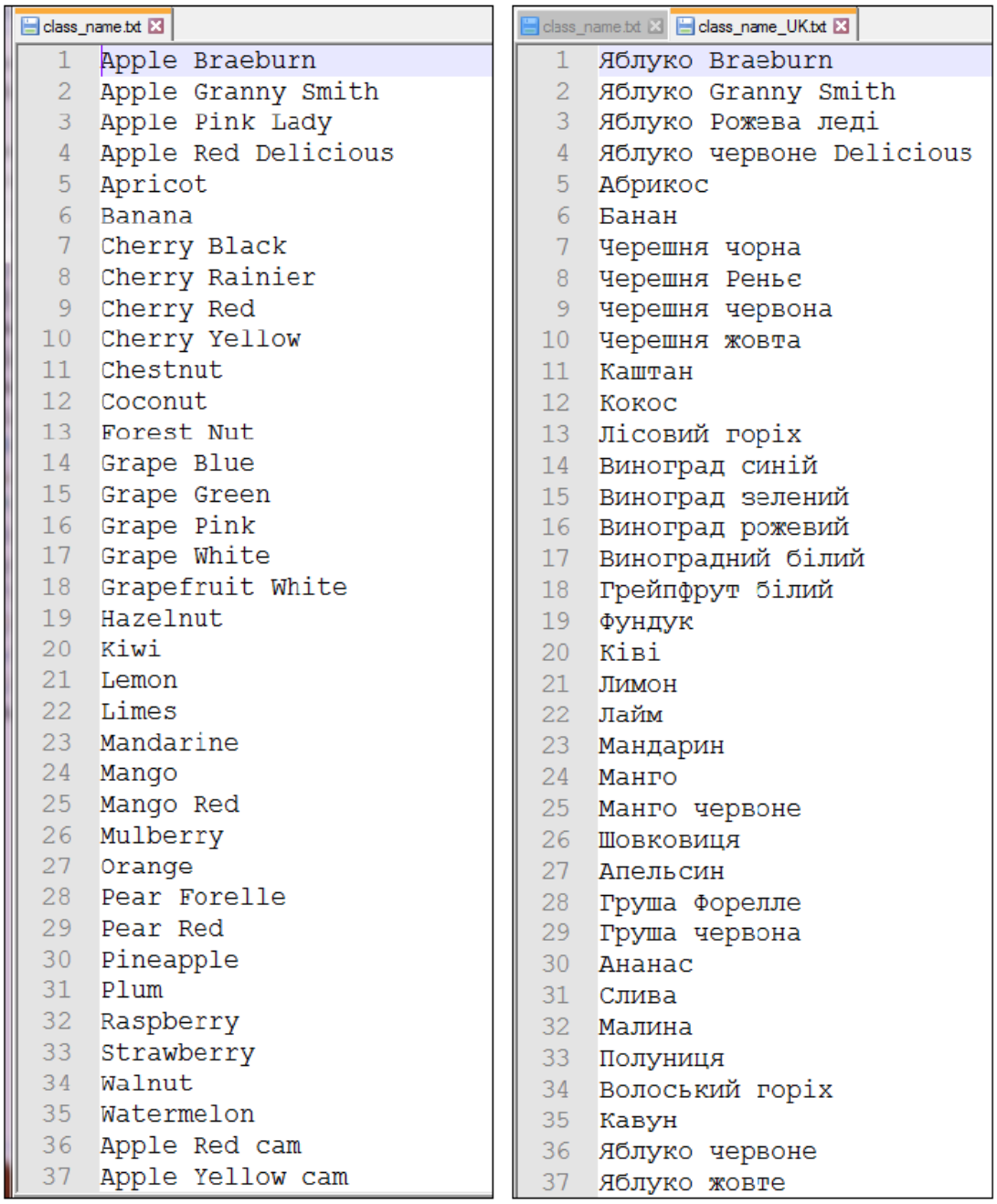
Із застосуванням розробленого програмного забезпечення було здійснено процедуру навчання ЗНМ на базі масивів даних другого варіанта тренувальної та валідаційної вибірок, що додатково задокументовано у додатку Б. Структуру

та елементи зазначеного тренувального набору даних проілюстровано на рисунках 3.14 та 3.15, тоді як характеристики валідаційної множини відображено на рисунку 3.16. З метою об'єктивної верифікації результатів оптимізації вагових коефіцієнтів та оцінювання ефективності функціонування розробленої системи класифікації було залучено незалежну екзаменаційну вибірку, візуальне представлення якої наведено на рисунку 3.17.



Рисунок 3.14 – Візуальне представлення фрагмента другого варіанта тренувальної вибірки

Сформовані інформаційні масиви містять цифрові зображення плодів, розподілені за тридцятьма сімома класифікаційними категоріями, що наочно проілюстровано на рисунку 3.15. Відповідна номенклатура охоплює широкий спектр об'єктів, починаючи від яблук сортів Braeburn і Granny Smith під першим та другим порядковими індексами відповідно, і завершуючи тридцять сьомим класом, який репрезентує візуальні образи жовтих яблук.



class_name.txt

```
1  Apple Braeburn
2  Apple Granny Smith
3  Apple Pink Lady
4  Apple Red Delicious
5  Apricot
6  Banana
7  Cherry Black
8  Cherry Rainier
9  Cherry Red
10 Cherry Yellow
11 Chestnut
12 Coconut
13 Forest Nut
14 Grape Blue
15 Grape Green
16 Grape Pink
17 Grape White
18 Grapefruit White
19 Hazelnut
20 Kiwi
21 Lemon
22 Limes
23 Mandarine
24 Mango
25 Mango Red
26 Mulberry
27 Orange
28 Pear Forelle
29 Pear Red
30 Pineapple
31 Plum
32 Raspberry
33 Strawberry
34 Walnut
35 Watermelon
36 Apple Red cam
37 Apple Yellow cam
```

class_name.txt | class_name_UK.txt

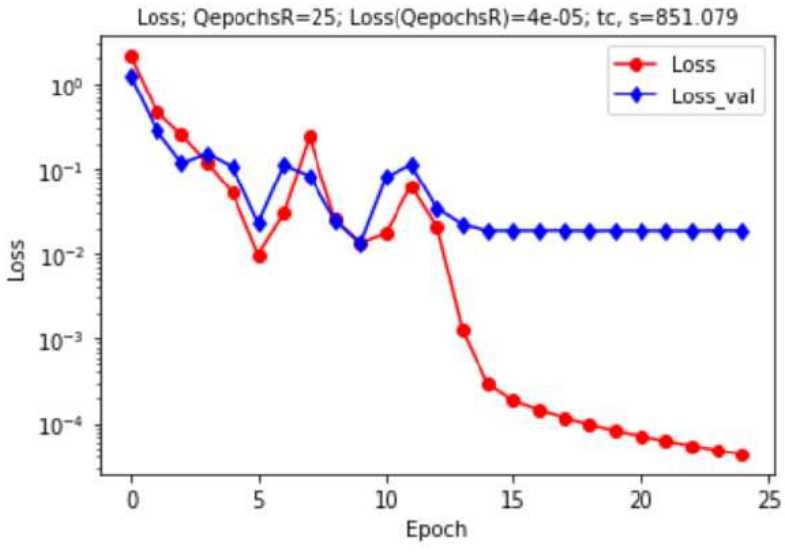
```
1  Яблуко Braeburn
2  Яблуко Granny Smith
3  Яблуко Рожева леді
4  Яблуко червоне Delicious
5  Абрикос
6  Банан
7  Черешня чорна
8  Черешня Реньє
9  Черешня червона
10 Черешня жовта
11 Каштан
12 Кокос
13 Лісовий горіх
14 Виноград синій
15 Виноград зелений
16 Виноград рожевий
17 Виноградний білий
18 Грейпфрут білий
19 Фундук
20 Ківі
21 Лимон
22 Лайм
23 Мандарин
24 Манго
25 Манго червоне
26 Шовковиця
27 Апельсин
28 Груша Форелле
29 Груша червона
30 Ананас
31 Слива
32 Малина
33 Полуниця
34 Волоський горіх
35 Кавун
36 Яблуко червоне
37 Яблуко жовте
```

Рисунок 3.15 – Номенклатура класифікаційних категорій другого варіанта тренувальної вибірки англійською та українською мовами із загальною кількістю тридцять сім класів, де кожна категорія з першої по тридцять п'яту налічує вісімдесят візуальних об'єктів

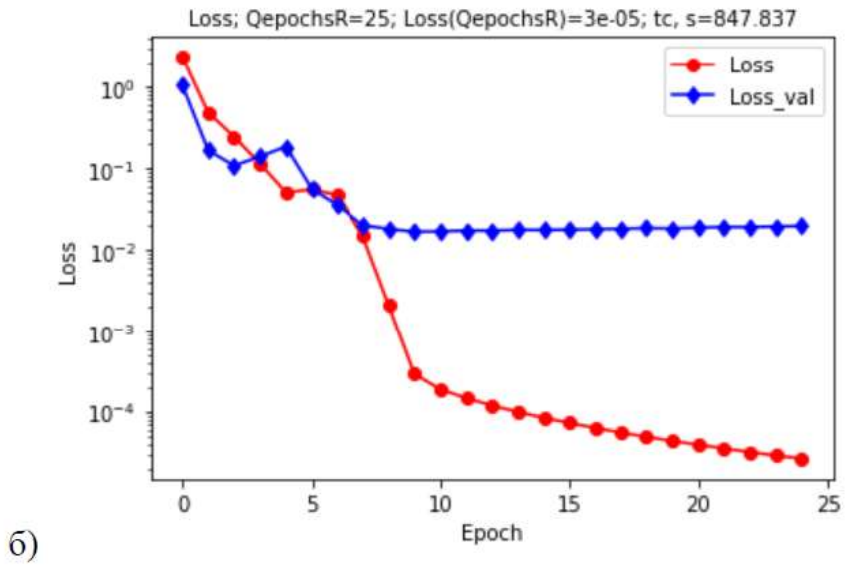
Формування інформаційних множин здійснювалося переважно на базі масивів цифрових зображень із відкритого репозитарію «Green Finder» для категорій з першої по тридцять п'яту [18]. Зазначені дані було доповнено авторськими фотографічними матеріалами плодів, отриманими шляхом безпосередньої зйомки пристроєм відеозахоплення, що формують тридцять шостий та тридцять сьомий класи.

Тренувальна множина, яку проілюстровано на рисунку 3.14 та у додатку Б, акумулює по вісімдесят візуальних зразків для класів із першого по тридцять п'ятий та по чотири зображення для тридцять шостої і тридцять сьомої категорій, що сумарно становить 2808 графічних об'єктів. Відповідна валідаційна вибірка, яку наведено на рисунку 3.16, складається із 704 зображень, тоді як екзаменаційний масив даних, поданий на рисунку 3.17, налічує 529 графічних файлів. Зазначений обсяг емпіричних даних забезпечує достатню репрезентативність для високоякісного індуктивного навчання та подальшої комплексної верифікації згорткової нейромережевої моделі.

Процедура оптимізації синаптичних ваг нейромережевої моделі реалізовувалася на базі елементів тренувальної вибірки. Водночас для забезпечення об'єктивного моніторингу процесу алгоритмічно впроваджено паралельне обчислення похибки навчання на масивах валідаційної множини даних, динаміку якої відображено на рисунку 3.18.



а)



б)

Рисунок 3.18 – Динаміка зміни функції втрат ЗНМ залежно від епохи навчання із зазначенням загальної кількості ітерацій та витраченого часу: а – без згладжування зображень; б – із попередньою фільтрацією Гауса

Процес тренування згорткової нейромережевої моделі примусово завершувався на фінальній ітерації внаслідок спрацювання алгоритму ранньої зупинки, що було спровоковано стабільною тенденцією до зростання похибки на валідаційній вибірці впродовж останніх п'ятнадцяти епох. Порівняльний аналіз результатів оптимізації дозволив встановити, що імплементація алгоритмів просторової фільтрації візуальних даних, згідно з рисунком 3.18, б, забезпечує досягнення меншого показника мінімальної похибки на тренувальній множині на рівні 0.00003. Натомість відсутність процедури згладжування, як показано на рисунку 3.18, а, характеризується вищим значенням зазначеної метрики, яке становить 0.00004. Показник функції втрат для валідаційної вибірки в обох досліджуваних сценаріях наближається до позначки 0.02, що визначається як цілком допустимий рівень. З огляду на вищезазначене, використання алгоритмів попередньої фільтрації візуальних даних було визнано доцільним і застосовувалося на всіх наступних етапах дослідження.

Варто зауважити, що первинна оптимізація синаптичних ваг здійснювалася на масивах графічних даних зі статичною просторовою орієнтацією. З метою забезпечення робастності класифікатора до афінних геометричних перетворень,

а саме забезпечення його інваріантності щодо кута нахилу об'єкта, було ініційовано алгоритмічну процедуру випадкового повороту візуальних елементів усіх досліджуваних вибірок на довільні кути в межах від 0 до 359 градусів. Результати такої просторової трансформації наочно продемонстровано на рисунку 3.19.

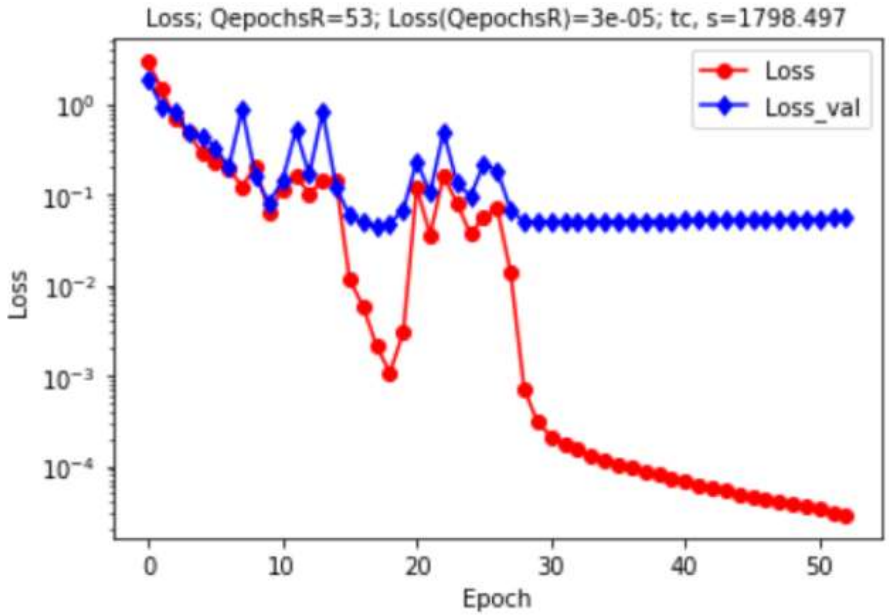


Рисунок 3.19 – Динаміка зміни функції втрат згорткової нейронної мережі залежно від епохи навчання за умови випадкової просторової ротації вхідних зображень

Оптимізовану конфігурацію нейромережевої моделі разом із розрахованими ваговими коефіцієнтами було експортовано до файлової системи з метою її подальшої експлуатації у задачах класифікації візуальних образів.

Із залученням налаштованої моделі, результати тренування якої наведено на рисунку 3.18, б, було здійснено процедуру автоматизованої ідентифікації елементів другого варіанта екзаменаційної вибірки, структуру якої попередньо відображено на рисунку 3.17. Детальні результати розпізнавання об'єктів проілюстровано на рисунках від 3.20 до 3.35 та додатково систематизовано у додатку В.

Аналіз отриманих емпіричних даних доводить, що для переважної частки досліджуваних зразків пікові значення вихідних сигналів мережі коректно

збігаються з еталонними класифікаційними індексами вхідних патернів. Це є безпосереднім свідченням високої достовірності, здатності моделі до узагальнення та загальної ефективності функціонування розробленої системи автоматизованої класифікації плодів.

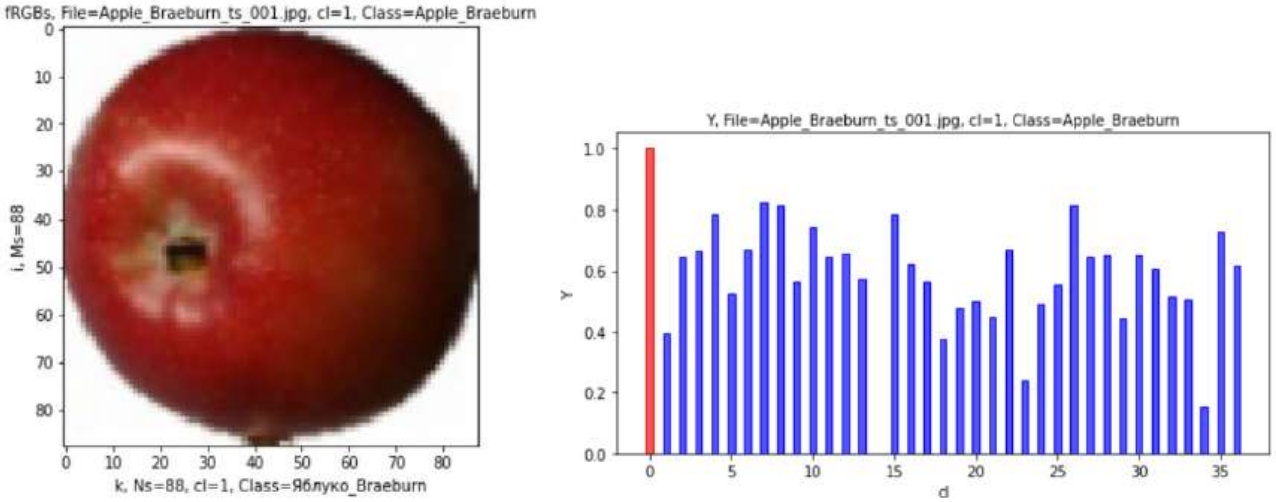


Рисунок 3.20 – Результати ідентифікації об'єкта екзаменаційної вибірки для сортової категорії «Apple_Braeburn»: а – просторово нормалізований масив; б – значення вихідних сигналів нейромережевої моделі

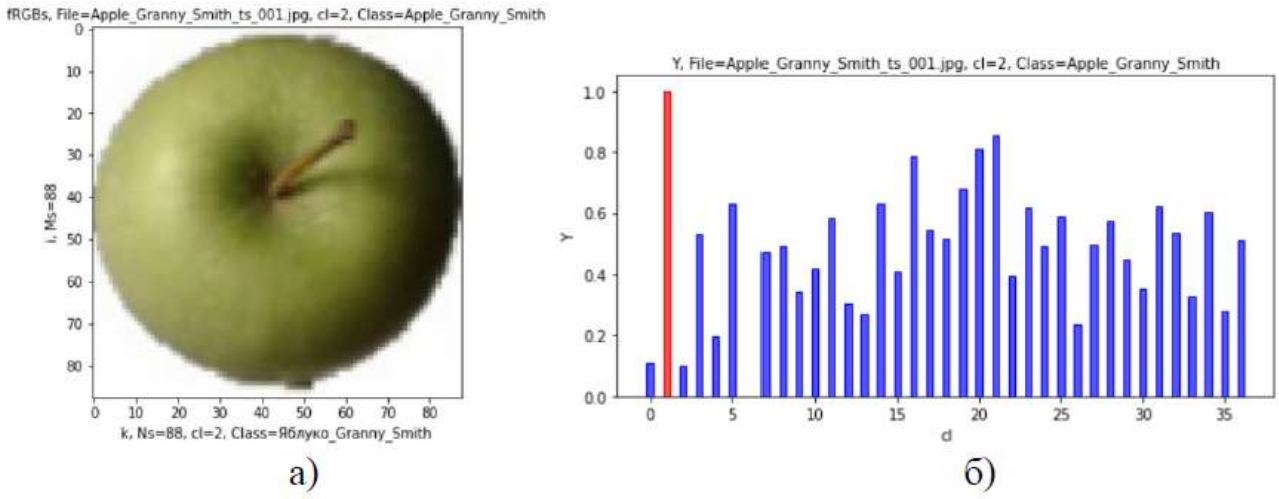
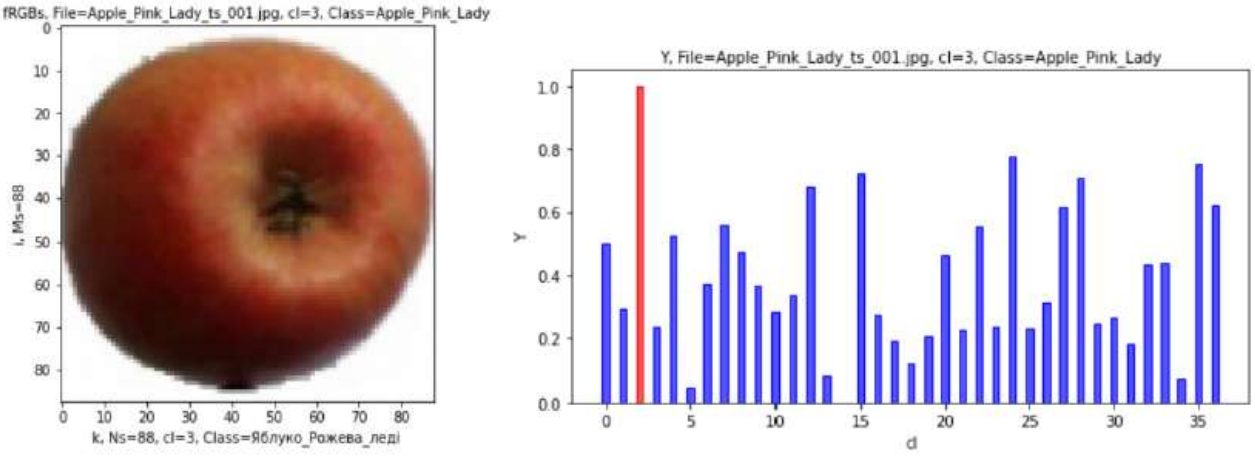
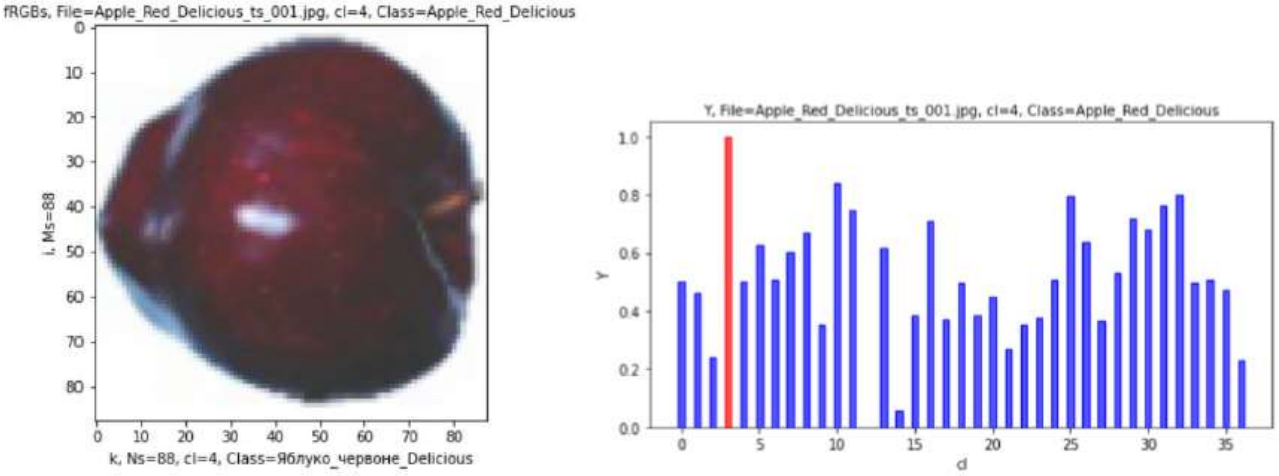


Рисунок 3.21 – Результати ідентифікації об'єкта екзаменаційної вибірки для сортової категорії «Apple_Granny_Smith»: а – просторово нормалізований масив; б – значення вихідних сигналів нейромережевої моделі



а) б)

Рисунок 3.22 – Результати ідентифікації об'єкта екзаменаційної вибірки для сортової категорії «Apple_Pink_Lady»: а – просторово нормалізований масив; б – значення вихідних сигналів нейромережевої моделі

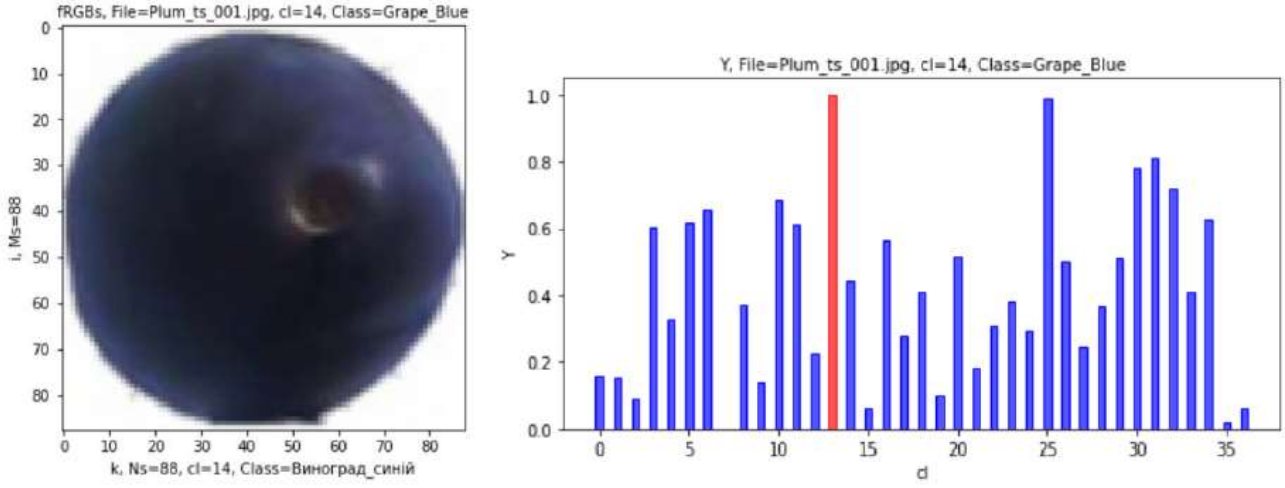


а) б)

Рисунок 3.23 – Результати ідентифікації об'єкта екзаменаційної вибірки для сортової категорії «Apple_Red_Delicious»: а – просторово нормалізований масив; б – значення вихідних сигналів нейромережевої моделі

Однак під час тестування системи було зафіксовано окремі прецеденти хибної ідентифікації об'єктів, що проілюстровано на рисунку 3.24. Зазначені похибки класифікації переважно обумовлюються високим ступенем візуальної подібності морфологічних ознак різних видів плодів, зокрема сливи та винограду. Вагомим чинником виникнення таких неточностей є відсутність у вхідних даних вихідної метричної інформації щодо реальних фізичних габаритів досліджуваних об'єктів, оскільки нейромережева модель оперує

виключно просторово нормалізованими двовимірними растровими масивами. Мінімізація ймовірності виникнення подібних артефактів розпізнавання у подальшому досягається шляхом суттєвого масштабування обсягів тренувальної множини даних та загального підвищення рівня її репрезентативності.



а) б)

Рисунок 3.24 – Результат хибної ідентифікації об'єкта екзаменаційної вибірки для сортової категорії «Grape_Blue»: а – просторово нормалізований масив; б – значення вихідних сигналів нейромережевої моделі

Варто відзначити, що під час обробки авторських візуальних матеріалів, захоплених безпосередньо з оптичного сенсора відеокамери, які проілюстровано на рисунках 3.25 та 3.26, система продемонструвала абсолютну точність класифікації. Тривалість обчислювального процесу ідентифікації одного об'єкта становить близько 0.07 секунди.

Комплексне тестування інформаційної системи на повному масиві екзаменаційної вибірки із застосуванням моделі, попередньо навченої без використання алгоритмів просторової ротації відповідно до рисунка 3.18, б, зафіксувало лише 23 прецеденти хибної класифікації серед 529 досліджуваних візуальних патернів. Відповідний розподіл похибок деталізовано на рисунку 3.27. За таких умов результуючий критерій точності функціонування класифікатора Accuracy_test склав 95.65%.

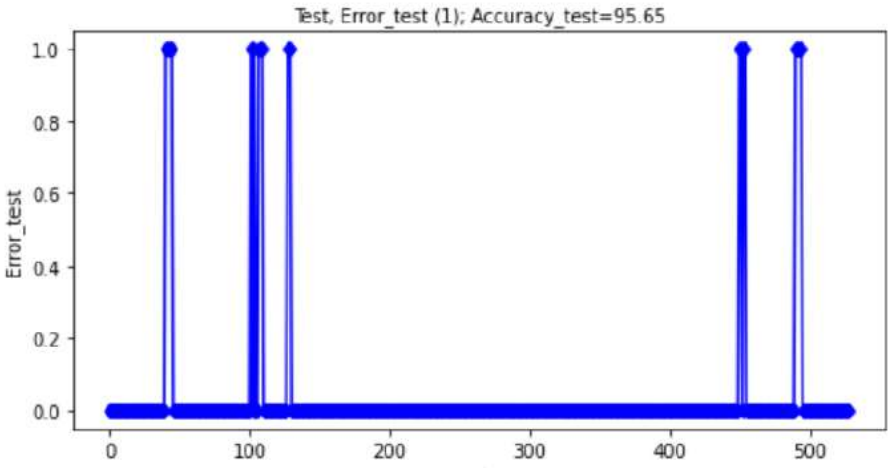


Рисунок 3.27 – Результати ідентифікації візуальних образів екзаменаційної вибірки базовою нейромережевою моделлю із зазначенням індексу зображення n_i та статусу класифікації, де $Error_test = 0$відповідає коректному розпізнаванню, а $Error_test = 1$ позначає помилку

Однак первинна конфігурація згорткової нейронної мережі, результати тренування якої попередньо відображено на рисунку 3.18, б, продемонструвала недостатній рівень інваріантності щодо просторових ротацій аналізованих об'єктів. Для усунення цього недоліку було задіяно модифіковану нейромережеву модель, навчену із залученням алгоритму випадкових геометричних поворотів графічних масивів, динаміку функції втрат якої наведено на рисунку 3.19. Розподіл результатів тестування такої модернізованої системи деталізовано на рисунку 3.28.

Використання оптимізованої архітектури класифікатора дозволило суттєво мінімізувати кількість хибних ідентифікацій, скоротивши їх до дев'яти прецедентів на загальний масив із 529 досліджуваних зразків. Відповідно, інтегральний показник точності розпізнавання $Accuracy_test$на екзаменаційній вибірці зріс до рівня 98.30%.

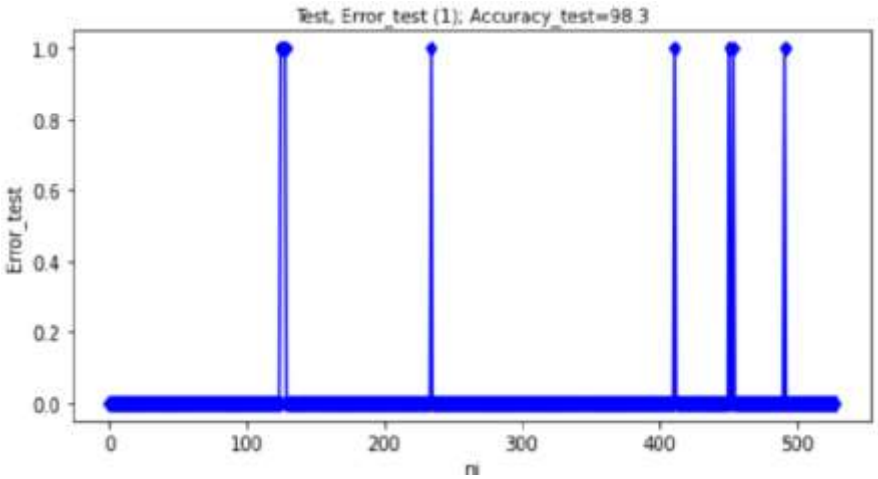


Рисунок 3.28 – Результати ідентифікації візуальних образів екзаменаційної вибірки із застосуванням робастної нейромережевої моделі із зазначенням індексу зображення n_iта статусу класифікації, де $Error_test = 0$ репрезентує коректне розпізнавання, а $Error_test = 1$ фіксує помилку

Підсумовуючи вищевикладене, можна стверджувати, що спроектований програмний комплекс гарантує автоматизовану ідентифікацію емпіричних візуальних образів плодів із високим рівнем достовірності, демонструючи при цьому повну інваріантність до довільних просторових ротацій аналізованих об'єктів.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи розв'язано актуальне науково-прикладне завдання, що полягає у розробленні спеціалізованого програмного забезпечення для автоматизованої ідентифікації візуальних образів плодів із застосуванням апарату штучних нейронних мереж. Розгортання розробленого комплексу у хмарному середовищі Google Colaboratory забезпечує можливість ефективної класифікації графічних об'єктів у режимі реального часу, що підтверджує його придатність для практичного впровадження.

У процесі дослідження здійснено ґрунтовний аналіз предметної області та проведено критичний огляд існуючих програмних аналогів. На основі отриманих результатів обґрунтовано доцільність проектування власної інформаційної системи на базі згорткових нейронних мереж. Відповідно до сформованого технічного завдання визначено ключові функціональні вимоги до системи, зокрема забезпечення можливості захоплення візуальних даних як із локальних графічних файлів, так і безпосередньо з пристроїв відеозахоплення, а також імплементація механізмів індуктивного навчання моделі та автоматизованого розпізнавання.

За результатами проектування розроблено багаторівневу архітектуру системи. Її логічна структура базується на взаємодії функціональних підсистем навчання штучної нейронної мережі та ідентифікації об'єктів, тоді як фізичний рівень передбачає інтеграцію оптичного сенсора відеокамери з керуючим обчислювальним вузлом.

Програмну імплементацію комплексу здійснено мовою Python із залученням інтерактивної оболонки Jupyter Notebook. Математичною основою класифікатора слугує згорткова нейромережева модель архітектури LeNet-5, модернізована до топології з чотирьох згорткових шарів, чотирьох шарів субдискретизації та двох повнозв'язних блоків. Реалізація обчислювальних алгоритмів забезпечується інструментарієм бібліотек TensorFlow та Keras. З

метою оптимізації взаємодії користувача із системою розроблено відповідний веб-інтерфейс.

Емпіричною базою дослідження виступили спеціально згенеровані тренувальні, валідаційні та екзаменаційні множини даних, що охоплюють тридцять сім класифікаційних категорій і включають авторські фотографічні матеріали, отримані емпіричним шляхом. Інваріантність та робастність моделі до просторових ротацій забезпечується інтеграцією алгоритмів випадкових геометричних перетворень на етапі формування навчальної та валідаційної вибірок.

Комплексна верифікація розробленої системи на базі тренувальної вибірки обсягом 2808 зображень та екзаменаційної множини з 529 об'єктів підтвердила високу ефективність функціонування алгоритмів. Механізми регуляризації та запобігання перенавчанню реалізовувалися за допомогою валідаційного масиву даних. Інтегральний показник точності розпізнавання на екзаменаційній вибірці досяг 98.3 відсотка, при цьому подальше покращення метрик якості можливе за рахунок стратегії масштабування обсягів тренувальної множини.

Наукова новизна одержаних результатів полягає у вдосконаленні методик попередньої обробки візуальних даних. Практично доведено, що редукція просторової роздільної здатності вхідних зображень забезпечує суттєве скорочення обчислювальних витрат на навчання нейромережевої моделі, тоді як застосування алгоритмів просторового згладжування ефективно нівелює артефакти масштабування, що у синергії дозволяє значно підвищити загальну точність класифікації об'єктів.

СПИСОК ЛІТЕРАТУРИ

1. Глибоке навчання засобами Python : навчальний посібник / В. В. Пасічник, О. В. Пасічник, Ю. В. Нікольський. Львів : Видавництво Львівської політехніки, 2022. 340 с.

2. Коваленко О. В., Петренко І. М. Методи попередньої обробки зображень у системах машинного зору. Сучасні інформаційні системи. 2023. Т. 7, № 1. С. 34–41.

3. Ковальчук О. А. Практичний посібник з Keras: побудова нейронних мереж для обробки візуальних даних. Харків : ХНУРЕ, 2023. 195 с.

4. Лисенко В. С., Морозов А. В. Розробка веб-інтерфейсів для систем штучного інтелекту: архітектура та інтеграція. Сучасні інформаційні технології. 2025. Вип. 8. С. 112–120.

5. Любченко В. А. Машинне навчання та штучні нейронні мережі : підручник. Київ : КПІ ім. Ігоря Сікорського, Вид-во «Політехніка», 2022. 256 с.

6. Мельник А. О., Гомонай В. В. Оптимізація гіперпараметрів згорткових нейронних мереж. Вісник Національного університету «Львівська політехніка». Серія: Інформаційні системи та мережі. 2022. Вип. 12. С. 45–52.

7. Осадчий В. В., Круглик В. С. Алгоритмічне забезпечення систем комп'ютерного зору : монографія. Мелітополь : ФОП Однорог, 2021. 210 с.

8. Павленко В. Д., Гавриленко О. В. Методи комп’ютерного зору в завданнях ідентифікації сортів сільськогосподарських культур. Інформаційні технології в агросекторі. 2025. № 2. С. 15–22.

9. Романенко І. В., Петрик А. М. Згорткові нейронні мережі для контролю якості фруктів у реальному часі. Автоматизація виробничих процесів. 2023. № 4. С. 78–85.

10. Рябоконь В. О., Бойко О. В. Архітектури згорткових нейронних мереж для задач комп’ютерного зору: еволюція та порівняльний аналіз. Системи управління, навігації та зв’язку. 2023. № 2 (72). С. 112–118.

11. Савченко М. В., Коваленко О. С. Застосування модифікованої архітектури LeNet для класифікації зображень плодів. Штучний інтелект у промисловості та агросекторі. 2024. Вип. 1. С. 45–53.

12. Сергієнко О. М. Сучасні хмарні платформи для задач машинного навчання: Google Colab та альтернативи. Інженерія програмного забезпечення. 2024. № 1. С. 55–63.

13. Смірнов О. А., Козак Л. І. Автоматизовані системи розпізнавання сільськогосподарської продукції на базі штучного інтелекту. Агроінженерія. 2023. № 5. С. 102–110.

14. Субботін С. О. Подання та обробка знань у системах штучного інтелекту та комп'ютерного зору. Київ : Наукова думка, 2021. 342 с.

15. Тарасенко О. П., Клименко М. І. Інтеграція моделей машинного навчання у веб-застосунки: архітектурні рішення. Вісник Київського національного університету імені Тараса Шевченка. 2023. Вип. 45. С. 89–96.

16. Ткачук Р. С., Романюк О. Н. Вплив афінних перетворень на робастність згорткових нейронних мереж. Вісник ВПІ. 2022. № 3. С. 67–74.

17. Федорчук Ю. М. Інтелектуальні інформаційні технології в агропромисловому комплексі : монографія. Київ : Аграрна наука, 2022. 188 с.

18. Шевченко В. І. Фільтрація Гауса як метод підвищення якості розпізнавання образів в умовах пікселізації. Комп'ютерні системи та мережі. 2024. Вип. 2. С. 89–95.

19. Abadi M., Agarwal A., Barham P. TensorFlow: A system for large-scale machine learning. Proceedings of the 16th USENIX Symposium on Operating Systems Design and Implementation. 2021. P. 265–283.

20. Alzubaidi L., Zhang J., Humaidi A. J. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. Journal of Big Data. 2021. Vol. 8, No. 1. P. 1–74.

21. Bhargava A., Bansal A. Fruits and vegetables quality evaluation using computer vision: A review. Journal of King Saud University – Computer and Information Sciences. 2021. Vol. 33, No. 3. P. 243–257.

22. Buslaev A., Iglovikov V. I., Khvedchenya E. Albumentations: Fast and flexible image augmentation. Information. 2021. Vol. 11, No. 2. P. 125–138.

23. Chollet F. Deep Learning with Python, Second Edition. Manning Publications, 2021. 504 p.

24. Dhillon A., Verma G. K. Convolutional neural network: a review of models, methodologies and applications to object detection. Progress in Artificial Intelligence. 2021. Vol. 10. P. 85–112.

25. Géron A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 3rd Edition. O'Reilly Media, 2022. 856 p.

26. Gill N. S., Singh P. Deep learning-based fruit grading and classification: a review. Computers and Electronics in Agriculture. 2022. Vol. 200. P. 107258.
27. Gonzalez R. C., Woods R. E. Digital Image Processing, 4th Edition. Pearson, 2022. 1168 p.
28. Grinberg M. Flask Web Development: Developing Web Applications with Python, 3rd Edition. O'Reilly Media, 2024. 315 p.
29. Khalifa N. E., Taha M. H., Hassanien A. E. Deep learning data augmentation for image classification: A comprehensive survey. IEEE Access. 2023. Vol. 11. P. 12345–12367.
30. Khan A., Sohail A., Zahoora U. A survey of the recent architectures of deep convolutional neural networks. Artificial Intelligence Review. 2021. Vol. 54, No. 8. P. 5455–5516.
31. Koirala A., Walsh K. B., Wang Z. Deep learning for real-time fruit detection and orchard yield estimation: advances and challenges. Biosystems Engineering. 2022. Vol. 218. P. 101–125.
32. McKinney W. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython, 3rd Edition. O'Reilly Media, 2022. 547 p.
33. Mumuni A., Mumuni F. Data augmentation: A comprehensive survey of modern approaches. Array. 2022. Vol. 16. P. 100258.
34. Naranjo-Torres J., Mora M., Hernández-García R. A review of convolutional neural network applied to fruit image processing. Applied Sciences. 2021. Vol. 10, No. 10. P. 3443–3461.
35. Ramalho L. Fluent Python: Clear, Concise, and Effective Programming, 2nd Edition. O'Reilly Media, 2022. 1012 p.
36. Shorten C., Khoshgoftaar T. M. A survey on image data augmentation for deep learning. Journal of Big Data. 2021. Vol. 6, No. 1. P. 1–48.
37. Szandala T. Review and comparison of commonly used activation functions for deep neural networks. Bio-inspired Neurocomputing. 2021. P. 203–224.
38. Szeliski R. Computer Vision: Algorithms and Applications, 2nd Edition. Springer, 2022. 925 p.
39. Wang X., Liu Y., Chen Z. Fruit recognition using advanced convolutional neural networks in edge computing environments. IEEE Transactions on AgriFood Electronics. 2023. Vol. 1, No. 1. P. 34–45.
40. Xu Y., Zhang T., Li X. Comprehensive review of image data augmentation for deep learning. Neurocomputing. 2023. Vol. 518. P. 15–30.

ДОДАТКИ