

ЧІП Святослав Андрійович

**Модуль розпізнавання якості фруктів з
використанням глибокого навчання/Fruit quality
recognition module using deep learning**

спеціальність: 122 - Комп'ютерні науки
освітньо-професійна програма - Комп'ютерні науки

Кваліфікаційна робота

Виконав студент групи КН-42
С. А. Чіп

Науковий керівник:
к.і.н., Г. В. Сапожник

Кваліфікаційну роботу
допущено до захисту:

" ____ " _____ 20 ____ р.

Завідувач кафедри
_____ **М. П. Комар**

ТЕРНОПІЛЬ - 2024

Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Освітній ступінь «бакалавр»
спеціальність 122 – Комп'ютерні науки
освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ
Завідувач кафедри
М.П. Комар
«_____» _____ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
ЧПУ Святославу Андрійовичу
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: Модуль розпізнавання якості фруктів з використанням глибокого навчання/Fruit quality recognition module using deep learning

керівник роботи Сапожник Григорій Вікторович, к.іст.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від 12 грудня 2023 р. № 753.

2. Строк подання студентом закінченої кваліфікаційної роботи 15 травня 2024 р.

3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити:

- Огляд теоретичних основ і практичних методів класифікації якості фруктів.

- Аналіз наукових підходів та існуючих технологічних рішень у галузі визначення якості агропромислової продукції.

- Огляд теоретичних основ і практичних методів класифікації якості фруктів.

- Аналіз наукових підходів та існуючих технологічних рішень у галузі визначення якості агропромислової продукції.

- Огляд теоретичних основ і практичних методів класифікації якості фруктів.

- Аналіз наукових підходів та існуючих технологічних рішень у галузі визначення якості агропромислової продукції.

5. Перелік графічного матеріалу в роботі:

- Архітектура модуля розпізнавання якості фруктів з використанням глибокого навчання

- Алгоритм розпізнавання якості фруктів з використанням глибокого навчання

- графіки з результатами експериментальних досліджень.

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 12 грудня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Затвердження теми кваліфікаційної роботи, ознайомлення з літературними джерелами та складання плану роботи.	до 01.01. 2024 р.	
2	Написання 1 розділу кваліфікаційної роботи	до 01.03. 2024 р.	
3	Написання 2 розділу кваліфікаційної роботи	до 01.04.2024 р.	
4	Написання 3 розділу кваліфікаційної роботи	до 01.05. 2024 р.	
5	Представлення попереднього варіанту кваліфікаційної роботи, перевірка та внесення змін керівником	до 15.05.2024 р.	
6	Опрацювання зауважень та представлення завершеного варіанту кваліфікаційної роботи. Підготовка супроводжуючих документів.	до 20.05.2024 р.	
7	Перевірка кваліфікаційної роботи на оригінальність тексту у системі «Unichек».	до 10.06.2024 р.	
8	Оформлення кваліфікаційної роботи та отримання допуску до захисту	до 14.06.2024 р.	
9	Подання кваліфікаційної роботи до захисту на засіданні атестаційної комісії.	до 14.06. 2024 р.	

Студент _____ С. А. Чіп
(підпис) (прізвище та ініціали)

Керівник кваліфікаційної роботи _____ Г.В. Сапожник
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота на тему «Модуль розпізнавання якості фруктів з використанням глибокого навчання» на здобуття освітнього ступеня «бакалавр» зі спеціальності 122 «Комп'ютерні науки» освітньої програми «Комп'ютерні науки» написана обсягом в 39 сторінок і містить 13 ілюстрацій, 2 таблиці та 14 використаних джерел.

Метою роботи є розробка та валідація модуля розпізнавання якості фруктів з використанням глибокого навчання.

Методами розроблення обрано метод аналізу (для дослідження існуючих підходів до класифікації якості фруктів), метод синтезу (для поєднання переваг існуючих методів), методи моделювання (для представлення та дослідження процесів класифікації якості фруктів), метод порівняльного аналізу (для оцінювання адекватності моделі класифікації якості фруктів).

Внаслідок виконання роботи обґрунтовано раціональний підхід до розроблення моделей класифікації якості фруктів та розроблено програмний засіб автоматизації процесу класифікації якості фруктів, який дозволяє створювати і досліджувати моделі класифікації якості фруктів.

Результати дослідження можуть бути використані в науково-дослідних установах і підрозділах підприємств, що займаються розробленням моделей класифікації якості фруктів.

Ключові слова: РОЗПІЗНАВАННЯ ЯКОСТІ ФРУКТІВ, НЕЙРОННІ МЕРЕЖІ, ГЛИБИННЕ НАВЧАННЯ, КОМП'ЮТЕРНИЙ ЗІР, МОДЕЛЬ АДЕКВАТНОСТІ.

ANNOTATION

Qualification work on the topic «Fruit quality recognition module using deep learning» for Bachelor's degree on speciality 122 «Computer Science» educational and professional program «Computer Science» is written on 39 pages and it contains 13 figures, 2 tables, and 14 sources.

The purpose of the work is to develop and validate a fruit quality recognition module using deep learning.

Research methods include analysis (to study existing approaches to fruit quality classification), synthesis (to combine the advantages of existing methods), modeling (to represent and study fruit quality classification processes), and comparative analysis (to evaluate the adequacy of the fruit quality classification model).

As a result of the work, a rational approach to the development of fruit quality classification models was substantiated, and a software tool for automating the fruit quality classification process was developed, allowing the creation and research of fruit quality classification models.

The research results can be used in research institutions and enterprise departments involved in the development of fruit quality classification models.

Keywords: FRUIT QUALITY RECOGNITION, NEURAL NETWORKS, DEEP LEARNING, COMPUTER VISION, MODEL ADEQUACY.

ЗМІСТ

Вступ	7
1 Аналіз предметної області і постановка задачі дослідження	10
1.1 Огляд предметної області.....	10
1.2 Огляд існуючих рішень	12
1.3 Вибір перспективного шляху і постановка задачі дослідження	14
2 Алгоритмічне та інформаційне забезпечення розпізнавання якості фруктів з використанням глибокого навчання.....	17
2.1 Архітектура модуля розпізнавання якості фруктів з використанням глибокого навчання.....	17
2.2 Архітектура глибокої нейронної мережі ResNet	19
2.3 Алгоритм розпізнавання якості фруктів з використанням глибокого навчання	20
3 Програмно-технологічне забезпечення	25
3.1 Підготовка даних.....	25
3.2 Навчання моделі.....	28
3.3 Отримані результати.....	30
3.4 Оптимізація моделі	33
3.5 Оцінка отриманих результатів.....	35
Висновки	39
Список використаних джерел	42
Додаток А Псевдокод алгоритму	44
Додаток Б Код для реалізації	45
Додаток В Апробація отриманих результатів.....	49

ВСТУП

Актуальність досліджень у сфері автоматизованої класифікації якості фруктів є незаперечною в умовах сучасного агропромислового виробництва та продовольчої промисловості. Зростання світового населення та збільшення попиту на високоякісні продовольчі товари спонукають до інновацій у методах обробки та контролю якості продуктів. Традиційні методи класифікації, які часто залежать від суб'єктивної оцінки та ручної праці, виявляються неефективними, особливо в умовах необхідності швидкої обробки великих обсягів продукції. Впровадження автоматизованих систем, які використовують машинне та глибоке навчання, може значно покращити якість та швидкість процесів оцінювання та сортування фруктів.

Технологічний прогрес у сфері комп'ютерного зору та штучного інтелекту відкриває нові можливості для розв'язання цих проблем. Сучасні алгоритми глибокого навчання, такі як ResNet і CNN, демонструють вражаючу здатність аналізувати та класифікувати зображення з високою точністю, що є ідеальним для задач оцінювання якості фруктів. Використання цих технологій дозволяє автоматизувати процеси, мінімізувати людський фактор, знизити витрати та збільшити продуктивність виробництва.

Однак, попри наявність значних досягнень в області машинного навчання, багато поточних рішень стикаються з проблемами універсальності та адаптації до різноманітних умов роботи. Більшість систем розроблені для специфічних типів фруктів або певних умов експлуатації, що обмежує їх застосування на загальнопромисловому рівні. Тому існує критична потреба в розробці більш гнучких, адаптивних моделей, які можуть бути ефективно налаштовані та використані для широкого спектру продукції.

У цьому контексті, наше дослідження, спрямоване на використання архітектури ResNet34 для класифікації якості різних типів фруктів, є вкрай актуальним. Ми прагнемо не тільки покращити загальну точність

класифікації, але й забезпечити високий рівень адаптивності та ефективності в реальних виробничих умовах, що може внести значний вклад у технологічний розвиток агропромислового сектору.

Метою роботи є розробка та валідація модуля розпізнавання якості фруктів з використанням глибокого навчання. Для досягнення цієї мети необхідно вирішити наступні завдання:

1. Огляд теоретичних основ і практичних методів класифікації якості фруктів.
2. Аналіз наукових підходів та існуючих технологічних рішень у галузі визначення якості агропромислової продукції.
3. Огляд теоретичних основ і практичних методів класифікації якості фруктів.
4. Аналіз наукових підходів та існуючих технологічних рішень у галузі визначення якості агропромислової продукції.
5. Огляд теоретичних основ і практичних методів класифікації якості фруктів.
6. Аналіз наукових підходів та існуючих технологічних рішень у галузі визначення якості агропромислової продукції.

Об'єктом дослідження є процес автоматизованої класифікації якості фруктів з використанням методів глибокого навчання. Цей процес включає збір та аналіз візуальних даних фруктів, їх обробку за допомогою передових алгоритмів машинного зору та глибоких нейронних мереж, які спроможні точно визначати різні аспекти якості, включно зі ступенем зрілості, наявністю дефектів та загальним зовнішнім виглядом.

Предметом дослідження є методи глибокого навчання застосовані до процесу класифікації якості фруктів. В деталях, дослідження зосереджується на вивченні та оптимізації архітектури глибоких нейронних мереж, зокрема ResNet34, для ефективного аналізу візуальних даних і точного визначення різних якісних характеристик фруктів.

Методи дослідження включають використання глибоких нейронних мереж, зокрема модифікації архітектури ResNet34, для аналізу та класифікації візуальних даних фруктів, їх передобробку, аугментацію зображень, та евалюацію результатів за допомогою статистичних методів порівняння та валідації.

Практичне значення даного дослідження полягає у розробці автоматизованої системи для класифікації якості фруктів, що значно підвищує ефективність та точність виробничих процесів у агропромисловому секторі. Це дозволяє агробізнесам зменшити втрати продукції, оптимізувати логістику та підвищити загальну якість продуктів, які постачаються на ринок. Впровадження таких технологій може також сприяти дотриманню міжнародних стандартів безпеки та якості, забезпечуючи компаніям конкурентні переваги на глобальному ринку. В результаті, дослідження має значний вплив на підвищення прибутковості та сталого розвитку в агропромисловому секторі.

Структура та обсяг роботи. Кваліфікаційна робота складається з вступу, трьох розділів, висновків та списку використаних джерел. Загальний обсяг роботи складає 39 сторінок тексту, включаючи 13 рисунків та 2 таблиці. У списку використаних джерел наведено 14 найменувань, які займають 2 сторінки.

Апробація результатів дослідження. Основні теоретичні положення роботи й практичні результати дослідження доповідалися й обговорювалися на V Всеукраїнської мультидисциплінарної студентської наукової конференції «Науковий простір: аналіз, сучасний стан, тренди та перспективи», яка відбулася 17 травня 2024 року у місті Київ, Україна.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Огляд предметної області

Сучасні технології в області комп'ютерного зору значною мірою трансформували багато галузей промисловості, зокрема агропромисловий сектор. Застосування методів машинного та глибокого навчання для класифікації якості фруктів і овочів відіграє важливу роль у підвищенні продуктивності, контролі якості та зменшенні втрат врожаю. Розвиток точних і надійних систем автоматичного розпізнавання не тільки сприяє оптимізації ланцюжків поставок, але й підвищує загальну безпеку та якість харчових продуктів.

Історично методи класифікації якості фруктів базувались на ручній інспекції та суб'єктивних оцінках. Однак, з появою цифрових зображень та комп'ютерного аналізу, почали розвиватися автоматизовані системи, здатні з високою точністю визначати якість продукції. Сучасні системи використовують глибокі нейронні мережі, такі як конволюційні нейронні мережі (CNN), для аналізу візуальних даних, забезпечуючи об'єктивність і повторюваність оцінок.

Ключовими аспектами застосування комп'ютерного зору в класифікації якості фруктів є здатність системи до збору та обробки великих обсягів візуальних даних. Сучасні технології дозволяють виявляти різноманітні дефекти на поверхні фруктів, такі як плями, подряпини, неправильні форми або стадії зрілості. Значення цих технологій зростає з підвищенням потреби в автоматизації великих агропромислових підприємств та підвищенням вимог до якості продуктів.

Дослідження у галузі комп'ютерного зору продовжують розширювати можливості класифікації якості фруктів. Зокрема, інтеграція з технологіями

штучного інтелекту та машинного навчання відкриває нові шляхи для підвищення точності та ефективності обробки даних. Впровадження систем автоматизованого контролю якості на всіх етапах виробництва і постачання може суттєво знизити втрати продукції та підвищити загальну якість продуктів на ринку.

Комп'ютерний зір використовується для автоматичного виявлення візуальних дефектів на фруктах, таких як плями, подряпини, вм'ятини та інші поверхневі недоліки. Застосування згорткових нейронних мереж (CNN) дозволяє системам глибокого навчання аналізувати зображення фруктів на предмет виявлення аномалій, які можуть свідчити про псування чи непридатність для продажу або споживання. Моделі тренуються на великих наборах даних з зображеннями фруктів, які містять різні типи дефектів, щоб забезпечити високу точність та спроможність до узагальнення в реальних умовах.

Оцінка стадії зрілості фруктів є ключовим фактором у визначенні їх готовності до споживання, обробки або транспортування. Системи комп'ютерного зору використовують різні візуальні характеристики, такі як колір шкірки, розмір і форма, для автоматичного визначення стадії зрілості. Наприклад, зелений колір бананів вказує на незрілість, тоді як яскраво-жовтий колір з коричневими плямами свідчить про високу зрілість. Алгоритми машинного навчання, зокрема нейронні мережі, можуть бути навчені класифікувати ці візуальні дані, що дозволяє автоматизувати процес контролю якості.

Точне вимірювання розмірів та форми фруктів є важливим для класифікації, сортування та пакування. Комп'ютерний зір дозволяє автоматично оцінювати ці параметри за допомогою алгоритмів обробки зображень. Використовуючи техніки, такі як краєвий детектор або контурний аналіз, системи можуть точно визначати контури фрукта на зображенні та розраховувати їх геометричні характеристики, включно з довжиною,

шириною, периметром, і площею. Ці вимірювання можуть бути використані для встановлення стандартів якості, визначення відповідності фруктів певним розмірним категоріям, а також для планування упаковки та транспортування.

Розвиток технологій комп'ютерного зору і глибокого навчання сприяє значному підвищенню ефективності та точності процесів контролю якості у агропромисловості. Від виявлення дефектів до оцінки зрілості та точного вимірювання параметрів фруктів — ці технології забезпечують високу автоматизацію та сприяють зниженню витрат, збільшенню продуктивності та покращенню якості продуктів.

1.2 Огляд існуючих рішень

В контексті розробки та впровадження інноваційних рішень для класифікації якості фруктів, існує широкий спектр досліджень, що вивчають різноманітні підходи та технології. Останнім часом значну увагу привертають методи машинного та глибокого навчання, які демонструють значний потенціал у вирішенні задач автоматизації сортування та оцінювання якості фруктів. Таблиця 1.1 надає огляд існуючих рішень у цій області, підкреслюючи їхні переваги та недоліки. Важливо зазначити, що багато з цих підходів концентруються на використанні новітніх алгоритмів для точного виявлення та класифікації фруктів, однак часто не враховують практичні обмеження або деталізацію специфічних умов застосування. Цей огляд покликаний на підставі аналізу існуючих досліджень формулювати напрямки подальших робіт, які дозволять оптимізувати процеси оцінювання якості, розширити їх застосування та підвищити загальну точність класифікації.

Таблиця 1.1 – Огляд існуючих рішень

Назва дослідження	Переваги	Недоліки
Виявлення та класифікація стиглих фруктів за допомогою машинного навчання [1]	Використання Mask R-CNN для точного виявлення бананів.	Не вказані можливі обмеження або недоліки методу.

Оптимізація та класифікація фруктів за допомогою алгоритму машинного навчання [2]	Використання алгоритмів GA та SVM для оптимізації.	Обмежена інформація про точність та ефективність методів.
Класифікація форми полуниць за допомогою машинного навчання [3]	Висока точність класифікації форми полуниць.	Не описано можливі проблеми зі збором або обробкою даних.
Автоматизована система оцінювання фруктів: технічний огляд [4]	Огляд різних систем на основі обробки зображень.	Можлива нестача практичних порівнянь алгоритмів.
Визначення якості фруктів за допомогою обробки зображень, МН та глибокого навч. [5]	Огляд класифікації яблук з використанням глибокого навчання.	Оглядовий характер, без детального аналізу специфічних кейсів.
Техніки оцінки та сортування драконових фруктів з використанням МН [6]	Застосування CNN, ANN, та SVM для оцінки якості.	Обмеження відносно видів фруктів, які можуть бути оцінені.
Останні досягнення в детекції фруктів та їх класифікації з використанням ГН [7]	Глибокий аналіз використання "Fruit 360" для навчання моделі	Можливі виклики з загальною узагальненістю моделей.
Класифікація фруктів за допомогою глибокого навчання [8]	Використання великого набору даних (Fruits 360) для тренування моделей.	Обмеження у деталізації методології тренування моделей.
Автоматична класифікація фруктів за допомогою алгоритму випадкового лісу [9]	Застосування комплексної передобробки та вилучення ознак.	Відсутність порівняння з іншими алгоритмами класифікації.
Класифікація зображень фруктів за допомогою глибокого навчання [10]	Аналіз трьох різних моделей глибокого навчання для класифікації.	Недостатньо інформації про ефективність окремих моделей.
Класифікація фізалису за ступенем зрілості за допомогою машинного навчання [11]	Використання різних кольорових просторів для підвищення точності.	Специфічність для одного типу фрукту може обмежувати застосування.
Автоматична класифікація фруктів за допомогою глибокого навчання у промисловості [12]	Практичне застосування глибокого навчання у промислових умовах.	Можливі виклики із масштабуванням та адаптацією до різних умов.

Наше дослідження вирізняється застосуванням глибокої нейронної мережі з архітектурою ResNet34 для класифікації якості фруктів, де особлива увага приділяється оптимізації точності та ефективності моделі через ретельний вибір та налаштування параметрів навчання. На відміну від багатьох існуючих рішень, які часто обмежуються одним видом фруктів або не розглядають практичні аспекти впровадження в промислові умови, наше дослідження включає роботу з різними категоріями фруктів та розробку

рішення, призначеного для масштабування. Також, ми акцентуємо увагу на глибокому аналізі помилок класифікації, використовуючи візуалізацію топ-вtrat для вдосконалення моделі та підвищення її реалістичності і адаптивності до змінних умов роботи. Це дозволяє не тільки збільшити точність класифікації, але й забезпечити більш високий рівень надійності та узагальненості у реальних умовах експлуатації.

1.3 Вибір перспективного шляху і постановка задачі дослідження

Якість продукції в агропромисловому секторі безпосередньо впливає на здоров'я споживачів, економічну вигоду виробників та стабільність продовольчих ресурсів. Підвищення вимог до якості та безпеки продуктів харчування, зокрема фруктів, в умовах глобалізації ринків, вимагає вдосконалення технологій їх обробки та класифікації. Відтак, розробка ефективних автоматизованих систем, що здатні швидко та точно оцінювати якість фруктів, стає вкрай актуальною задачею.

Традиційні методи класифікації фруктів зазвичай залежать від ручної праці, що робить процес трудомістким, суб'єктивним та схильним до помилок. Це призводить до значних фінансових втрат через неправильне сортування та зниження продуктивності. Окрім того, недостатня швидкість та точність у визначенні ступеня зрілості чи виявленні вад можуть негативно вплинути на загальну якість продукції, що потрапляє на ринок.

Глибоке навчання відкриває нові можливості для розробки передових систем класифікації, які можуть автоматично аналізувати великі обсяги візуальних даних із високою точністю та об'єктивністю. Завдяки здатності мереж обробляти складні шаблони і візерунки, такі системи можуть значно покращити якість виробництва, зменшити відходи та оптимізувати процеси контролю якості. Це дозволяє досягти вищої однорідності продукції та забезпечити краще дотримання стандартів якості.

Інтеграція технологій машинного зору та глибокого навчання у системи контролю якості фруктів також сприяє адаптації до змінюваних глобальних стандартів безпеки харчових продуктів. Здатність швидко адаптуватися до нових вимог ринку та впроваджувати технологічні нововведення вирізняє лідерів індустрії, забезпечуючи їм конкурентні переваги. Розробка та впровадження моделей глибокого навчання для класифікації якості фруктів є кроком вперед до створення стійких і інноваційних агропромислових систем.

Метою цієї бакалаврської роботи є розробка та валідація модуля розпізнавання якості фруктів з використанням глибокого навчання. Для досягнення цієї мети необхідно вирішити наступні завдання:

1. Аналіз предметної області та постановка задачі дослідження: Вивчення сучасного стану технологій класифікації якості фруктів і визначення ключових проблем, які можуть бути вирішені за допомогою глибокого навчання.
2. Огляд існуючих рішень у сфері розпізнавання якості фруктів: Проаналізувати поточні наукові та практичні підходи до розпізнавання якості фруктів, включаючи традиційні методи і новітні розробки на основі машинного навчання.
3. Проектування алгоритмічного та інформаційного забезпечення системи: Спроекувати архітектуру модуля та вибрати архітектуру глибокої нейронної мережі, зокрема ResNet, для розпізнавання якості фруктів. Розробити алгоритм розпізнавання якості фруктів.
4. Розробка програмно-технологічного забезпечення: Виконати підготовку даних, навчання моделі, оцінку отриманих результатів та оптимізацію моделі. Забезпечити документацію всього процесу розробки та результатів експериментів.
5. Аналіз результатів та валідація системи: Провести тестування розробленої системи на реальних даних, аналізувати її ефективність та

точність. Запропонувати рекомендації щодо подальших напрямків дослідження та комерціалізації проекту..

Завдання 1-2 мають бути розглянуті в теоретичних розділах дипломної роботи, а завдання 3-5 – у практичних розділах, що охоплюють розробку, оцінку та тестування алгоритму.

2 АЛГОРИТМІЧНЕ ТА ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ РОЗПІЗНАВАННЯ ЯКОСТІ ФРУКТІВ З ВИКОРИСТАННЯМ ГЛИБОКОГО НАВЧАННЯ

2.1 Архітектура модуля розпізнавання якості фруктів з використанням глибокого навчання

Архітектура (рис.2.1) модуля розпізнавання якості фруктів, розроблена на базі глибокого навчання, розбивається на три основні складові частини, які оптимізують процес розробки та дозволяють зосередитися на специфічних задачах у кожному з етапів. Описана нижче архітектура поєднує в собі різноманітні компоненти, що взаємодіють між собою, щоб забезпечити ефективну обробку даних, тренування моделі, а також її оцінку і деплоймент.

1. Дані і підготовка. Цей модуль відповідає за збір та підготовку даних, які необхідні для подальшого тренування моделі глибокого навчання. Він включає наступні компоненти:

- Імпорт бібліотек: Використовується для завантаження всіх необхідних зовнішніх бібліотек, таких як Pandas, Numpy, Matplotlib, що використовуються для обробки та візуалізації даних.
- Імпорт фреймворків: Залучення фреймворків глибокого навчання, таких як TensorFlow, Keras або PyTorch, які надають необхідні інструменти та API для розробки згорткових нейронних мереж.
- Завантаження даних: Здійснення завантаження та першої обробки даних з зовнішніх джерел або локального сховища.
- Підготовка завантажувачів даних: Створення об'єктів для ітеративного доступу до даних під час тренування, що включає розділення на тренувальні та валідаційні набори.

- Трансформація та нормалізація: Застосування предобробки даних, таких як масштабування, нормалізація, аугментація, щоб забезпечити кращі умови для ефективного навчання.

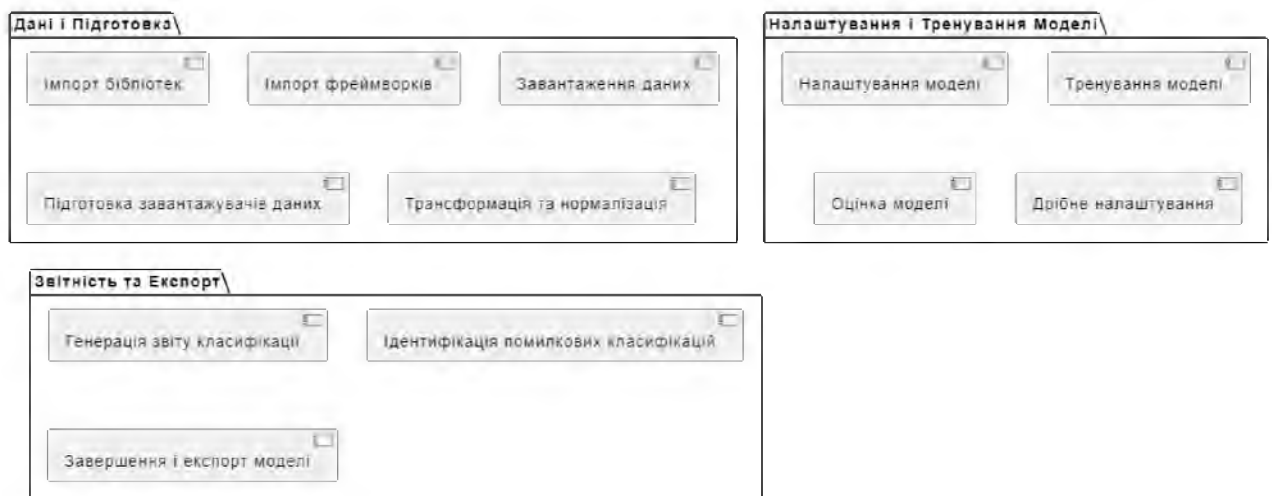


Рисунок 2.1. Архітектура модуля розпізнавання якості фруктів з використанням глибокого навчання

2. Налаштування і тренування моделі. Цей блок охоплює всі аспекти конфігурації та оптимізації моделі нейронної мережі:

- Налаштування моделі: Вибір архітектури моделі, такої як ResNet, VGG або інша згортова нейронна мережа, та її ініціалізація з використанням попередньо навчених ваг.
- Тренування моделі: Використання алгоритмів оптимізації для мінімізації помилок прогнозування на основі тренувальних даних.
- Оцінка моделі: Використання валідаційних наборів для перевірки ефективності моделі та її здатності до узагальнення.
- Дрібне налаштування: Додаткове тренування моделі з відкритими всіма шарами для доопрацювання та підвищення точності.

3. Звітність та експорт. Фінальний етап, який забезпечує аналіз результатів та їх представлення у вигляді звітів, а також експорт тренованої моделі для її подальшого використання:

- Генерація звіту класифікації: Вивчення та оцінка продуктивності моделі за допомогою метрик, таких як точність, відгук, F1-оцінка.
- Ідентифікація помилкових класифікацій: Аналіз причин неправильних прогнозів та їх вплив на загальну точність.
- Завершення і експорт моделі: Збереження моделі у форматі, зручному для деплоювання, разом із усіма необхідними метаданими та налаштуваннями.

Розроблена архітектура відображає інтегрований підхід до вирішення задачі класифікації фруктів, поєднуючи в собі передові технології обробки даних, машинного навчання та глибокого навчання.

2.2 Архітектура глибокої нейронної мережі ResNet

Архітектура ResNet (Residual Network) була представлена дослідниками з Microsoft Research у 2015 році та виграла змагання ImageNet того року завдяки своїй винятковій точності та ефективності у глибокому навчанні зображень. Основна інновація ResNet полягає у використанні так званих резидуальних блоків, які допомагають зменшити проблему зникнення градієнту, яка є значною перешкодою у тренуванні дуже глибоких нейронних мереж.

Резидуальні блоки є основною інновацією архітектури ResNet. Ці блоки вирішують проблему зникнення градієнту, яка стає дедалі більш критичною зі збільшенням глибини нейронної мережі.

У контексті нейронних мереж, резидуальний блок можна представити як блок, що містить один або декілька звичайних шарів разом із скіп-з'єднанням (skip connection), яке "перескакує" через ці шари. Якщо ввести позначення x для вхідних даних у резидуальний блок, а $F(x)$ для функції, що представляє послідовність шарів всередині блоку, вихід резидуального блоку визначається як:

$$H(x) = F(x) + x$$

де $H(x)$ — це вихідний сигнал резидуального блоку.

Функція $F(x)$ може бути представлена як послідовність операцій нейронної мережі, таких як згортки (convolutions), активаційні функції (ReLU, інші), пулінг (pooling) або нормалізації. Наприклад, у типовому резидуальному блоку для ResNet $F(x)$ може бути визначена так:

$$F(x) = \sigma(W_2 \sigma(W_1 x + b_1) + b_2)$$

де:

- W_1, W_2 — ваги шарів,
- b_1, b_2 — вектори зсуву,
- σ — активаційна функція (зазвичай ReLU).

Скіп-з'єднання у резидуальному блоку дозволяє вихідним даним x бути доданим напряму до виходу послідовності шарів, що ефективно усуває потенційну проблему зникнення градієнту під час тренування, оскільки дозволяє градієнту передаватися назад безпосередньо через ці з'єднання. Це особливо важливо у глибоких мережах, де стандартний потік градієнту через багато шарів може призводити до дуже малих градієнтів, що робить неможливим ефективне оновлення ваг.

З введенням скіп-з'єднань, мережа ефективно навчається адаптувати $F(x)$, контролюючи, як додаткові шари змінюють вхідні дані, замість спроби напряму апроксимувати $H(x)$. Це призводить до більш стабільного навчання і зменшення ризику перенавчання, оскільки мережа зосереджується на мінімізації лише залишкових відмінностей між входом та виходом.

2.3 Алгоритм розпізнавання якості фруктів з використанням глибокого навчання

У сучасному світі агротехнологій автоматизація процесів обробки та класифікації сільськогосподарської продукції стає все більш актуальною. Впровадження глибокого навчання для розпізнавання якості фруктів може

суттєво підвищити ефективність виробничих ліній та забезпечити вищу якість продукції. Цей алгоритм демонструє процес створення моделі з використанням бібліотеки FastAI, що класифікує фрукти на основі їх зображень.

Далі алгоритм (рис.2.2) розпізнавання якості фруктів з використанням глибокого навчання на базі навчального записника:

1. Налаштування середовища:

- Імпорт необхідних бібліотек, таких як pandas, numpy, seaborn, matplotlib тощо.
- Імпорт фреймворків машинного навчання і глибокого навчання, таких як FastAI, OpenCV та scikit-learn.
- Вимкнення попереджень для підтримки чистоти записника.

2. Завантаження даних:

- Визначення шляху до папки з набором даних, що містить зображення фруктів.
- Генерація списку усіх файлів зображень у наборі даних за допомогою шаблону glob.

3. Підготовка завантажувачів даних:

- Створення ImageDataLoaders з папки набору даних. Розділення даних на навчальний та валідаційний набори.
- Застосування трансформацій та нормалізації до зображень, щоб вони були придатні для навчання нейронної мережі.

4. Налаштування моделі:

- Ініціалізація згорткової нейронної мережі за допомогою vision_learner з попередньо навченою архітектурою ResNet34 від FastAI.
- Визначення каталогу, де будуть зберігатися ваги моделі.

5. Пошук оптимальної швидкості навчання: Використання методу lr_find для пошуку оптимальної швидкості навчання для тренування моделі.



Рисунок 2.2 - Алгоритм розпізнавання якості фруктів з використанням глибокого навчання

Цей метод допомагає уникнути надто повільного або надто швидкого навчання.

6. Тренування моделі:

- Тренування моделі за допомогою методу `fit_one_cycle` з раніше знайденою швидкістю навчання.
- Візуалізація графіків втрат навчання та валідації для оцінки навчання моделі.

7. Оцінка моделі:

- Використання моделі для роботи прогнозів на нових даних.
- Відображення результатів для візуалізації продуктивності моделі на валідаційному наборі.

8. Дрібне налаштування:

- Розморожування моделі для тренування всіх шарів (а не тільки кінцевих).
- Пошук нової оптимальної швидкості навчання для повністю розмороженої моделі.
- Продовження тренування (дрібне налаштування) моделі з новою швидкістю навчання для підвищення точності.

9. Генерація звіту класифікації:

- Прогнозування міток для валідаційного набору даних і їх порівняння з істинними мітками.
- Генерація звіту класифікації для оцінки продуктивності моделі з точки зору точності, відгуку та F1-оцінки для кожного класу.

10. Візуалізація матриці помилок: Візуалізація матриці помилок для оцінки продуктивності моделі по різних класах, вказуючи, де модель плутає одну мітку з іншою.

11. Ідентифікація помилкових класифікацій: Аналіз і відображення зображень з найбільшою кількістю помилок для розуміння потенційних причин помилок і для поліпшення моделі в наступних ітераціях.

12. Завершення і експорт моделі:

- Закінчення проекту збереженням навченої моделі та важливих метрик.
- За бажанням, надання посилання для завантаження або збереження моделі у директорії для розгортання.

Цей алгоритм надає детальний план будівництва моделі класифікації фруктів з використанням бібліотеки FastAI, від завантаження та підготовки даних до тренування, дрібного налаштування та оцінки моделі.

3 ПРОГРАМНО-ТЕХНОЛОГІЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Підготовка даних

Для роботи з даними використовувалися такі інструменти та бібліотеки, як pandas, NumPy, OpenCV та FastAI. Дані були завантажені з використанням Python бібліотеки Pathlib, що дозволило ефективно обробити зображення та підготувати їх до тренування моделі. Загальна кількість зображень склала 1655, що було підтверджено за допомогою функції виведення.

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import os
import random
import cv2
import sklearn
from PIL import Image
from pathlib import Path
from fastai.vision.all import *
from fastai.callback import *
from fastai.metrics import error_rate
from sklearn.metrics import classification_report

import warnings
warnings.filterwarnings('ignore')
```

У цьому дослідженні розглядається використання глибокої нейронної мережі з архітектурою ResNet для класифікації зображень фруктів на основі їхньої якості. Аналіз проводиться на великому наборі даних, що містить 1655 зображень, категоризованих за свіжістю та станом (наприклад, свіжі або зіпсовані персики, гранати та полуниці).

```
files = list(path.glob("*/*.jpg"))
print(f'Total Images = {len(files)}')
```

Total Images = 1655

Оптимізація вхідних даних є критичною фазою у тренуванні глибоких нейронних мереж, особливо при класифікації зображень. Використання ефективних методів передобробки та аугментації може значно підвищити здатність моделі узагальнювати дані, не бачені під час тренування.

Для класифікації якості фруктів було використано функцію `ImageDataLoaders.from_folder` бібліотеки `FastAI`, яка автоматично розділяє зображення на навчальний та валідаційний набори. Навчальний набір містить приблизно 80% даних (1324 зображення), а валідаційний — 20% (331 зображення). Це розділення є стандартним підходом для перевірки ефективності моделі на неопублікованих даних.

```
dls = ImageDataLoaders.from_folder(path, train='train', valid_pct=0.2,
                                   item_tfms=Resize(460),
                                   batch_tfms=[*aug_transforms(size=224), Normalize.from_stats(*imagenet_stats)])

len(dls.train_ds)

1324

len(dls.valid_ds)

331
```

Перед тренуванням зображення були оброблені за допомогою кількох ключових кроків передобробки:

1. Зміна розміру зображень: Всі зображення були масштабовані до розміру 460 пікселів по довшій стороні для забезпечення уніформності вхідних даних та покращення ефективності обчислень.
2. Аугментація зображень: Використовувалися стандартні техніки аугментації, такі як повороти, зміщення та зміна масштабу до розміру 224 пікселів. Це дозволяє моделі краще узагальнювати на різних варіаціях зображень, що підвищує її стійкість до переоснащення.
3. Нормалізація: Зображення були нормалізовані за допомогою статистики ImageNet, що дозволяє використовувати ваги попередньо навченої моделі ефективніше, забезпечуючи кращий стартовий пункт для тренування.

Після завершення процесу передобробки та розділення даних, було проведено візуалізацію партії зображень (рис.3.1), що дозволяє оцінити якість передобробки та різноманітність аугментацій, застосованих до зображень. Візуалізація використовується для перевірки того, чи зображення правильно оброблені та чи готові вони до тренування моделі.



Рисунок 3.1 - Оцінка якості передобробки та різноманітність аугментацій, застосованих до зображень

3.2 Навчання моделі

Модель була ініційована за допомогою архітектури ResNet34, яка є однією з передових архітектур для задач класифікації зображень у галузі комп'ютерного зору. Це рішення було вибрано через його здатність до глибокого навчання з великою кількістю параметрів та ефективного управління зникненням градієнтів через використання резидуальних з'єднань.

Для початку навчання моделі було використано ваги з попередньо навченої моделі ResNet34, доступної в репозиторії PyTorch. Це дозволило скоротити час та ресурси, необхідні для досягнення високої точності, оскільки попереднє навчання на великих наборах даних забезпечує хорошу ініціалізацію для ваг моделі.

Основна структура моделі включає послідовно розміщені конволюційні шари, нормалізацію пакетів, ReLU активаційні функції, та методи пулінгу. Детальний опис шарів та параметрів моделі представлено на Рисунку 3.1, що дозволяє глибше зрозуміти структуру та функціонування мережі.

Модель ResNet34, що була адаптована для задачі класифікації фруктів, містить важливі архітектурні елементи, які сприяють її ефективності. Конфігурація моделі включає різні типи шарів: згорткові (Conv2d), нормалізацію пакетів (BatchNorm2d), активаційні функції (ReLU), та методи пулінгу (MaxPool). Ключові параметри кожного шару зазначені в архітектурному зведенні, що представлено у моделі, де загальна кількість параметрів складає 21,815,104, з яких лише 547,456 є навчуваними.

Тренування моделі здійснювалось із використанням оптимізатора Adam, що відомий своєю здатністю ефективно коригувати швидкість навчання на основі моментних оцінок градієнтів. Вибір функції втрати — Flattened Loss of CrossEntropyLoss, був спрямований на оптимізацію роботи з категоріальними мітками, що є типовим для задач класифікації.

Sequential (Input shape: 64 x 3 x 224 x 224)				64 x 128 x 28 x 28			
Layer (type)	Output Shape	Param #	Trainable				
				Conv2d	73728	False	
				BatchNorm2d	256	True	
	64 x 64 x 112 x 112			ReLU			
Conv2d		9408	False	Conv2d	147456	False	
BatchNorm2d		128	True	BatchNorm2d	256	True	
ReLU				Conv2d	8192	False	
	64 x 64 x 56 x 56			BatchNorm2d	256	True	
MaxPool2d				Conv2d	147456	False	
Conv2d		36864	False	BatchNorm2d	256	True	
BatchNorm2d		128	True	ReLU			
ReLU				Conv2d	147456	False	
Conv2d		36864	False	BatchNorm2d	256	True	
BatchNorm2d		128	True	Conv2d	147456	False	
Conv2d		36864	False	BatchNorm2d	256	True	
BatchNorm2d		128	True	ReLU			
ReLU				Conv2d	147456	False	
Conv2d		36864	False	BatchNorm2d	256	True	
BatchNorm2d		128	True	Conv2d	147456	False	
Conv2d		36864	False	BatchNorm2d	256	True	
BatchNorm2d		128	True	ReLU			
ReLU				Conv2d	147456	False	
Conv2d		36864	False	BatchNorm2d	256	True	
BatchNorm2d		128	True				

Conv2d	294912	False	Conv2d	1155144	False
BatchNorm2d	512	True	BatchNorm2d	1024	True
ReLU			ReLU		
Conv2d	500032	False	Conv2d	739976	False
BatchNorm2d	512	True	BatchNorm2d	1024	True
Conv2d	500032	False	Conv2d	51872	False
BatchNorm2d	512	True	BatchNorm2d	1024	True
Conv2d	500032	False	Conv2d	259676	False
BatchNorm2d	512	True	BatchNorm2d	1024	True
ReLU			ReLU		
Conv2d	500032	False	Conv2d	739976	False
BatchNorm2d	512	True	BatchNorm2d	1024	True
Conv2d	500032	False	BatchNorm2d	1024	True
BatchNorm2d	512	True	ReLU		
ReLU			Conv2d	259676	False
Conv2d	500032	False	BatchNorm2d	1024	True
BatchNorm2d	512	True			
Conv2d	500032	False	AdaptiveAvgPool2d		
BatchNorm2d	512	True	AdaptiveMaxPool2d		
ReLU					
Conv2d	500032	False			
BatchNorm2d	512	True	Flatten		
Conv2d	500032	False	BatchNorm1d	2048	True
BatchNorm2d	512	True	Dropout		
ReLU					
Conv2d	500032	False			
BatchNorm2d	512	True	Linear	8832	True
Conv2d	500032	False			
BatchNorm2d	512	True			

Total params: 21,815,104

Total trainable params: 547,456

Total non-trainable params: 21,267,648

Optimizer used: <function Adam at 0x78ee146f2200>

Loss function: FlattenedLoss of CrossEntropyLoss()

Model frozen up to parameter group #2

Callbacks:

- TrainEvalCallback
- CastToTensor
- Recorder
- ProgressCallback

Рисунок 3.1 – Результат навчання моделі

Після завершення процесу навчання, модель продемонструвала високий рівень точності на навчальному та валідаційному наборах даних. Детальні метрики, включаючи точність, втрати та інші важливі показники, були зафіксовані і занотовані в системі збору даних (згадані в розділі 'Callbacks'). Прогрес у навчанні та адаптації моделі до даних можна відстежити за допомогою візуалізаційних інструментів, таких як зведені графіки втрат, представлені у модулі Recorder.

Особливості архітектури ResNet34, зокрема використання резидуальних з'єднань та глибоких згорткових шарів, істотно вплинули на здатність моделі ефективно класифікувати фрукти на основі їхніх зображень. Резидуальні з'єднання зменшують проблему зникнення або вибуху градієнтів під час тренування, що дозволяє моделі вивчати більш складні візуальні патерни без значної втрати продуктивності.

3.3 Отримані результати

З використанням методу `lr_find` від FastAI, модель ResNet34 проходила процес автоматичного визначення оптимального діапазону швидкості навчання. Для цього методу використовувалися спеціально визначені функції `suggest_funcs`, що включали `minimum`, `steep`, `valley`, та `slide` для точнішого визначення критичних точок зміни швидкості навчання. Згідно з результатами, були запропоновані такі швидкості навчання: мінімальна - 0.00831763744354248, круто - 0.002511886414140463, долина - 0.0010000000474974513, і слайд - 0.0012022644514217973 (див. Рисунок 3.2).

Модель була тренувана використовуючи метод `fit_one_cycle`, що включав п'ять епох з використанням діапазону швидкості навчання від 'долини' до 'слайду'. Значення втрат (loss) та точності на навчальних та валідаційних наборах даних поступово зменшувалися та покращувалися відповідно, вказуючи на ефективність підходу до тренування. Втрати на

тренуванні знизилися з 1.852164 до 0.440642, а точність зросла до 0.918429 за останню епоху (Таблиця 3.1).

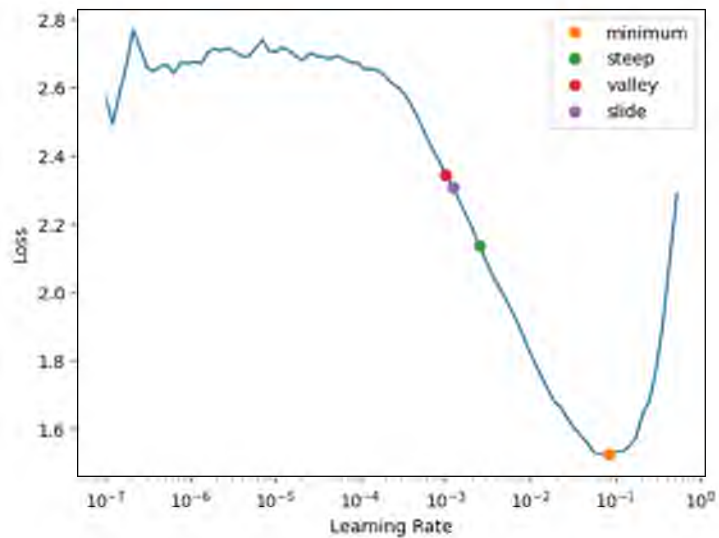


Рисунок 3.2 – Візуалізація процесу навчання

Таблиця 3.1 - Втрати на тренуванні

epoch	train_loss	valid_loss	accuracy	time
0	1.852164	0.591013	0.812689	00:06
1	1.083867	0.326401	0.885196	00:06
2	0.746740	0.294479	0.894260	00:06
3	0.558704	0.273870	0.915408	00:06
4	0.440642	0.260459	0.918429	00:06

Графік динаміки втрат, наданий **learn.recorder.plot_loss**, ілюструє тенденції зменшення втрат та покращення точності моделі під час навчання. Цей графік (Рисунок 3.3) відображає важливість адаптації швидкості навчання та її вплив на ефективність навчання.

Після тренування моделі було проведено тестування з використанням зображень із датасету. Перше тестування показало, що зразок зіпсованих персиків було успішно класифіковано з високою вірогідністю (передбачена категорія "rotten_peaches_done" з імовірністю 76.943%). Такий результат демонструє здатність моделі ефективно розпізнавати та класифікувати стани фруктів, базуючись на візуальних даних (Рисунок 3.4).

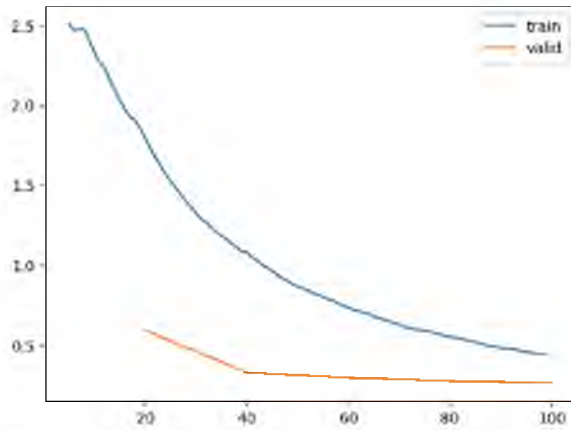


Рисунок 3.3 – Візуалізація швидкості навчання



Рисунок 3.4 – Візуалізація передбачених результатів

3.4 Оптимізація моделі

Після розморожування всіх шарів моделі, було проведено повторний пошук оптимальної швидкості навчання за допомогою функції **lr_find**. Результати цього пошуку запропонували різні точки швидкості навчання для ефективного тренування: мінімум ($5.75e-07$), круто ($1.90e-06$), долина (0.000229), та слайд (0.0000912). Ці значення були відображені на рисунку 3.5, що демонструє графік зміни швидкості навчання.

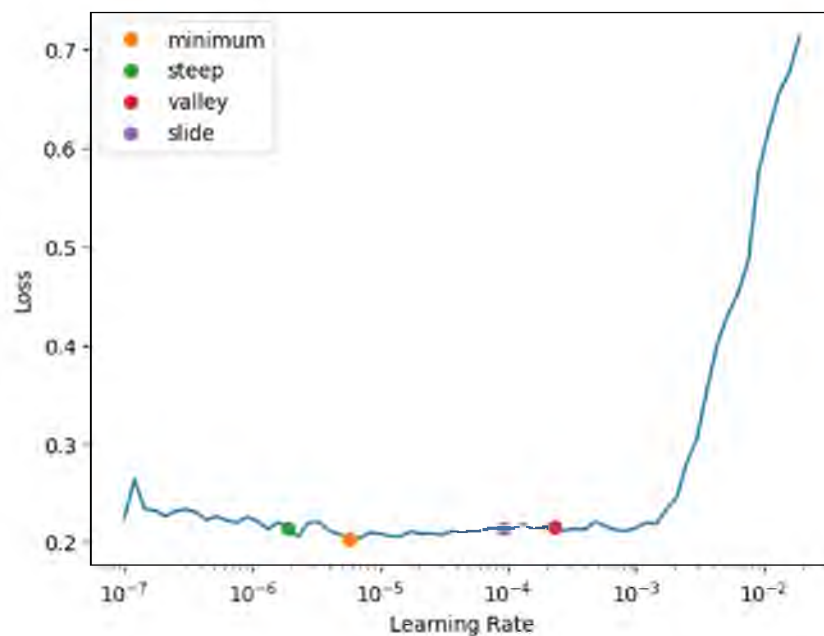


Рисунок 3.5 – Візуалізація зміни швидкості навчання після оптимізації

Модель була тренована за допомогою методу **fit_one_cycle** на 10 епохах, використовуючи швидкості навчання в діапазоні від 'долини' до 'слайду'. Під час цього процесу спостерігалася значна поліпшення точності та зменшення втрат, як в навчальному, так і в валідаційному наборах даних. Результати навчання показали зростання точності до 96.978% та зниження втрат до 0.051618 у фінальній епосі, що є відображені в таблиці 3.2.

Графічне зображення динаміки втрат під час навчання (Рисунок 3.6) відображає як зменшення втрат, так і зростання точності, підкреслюючи

ефективність проведеної оптимізації швидкості навчання. Найвища точність та найнижчі втрати були зафіксовані у восьмій епосі, демонструючи значний прогрес у навчанні моделі.

Таблиця 3.2 - Втрати на тренуванні після оптимізації

epoch	train_loss	valid_loss	accuracy	time
0	0.239641	0.251509	0.903323	00:06
1	0.215540	0.186518	0.930514	00:06
2	0.213415	0.312235	0.900302	00:06
3	0.185759	0.258474	0.918429	00:06
4	0.168092	0.186935	0.930514	00:07
5	0.141757	0.093001	0.960725	00:06
6	0.116545	0.115439	0.963746	00:06
7	0.084456	0.103935	0.972810	00:06
8	0.064183	0.088723	0.966767	00:06
9	0.051618	0.089892	0.969788	00:07

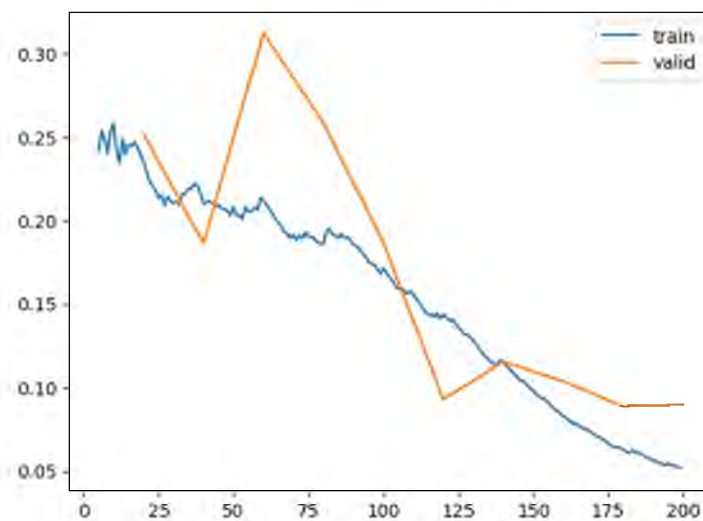


Рисунок 3.6 – Візуалізація динаміки втрат під час навчання після оптимізації

Після завершення тренування моделі були представлені результати класифікації на валідаційному наборі, що включає візуалізацію передбачень моделі в порівнянні з фактичними мітками (див. Рисунок 3.7). Ця візуалізація включає зразки зображень, їх передбачені мітки та ймовірності, що

підтверджують високу відповідність між передбаченнями моделі та фактичними мітками.



Рисунок 3.7 – Візуалізація передбачених результатів після оптимізації

3.5 Оцінка отриманих результатів

Після завершення тренування моделі було здійснено оцінку класифікації на валідаційному наборі даних. Звіт про класифікацію показав високу точність у розпізнаванні різних категорій фруктів. Значення точності (precision), відгуку (recall) та F1-балу для кожної категорії зазначені в рисунку 3.8. Загальна точність моделі склала 97%, з високими показниками точності по всім класам, зокрема 99% для свіжих гранатів та 98% для зіпсованих персиків і гранатів.

	precision	recall	f1-score	support
fresh_peaches_done	0.92	0.98	0.95	46
fresh_pomegranates_done	0.99	1.00	0.99	75
fresh_strawberries_done	0.97	0.95	0.96	38
rotten_peaches_done	0.98	0.94	0.96	62
rotten_pomegranates_done	0.98	0.96	0.97	57
rotten_strawberries_done	0.96	0.98	0.97	53
accuracy			0.97	331
macro avg	0.97	0.97	0.97	331
weighted avg	0.97	0.97	0.97	331

Рисунок 3.8 – Звіт про класифікацію

Матриця помилок, яка була отримана з використанням інструмента **ClassificationInterpretation**, відображає розподіл передбачених та фактичних класів (Рисунок 3.9). Це візуальне представлення підтверджує високу здатність моделі розрізняти між свіжими та зіпсованими фруктами з високою точністю, особливо виділяючись у класах зіпсованих персиків та гранатів.

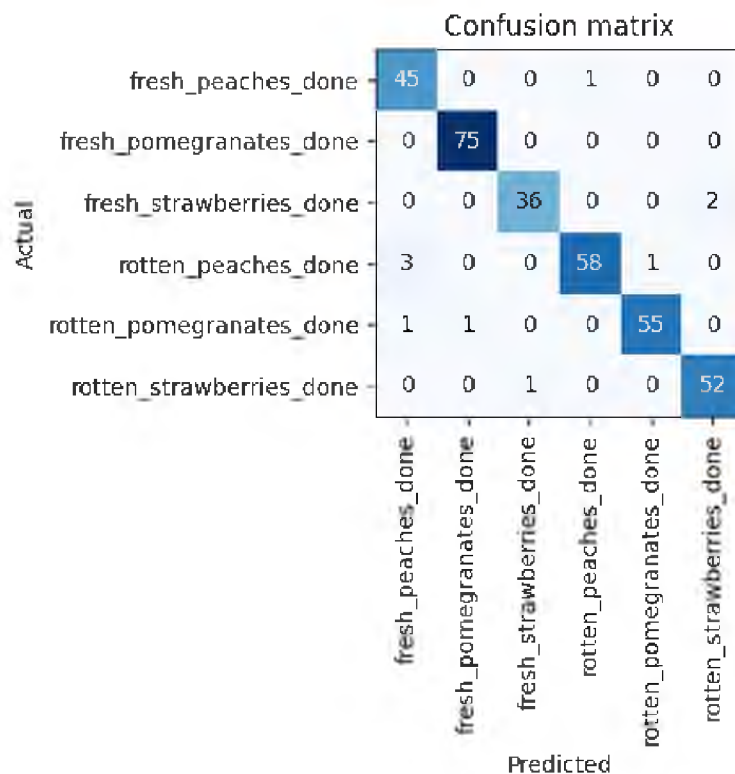


Рисунок 3.9 – Матриця помилок

Для глибшого розуміння помилок класифікації було використано функцію **plot_top_losses**, яка відображає зображення з найбільшими втратами

(Рисунок 3.10). Аналіз цих випадків дозволяє ідентифікувати специфічні характеристики зображень, що можуть викликати помилки у моделі, та надає інсайти для подальшої оптимізації моделі.

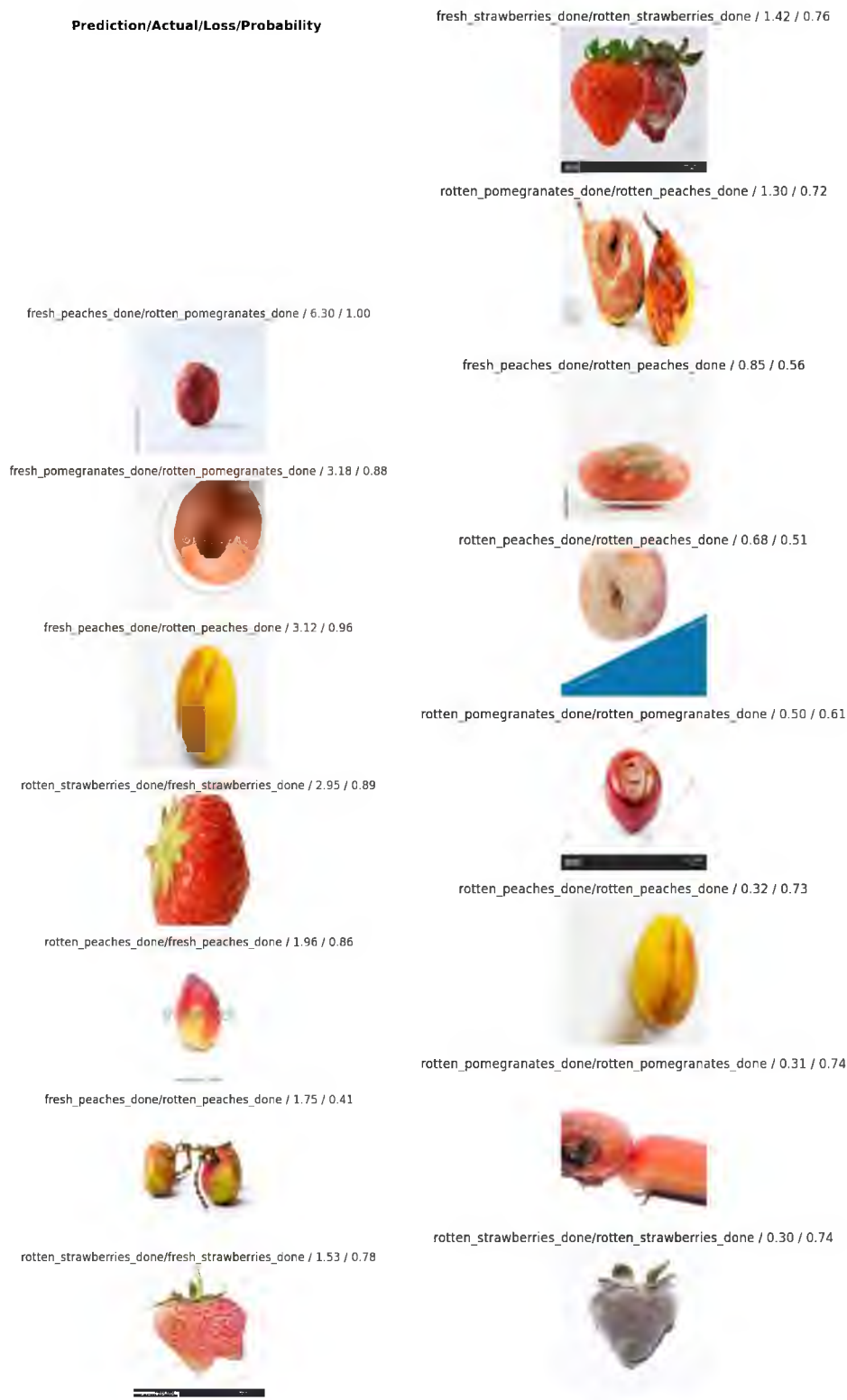


Рисунок 3.9 – Візуалізація зображення з найбільшими втратами

Отримані результати демонструють високу ефективність моделі ResNet34 у задачі класифікації якості фруктів. Висока точність класифікації, зокрема відмінна спроможність розпізнавати стан зіпсованих фруктів, свідчить про потенціал моделі у практичних застосуваннях у сфері агропромисловості та продовольчої безпеки. Розуміння та вдосконалення моделі через аналіз втрат та помилок може допомогти у подальшому підвищенні її точності та надійності.

ВИСНОВКИ

У світлі постійно зростаючих вимог до ефективності агропромислового сектору та покращення контролю якості харчових продуктів, наше дослідження набуває особливої актуальності. Використання архітектури ResNet34 для класифікації якості фруктів демонструє значні переваги у точності, швидкості та узагальненості порівняно з традиційними методами. Це дослідження не тільки виправдовує інвестиції у передові технології машинного зору та глибокого навчання, але й вказує на широкий спектр потенційних застосувань у сферах за межами агропромисловості. Впровадження таких систем може кардинально трансформувати процеси оцінки та сортування фруктів, значно зменшуючи втрати та підвищуючи загальну якість продукції, що, у свою чергу, сприятиме задоволенню та збереженню здоров'я кінцевих споживачів.

У ході нашого дослідження ми розробили та впровадили передову систему розпізнавання якості фруктів, використовуючи архітектуру глибокої нейронної мережі ResNet34. Використання цієї технології дозволило нам автоматизувати процес оцінки якості, значно підвищивши точність класифікації та забезпечивши об'єктивність у визначенні стандартів якості продукції. Реалізація даної системи демонструє, як сучасні досягнення у галузі комп'ютерного зору та машинного навчання можуть ефективно вирішувати проблеми, з якими стикаються агропромислові підприємства, сприяючи оптимізації виробничих процесів та покращенню якості продуктів.

Систематичний підхід до розробки моделі, включаючи детальну підготовку та обробку даних, налаштування та тренування нейронної мережі, а також глибокий аналіз отриманих результатів, забезпечив високий рівень точності та надійності розробленої системи. Важливим аспектом дослідження було виявлення та аналіз помилкових класифікацій, що допомогло в удосконаленні моделі, роблячи її більш адаптивною до різноманітних умов

експлуатації. Це, в свою чергу, підкреслює важливість континуального навчання та постійного вдосконалення систем на основі зібраних даних і зворотного зв'язку від кінцевих користувачів.

На завершення, реалізована система розпізнавання якості фруктів стала вагомим внеском у технологічний розвиток агропромислового сектору, демонструючи ефективність застосування передових алгоритмів машинного навчання у практичних сценаріях. Перспективи подальшого розвитку і вдосконалення таких систем відкривають нові можливості для їх застосування не тільки у сільському господарстві, а й у інших галузях, де потрібна висока точність класифікації та обробки великих обсягів даних.

У рамках нашого дослідження було розроблено та впроваджено модуль розпізнавання якості фруктів з використанням глибокого навчання на базі архітектури ResNet34. Результати тренування показали значні досягнення в точності класифікації, досягаючи 96.978% на валідаційному наборі даних із загальним зниженням втрат до 0.051618 у фінальній епосі навчання. Ці кількісні показники вказують на високу ефективність використаної моделі та її спроможність адекватно реагувати на складні варіації в зовнішньому вигляді фруктів, що забезпечується завдяки глибокому навчанню та ретельно розробленій архітектурі.

Оптимізація параметрів моделі дозволила домогтися не тільки високої точності, але й стабільності в результатах. Передбачення якості фруктів, як показали тести, включаючи аналіз помилок, є високо вірогідними, з високими показниками precision та recall, що свідчить про мінімальну кількість хибнопозитивних та хибнонегативних результатів. Модуль зміг ефективно визначати стан фруктів, включаючи складні випадки, де зовнішні ознаки пошкоджень або стиглості не були виражені явно. Це стало можливим завдяки адаптації моделі до різноманітних зображень через аугментацію та передобробку даних.

Висновок дослідження підкреслює значний потенціал впровадження глибокого навчання в агропромисловому секторі для покращення якості продукції та оптимізації виробничих процесів. Розроблена система може бути адаптована для різноманітних типів фруктів та умов експлуатації, що робить її універсальним інструментом для різних агропромислових підприємств. Крім того, аналіз та оптимізація моделі на основі зібраних даних та зворотного зв'язку від користувачів можуть допомогти в подальшому підвищенні точності та ефективності системи, сприяючи її впровадженню як стандарту для контролю якості в індустрії.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Africa, A. D. M., Tabalan, A. R. V., & Tan, M. A. A. (2020). Ripe fruit detection and classification using machine learning. Retrieved from <https://www.academia.edu/download/63555342/ijeter60852020200607-61696-z74qi4.pdf>
2. Mahajan, M. S., & Patil, S. T. (2016). Optimization and classification of fruit using machine learning algorithm. *International Journal*. Retrieved from <https://www.academia.edu/download/47640418/IJRSTV311139.pdf>
3. Ishikawa, T., Hayashi, A., & Others. (2018). Classification of strawberry fruit shape by machine learning. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Retrieved from <https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLII-2/463/2018/isprs-archives-XLII-2-463-2018.pdf>
4. Pandey, R., Naik, S., & Marfatia, R. (2013). Image processing and machine learning for automated fruit grading system: A technical review. *International Journal of Computer Applications*. Retrieved from <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=aa50c9a262d58ba876b3be2e1a0472a2dc799277>
5. Aherwadi, N., & Mittal, U. (2022). Fruit quality identification using image processing, machine learning, and deep learning: A review. *Advances in Applied Mathematics and Science*. Retrieved from https://www.mililink.com/upload/article/2061129010aams_vol_215_march_2022_a24_p2645-2660_nagnath_aherwadi_and_usha_mittal.pdf
6. Patil, P. U., Lande, S. B., Nagalkar, V. J., & Nikam, S. B. (2021). Grading and sorting technique of dragon fruits using machine learning algorithms. *Journal of Agriculture and Food Research*. Retrieved from <https://www.sciencedirect.com/science/article/pii/S266615432100020X>

7. Ukwuoma, C. C., Zhiguang, Q., Bin Heyat, M. B., & Others. (2022). Recent advancements in fruit detection and classification using deep learning techniques. *Mathematical Problems in Engineering*. Retrieved from <https://www.hindawi.com/journals/mpe/2022/9210947/>
8. Joseph, J. L., Kumar, V. A., & Mathew, S. P. (2021). Fruit classification using deep learning. Retrieved from https://link.springer.com/chapter/10.1007/978-981-16-0749-3_62
9. Zawbaa, H. M., Hazman, M., & Abbass, M. (2014). Automatic fruit classification using random forest algorithm. Retrieved from <https://ieeexplore.ieee.org/abstract/document/7086191/>
10. Gill, H. S., & Khehra, B. S. (2022). Fruit image classification using deep learning. Retrieved from <https://www.researchsquare.com/article/rs-574901/latest>
11. Castro, W., Oblitas, J., De-La-Torre, M., & Cotrina, C. (2019). Classification of cape gooseberry fruit according to its level of ripeness using machine learning techniques and different color spaces. Retrieved from <https://ieeexplore.ieee.org/abstract/document/8657936/>
12. Hossain, M. S., & Al-Hammadi, M. (2018). Automatic fruit classification using deep learning for industrial applications. Retrieved from <https://ieeexplore.ieee.org/abstract/document/8488544/>
13. Комар М.П., Саченко А.О., Васильків Н.М., Гладій Г.М., Коваль В.С., Лип'яніна-Гончаренко Х.В. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» за першим (бакалаврським) рівнем вищої освіти. Тернопіль: ЗУНУ, 2024. 52 с.
14. Загальні методичні рекомендації з підготовки, оформлення, захисту та оцінювання кваліфікаційних робіт здобувачів вищої освіти першого (бакалаврського) і другого (магістерського) рівнів. Тернопіль: ЗУНУ, 2024. 83 с.

ДОДАТОК А

ПСЕВДОКОД АЛГОРИТМУ

```
// Import necessary libraries
import libraries including pandas, numpy, matplotlib, opencv, fastai

// Suppress warnings
ignore warnings

// Load dataset from a specified path
define path to dataset
list all image files in the dataset
print total number of images

// Prepare DataLoaders for training and validation
create ImageDataLoaders from the dataset directory
    specify path, train directory, validation percentage, item transformations, batch transformations
print number of training and validation samples
display sample batch of images

// Model Training
initialize a vision learner using a pre-trained model (e.g., resnet34) and specify accuracy as the metric
print summary of the model architecture
find the best learning rate using the learning rate finder
train the model using the 'fit_one_cycle' method for a specified number of epochs and learning rates

// Display training results
plot training and validation loss curves
make predictions on sample images
show results of predictions on a batch

// Model Optimization and Final Training
unfreeze all layers of the model to allow fine-tuning
find the best learning rate again
train the entire model for more epochs using the new learning rates
plot the final loss curves
show final results of predictions on a batch

// Evaluate the Model
get predictions and actual targets for the validation set
generate a classification report showing precision, recall, f1-score for each class
print the classification report

// Confusion Matrix and Error Analysis
create an interpretation object from the learner
plot the confusion matrix to see class-wise performance
plot top losses to identify where the model is making the most mistakes

// Conclude
```

ДОДАТОК Б

КОД ДЛЯ РЕАЛІЗАЦІЇ

```
# %% [markdown]
# ![fruitytypes-header.jpg](attachment:14f3e48d-1b6a-4231-9fa2-109d49b53c2f.jpg)

# %% [markdown]
# <div style="background-color: #6A0DAD; padding: 15px; border-radius: 10px; box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.1);">
#   <p style="color: white; font-size: 18px; font-weight: bold; text-align: center; text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5); margin: 0;">
#     Import Libraries
#   </p>
# </div>
#

# %% [code] {"execution": {"iopub.status.busy": "2024-03-12T05:51:31.148534Z", "iopub.execute_input": "2024-03-12T05:51:31.149423Z", "iopub.status.idle": "2024-03-12T05:51:31.158512Z", "shell.execute_reply.started": "2024-03-12T05:51:31.149382Z", "shell.execute_reply": "2024-03-12T05:51:31.157168Z"}}
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import os
import random
import cv2
import sklearn
from PIL import Image
from pathlib import Path
from fastai.vision.all import *
from fastai.callback import *
from fastai.metrics import error_rate
from sklearn.metrics import classification_report

import warnings
warnings.filterwarnings('ignore')

# %% [markdown]
# <div style="background-color: #6A0DAD; padding: 15px; border-radius: 10px; box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.1);">
#   <p style="color: white; font-size: 18px; font-weight: bold; text-align: center; text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5); margin: 0;">
#     Load Data
#   </p>
# </div>
#

# %% [code] {"execution": {"iopub.status.busy": "2024-03-12T05:51:31.173533Z", "iopub.execute_input": "2024-03-12T05:51:31.173985Z", "iopub.status.idle": "2024-03-12T05:51:31.179371Z", "shell.execute_reply.started": "2024-03-12T05:51:31.17394Z", "shell.execute_reply": "2024-03-12T05:51:31.178097Z"}}
path = Path("/kaggle/input/fruits-dataset-for-classification")

# %% [code] {"execution": {"iopub.status.busy": "2024-03-12T05:51:31.182261Z", "iopub.execute_input": "2024-03-12T05:51:31.182685Z", "iopub.status.idle": "2024-03-12T05:51:31.19632Z", "shell.execute_reply.started": "2024-03-12T05:51:31.182647Z", "shell.execute_reply": "2024-03-12T05:51:31.195011Z"}}
path.ls()

# %% [code] {"execution": {"iopub.status.busy": "2024-03-12T05:51:31.197883Z", "iopub.execute_input": "2024-03-12T05:51:31.19867Z", "iopub.status.idle": "2024-03-12T05:51:31.21791Z", "shell.execute_reply.started": "2024-03-12T05:51:31.198633Z", "shell.execute_reply": "2024-03-12T05:51:31.216794Z"}}
files = list(path.glob("*/*.jpg"))

print(f'Total Images = {len(files)}')

# %% [markdown]
# <div style="background-color: #6A0DAD; padding: 15px; border-radius: 10px; box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.1);">
#   <p style="color: white; font-size: 18px; font-weight: bold; text-align: center; text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5); margin: 0;">
#     Fast AI
#   </p>
# </div>
#

# %% [markdown]
# ![ai_banner.jpg](attachment:57c6ee87-2290-43c1-9c8a-e79a582b15a1.jpg)

# %% [code] {"execution": {"iopub.status.busy": "2024-03-12T05:51:31.219494Z", "iopub.execute_input": "2024-03-12T05:51:31.219902Z", "iopub.status.idle": "2024-03-12T05:51:31.425199Z", "shell.execute_reply.started": "2024-03-12T05:51:31.219865Z", "shell.execute_reply": "2024-03-12T05:51:31.424098Z"}}
dls = ImageDataLoaders.from_folder(path, train='train', valid_pct=0.2,
                                   item_tfms=Resize(460),
                                   batch_tfms=[*aug_transforms(size=224), Normalize.from_stats(*imagenet_stats)])
```

```

# %% [code] {"execution":{"iopub.status.busy":"2024-03-12T05:51:31.428981Z","iopub.execute_input":"2024-03-12T05:51:31.429415Z","iopub.status.idle":"2024-03-12T05:51:31.436479Z","shell.execute_reply.started":"2024-03-12T05:51:31.429376Z","shell.execute_reply":"2024-03-12T05:51:31.435363Z"}}
len(dls.train_ds)

# %% [code] {"execution":{"iopub.status.busy":"2024-03-12T05:51:31.438276Z","iopub.execute_input":"2024-03-12T05:51:31.438681Z","iopub.status.idle":"2024-03-12T05:51:31.447018Z","shell.execute_reply.started":"2024-03-12T05:51:31.43865Z","shell.execute_reply":"2024-03-12T05:51:31.445797Z"}}
len(dls.valid_ds)

# %% [code] {"execution":{"iopub.status.busy":"2024-03-12T05:51:31.448276Z","iopub.execute_input":"2024-03-12T05:51:31.448644Z"}}
dls.show_batch()

# %% [markdown]
# <div style="background-color: #6AoDAD; padding: 15px; border-radius: 10px; box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.1);">
#   <p style="color: white; font-size: 18px; font-weight: bold; text-align: center; text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5); margin: 0;">
#     Training Data
#   </p>
# </div>
#

# %% [code]
learn = vision_learner(dls, resnet34, metrics=accuracy)

# %% [code] {"execution":{"iopub.status.idle":"2024-03-12T05:51:33.804702Z","shell.execute_reply.started":"2024-03-12T05:51:33.795627Z","shell.execute_reply":"2024-03-12T05:51:33.799126Z"}}
learn.model_dir = "/kaggle/working/"

# %% [code] {"execution":{"iopub.status.busy":"2024-03-12T05:51:33.806844Z","iopub.execute_input":"2024-03-12T05:51:33.807216Z","iopub.status.idle":"2024-03-12T05:51:34.720983Z","shell.execute_reply.started":"2024-03-12T05:51:33.807181Z","shell.execute_reply":"2024-03-12T05:51:34.719692Z"}}
learn.summary()

# %% [markdown]
# <div style="background-color: #6AoDAD; padding: 15px; border-radius: 10px; box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.1);">
#   <p style="color: white; font-size: 18px; font-weight: bold; text-align: center; text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5); margin: 0;">
#     Model Building
#   </p>
# </div>
#

# %% [code] {"execution":{"iopub.status.busy":"2024-03-12T05:51:34.722553Z","iopub.execute_input":"2024-03-12T05:51:34.723615Z","iopub.status.idle":"2024-03-12T05:52:11.407364Z","shell.execute_reply.started":"2024-03-12T05:51:34.723577Z","shell.execute_reply":"2024-03-12T05:52:11.406167Z"}}
suggest_funcs = (minimum, steep, valley, slide)

lrs = learn.lr_find(suggest_funcs=suggest_funcs)

# %% [code] {"execution":{"iopub.status.busy":"2024-03-12T05:52:11.408992Z","iopub.execute_input":"2024-03-12T05:52:11.409363Z","iopub.status.idle":"2024-03-12T05:52:11.419032Z","shell.execute_reply.started":"2024-03-12T05:52:11.409319Z","shell.execute_reply":"2024-03-12T05:52:11.417955Z"}}
lrs

# %% [code] {"execution":{"iopub.status.busy":"2024-03-12T05:52:11.420243Z","iopub.execute_input":"2024-03-12T05:52:11.420627Z","iopub.status.idle":"2024-03-12T05:52:59.583372Z","shell.execute_reply.started":"2024-03-12T05:52:11.420602Z","shell.execute_reply":"2024-03-12T05:52:59.582266Z"}}
learn.fit_one_cycle(5, lr_max=slice(lrs.valley, lrs.slide))
learn.recorder.plot_loss()

# %% [code] {"execution":{"iopub.status.busy":"2024-03-12T05:52:59.587944Z","iopub.execute_input":"2024-03-12T05:52:59.588271Z","iopub.status.idle":"2024-03-12T05:52:59.64738Z","shell.execute_reply.started":"2024-03-12T05:52:59.588239Z","shell.execute_reply":"2024-03-12T05:52:59.64623Z"}}
learn.predict(files[0])

# %% [code] {"execution":{"iopub.status.busy":"2024-03-12T05:52:59.649199Z","iopub.execute_input":"2024-03-12T05:52:59.649648Z","iopub.status.idle":"2024-03-12T05:53:01.520687Z","shell.execute_reply.started":"2024-03-12T05:52:59.649608Z","shell.execute_reply":"2024-03-12T05:53:01.519504Z"}}
learn.show_results()

# %% [markdown]
# <div style="background-color: #6AoDAD; padding: 15px; border-radius: 10px; box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.1);">

```

```

# <p style="color: white; font-size: 18px; font-weight: bold; text-align: center; text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5); margin: 0;">
#   Optimize Model
# </p>
# </div>
#

# %% [code] {"execution":{"iopub.status.busy":"2024-03-12T05:53:01.522168Z","iopub.execute_input":"2024-03-
12T05:53:01.522559Z","iopub.status.idle":"2024-03-12T05:53:36.121841Z","shell.execute_reply.started":"2024-03-
12T05:53:01.522525Z","shell.execute_reply":"2024-03-12T05:53:36.120521Z"}}
learn.unfreeze()
lrs = learn.lr_find(suggest_funcs=(minimum, steep, valley, slide))

# %% [code] {"execution":{"iopub.status.busy":"2024-03-12T05:53:36.123693Z","iopub.execute_input":"2024-03-
12T05:53:36.12408Z","iopub.status.idle":"2024-03-12T05:53:36.131846Z","shell.execute_reply.started":"2024-03-
12T05:53:36.124041Z","shell.execute_reply":"2024-03-12T05:53:36.130846Z"}}
lrs

# %% [code] {"execution":{"iopub.status.busy":"2024-03-12T05:53:36.133434Z","iopub.execute_input":"2024-03-
12T05:53:36.134288Z","iopub.status.idle":"2024-03-12T05:55:14.902733Z","shell.execute_reply.started":"2024-03-
12T05:53:36.13425Z","shell.execute_reply":"2024-03-12T05:55:14.901378Z"}}
learn.fit_one_cycle(10, lr_max=slice(lrs.valley, lrs.slide))
learn.recorder.plot_loss()

# %% [code] {"execution":{"iopub.status.busy":"2024-03-12T05:55:14.905077Z","iopub.execute_input":"2024-03-
12T05:55:14.905525Z","iopub.status.idle":"2024-03-12T05:55:17.347411Z","shell.execute_reply.started":"2024-03-
12T05:55:14.905478Z","shell.execute_reply":"2024-03-12T05:55:17.346216Z"}}
learn.show_results()

# %% [markdown]
# <div style="background-color: #6AoDAD; padding: 15px; border-radius: 10px; box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.1);">
#   <p style="color: white; font-size: 18px; font-weight: bold; text-align: center; text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5); margin: 0;">
#     Classification Report
#   </p>
# </div>
#

# %% [code] {"execution":{"iopub.status.busy":"2024-03-12T05:55:17.34928Z","iopub.execute_input":"2024-03-
12T05:55:17.35003Z","iopub.status.idle":"2024-03-12T05:55:19.680648Z","shell.execute_reply.started":"2024-03-
12T05:55:17.349991Z","shell.execute_reply":"2024-03-12T05:55:19.678102Z"}}
preds, targets = learn.get_preds()

class_report = classification_report(targets, torch.argmax(preds, dim=1), target_names=dls.vocab)
print(class_report)

# %% [markdown]
# <div style="background-color: #6AoDAD; padding: 15px; border-radius: 10px; box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.1);">
#   <p style="color: white; font-size: 18px; font-weight: bold; text-align: center; text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5); margin: 0;">
#     Confusion Matrix
#   </p>
# </div>
#

# %% [code] {"execution":{"iopub.status.busy":"2024-03-12T05:55:19.683263Z","iopub.execute_input":"2024-03-
12T05:55:19.683774Z","iopub.status.idle":"2024-03-12T05:55:24.993405Z","shell.execute_reply.started":"2024-03-
12T05:55:19.683722Z","shell.execute_reply":"2024-03-12T05:55:24.992233Z"}}
interp = ClassificationInterpretation.from_learner(learn)
interp.plot_confusion_matrix()

# %% [markdown]
# <div style="background-color: #6AoDAD; padding: 15px; border-radius: 10px; box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.1);">
#   <p style="color: white; font-size: 18px; font-weight: bold; text-align: center; text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5); margin: 0;">
#     Predictions - Actual
#   </p>
# </div>
#

# %% [code] {"execution":{"iopub.status.busy":"2024-03-12T05:55:24.994978Z","iopub.execute_input":"2024-03-
12T05:55:24.995472Z","iopub.status.idle":"2024-03-12T05:55:29.709311Z","shell.execute_reply.started":"2024-03-
12T05:55:24.995436Z","shell.execute_reply":"2024-03-12T05:55:29.707951Z"}}
interp = Interpretation.from_learner(learn)

interp.plot_top_losses(15, nrows=15)

```

```

# %% [markdown]
# <div style="background-color: #6A0DAD; padding: 15px; border-radius: 10px; box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.1);">
#   <p style="color: white; font-size: 18px; font-weight: bold; text-align: center; text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5); margin: 0;">
#     Finish
#   </p>
# </div>
#

# %% [markdown]
# <div style="background-color: #FFD700; color: #000000; padding: 15px; border-radius: 8px; text-align: center; box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.1);">
#   <p style="font-size: 18px; font-weight: bold; margin: 0;">
#
#   </p>
# </div>
#

# %% [markdown]
# ![360_F_543095125_SFIK1iuFVorJ7yGb10hq7DEWmSF9RjuG.jpg](attachment:44394637-4bb4-4610-9c37-82fa7f8d173e.jpg)

```

ДОДАТОК В
АПРОБАЦІЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ

МАТЕРІАЛИ

У ВСЕУКРАЇНСЬКОЇ СТУДЕНТСЬКОЇ НАУКОВОЇ

КОНФЕРЕНЦІЇ

17 ТРАВНЯ 2024 РІК • М. КИЇВ, УКРАЇНА

НАУКОВИЙ ПРОСТІР: АНАЛІЗ,
СУЧАСНИЙ СТАН, ТРЕНДИ ТА
ПЕРСПЕКТИВИ

ISBN 978-617-8312-44-2
DOI 10.36074/liga-uki-17.05.2024



УДК 082:001
Н 34

Голова оргкомітету: Коренюк І.О.
Верстка: Зрада С.І.
Дизайн: Бондаренко І.В.

Рекомендовано до видання Вченою Радою Інституту науково-технічної інтеграції та співпраці. Протокол № 36 від 16.05.2024 року.



Конференцію зареєстровано Державною науковою установою «УкрІНТЕІ» в базі даних науково-технічних заходів України та інформаційному бюлетені «План проведення наукових, науково-технічних заходів в Україні» (Посвідчення №29 від 05.01.2024).

Матеріали конференції знаходяться у відкритому доступі на умовах ліцензії CC BY-SA 4.0 International.

Науковий простір: аналіз, сучасний стан, тренди та перспективи:
Н 34 матеріали V Всеукраїнської студентської наукової конференції, м. Київ, 17 травня, 2024 рік / ГО «Молодіжна наукова ліга». — Вінниця: ТОВ «УКРЛОГОС Груп», 2024. — 586 с.

ISBN 978-617-8312-44-2
DOI 10.62732/liga-ukr-17.05.2024

Викладено матеріали учасників V Всеукраїнської мультидисциплінарної студентської наукової конференції «Науковий простір: аналіз, сучасний стан, тренди та перспективи», яка відбулася 17 травня 2024 року у місті Київ, Україна.

УДК 082:001

ISBN 978-617-8312-44-2

© Колектив учасників конференції, 2024
© ГО «Молодіжна наукова ліга», 2024
© ТОВ «УКРЛОГОС Груп», 2024

АРХІТЕКТУРА МОДУЛЯ ПРОГНОЗУВАННЯ СЕРЦЕВИХ ЗАХВОРЮВАНЬ Закалюк П., Науковий керівник: <i>Констанкевич Ю.В.</i>	374
АРХІТЕКТУРА МОДУЛЯ РЕКОМЕНДАЦІЙ МУЗИКИ НА ОСНОВІ МАШИННОГО НАВЧАННЯ Черній І., Науковий керівник: <i>Констанкевич Ю.В.</i>	376
АРХІТЕКТУРА МОДУЛЯ РЕКОМЕНДАЦІЙ НА ОСНОВІ МАШИННОГО НАВЧАННЯ Філюк В., Науковий керівник: <i>Констанкевич Ю.В.</i>	378
АРХІТЕКТУРА МОДУЛЯ РОЗПІЗНАВАННЯ ВІКУ ТА СТАТІ ЛЮДЕЙ Мартиник В., Науковий керівник: <i>Сапожник Г.В.</i>	380
АРХІТЕКТУРА МОДУЛЯ РОЗПІЗНАВАННЯ ЯКОСТІ ФРУКТІВ З ВИКОРИСТАННЯМ ГЛИБОКОГО НАВЧАННЯ Чіп С., Науковий керівник: <i>Сапожник Г.В.</i>	382
ВИБІР ТА ОБҐРУНТУВАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ КОНТРОЛЮ ПАРАМЕТРІВ МІКРОКЛІМАТУ УКРИТТЯ Заровський С.В., Науковий керівник: <i>Хімічева Г.І.</i>	384
ВИКОРИСТАННЯ МАШИННОГО НАВЧАННЯ ДЛЯ ОПТИМІЗАЦІЇ ПРОВЕДЕННЯ СПІВБЕСІД Бурлаков О.О., Науковий керівник: <i>Левус Є.В.</i>	387
ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ ОСВІТНІХ КУРСІВ Горбовий В.Р., Науковий керівник: <i>Левус Є.В.</i>	389
ІНФОРМАЦІЙНА СИСТЕМА НАДАННЯ ТЕЛЕМЕДИЧНИХ КОНСУЛЬТАЦІЙ З ПІДТРИМКОЮ СТАНДАРТУ HL7 Смерек Л.Л., Науковий керівник: <i>Кісь Я.П.</i>	391
МЕТОДИ ТА ЗАСОБИ МОДУЛЯЦІЇ РАДІОТЕХНІЧНИХ СИГНАЛІВ АВТОГЕНЕРАТОРІВ Овчарук А.О., Науковий керівник: <i>Осадчук В.С.</i>	394
ПРОГНОЗУВАННЯ ЦІН НА АКЦІЇ ЗА ДОПОМОГОЮ ШТУЧНОГО ІНТЕЛЕКТУ Воробйов А.Р.....	396
СИСТЕМА ДЛЯ ЗАБЕЗПЕЧЕННЯ ПЕРЕНАВЧАННЯ НЕЙРОННИХ СТРУКТУР МОЗКУ ЩОДО ФУНКЦІОНУВАННЯ ШЛЯХІВ ПЕРЕДАЧІ ЗОБРАЖЕННЯ ВІД ЗОРОВОГО НЕРВУ Гема О.Г., Науковий керівник: <i>Кузьомін О.Я.</i>	398

СЕКЦІЯ 17. ФІЗИКО-МАТЕМАТИЧНІ НАУКИ

ОСОБЛИВОСТІ ДИСТАНЦІЙНОЇ ФОРМИ НАВЧАННЯ ПРЕДМЕТУ “ВИЩА МАТЕМАТИКА” НА СПЕЦІАЛЬНОСТІ “АРХІТЕКТУРА” Бондарець В.Ю., Науковий керівник: <i>Турчанінова Л.І.</i>	400
---	-----

Чіп Святослав, здобувач вищої освіти факультету
комп'ютерних інформаційних технологій
Західноукраїнський національний університет, Україна

Науковий керівник: Сапожник Г.В., канд. іст. наук, канд. техн. наук, доцент,
доцент кафедри інформаційно-обчислювальних систем і управління
Західноукраїнський національний університет, Україна

АРХІТЕКТУРА МОДУЛЯ РОЗПІЗНАВАННЯ ЯКОСТІ ФРУКТІВ З ВИКОРИСТАННЯМ ГЛИБОКОГО НАВЧАННЯ

Архітектура (рис.1) модуля розпізнавання якості фруктів, розроблена на базі глибокого навчання, розбивається на три основні складові частини, які оптимізують процес розробки та дозволяють зосередитися на специфічних задачах у кожному з етапів. Описана нижче архітектура поєднує в собі різноманітні компоненти, що взаємодіють між собою, щоб забезпечити ефективну обробку даних, тренування моделі, а також її оцінку і деплоймент.

1. Дані і підготовка. Цей модуль відповідає за збір та підготовку даних, які необхідні для подальшого тренування моделі глибокого навчання. Він включає наступні компоненти:

- **Імпорт бібліотек:** Використовується для завантаження всіх необхідних зовнішніх бібліотек, таких як Pandas, Numpy, Matplotlib, що використовуються для обробки та візуалізації даних.

- **Імпорт фреймворків:** Залучення фреймворків глибокого навчання, таких як TensorFlow, Keras або PyTorch, які надають необхідні інструменти та API для розробки згорткових нейронних мереж.

- **Завантаження даних:** Здійснення завантаження та першої обробки даних з зовнішніх джерел або локального сховища.

- **Підготовка завантажувачів даних:** Створення об'єктів для ітеративного доступу до даних під час тренування, що включає розділення на тренувальні та валідаційні набори.

- **Трансформація та нормалізація:** Застосування предобробки даних, таких як масштабування, нормалізація, аугментація, щоб забезпечити кращі умови для ефективного навчання.

2. Налаштування і тренування моделі. Цей блок охоплює всі аспекти конфігурації та оптимізації моделі нейронної мережі:

- **Налаштування моделі:** Вибір архітектури моделі, такої як ResNet, VGG або інша згорткова нейронна мережа, та її ініціалізація з використанням попередньо навчених ваг.

- **Тренування моделі:** Використання алгоритмів оптимізації для мінімізації помилок прогнозування на основі тренувальних даних.

- **Оцінка моделі:** Використання валідаційних наборів для перевірки ефективності моделі та її здатності до узагальнення.

- **Дрібне налаштування:** Додаткове тренування моделі з відкритими всіма шарами

для доопрацювання та підвищення точності.



Рис. 1. Архітектура модуля розпізнавання якості фруктів з використанням глибокого навчання

3. Звітність та експорт. Фінальний етап, який забезпечує аналіз результатів та їх представлення у вигляді звітів, а також експорт тренованої моделі для її подальшого використання:

- Генерація звіту класифікації: Вивчення та оцінка продуктивності моделі за допомогою метрик, таких як точність, відгук, F1-оцінка.
- Ідентифікація помилкових класифікацій: Аналіз причин неправильних прогнозів та їх вплив на загальну точність.
- Завершення і експорт моделі: Збереження моделі у форматі, зручному для деплою, разом із усіма необхідними метаданими та налаштуваннями.

Розроблена архітектура відображає інтегрований підхід до вирішення задачі класифікації фруктів, поєднуючи в собі передові технології обробки даних, машинного навчання та глибокого навчання.

Список використаних джерел:

1. Архітектура програмного забезпечення для математичного моделювання на основі аналізу інтервальних даних з використанням хмарних технологій / М. Дивак та ін. Measuring and computing devices in technological processes. 2024. № 1. С. 125–139. URL: <https://doi.org/10.31891/2219-9365-2024-77-15> (дата звернення: 13.05.2024).