Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)
Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження моделей класифікації об'єктів на основі мультмодальних даних
(тема)

Виконав:
здобувач другого року навчання,
групи СШІм-23-1
Данило Кривошеїн
(власне ім'я, прізвище)
Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)
Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)
Освітня програма Системи штучного інтелекту
(повна назва освітньої програми)
Керівник доц. Олексій Турута
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ШІ _____ Олег ЗОЛОТУХІН
(підпис) (власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Штучного інтелекту

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного інтелекту
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«_____» _____ 20 ___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Кривошеїну Данилу Дмитровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження моделей класифікації об'єктів на основі мультимодальних даних

затверджена наказом університету від 21 квітня 20 25 р. № 295Ст

2. Термін подання студентом роботи до екзаменаційної комісії 5 червня 20 25 р.

3. Вихідні дані до роботи Наукові публікації про мультимодальну класифікацію об'єктів, дослідження методів інтеграції, наукові роботи моделей CLIP, FLAVA, ViLT, наукові статті про стратегії злиття модальностей, документація до бібліотек PyTorch та TensorFlow для реалізації мультимодальних архітектур, методи оцінки ефективності мультимодальних моделей, використання трансформерних архітектур для задач класифікації, набори даних для навчання та тестування мультимодальних класифікаторів товарів.

4. Перелік питань, що потрібно опрацювати в роботі
1) Аналіз предметної галузі мультимодальної класифікації, теоретичні основі обробки візуальної та текстової інформації
2) Дослідження методів комбінування різних модальностей
3) Підходи з ансамблем простих моделей
4) Імплементація підходу на основі трансформерної архітектури
5) Порівняльний аналіз розроблених методів

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	21.04.2025	виконано
2	Загальний огляд предметної галузі	22.04.2025 – 24.04.2025	виконано
3	Аналіз існуючих підходів до класифікації	25.04.2025 – 28.04.2025	виконано
	об’єктів		
4	Постановка задачі дослідження	29.04.2025 – 01.05.2025	виконано
5	Аналіз датасетів та формування набору даних	02.05.2025 – 07.05.2025	виконано
6	Підготовка даних для навчання моделей	08.05.2025 – 10.05.2025	виконано
7	Аналіз методів вирішення задачі	11.05.2025 – 14.05.2025	виконано
	мультимодальної класифікації		
8	Вибір підходів та теоретичне обґрунтування	15.05.2025 – 18.05.2025	виконано
9	Реалізація першого підходу: ансамбль моделей	19.05.2025 – 22.05.2025	виконано
	по модальностях		
10	Реалізація другого підходу: трансформер з	23.05.2025 – 27.05.2025	виконано
	модально-залежними ембедингами		
11	Порівняння результатів моделей	28.05.2025 – 30.05.2025	виконано
12	Написання пояснювальної записки	31.05.2025 – 03.06.2025	виконано
13	Попередній захист	04.06.2025	виконано
14	Захист перед ЕК	05.06.2025	

Дата видачі завдання 21 квітня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ доц. Олексій Турута
(підпис) (посада, власне ім’я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 108 с., 22 рис., 8 табл., 1 дод., 52 джерела.

ГЛИБОКЕ НАВЧАННЯ, КЛАСИФІКАЦІЯ ОБ’ЄКТІВ, КОМП’ЮТЕРНИЙ ЗІР, МУЛЬТИМОДАЛЬНІ ДАНІ, ТРАНСФОРМЕР.

Об’єкт дослідження – процес класифікації об'єктів на мультимодальних даних, які поєднують зображення та елементи текстових описів.

Предмет дослідження – методи класифікації об’єктів на основі зображень та текстових описів.

Мета роботи – дослідження та порівняння сучасних підходів до мультимодальної класифікації товарів, зокрема ансамблевих архітектур та трансформерних моделей зі спільним простором ознак, а також створення комбінованого датасету, адаптованого до українського ринку.

Методи дослідження – теоретичний аналіз (огляд літератури, класифікація існуючих підходів), експериментальний (побудова моделей, формування датасету, оцінка точності), емпіричний (аналіз ефективності підходів у різних сценаріях).

У результаті роботи проведено огляд наявних датасетів та мультимодальних моделей, сформовано комбінований датасет із урахуванням специфіки українського ринку. Реалізовано два підходи до класифікації: класичне ансамблеве поєднання спеціалізованих моделей та трансформерну архітектуру зі спільним простором ознак. Експериментальна оцінка продемонструвала переваги мультимодального підходу в задачах із нечіткими або неоднозначними вхідними даними. Запропоновано рекомендації щодо вибору архітектур залежно від ресурсів та бізнес-вимог.

ABSTRACT

Master’s thesis contains: 108 pp., 22 fig., 8 tabl., 1 ann., 52 references.

COMPUTER VISION, DEEP LEARNING, MULTIMODAL DATA, OBJECT CLASSIFICATION, TRANSFORMER.

The object of research is the process of automated object classification in the retail domain.

The subject of the research is the use of multimodal data (visual and textual) for product classification using deep learning models.

The purpose of the work is to investigate and compare modern approaches to multimodal classification of products, including ensemble-based models and transformer architectures with joint embedding space, as well as to develop a combined dataset adapted to the Ukrainian retail market.

Research methods: theoretical (literature review, classification of existing approaches), experimental (model implementation, dataset construction, performance evaluation), empirical (analysis of accuracy and robustness in real-world scenarios).

As a result, a combined multimodal dataset was constructed based on existing public datasets and extended with Ukrainian products. Two classification approaches were implemented: an ensemble of separate modality-specific models and a transformer-based architecture with multimodal alignment. Comparative experiments demonstrated the advantages of multimodal learning, especially in cases with ambiguous inputs. Practical recommendations were proposed for selecting suitable models based on system constraints and business goals.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

BERT – Bidirectional Encoder Representations from Transformers – бінаправлені кодувальні представлення з трансформерів;

Bi-LSTM – Bidirectional Long Short-Term Memory – бінаправлена довга короткочасна пам’ять;

CLIP – Contrastive Language–Image Pre-training – модель контрастивного навчання мови й зображення;

CNN – Convolutional Neural Network – згорткова нейронна мережа;

FLAVA – Fusion of Language and Vision Architecture – архітектура для мультимодального навчання мови та зору;

FN – False Negative – хибно негативне передбачення;

FP – False Positive – хибно позитивне передбачення;

LSTM – Long Short-Term Memory – довга короткочасна пам’ять (тип рекурентної нейронної мережі);

OCR – Optical Character Recognition – оптичне розпізнавання символів;

ResNet50 – Residual Network 50 – згорткова нейронна мережа з 50 шарами та залишковими з'єднаннями;

RPC – Retail Product Checkout – датасет для розпізнавання товарів на касі;

TP – True Positive – істинно позитивне передбачення;

ViLT – Vision-and-Language Transformer – трансформер для спільної обробки зображення і тексту.

ВСТУП

Сучасні технології штучного інтелекту та комп'ютерного зору відіграють ключову роль у автоматизації бізнес-процесів, зокрема у сфері рітейлу. Автоматизоване детектування та ідентифікація товарів на полицях магазинів дозволяє підвищити ефективність управління асортиментом, зменшити людський фактор у процесах інвентаризації та забезпечити кращу аналітику продажів.

Одним із головних викликів у цьому напрямі є вибір та оптимізація алгоритмів комп'ютерного зору, які можуть працювати з різноманітними умовами освітлення, перспективами камер, частковими перекриттями об'єктів та варіативністю упаковки товарів. Сучасні методи, зокрема нейромережеві моделі, демонструють значні успіхи, проте їхня ефективність залежить від специфічних параметрів реалізації, якості даних та вимог до обчислювальних ресурсів.

Актуальність даної теми зумовлена зростаючою потребою рітейл-мереж у точних та швидких системах автоматичного моніторингу товарів. Успішна імплементація таких технологій може сприяти покращенню контролю за наявністю товарів, зменшенню втрат через відсутність товару на полиці та оптимізації логістичних процесів.

Метою даної роботи є аналіз ефективності сучасних алгоритмів комп'ютерного зору для задач ідентифікації товарів у торгових залах. У ході дослідження будуть розглянуті ключові підходи, проведене порівняння їхньої точності та швидкості роботи, а також оцінена їхня придатність до використання в реальних умовах.

Результати цього дослідження мають практичне значення для підприємств роздрібної торгівлі та розробників систем автоматизованого моніторингу, оскільки дозволять визначити найефективніші алгоритми для конкретних сценаріїв використання.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАВНОКА ЗАДАЧІ

1.1 Загальний огляд предметної галузі

У сучасних умовах розвиток роздрібної торгівлі неможливий без використання технологічних інновацій, що дозволяють ефективно управляти товарними запасами та оптимізувати процеси обслуговування клієнтів. Одним із ключових напрямків цифрової трансформації галузі є застосування передових рішень для управління асортиментом, які сприяють підвищенню ефективності бізнес-процесів та створенню конкурентних переваг.

Сучасна роздрібна торгівля характеризується стрімким розвитком технологій та змінами в поведінці споживачів. Традиційні магазини поступово трансформуються у високотехнологічні торгові центри, де ключову роль відіграють автоматизовані системи управління асортиментом і обслуговуванням клієнтів [1].

Умови сучасної економіки вимагають від підприємств оперативного реагування на змінні ринкові умови, що спричиняє потребу у впровадженні інноваційних підходів для оптимізації логістичних процесів, зниження операційних витрат і підвищення конкурентоспроможності.

Особливістю сучасного ринку є не лише зростання конкуренції, а й швидка динаміка змін у вподобаннях споживачів, що обумовлює необхідність постійного оновлення асортименту та адаптації торгового простору до актуальних тенденцій.

Впровадження сучасних інформаційних технологій дозволяє значно покращити процеси управління асортиментом. Системи автоматичного розпізнавання товарів, побудовані на основі алгоритмів комп’ютерного зору та машинного навчання, сприяють:

– оперативному моніторингу запасів: завдяки використанню камер та сенсорів, інформація про наявність товарів передається в реальному часі, що дозволяє швидко виявляти недостачу або надлишок продукції;

– покращенню точності інвентаризації: автоматизоване розпізнавання зображень зменшує людський фактор, знижує кількість помилок і сприяє більш точному обліку товарних одиниць;

– аналізу поведінки споживачів: інтеграція систем розпізнавання з аналітичними платформами дозволяє виявляти патерни купівельної активності, прогнозувати попит і відповідно коригувати асортимент.

Застосування таких технологій стає особливо актуальним для великих торгових мереж, де масштаб даних і потреба у швидкому прийнятті рішень визначають успіх бізнесу. Таким чином, оптимізація управління асортиментом є одним із ключових завдань сучасної роздрібної торгівлі [2].

Актуальність даного дослідження обумовлена низкою чинників, що безпосередньо впливають на роботу сучасних торгових мереж.

По-перше, постійні зміни умов зйомки, спричинені різними параметрами освітлення та розташування камер, ускладнюють процес автоматичної класифікації товарів.

По-друге, розширення асортименту супроводжується збільшенням кількості товарів із подібним зовнішнім виглядом, що створює додаткові виклики для існуючих систем розпізнавання.

По-третє, динамічність ринку, зумовлена сезонними коливаннями та регулярним оновленням асортименту, вимагає створення адаптивних систем, здатних швидко перенавчатися на основі нових даних.

Зважаючи на ці аспекти, дослідження спрямоване на розробку вдосконалених підходів до класифікації товарів є надзвичайно актуальним. Розроблена модель з використанням сучасних методів машинного навчання і глибоких нейронних мереж може стати ефективним інструментом для підвищення точності автоматичного розпізнавання товарів, що, у свою

чергу, сприятиме оптимізації управління запасами та покращенню обслуговування клієнтів у торгових мережах.

1.2 Аналіз існуючих підходів до класифікації об’єктів

Задача класифікації об’єктів є ключовою у багатьох сферах, включаючи автоматизоване розпізнавання товарів у роздрібній торгівлі. Існуючі підходи до класифікації можна розділити на класичні методи обробки зображень, традиційні алгоритми машинного навчання та сучасні глибокі нейронні мережі. Кожен із цих підходів має свої переваги та обмеження, що визначає доцільність їх використання залежно від специфіки завдання.

Класичні методи обробки зображень мають давню історію у сфері аналізу візуальної інформації і ґрунтуються на математичних алгоритмах, що дозволяють витягувати базові ознаки без використання навчання. До основних підходів цього напряму належать:

– фільтрація зображень. Використовуються середні, медіанні та гауссові фільтри для зменшення шуму та згладжування зображення, що допомагає покращити якість подальшої обробки;

– оператори для виявлення контурів. Алгоритми, такі як оператори Собеля, Лапласа та метод Кенні, дозволяють виділити ключові контури та границі об’єктів, що є необхідним кроком для подальшої сегментації [3];

– сегментація зображень. За допомогою методів порогової обробки, кластеризації та регіональних підходів відбувається розбиття зображення на логічні сегменти, що відповідають окремим об’єктам;

– виділення ключових точок і дескрипторів. Техніки SIFT, SURF, ORB дозволяють визначити стабільні точки на зображенні та побудувати дескриптори, що використовуються для зіставлення з іншими зображеннями або для класифікації.

Основною перевагою класичних методів є їхня простота, відносно низькі обчислювальні витрати та можливість інтерпретації отриманих ознак. Однак, ці методи виявляються недостатньо ефективними у випадках, коли зображення мають низьку якість, містять шум, розмиття або нерівномірне освітлення. Така вразливість до змін умов зйомки обмежує їх застосування в реальних системах розпізнавання товарів у торгових мережах.

До традиційних методів машинного навчання відносять алгоритми, які використовують вручну витягнуті ознаки зображень для побудови моделей класифікації. Серед основних підходів можна виділити:

– метод опорних векторів (SVM). Цей метод дозволяє знаходити оптимальну гіперплощину, що розділяє простір ознак, забезпечуючи хорошу узагальнюваність при відносно невеликій кількості параметрів [4];

– рішучі дерева та ансамблеві методи. Використання рішучих дерев, а згодом ансамблевих моделей, таких як Random Forest, сприяє підвищенню точності за рахунок комбінування рішень багатьох простих моделей;

– логістична регресія. Цей підхід застосовується для бінарної або мультикласової класифікації, дозволяючи визначити ймовірність належності об’єкта до певного класу;

– методи кластеризації. Застосовуються для попереднього групування схожих об’єктів, що спрощує задачу класифікації при подальшій обробці.

Традиційні методи машинного навчання мають переваги у відносній простоті реалізації та менших вимогах до обчислювальних ресурсів порівняно з сучасними підходами. Однак їх успішність значною мірою залежить від якості витягнутих ознак, що, у свою чергу, вимагає ретельної попередньої обробки зображень. При наявності складних або варіативних візуальних патернів ці методи часто виявляються недостатньо точними, що обмежує їх застосування у реальних умовах.

Останнім часом глибоке навчання стало основним підходом у завданнях класифікації та виявлення об'єктів, оскільки воно здатне автоматично визначати релевантні ознаки без необхідності попередньої ручної обробки. Існують як архітектури, орієнтовані виключно на класифікацію, так і комплексні методи, що поєднують класифікацію та детекцію.

Одним із популярних підходів є застосування згорткових нейронних мереж (CNN), таких як VGG, ResNet (рисунок 1.1) або EfficientNet, які демонструють високу точність завдяки багаторівневій структурі, що дозволяє виділяти дедалі абстрактніші особливості зображення. Однак для задач, що потребують не лише розпізнавання класів, а й точного визначення місцезнаходження об'єктів, розроблено спеціалізовані алгоритми, серед яких особливо виділяються YOLO (You Only Look Once) та Faster R-CNN.

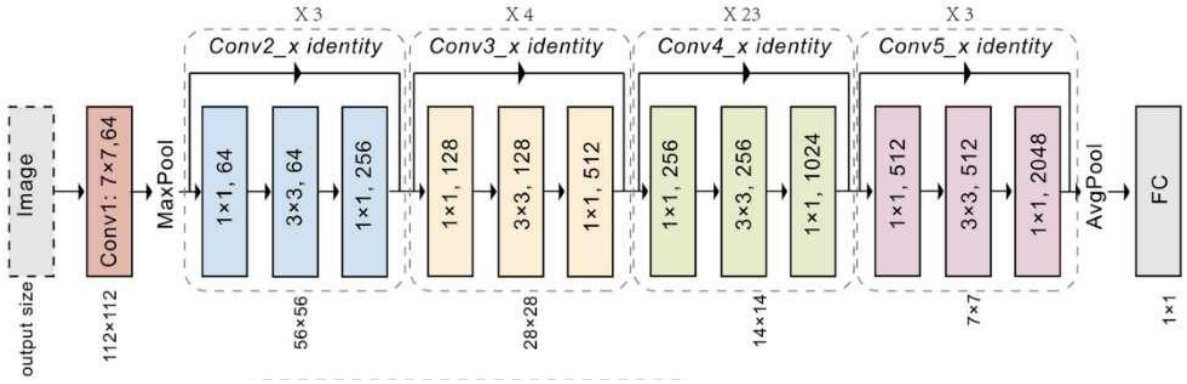

Рисунок 1.1 – Архітектура ResNet

Модель YOLO відрізняється високою швидкістю обробки та можливістю працювати в режимі реального часу, що робить її чудовим вибором для завдань, де швидкодія є критично важливою. Вона розглядає детекцію як задачу регресії, одночасно визначаючи координати та клас об'єкта, що особливо корисно для систем, які працюють із потоковими даними [5].

У той же час Faster R-CNN пропонує більш точний, хоча й ресурсозатратний підхід. Ця модель використовує регіон-пропозиції для знаходження об'єктів, після чого виконує їх класифікацію. Її головна перевага – висока точність у визначенні місцезнаходження та категорій об'єктів, що робить її незамінною у випадках, коли якість детекції є пріоритетною навіть за умови значних обчислювальних витрат [6].

Сучасні глибокі нейронні мережі, зокрема CNN, YOLO (рисунок 1.2) та Faster R-CNN, не лише забезпечують високу точність класифікації, а й ефективно знаходять об'єкти навіть за складних умов, таких як зміни освітлення, шум чи неоднорідний фон. Завдяки методам transfer learning та тонкого налаштування попередньо натренованих моделей, їх можна адаптувати до конкретних сфер, наприклад, для класифікації товарів у роздрібній торгівлі. Проте їх впровадження в реальні системи потребує значних обчислювальних ресурсів та ретельної оптимізації, що залишає простір для подальших удосконалень.

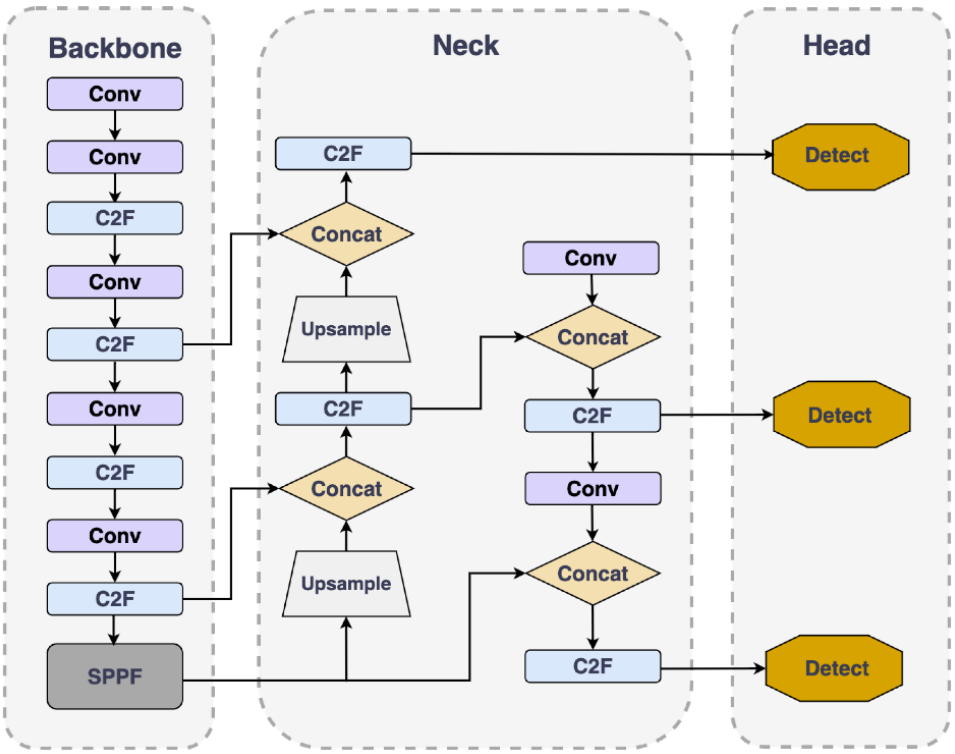

Рисунок 1.2 – Архітектура YOLOv8

1.3 Проблематика класифікації товарів у магазині

Класифікація товарів у роздрібних магазинах є складним завданням, яке потребує врахування багатьох факторів, таких як візуальні характеристики продуктів, складність категоризації та технічні обмеження системи. Ефективна класифікація товарів відіграє ключову роль у процесах автоматизації, управління запасами та оптимізації логістики, а також значно покращує споживчий досвід. Проте існує низка викликів, які необхідно подолати для успішного впровадження цієї технології.

Одним із основних викликів класифікації товарів є варіативність їхнього зовнішнього вигляду. Навіть однакові продукти можуть відрізнятися за:

– освітленням та кутом зйомки під час зчитування штрих-кодів або візуального розпізнавання;

– наявністю різних пакувань для одного товару (наприклад, зміна дизайну або обмежені серії);

– деформаціями упаковки, що можуть спричинити помилки в класифікації.

З технічної точки зору, виклики включають:

– обмежені обчислювальні ресурси для обробки зображень у режимі реального часу;

– необхідність високої точності при розпізнаванні схожих за зовнішнім виглядом товарів;

– інтеграцію алгоритмів глибокого навчання з існуючими системами управління магазинами.

Роздрібні мережі використовують складні ієрархічні системи для організації товарів. Це може створювати додаткові труднощі під час автоматичної класифікації, оскільки один товар може належати до кількох категорій. Наприклад, молочний напій може бути одночасно віднесений до категорій «молочні продукти» та «напої» [7].

Основні проблеми ієрархічної класифікації:

– висока варіативність категорій та їх динамічні зміни;

– складність побудови універсальної моделі, що враховує всі можливі підкатегорії товарів;

– використання різних стандартів класифікації в різних магазинах та регіонах.

Ієрархічна класифікація товарів зазвичай поділяється на дві основні складові: родову класифікацію та видову класифікацію.

Родова класифікація визначає загальні групи товарів, об'єднані спільними характеристиками. Наприклад, усі молочні продукти можуть бути згруповані в категорію «молочні продукти», а всі газовані напої – у категорію «напої» [8].

Видова класифікація деталізує родові категорії до конкретних товарів, враховуючи їхні характеристики, такі як бренд, літраж, склад, ціну, форму випуску тощо. Наприклад, у категорії «молоко» може бути конкретний товар: «Молоко ультрапастеризоване, 1л, ТМ Яготинське». Подібний підхід використовується для всіх товарів, щоб забезпечити точну ідентифікацію кожного найменування.

Поділ товарів за родовими та видовими ознаками є ключовим аспектом ефективної класифікації, оскільки він дозволяє:

– впорядкувати асортимент, спрощуючи пошук товарів;

– забезпечити правильне відображення продуктів у системах управління магазином;

– покращити роботу автоматичних алгоритмів класифікації за допомогою навчання моделей на чітко структурованих даних.

При впровадженні систем автоматичної класифікації необхідно враховувати низку практичних аспектів, зокрема:

– системи класифікації повинні коректно працювати з наявними базами товарів, штрих-кодами та внутрішніми стандартами магазинів;

– моделі повинні мати можливість адаптуватися до змін у товарах, появи нових продуктів та оновлення упаковок;

– у магазинах із великим потоком клієнтів необхідно забезпечити швидку та точну обробку даних без затримок;

– важливим аспектом є навчання співробітників магазину роботі з автоматичними системами класифікації для швидкого виправлення можливих помилок.

Таким чином, успішне впровадження автоматизованих методів класифікації товарів потребує поєднання сучасних технологій глибокого навчання, гнучких алгоритмів категоризації та ефективної інтеграції з існуючими роздрібними системами. Це дозволить підвищити точність розпізнавання, оптимізувати управління запасами та покращити загальну ефективність роботи магазинів [9].

1.4 Постановка задачі дослідження

Автоматизація класифікації товарів є актуальною проблемою для роздрібної торгівлі, оскільки правильне групування продукції впливає на ефективність управління асортиментом, точність пошуку товарів у базах даних та покращення рекомендаційних систем. Сучасні методи детекції об’єктів дозволяють визначати товар на зображенні та його межі (bounding box), проте проблема класифікації товарів за родовими та видовими ознаками залишається відкритою.

Дослідження спрямоване на вирішення завдання родової-видової класифікації товарів, що передбачає автоматичне визначення не лише загальної категорії товару, а й його конкретних характеристик, таких як бренд, об’єм, склад та інші параметри.

Основна проблема дослідження полягає в розробці підходу до автоматичної родової-видової класифікації товарів, який враховує:

– велику кількість товарних категорій та їх динамічні зміни;

– неоднозначність класифікації через належність одного товару до кількох категорій;

– відсутність єдиного стандарту категоризації в різних магазинах та регіонах.

Рішення цієї проблеми має базуватися на поєднанні глибокого навчання та методів обробки даних, що дозволить створити адаптивну систему класифікації.

Для вирішення поставленої проблеми необхідно виконати такі завдання:

– дослідити сучасні підходи до класифікації товарів, включаючи нейронні мережі (ResNet, EfficientNet, Vision Transformers) та методи семантичного аналізу [10];

– сформувати датасет, що міститиме зображення товарів, їх родові та видові категорії, а також супутню інформацію (бренд, об’єм, склад тощо);

– обрати та порівняти різні підходи до класифікації, зокрема методи машинного та глибокого навчання;

– оцінити ефективність запропонованих підходу шляхом проведення експериментів, аналізу отриманих результатів та їх порівняння.

2 ФОРМУВАННЯ НАБОРУ ДАНИХ

2.1 Аналіз існуючих датасетів товарів

У контексті завдань комп'ютерного зору та мультимодальної класифікації об'єктів важливе місце займає якість та репрезентативність датасетів, на яких відбувається навчання моделей. Особливо актуальним це питання стає при розробці систем розпізнавання товарів у роздрібній торгівлі, де кінцевою метою є створення надійних алгоритмів ідентифікації та класифікації продуктів за їхніми візуальними та текстовими характеристиками. У цьому підрозділі буде проведено детальний аналіз існуючих датасетів товарів, які можуть бути використані для вирішення поставленого завдання.

Проєкт Open Food Facts представляє собою відкриту базу даних харчових продуктів, яка містить інформацію про складники, поживну цінність, країни походження та інші характеристики продуктів харчування з усього світу. Цей ресурс розвивається як громадська ініціатива, де користувачі можуть додавати нові продукти та оновлювати інформацію про існуючі [11].

Open Food Facts відрізняється від інших датасетів тим, що представляє собою не просто фіксований набір даних, а динамічну систему, яка постійно оновлюється (щомісяця) та розширюється силами спільноти. Станом на момент дослідження, база даних містить інформацію про понад 2,5 мільйона продуктів з більш ніж 150 країн світу.

Структура даних Open Food Facts включає:

– текстову інформацію (назва продукту, бренд, категорія, країна походження);

– харчову цінність (калорійність, вміст білків, жирів, вуглеводів);

– перелік інгредієнтів;

– штрих-коди продуктів;

– упаковка та екологічна інформація;

– зображення продуктів (переважно фронтальна частина упаковки).

Основна перевага цього ресурсу полягає у великому обсязі та різноманітності представлених даних, що потенційно дозволяє створювати моделі з широким охопленням товарів. Крім того, відкритий характер даних та можливість використання API спрощує інтеграцію цієї інформації в дослідницькі проєкти.

Кількість товарів по різним категоріям можна побачити на рисунку 2.1.

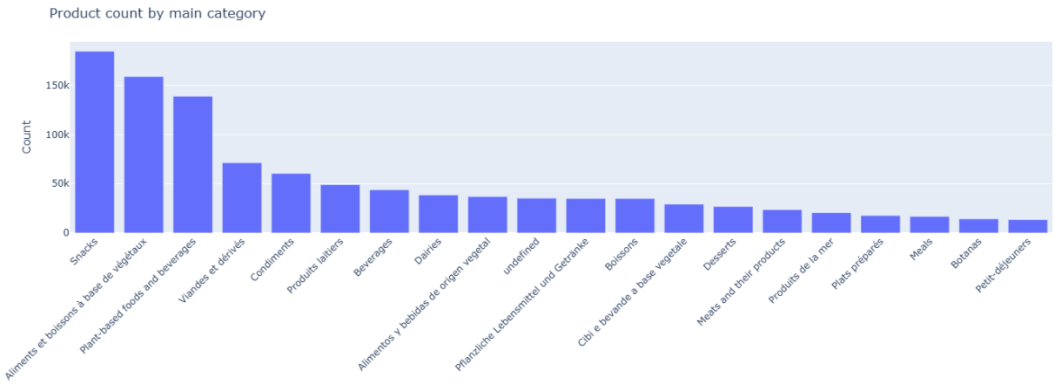

Рисунок 2.1 – Розподіл продуктів по категоріям представлених в даних від Open Food

Однак, незважаючи на очевидні переваги, Open Food Facts має ряд суттєвих обмежень для задач мультимодальної класифікації:

– неоднорідна якість даних. Оскільки інформація додається користувачами, точність та повнота даних може варіюватися [12];

– нерівномірне представлення продуктів з різних регіонів (європейські країни представлені краще, ніж інші);

– відсутність систематизованого підходу до збору візуальних даних – більшість продуктів представлені лише одним зображенням, що є недостатнім для надійного навчання моделей комп'ютерного зору [13];

– критично важливе обмеження: неможливість знайти декілька фотографій одного й того ж товару під різними кутами, за різного освітлення чи в різних умовах, що значно ускладнює створення робастних моделей класифікації.

Ця остання характеристика є ключовим недоліком, який робить Open Food Facts менш придатним для задач, які потребують різноманітних візуальних представлень одного й того ж товару. Такі задачі включають розпізнавання товарів у реальних умовах магазинів, коли продукт може бути сфотографований з різних ракурсів, за різного освітлення або частково закритим.

Аналіз статистичного розподілу категорій продуктів у датасеті Open Food Facts дозволяє виявити ще один важливий аспект – значну незбалансованість класів. Деякі категорії продуктів представлені тисячами зразків, тоді як інші – лише кількома десятками, що створює додаткові виклики при навчанні моделей. З країнами така-ж проблема, що можна побачити на рисунку 2.2.

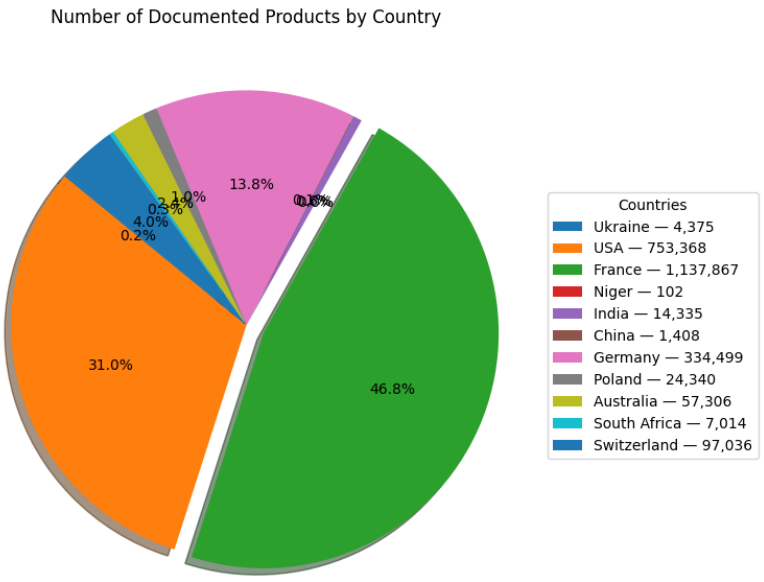

Рисунок 2.2 – Розподіл продуктів по країнам представлених в даних від Open Food

Таким чином, Open Food Facts, попри свій великий обсяг та широке охоплення, краще підходить для завдань, пов'язаних з аналізом текстової інформації про продукти, харчової цінності та складників, але має суттєві обмеження для задач мультимодальної класифікації об'єктів, де потрібна різноманітність візуальних представлень продуктів [14].

Приклад одного заповненого продукту можна побачити у лістингу 2.1

Лістинг 2.1 – Вигляд структури JSON заповненого продукту

```
{
  "code": "0000101209159",
  "product_name": [
    {
      "lang": "main",
      "text": "Véritable pâte à tartiner noisettes chocolat noir"
    },
    {
      "lang": "fr",
      "text": "Véritable pâte à tartiner noisettes chocolat noir"
    }
  ],
  "brands": "Bovetti",
  "categories": "Petit-déjeuners,Produits à tartiner,Produits à tartiner sucrés,Pâtes à tartiner,Pâtes à tartiner aux noisettes,Pâtes à tartiner au chocolat,Pâtes à tartiner aux noisettes et au cacao",
  "nutriscore_grade": "e",
  "nutriments": …,
"completeness": 0.6875
}
```

Перейдемо до іншого, Grocery Store Dataset, який представляє собою спеціалізований набір даних, розроблений саме для завдань комп'ютерного зору і класифікації товарів у контексті роздрібної торгівлі. На відміну від Open Food Facts, цей датасет створювався цілеспрямовано для дослідницьких цілей і має більш структурований характер [15].

Датасет складається з 5125 зображень, що належать до 81 класу продуктів, організованих у 43 категорії. Кожен клас містить від 40 до 97 зображень, що представляють різні екземпляри відповідного товару. Зображення мають роздільну здатність 1024×768 пікселів, що забезпечує достатню деталізацію для аналізу візуальних особливостей продуктів.

Особливість цього датасету полягає в тому, що зображення були отримані в реальних умовах продуктового магазину, з фіксованої точки огляду (імітуючи стаціонарну камеру спостереження), але з варіаціями освітлення, розташування товарів та часткового перекриття об'єктів. Такий підхід до збору даних дозволяє створювати моделі, більш пристосовані до реальних умов експлуатації.

Структура каталогізації в Grocery Store Dataset є ієрархічною: продукти спочатку поділяються на категорії (наприклад, фрукти, овочі, упаковані продукти), а потім на підкатегорії або конкретні типи товарів. Така організація даних дозволяє проводити експерименти як з бінарною класифікацією, так і з мультикласовою, а також досліджувати ієрархічні підходи до класифікації.

Одним із ключових переваг Grocery Store Dataset є наявність декількох зображень для кожного типу продукту, що дозволяє моделям вивчати інваріантність до певних змін у візуальній репрезентації. Крім того, для багатьох зображень доступні анотації у вигляді обмежувальних рамок (bounding boxes), що дозволяє використовувати цей датасет не лише для задач класифікації, але й для об'єктного детектування.

Проте, незважаючи на високу якість та цільове призначення, Grocery Store Dataset має суттєве обмеження – порівняно невеликий розмір. З 5125

зображень та лише 81 класом товарів, цей датасет не може забезпечити достатнє охоплення різноманітності продуктів, представлених у сучасних супермаркетах, де асортимент може сягати десятків тисяч найменувань. Це значно обмежує можливості масштабування моделей, навчених виключно на цьому датасеті, до реальних промислових застосувань, але повністю охоплює дослідницький потенціал [16].

Ще одним аспектом, який варто враховувати, є географічна специфіка представлених товарів – датасет містить переважно продукти, характерні для західноєвропейських ринків, і може не відображати особливості асортименту в інших регіонах, зокрема в Україні.

Аналіз структури Grocery Store Dataset також виявив нерівномірний розподіл зразків між різними класами, хоча ця незбалансованість не така критична, як у випадку з Open Food Facts. Різниця в кількості зображень між найбільш та найменш представленими класами становить близько 2,5 рази, що є значно кращим показником з точки зору навчання збалансованих моделей. Це можна побачити у таблиці 2.1.

Таблиця 2.1 – Розподіл класів у Grocery Store Dataset

Рід	Клас	Кількість зображень
Apple	Golden-Delicious	45
	Granny-Smith	59
	Pink-Lady	59
	Red-Delicious	50
	Royal-Gala	65
Plum	Plum	22
Soyghurt	Alpro-Blueberry-Soyghurt	28
	Alpro-Vanilla-Soyghurt	23
...	...	5125

Таким чином, Grocery Store Dataset добре підходить для початкових експериментів та створення базових моделей класифікації товарів, але потребує доповнення додатковими даними для створення більш масштабних і універсальних систем розпізнавання. Приклади фото можна побачити на рисунку 2.3.

Рисунок 2.3 – Приклади фото з датасету Grocery Store Dataset

Наступним важливим ресурсом для дослідження є датасет Retail Product Checkout (RPC), який був розроблений спеціально для вирішення задач класифікації товарів у контексті автоматизованих касових систем. На відміну від попередньо розглянутих колекцій даних, RPC фокусується на сценарії розпізнавання товарів безпосередньо в процесі оформлення покупок, що має свої специфічні вимоги та обмеження [17].

Датасет Retail Product Checkout був представлений дослідниками з AiFi Inc. та Стенфордського університету в 2019 році і позиціонується як перший великомасштабний датасет, спеціально призначений для задач розпізнавання товарів на касі. Він містить 30,000 зображень, що охоплюють 200 унікальних товарів роздрібної торгівлі. Особлива цінність цього датасету полягає в тому, що він імітує реальні умови процесу оформлення покупок, з характерними для цього контексту варіаціями та складнощами.

Структурно датасет RPC організований досить чітко: 200 товарів поділені на 17 категорій (рисунок 2.4), що дозволяє проводити як родову (за категоріями), так і детальну (за конкретними товарами) класифікацію. Це суттєва перевага з точки зору дослідження ефективності різних підходів до класифікації на різних рівнях абстракції. Категорії включають такі групи як упаковані харчові продукти, напої, свіжі фрукти та овочі, товари для дому та інші поширені типи продуктів, що зустрічаються в супермаркетах.

Процес збору даних для RPC датасету проводився зі значною увагою до відтворення реальних умов використання. Для кожного товару було зроблено близько 150 зображень у різних конфігураціях, що включали:

– варіації орієнтації товару (повороти, нахили);

– різні умови освітлення (враховуючи як природне, так і штучне джерело світла);

– різні фони (що імітують поверхні касових столів);

– часткове перекриття товарів (що є типовою ситуацією при скануванні декількох товарів одночасно).

Цей підхід до формування даних забезпечує високу варіативність візуальних представлень кожного товару, що є критично важливим фактором для створення робастних моделей класифікації, здатних працювати в умовах реального роздрібного середовища.

Важливою особливістю датасету RPC є детальна анотація зображень, яка включає не лише мітки класів, але й обмежувальні рамки для кожного товару на зображенні. Ця додаткова інформація дозволяє використовувати датасет не тільки для задач класифікації, але й для навчання моделей об'єктного детектування, що є важливим аспектом повноцінних систем автоматизованого оформлення покупок [18].

Аналізуючи статистичний розподіл зразків у межах датасету RPC, можна відзначити досить збалансований характер представлення різних товарів. Кожен з 200 товарів має приблизно однакову кількість зображень (близько 150), що мінімізує проблему незбалансованості класів

при навчанні моделей. Проте на рівні категорій спостерігається певна нерівномірність (рисунок 2.4) розподілу – деякі категорії, такі як упаковані харчові продукти, представлені значно більшою кількістю унікальних товарів порівняно з іншими категоріями, наприклад, свіжими овочами.

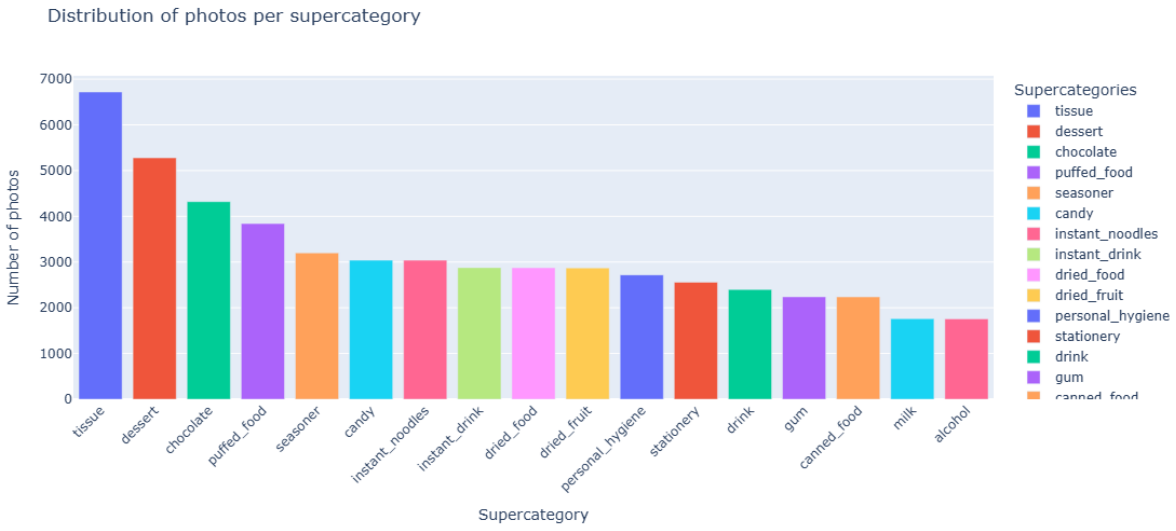

Рисунок 2.4 – Розподіл кількості унікальних товарів за категоріями в датасеті Retail Product Checkout

Детальний аналіз зображень у датасеті RPC виявив певні специфічні характеристики, які варто враховувати при використанні цих даних для навчання моделей. По-перше, всі зображення мають високу роздільну здатність (2048×1536 пікселів), що забезпечує добру деталізацію навіть дрібних елементів товарів, таких як текст на упаковці або текстура поверхні. По-друге, характерною особливістю датасету є наявність значної кількості зображень (рисунок 2.5) з багатьма товарами одночасно (до 17 об'єктів на одному зображенні), що створює додаткові виклики для алгоритмів детектування та класифікації [19].

Окрім кількісних характеристик, важливим аспектом аналізу датасету є оцінка його репрезентативності з точки зору охоплення реального асортименту роздрібної торгівлі. В цьому контексті RPC демонструє добру

збалансованість між популярними брендами міжнародного рівня та локальними товарами, що робить його потенційно придатним для використання в різних географічних регіонах. Проте, як і у випадку з іншими розглянутими датасетами, RPC має специфіку американського та європейського ринків, що може обмежувати його безпосереднє застосування для систем, орієнтованих на український ринок.

Важливим аспектом при оцінці придатності датасету для мультимодальної класифікації є наявність додаткової нетекстової інформації про товари. У випадку з RPC такої інформації немає.

Рисунок 2.5 – Приклади зображень з датасету Retail Product Checkout, що демонструють різні категорії товарів та умови зйомки

У контексті дослідження можливостей розширення датасету, RPC надає певні переваги порівняно з іншими розглянутими ресурсами. Висока якість анотацій та чітка методологія збору даних створюють надійну основу для потенційного розширення набору даних за рахунок додавання нових товарів, зокрема специфічних для українського ринку. Технічний протокол збору та обробки зображень, використаний при створенні RPC, може бути відтворений для додаткових товарів, забезпечуючи узгодженість даних.

Проте варто відзначити, що розширення датасету RPC потребуватиме значних ресурсів, оскільки для кожного нового товару необхідно створити близько 150 різноманітних зображень з дотриманням методології

оригінального датасету. Це створює певні практичні обмеження для маломасштабних дослідницьких проєктів.

Підсумовуючи аналіз датасету Retail Product Checkout, можна стверджувати, що він є цінним ресурсом для дослідження моделей класифікації товарів у контексті роздрібної торгівлі, особливо для задач родової класифікації. Його структура, якість зображень та методологія збору даних забезпечують надійну основу для експериментів з різними архітектурами моделей та підходами до обробки візуальної інформації. Водночас, для повноцінного вирішення задачі мультимодальної класифікації об'єктів у контексті українського ринку, цей датасет потребує доповнення додатковими даними, що охоплюють специфічні для нашого регіону товари та більш детальною текстовою інформацією для мультимодального аналізу [20].

Проведений аналіз трьох датасетів – Open Food Facts, Grocery Store Dataset та Retail Product Checkout – виявив їхні порівняльні характеристики для задач мультимодальної класифікації товарів. У таблиці 2.2 представлено узагальнену характеристику цих датасетів.

Таблиця 2.2 – Порівняльна характеристика аналізованих датасетів

Характеристика	Open Food Facts	Grocery Store Dataset	Retail Product Checkout
Загальна кількість зразків	> 2,5 млн	5125	~50 000
Кількість категорій	> 20 000	43	17
Кількість класів товарів	> 100 000	81	200
Наявність текстової інформації	Розширена	Середня	Мінімальна

Продовження таблиці 2.2

Якість зображень	Змінна	Середня	Висока
Варіативність умов зйомки	Низька	Висока	Середня
Наявність декількох зображень одного товару	Ні	Так	Так
Збалансованість класів	Дуже низька	Висока	Середня
Актуальність для українського ринку	Середня	Часткова	Низька
Можливість розширення	Висока	Середня	Середня

Аналіз показав, що Grocery Store Dataset має суттєві переваги для задач мультимодальної класифікації товарів. Цей датасет відрізняється високою якістю зображень, значною варіативністю умов зйомки та наявністю декількох зображень одного товару, що критично важливо для навчання стійких моделей розпізнавання. Також Grocery Store Dataset демонструє високу збалансованість класів, що забезпечує більш точне навчання моделей без систематичних помилок класифікації.

2.2 Методика формування комбінованого датасету

На основі проведеного аналізу існуючих датасетів було розроблено методику формування комбінованого датасету для задач мультимодальної класифікації товарів. Ця методика враховує переваги та обмеження кожного

з розглянутих наборів даних і спрямована на створення оптимального датасету для українського ринку.

В якості основи комбінованого датасету було обрано Grocery Store Dataset, що зумовлено його значними перевагами у контексті якості даних. Цей датасет відрізняється якістю зображень, значною варіативністю умов зйомки, наявністю декількох зображень одного товару з різних ракурсів та збалансованістю класів.

Непогана якість зображень забезпечує чітку видимість характерних ознак товарів – логотипів, текстових написів, кольорових схем та форм, що є критично важливим для точної класифікації. Варіативність умов зйомки у Grocery Store Dataset сприяє формуванню моделей, здатних розпізнавати товари в різних умовах освітлення, з різних ракурсів та на різних фонах, що значно підвищує їхню ефективність у реальних умовах застосування.

Особливо цінною характеристикою Grocery Store Dataset є наявність від 10 зображень кожного товару, зроблених з різних ракурсів. Це дозволяє моделям ефективно вивчати інваріантні ознаки товарів, які залишаються незмінними незалежно від умов спостереження. Збалансоване представлення різних класів товарів мінімізує ризик систематичного зміщення при навчанні моделей класифікації.

Для подолання обмеження Grocery Store Dataset щодо кількості представлених класів товарів було здійснено інтеграцію додаткової категорії з Retail Product Checkout, а саме Candy. Відбір цієї категорії здійснювався за критеріями відсутності у базовому датасеті та наявності міжнародних аналогів. Ця категорія налічує 10 різних класів.

Суттєвим викликом при інтеграції даних з Retail Product Checkout була наявність китайських написів на значній частині товарів. Для вирішення цієї проблеми було застосовано комбінований підхід з використанням технологій оптичного розпізнавання тексту (OCR) та подальшою транслітерацією і перекладом.

Варто зазначити, що в рамках даного дослідження не розглядається розробка OCR-системи. Для отримання текстової інформації з зображень використовувались відкриті моделі OCR, які забезпечують високоточне розпізнавання тексту різними мовами. Отримані результати OCR були додатково валідовані через ручну перевірку для забезпечення коректності текстової інформації.

Критичним компонентом комбінованого датасету стало включення зразків українських товарів, необхідність якого зумовлена їх низькою представленістю у міжнародних датасетах. Для цього було проведено фотографування понад 12 унікальних товарів українських виробників у 3 нових категоріях [21].

Кожен товар фотографувався з 7–12 різних ракурсів в умовах, що імітують реальні сценарії використання системи розпізнавання: у руці в магазину, в касовій зоні, у домашньому середовищі.

Для забезпечення максимальної інформативності, фотографування здійснювалося з варіативністю ракурсів, відстаней, освітлення, фону та положення товару. Така різноманітність умов зйомки є критично важливою для навчання робастних моделей класифікації (рисунок 2.6).

Рисунок 2.6 – Приклад зображень з власноруч згенерованого датасету

Для кожного товару було сформовано уніфікований набір метаданих у форматі JSON, який включав структуровану інформацію про товар. Структура JSON-файлу можна побачит у лістингу 2.2.

Лістинг 2.2 – Структура структури JSON для продукту

```
{
       "product_id": "milk-12345",
       "name ": “Milk Arla Standard Milk”,
       "classification": {
         "category": "Milk",
         "subcategory": " Arla-Standard-Milk ",
       },
       "ocr_text": "Sommarmjolk 1.4% 900ml"
       "images":    ["milk-12345_front.jpg",    "    milk-
12345_back.jpg", " milk-12345_side.jpg"]
}
```

Така структура метаданих дозволяє ефективно використовувати різні модальності даних при навчанні моделей класифікації, зокрема, поєднувати візуальну інформацію з текстовою для підвищення точності розпізнавання.

Для класів з недостатньою кількістю зразків було застосовано техніки аугментації даних, включаючи геометричні трансформації, колірні модифікації та симуляцію різних умов освітлення. Для класів з надмірною кількістю зразків було здійснено стратегічний відбір найбільш репрезентативних екземплярів вручну.

2.3 Аналіз структури та особливостей сформованого датасету

Сформований комбінований датасет має низку специфічних характеристик, які відрізняють його від існуючих наборів даних та забезпечують його ефективність для вирішення задач мультимодальної класифікації товарів у контексті українського ринку [22].

Комбінований датасет включає загалом 8132 зображення, що охоплюють 103 унікальних товарів у 85 категоріях. Розподіл зображень за джерелами наступний:

– Grocery Store Dataset: 5,125 зображень (81 унікальних товарів у 43 категоріях);

– Retail Product Checkout: 2,862 зображень (10 унікальних товарів у 1 категорії);

– українські товари: 145 зображень (12 унікальних товарів у 3 категоріях).

Середня кількість зображень на один товар становить 78.6, що забезпечує достатню варіативність представлення кожного товару для навчання робастних моделей. Найбільша кількість зображень на товар спостерігається у категорії фруктів (95.2) та молока (84.8), найменша – у категорії сметани (8.8).

Сформований датасет вирізняється мультимодальністю, адже поєднує в собі кілька типів даних. Він містить візуальну інформацію – зображення товарів, зроблені з різних ракурсів, у різних умовах освітлення та на різному фоні. До того ж, до нього включено текстові дані: як структуровані описи товарів, так і текст, видобутий за допомогою OCR-аналізу зображень. Окреме місце займають категоріальні дані, організовані у вигляді ієрархічної класифікації [23].

У порівнянні з базовими датасетами, цей комбінований набір демонструє помітні переваги. Він охоплює значно ширший спектр товарів – на 35% більше унікальних категорій, ніж у Grocery Store Dataset. Його також вирізняє локальна специфіка: включені товари, які є характерними саме для українського ринку, і яких зазвичай немає в міжнародних наборах даних. Крім того, структура метаданих тут є розширеною, з більшим набором параметрів, що дозволяє глибше аналізувати дані. І найголовніше – мультимодальний підхід дозволяє поєднувати зображення та текстову інформацію, що суттєво підвищує ефективність класифікації.

2.4 Підготовка даних для навчання моделей

Підготовка даних є критичним етапом, що значною мірою визначає ефективність моделей мультимодальної класифікації товарів. Розроблений процес підготовки даних включає ряд етапів, спрямованих на оптимізацію використання сформованого датасету.

Для забезпечення оптимальної якості вхідних даних було застосовано наступні процедури попередньої обробки зображень:

– стандартизація розміру: усі зображення були приведені до єдиного розміру 512×512 пікселів з використанням методу збереження пропорцій та заповнення пустих областей нейтральним фоном [24];

– нормалізація колірного простору: конвертація усіх зображень до колірного простору RGB та нормалізація значень пікселів до діапазону [0, 1];

– видалення фону: для частини зображень було застосовано алгоритми сегментації для видалення фону та виділення самого товару, що дозволяє моделям краще зосередитися на релевантних ознаках.

Текстові дані, отримані з OCR-обробки зображень та з метаданих товарів, пройшли наступні етапи підготовки:

– нормалізація тексту: приведення тексту до нижнього регістру, видалення зайвих пробілів, нормалізація пунктуації;

– мовна адаптація: обробка багатомовних текстів, зокрема українською, китайська, польська та англійською мовами;

– токенізація: розбиття тексту на токени з урахуванням особливостей кожної мови;

– видалення стоп-слів: виключення неінформативних слів, специфічних для кожної з мов;

– лематизація: приведення слів до базової форми з урахуванням морфологічних особливостей української мови [25].

Як уже згадувалося раніше, для отримання текстової інформації з зображень у рамках цього дослідження було використано готові рішення – зокрема, відкриту OCR-модель, реалізовану у Python-фреймворку EasyOCR. Створення власної системи розпізнавання тексту (OCR) не входило до переліку завдань, оскільки основна увага була зосереджена саме на подальшій мультимодальній класифікації вже зібраних та попередньо оброблених даних [26].

З метою підвищення стійкості моделей до змін вхідних даних (робастності) та для часткового усунення проблеми дисбалансу класів було впроваджено процедуру аугментації. Варто зазначити, що ця аугментація застосовувалася виключно до зображень українських товарів, а саме – фото сметани, майонезу та кетчупу. Інші категорії або мовні групи у процесі аугментації участі не брали.

Під час аугментації були реалізовані наступні підходи:

– геометричні трансформації: поворот зображення, зміна масштабу, зсув, горизонтальне відображення – ці методи дозволяють моделі навчитися розпізнавати об’єкти при різному положенні камери або упаковки;

– колірні трансформації: варіювання яскравості, контрасту, насиченості та колірного балансу, що імітує різні умови освітлення при зйомці;

– додавання шуму: застосування гаусівського шуму, ефекту «сіль-перець» та легке розмиття допомагає зробити модель менш чутливою до недосконалостей фото;

– симуляція різного фону та освітлення, що також моделює реальні умови фотографування товарів у побутових або торгових середовищах.

Для оцінки ефективності моделей та уникнення проблеми перенавчання було здійснено розбиття датасету на три підмножини:

– навчальна вибірка (70%): використовується безпосередньо для навчання моделей;

– валідаційна вибірка (15%): використовується для налаштування гіперпараметрів та раннього зупинення процесу навчання;

– тестова вибірка (15%): використовується для фінальної оцінки якості моделей.

Розбиття здійснювалося з використанням стратифікованого підходу, який забезпечує пропорційне представлення усіх класів у кожній з підмножин [27].

Важливою особливістю розбиття є забезпечення того, щоб різні зображення одного й того ж товару потрапляли до однієї підмножини, що запобігає «протіканню» інформації між навчальною та тестовою вибірками.

3 АНАЛІЗ ПІДХОДІВ КЛАСИФІКАЦІЇ МУЛЬТИМОДАЛЬНИХ ДАНИХ

3.1 Огляд сучасних методів обробки мультмиодальних даних

У сучасному світі даних, що постійно розширюється, одномодальні підходи до класифікації об'єктів дедалі частіше демонструють свої обмеження. Мультимодальний аналіз, що поєднує інформацію з різних джерел сприйняття (модальностей), став передовим напрямком досліджень, особливо в контексті класифікації об'єктів. У цьому підрозділі розглянемо теоретичні основи та практичні реалізації сучасних мультимодальних систем, зосереджуючись на інтеграції візуальної та текстової інформації.

Мультимодальні системи класифікації ґрунтуються на фундаментальному принципі, що різні типи даних можуть надавати взаємодоповнюючу інформацію про об'єкт. Інтеграція візуальних та текстових модальностей дає можливість подолати обмеження, притаманні кожній окремій модальності.

Основна теоретична концепція мультимодальних систем полягає у створенні спільного представлення або простору ознак, де семантично пов'язані елементи з різних модальностей знаходяться близько один до одного. Це досягається шляхом навчання проєкцій, які відображають дані різних модальностей у цей єдиний простір (рисунок 3.1) [28].

Математично це можна представити як пошук функцій відображення для кожної модальності (формули 3.1 – 3.2) :

$$f_{visual}: X_{visual} \rightarrow Z, \tag{3.1}$$

$$f_{text}: X_{text} \rightarrow Z, \tag{3.2}$$

де X_{visual} та X_{text} – вхідні простори для візуальних і текстових даних відповідно;

Z – спільний простір ознак.

Ключовою метою є мінімізація відстані між проєкціями семантично пов'язаних елементів у просторі Z (рисунок 3.1), що формалізується через функцію втрат (формула 3.3):

$$L = \sum d\left(f_{visual}(x_i), f_{text}(y_i)\right), \tag{3.3}$$

де d – функція відстані (найчастіше косинусна відстань);

x_i – візуальне представлення;

y_i – відповідний текстовий опис.

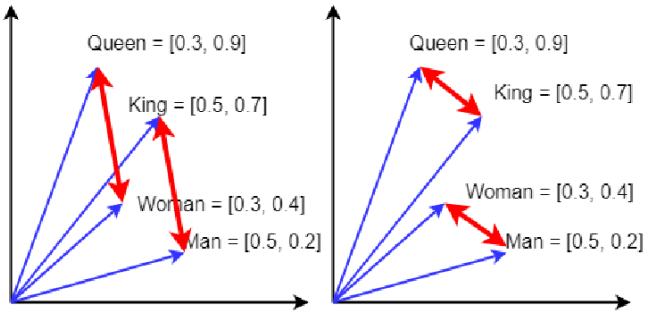

Рисунок 3.1 – Схематичне зображення проєкції текстових даних у спільний простір ознак, де семантично схожі елементи розташовані близько

Важливим теоретичним аспектом є також принцип контрастивного навчання, що лежить в основі багатьох сучасних мультимодальних моделей. Цей підхід оптимізує модель таким чином, щоб зменшити відстань між позитивними парами (відповідні одна одній візуальні та текстові репрезентації) і збільшити відстань між негативними парами [29].

Розглянемо три передові моделі, що демонструють різні підходи до мультимодальної класифікації: CLIP, FLAVA та ViLT.

Модель CLIP, розроблена OpenAI, використовує контрастивне навчання для створення універсальних візуальних класифікаторів. Архітектурно CLIP складається з двох окремих енкодерів (рисунок 3.2):

– візуального енкодера (зазвичай на основі Vision Transformer або ResNet);

– текстового енкодера (на основі Transformer).

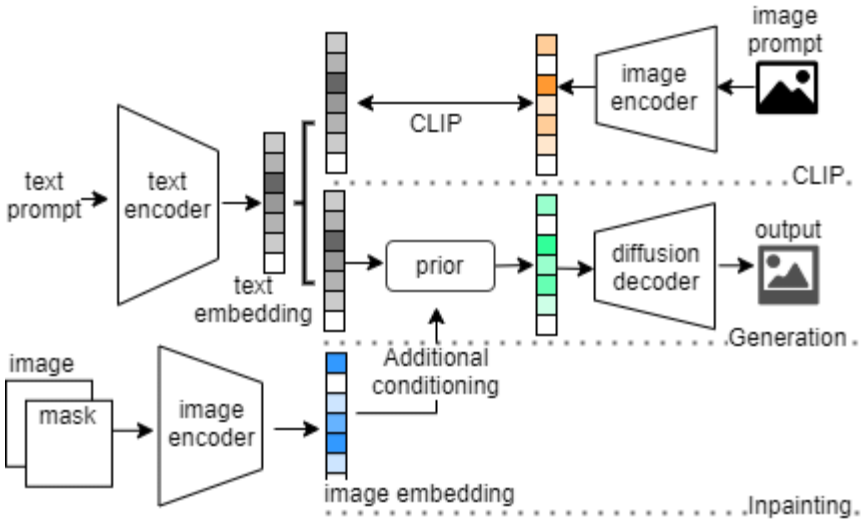

Рисунок 3.2 – Архітектурна схема моделі CLIP з візуальним та текстовим енкодерами

Особливістю CLIP є те, що модель навчається на масштабних наборах пар «зображення-текст», зібраних з інтернету, без явної розмітки категорій. Контрастивна функція втрат оптимізує близькість репрезентацій відповідних пар зображень і текстів у спільному просторі [30].

Головною перевагою CLIP для класифікації товарів є можливість нульового (zero-shot) або малозразкового (few-shot) навчання. Модель може класифікувати об'єкти за категоріями, яких не бачила під час навчання, просто порівнюючи вбудовування зображення з вбудовуваннями текстових описів можливих категорій.

FLAVA представляє більш інтегрований підхід до мультимодальної обробки, пропонуючи єдину модель для візуальних, текстових та мультимодальних завдань. Архітектурно FLAVA містить (рисунок 3.3):

– модульний енкодер зображень;

– модульний текстовий енкодер;

– мультимодальний енкодер для узгодження інформації з обох модальностей.

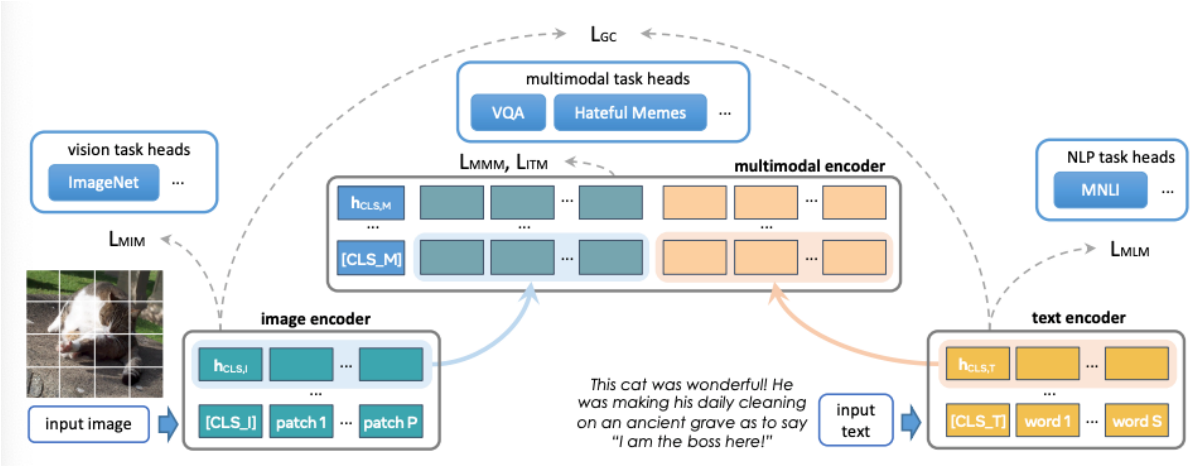

Рисунок 3.3 – Архітектурна схема моделі FLAVA з наголосом на мультимодальному енкодері та його взаємодії з модальними енкодерами

На відміну від CLIP, FLAVA використовує не лише контрастивне навчання, а й додаткові методи: маскування зображень, маскування тексту та мультимодальне маскування, що робить її більш гнучкою для різних задач [31].

Для класифікації товарів FLAVA демонструє особливу ефективність завдяки здатності враховувати тонкі взаємозв'язки між візуальними та текстовими характеристиками товарів.

ViLT відрізняється мінімалістичним підходом, уникаючи складних модулів попередньої обробки зображень. Архітектурно модель складається з (рисунок 3.4):

– простого перетворення зображення в послідовність патчів;

– єдиного трансформера для обробки конкатенованих послідовностей зображення та тексту.

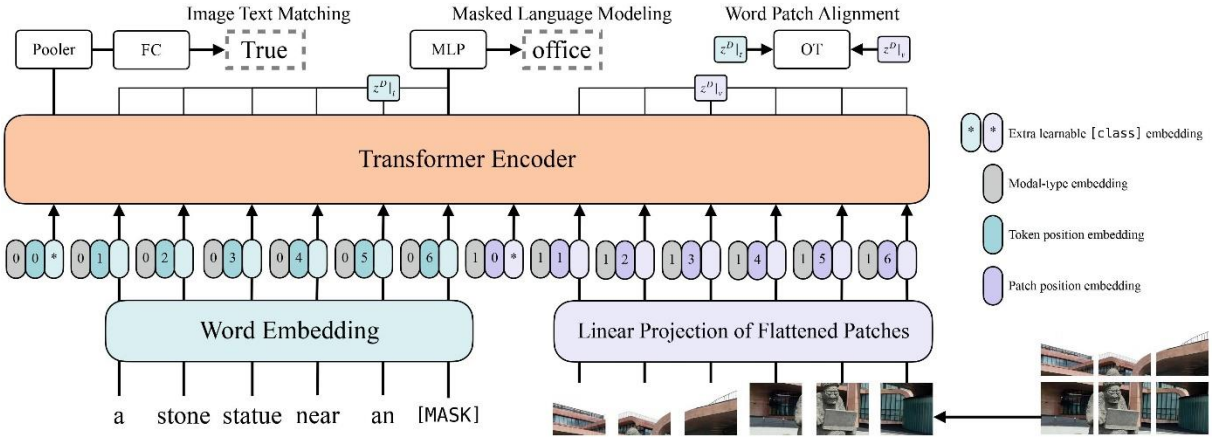

Рисунок 3.4 – Схема архітектури ViLT з патчами зображення та токенами тексту, що подаються в спільний трансформер

Ця архітектура значно зменшує обчислювальну складність, що особливо важливо для обробки великих наборів даних про товари. При цьому ViLT підтримує баланс між ефективністю та продуктивністю, що робить її привабливою для реальних систем класифікації товарів [32].

Однією з фундаментальних проблем у мультимодальній класифікації є семантичний розрив між візуальними та текстовими даними. Ця проблема виникає через принципово різну природу представлення інформації в різних модальностях [33].

Візуально-семантичний розрив (visual-semantic gap) проявляється в тому, що:

– візуальні ознаки зазвичай представлені низькорівневими характеристиками (кольори, текстури, форми), тоді як текстові описи містять високорівневі семантичні концепти;

– текстові описи можуть включати абстрактні властивості, які складно відобразити візуально (наприклад, «міцний», «надійний», «високоякісний»);

– візуальні зображення можуть містити деталі, які важко описати лаконічним текстом.

Сучасні підходи до подолання семантичного розриву включають:

– створення спільних вбудовувань, які зберігають семантичну схожість між модальностями;

– використання механізмів уваги для фокусування на релевантних частинах зображення та відповідних фрагментах тексту;

– ієрархічне представлення даних, яке дозволяє зіставляти різні рівні абстракції в різних модальностях;

– попередня обробка для нормалізації та стандартизації представлень у різних модальностях.

Ефективність мультимодальних систем класифікації товарів значною мірою залежить від того, наскільки вдало вони поєднують інформацію з різних джерел. Існує кілька основних стратегій інтеграції модальностей.

При ранньому злитті (Early fusion) інтеграція відбувається на рівні вхідних даних або низькорівневих ознак. Хоча цей підхід концептуально простий (рисунок 3.5), він має обмеження через різну природу представлення різних типів даних [34].

Пізнє злиття (Late fusion) передбачає окрему обробку різних модальностей і комбінування результатів на етапі прийняття рішень. Цей підхід часто реалізується як зважена комбінація результатів окремих класифікаторів (формула 3.3):

$$F_{combine} = \alpha * F_{visual} + (1 - \alpha) * F_{text}, \quad (3.3)$$

де α – коефіцієнт, що визначає вагу кожної модальності.

Гібридні методи (Hybrid fusion) поєднують переваги раннього та пізнього злиття, інтегруючи модальності на проміжних рівнях абстракції. До цієї категорії належать моделі з механізмами крос-модальної уваги, такі як FLAVA та ViLT.

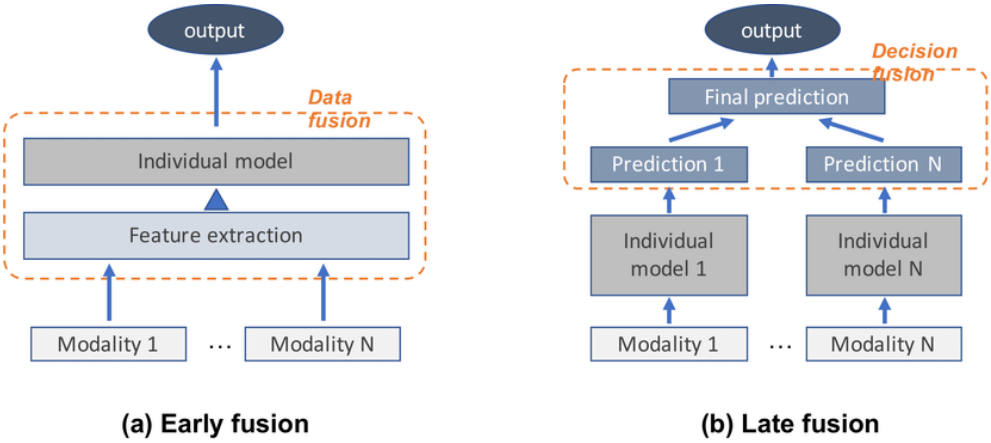

Рисунок 3.5 – Порівняльна схема різних стратегій злиття модальностей (раннє, пізнє)

При класифікації товарів кожна модальність надає унікальну інформацію:

– візуальна модальність найкраще передає фізичні характеристики товару: форму, колір, текстуру, дизайн;

– текстова модальність ефективніша для передачі технічних специфікацій, функціональних можливостей, брендових характеристик.

Взаємне доповнення проявляється, наприклад, коли текстовий опис допомагає розрізнити візуально схожі товари різних категорій, або коли зображення дає можливість оцінити естетичні якості, які складно описати словами.

Розвиток методів мультимодальної класифікації відбувався у кілька етапів, відображаючи загальну еволюцію алгоритмів машинного навчання та розуміння природи мультимодальних даних.

Ранні підходи часто базувалися на послідовній обробці різних типів даних, де результати обробки однієї модальності подавалися як вхідні дані для обробки іншої. Наприклад:

– використання текстових описів для створення шаблонів пошуку візуальних ознак;

– класифікація на основі зображень з подальшим уточненням результатів за допомогою текстової інформації.

Хоча такі методи були прості в реалізації, вони часто страждали від накопичення помилок і не могли повністю використати синергетичний потенціал різних модальностей.

Наступним кроком стала паралельна обробка модальностей з подальшим об'єднанням результатів:

– окремі мережі для обробки візуальних і текстових даних;

– зважена комбінація предикцій або ознак високого рівня;

– метод ансамблювання для підвищення надійності класифікації;

– цей підхід забезпечував кращу точність, але все ще не враховував повною мірою взаємодію між модальностями на ранніх етапах обробки.

Сучасні методи, представлені моделями CLIP, FLAVA і ViLT, реалізують більш інтегрований підхід:

– одночасна обробка обох модальностей;

– механізми крос-модальної уваги для моделювання взаємодії між модальностями;

– контрастивне навчання для створення узгоджених представлень.

Одночасна обробка дозволяє моделям вивчати складні взаємозв'язки між модальностями, що значно підвищує продуктивність класифікації, особливо для товарів зі складними характеристиками або тонкими відмінностями між категоріями.

Важливим аспектом еволюції є також перехід від моделей, що потребують великої кількості розмічених даних для конкретного домену, до більш універсальних моделей з можливістю трансферного та

малозразкового навчання, що особливо важливо для систем електронної комерції з постійно оновлюваним каталогом товарів.

3.2 Стратегії об'єднання результатів класифікації даних різних модальностей

Злиття (fusion) різних типів даних є ключовим аспектом мультимодальних систем класифікації, що визначає ефективність взаємодії та інтеграції інформації з різних джерел. У цьому підрозділі розглянемо математичні моделі різних стратегій злиття, їх вплив на точність класифікації та методи інтерпретації взаємодії між модальностями.

Стратегії злиття модальностей можна класифікувати за рівнем абстракції, на якому відбувається інтеграція, та за складністю механізмів взаємодії між модальностями. Розглянемо математичні формулювання основних підходів [35].

Найпростішим методом злиття є пряма конкатенація векторів ознак із різних модальностей (формула 3.4):

$$z = [\, x_v, x_t], \tag{3.4}$$

де [x_v ∈ R^{ d_v} – вектор ознак візуальної модальності;

x_t ∈ R^{ d_t} – вектор ознак текстової модальності;

z ∈ R^{ d_v + d_t} – об'єднаний вектор.

Попри простоту, цей метод має суттєві недоліки: він не враховує відносну важливість різних модальностей та створює вектори високої розмірності, що ускладнює подальше навчання.

Більш гнучкий підхід передбачає лінійну комбінацію проєкцій векторів ознак (формуал 3.5):

$$z = W_v\, x_v + \, W_t\, x_t + b, \tag{3.5}$$

де W_v ∈ R^{d_z * d_v} та W_t ∈ R^{d_z * d_v} – вагові матриці для відповідних модальностей;

b ∈ R^{d_v} – вектор зміщення;

d_z – розмірність спільного простору ознак.

Біллінійні моделі враховують взаємодію між елементами різних модальностей (формула 3.6):

$$z = x_y^T W x_t, \tag{3.6}$$

де W ∈ R^{d_v * d_t} – тензор, що кодує взаємозв'язки між елементами різних модальностей.

Більш загальна форма біллінійної моделі може бути представлена як (формула 3.7):

$$z = \sum^{ir} \sigma(W_v^i x_v) \odot \sigma(W_t^i x_t), \tag{3.7}$$

де σ – функція активації (наприклад, ReLU, tanh);

⊙ – поелементне множення (Hadamard product);

r – ранг взаємодії (кількість компонент у сумі);

W_v^i ∈ $\mathbb{R}$^{d × d_v}, W_t^i ∈ $\mathbb{R}$^{d × d_t} матриці для кожного компоненту i.

Гейтингові механізми забезпечують адаптивне зважування інформації з різних модальностей (формули 3.8 – 3.10):

$$g_v = \sigma(Wg\,[x_v;\, x_t]), \tag{3.8}$$

$$g_t = \sigma(Wg\,[x_v;\, x_t]), \tag{3.9}$$

$$z = g_v \odot (W_v x_v) + g_t \odot (W_t x_t), \tag{3.10}$$

де g_{v} та g_{t} – гейтингові вектори, що визначають вагу кожної модальності залежно від контексту [36].

Механізми уваги, особливо крос-модальна увага, дозволяють моделі фокусуватися на найбільш релевантних частинах кожної модальності (формули 3.11 – 3.12):

$$A = softmax\left(\frac{(Q_{\mathrm{v}} K_{\mathrm{t}}{}^{\mathrm{T}})}{\sqrt{d}_{\mathrm{k}}}\right), \tag{3.11}$$

$$z_{\mathrm{v}} = A\, V_{\mathrm{t}}, \tag{3.12}$$

де $Q_{\mathrm{v}} = W_Q$ * $\mathrm{x_v}$ – запити від візуальної модальності;

$\mathrm{K_t} = W_K$ * $\mathrm{x_t}$ та $\mathrm{V_t} = W_V$ * $\mathrm{x_t}$ – ключі та значення від текстової модальності;

d_k – масштабний коефіцієнт.

Аналогічно обчислюється увага від текстової до візуальної модальності z_t, а потім результати об'єднуються (формула 3.13):

$$z = W_z\,[z_{\mathrm{v}};\, z_{\mathrm{t}}]. \tag{3.13}$$

Сучасні моделі, такі як FLAVA та ViLT, використовують трансформерні блоки для інтеграції модальностей (формули 3.14 – 3.15):

$$Z^{(0)} = \left[z_{\mathrm{v}}{}^{(0)};\, z_{\mathrm{t}}{}^{(0)}\right], \tag{3.14}$$

$$Z^{(\mathrm{l}+1)} = TransformerBlock\left(Z^{(\mathrm{l})}\right), \tag{3.15}$$

де $\mathrm{Z^{(l)}}$ – стан мультимодальної моделі після l-го шару трансформера;

$\mathrm{z_v^{(0)}}$, $\mathrm{z_t^{(0)}}$ – початкові представлення з візуальної та текстової модальностей відповідно;

$[\mathrm{z_v^{(0)}};\, \mathrm{z_t^{(0)}}]$ – конкатенація векторів у спільну послідовність;

TransformerBlock(·) – один шар трансформера (self-attention + feedforward);

$Z^{(l+1)}$ – результат наступного шару.

Також є декілька підходів вагового злиття, їх насправді досить багато, але розглянут я вирішив лише кілька.

Найпростішим підходом є використання фіксованих вагових коефіцієнтів (формула 3.16):

$$z = \alpha\, x_{\mathrm{v}} + (1 - \alpha) x_{\mathrm{t}}, \tag{3.16}$$

де x_v – вектор ознак візуальної модальності;

x_t – вектор ознак текстової модальності;

$\alpha \in [0, 1]$ – скалярний коефіцієнт, що визначає важливість кожної модальності;

z – результат лінійної інтерполяції між модальностями.

Оптимальне значення α може бути визначене емпірично на валідаційному наборі даних або шляхом перехресної валідації.

Більш гнучкий підхід передбачає навчання вагових коефіцієнтів як параметрів моделі (формула 3.17):

$$z = softmax(w) \cdot [x_{\mathrm{v}};\, x_{\mathrm{t}}], \tag{3.17}$$

де x_v, x_t – вектори ознак візуальної та текстової модальностей;

$[x_v; x_t]$ – їх конкатенація;

w – вектор ваг (тієї ж розмірності, що й $[x_v; x_t]$), який навчається;

softmax(w) – розподіл ваг, що забезпечує суму 1.

Ну і адаптивні методи зважування враховують контекст або характеристики конкретного екземпляра даних (формули 3.18 – 3.19):

$$\alpha = f_{\theta(x_{\mathrm{v}}, x_{\mathrm{t}})}, \tag{3.18}$$

$$z = \alpha \cdot x_{\text{v}} + (1 - \alpha) \cdot x_{\text{t}}, \quad (3.19)$$

де $f_{\theta(x_{\text{v}},x_{\text{t}})}$ – параметризована функція, що визначає оптимальне співвідношення модальностей для кожного конкретного випадку.

Аналіз стратегій злиття модальностей показав їх критичну важливість для ефективної мультимодальної класифікації об'єктів, особливо в контексті категоризації товарів. Математичні моделі стратегій злиття варіюються від простих підходів, таких як конкатенація векторів, до складних механізмів, як трансформерні архітектури та гейтингові механізми.

Вибір оптимальної стратегії злиття залежить від характеристик товарних категорій, кількості доступних навчальних даних та якості інформації в різних модальностях. Експериментальні дослідження показують, що адаптивні стратегії злиття зазвичай переважають детерміністичні підходи для специфічних підкатегорій товарів, тоді як детерміністичні методи демонструють хорошу продуктивність та ефективність на верхніх рівнях ієрархії класифікації [37].

Методи зважування вкладів різних модальностей, від статичних до контекстно-залежних та основаних на невизначеності, забезпечують гнучкість у врахуванні відносної важливості візуальної та текстової інформації залежно від контексту.

Подальші дослідження в області стратегій злиття модальностей спрямовані на розробку більш ефективних адаптивних методів, що враховують специфіку даних та завдань, а також на вдосконалення методів інтерпретації взаємодії між модальностями для підвищення прозорості та зрозумілості мультимодальних систем класифікації.

3.3 Критерії оцінки ефективності мультимодальних моделей

Оцінка ефективності мультимодальної моделі – це не просто перевірка, чи правильно вона класифікує об’єкти. Враховуючи складність

поєднання візуальних і текстових ознак, необхідно враховувати цілу низку характеристик, які відображають як якість класифікації, так і здатність моделі працювати в реальних умовах.

Найпоширенішим показником є точність (accuracy), яка показує відсоток правильно класифікованих прикладів (формула 3.20) [38]:

$$Accuracy = \frac{\text{Кількість правильних передбачень}}{\text{Загальна кількість прикладів}}. \quad (3.20)$$

Однак, коли кількість товарів у різних класах дуже нерівномірна, точність може бути оманливою. У таких випадках доцільніше використовувати F1-міру, яка враховує як точність передбачень (precision), так і повноту (recall) (формули 3.21 – 3.23):

$$F^1 = 2 \cdot \frac{(Precision \cdot Recall)}{(Precision + Recall)}, \quad (3.21)$$

$$Precision = \frac{TP}{(TP + FP)}, \quad (3.22)$$

$$Recall = \frac{TP}{(TP + FN)}, \quad (3.23)$$

де TP – кількість істинно позитивних передбачень;

FP – хибнопозитивні;

FN – хибнонегативні;

Precision – точність;

Recall – повнота.

Для мультимодальних моделей важливо також оцінювати, як добре поєднуються ознаки з різних джерел. Одним із способів це зробити є

обчислення косинусної подібності між вбудовуванням зображення $\vec{v}$ і вбудовуванням тексту $\vec{t}$ (формула 3.24):

$$cos_{sim(\vec{v},\vec{t})} = \frac{(\vec{v} \cdot \vec{t})}{(\|\vec{v}\| \cdot \|\vec{t}\|)}, \tag{3.24}$$

де $\vec{v}$ – векторне представлення зображення;

$\vec{t}$ – векторне представлення тексту.

Цей показник близький до 1, коли зображення і текст мають схожий зміст у спільному векторному просторі, тобто модель правильно «зрозуміла» зв'язок між модальностями.

Ще одним корисним інструментом є функція втрат для контрастивного навчання, яка заохочує модель розташовувати правильні пари ближче у векторному просторі, ніж неправильні. Приклад такої функції втрат – triplet loss (формула 3.25) [39]:

$$L = \max(0, D(v_{\mathrm{a}}, v_{\mathrm{p}}) - D(v_{\mathrm{a}}, v_{\mathrm{n}}) + \alpha), \tag{3.25}$$

де v_{a} – вектор для зображення-оригіналу;

v_{p} – позитивна пара (відповідний текст);

v_{n} – негативна пара (невідповідний текст).

Крім точності класифікації, варто враховувати ще один фактор – робастність, або стійкість до змін у даних. Тобто, якщо зображення зняте під іншим кутом або на фоні іншого кольору, модель повинна залишатися точною. Цю властивість можна оцінити, наприклад, шляхом проведення класифікації на модифікованому тестовому наборі, створеному через аугментації.

Ще один критерій – швидкодія. Вона визначається як середній час обробки одного прикладу і залежить від обчислювальної складності моделі.

Для практичних застосувань, особливо у точках продажу, важливо, щоб цей показник був мінімальним, наприклад (формула 3.26):

$$T_{avg} = \left(\frac{1}{N}\right) \cdot \sum_{i=1}^{n} t_i , \quad (3.26)$$

де t_i – час, який модель витрачає на обробку і-го прикладу.

Окрему увагу заслуговує інтерпретованість моделі, тобто наскільки зрозумілою є логіка класифікації для користувача. Це особливо важливо у випадках помилок – щоб можна було виявити, що саме вплинуло на рішення моделі: текст, зображення чи їх поєднання. Тут часто використовують теплові карти важливості (наприклад, Grad-CAM), які показують, які частини зображення були найбільш релевантними для класифікації.

Загалом, ефективність мультимодальної моделі не вимірюється лише одним показником. Потрібно враховувати її точність, здатність працювати зі складними й нечіткими випадками, стійкість до змін, швидкодію і здатність до адаптації. І саме в поєднанні цих характеристик проявляється реальна цінність мультимодального підходу, який поєднує візуальну і текстову інформацію для точнішої класифікації товарів.

4 АНАЛІЗ МЕТОДІВ КЛАСИФІКАЦІЇ

4.1 Обґрунтування вибору методів класифікації

У контексті дослідження мультимодальної класифікації товарів постала необхідність обрання релевантних методів машинного навчання. Сучасний спектр технологій машинного навчання включає значну кількість підходів, кожен з яких характеризується певними перевагами та обмеженнями. З метою забезпечення практичної цінності дослідження для застосування в умовах реального магазину, було прийнято рішення зосередитися на виборі взаємодоповнюючих підходів, здатних всебічно відобразити специфіку проблеми.

На основі аналізу літератури було обрано наступні методи машинного навчання:

– ансамблеві методи;

– трансформерна архітектура з вирівнюванням мультимодальних ембедингів.

Перший обраний підхід передбачає ансамблеве поєднання окремих моделей, спеціалізованих для обробки кожної модальності даних. Цей класичний метод базується на принципі незалежної обробки різних типів інформації. У рамках даного підходу розробляються окремі моделі для аналізу зображень (наприклад, згорткові нейронні мережі), текстових описів (наприклад, рекурентні нейронні мережі або моделі на основі BERT) та, за необхідності, інших характеристик товарів. Після незалежного навчання, прогнози кожної моделі агрегуються за допомогою мета-алгоритму для прийняття остаточного рішення [40].

Зазначений підхід відрізняється модульністю. У випадку появи більш ефективного алгоритму для аналізу певної модальності, можлива його інтеграція без необхідності повної переробки системи. Крім того, ансамблеві методи зазвичай потребують менших обчислювальних ресурсів

на етапі навчання, що є важливим для малих і середніх підприємств. Також, вони демонструють стійкість до помилок, оскільки некоректні прогнози однієї моделі можуть бути компенсовані іншими.

Однак, ансамблевий підхід має обмеження, зокрема потенційну втрату інформації про взаємозв'язки між модальностями. Незалежна обробка даних може призвести до ігнорування важливих взаємодоповнюючих сигналів, що містяться в різних типах даних.

Другим об'єктом дослідження є трансформерна архітектура з вирівнюванням мультимодальних ембедингів. Цей підхід представляє парадигму наскрізного навчання, де обробка різних модальностей здійснюється в єдиному обчислювальному графі. Метою є створення спільного семантичного простору, в якому візуальні та текстові концепти інтегровані [41].

Застосування механізмів самоуваги дозволяє встановлювати зв'язки між елементами різних модальностей, сприяючи формуванню контекстуального розуміння товару. Фундаментальною ідеєю є створення спільного простору ознак шляхом контрастивного навчання, що мінімізує відстань між семантично пов'язаними елементами з різних модальностей та максимізує відстань між невідповідними.

У рамках дослідження розглядаються два варіанти організації такої архітектури:

– ахітектура з попередньо навченими енкодерами для обробки окремих модальностей з подальшою проекцією їх векторів у спільний простір;

– повністю інтегрована архітектура, де обробка зображень та тексту здійснюється єдиною моделлю з використанням крос-модальних механізмів уваги.

На відміну від ансамблевого підходу, трансформерна архітектура теоретично здатна виявляти складні взаємозалежності між модальностями, формуючи цілісне розуміння об'єкта. Проте, даний підхід вимагає значних

обчислювальних ресурсів та великих обсягів навчальних даних, а також може бути схильним до перенавчання та мати обмежену інтерпретованість.

Вибір зазначених двох підходів обумовлений їхньою репрезентацією двох протилежних філософій машинного навчання: модульності та інтеграції. Їх порівняльний аналіз дозволить не лише оцінити точність, але й визначити компроміси, пов'язані з вибором кожного методу, особливо в контексті реальних умов експлуатації з потенційно зашумленими та неповними даними, а також з урахуванням ієрархічної структури категорій товарів.

У контексті магістерської роботи, ці підходи розглядаються як взаємодоповнюючі інструменти. Ансамблевий підхід може бути оптимальним для систем з обмеженими ресурсами або у випадках, де важлива інтерпретованість рішень, тоді як трансформерна архітектура може забезпечити вищу точність у складних сценаріях, що потребують врахування складних міжмодальних залежностей.

Таким чином, порівняльний аналіз обраних підходів сприятиме формуванню всебічного розуміння можливостей мультимодальної класифікації товарів, враховуючи як теоретичні аспекти, так і практичні обмеження їх застосування.

4.2 Перший підхід: ансамблеве поєднання моделей для окремих модальностей

Ансамблеве поєднання спеціалізованих моделей для кожної модальності представляє собою класичний, але надзвичайно ефективний підхід до мультимодальної класифікації. Цей метод ґрунтується на принципі, що кожна модальність має свою унікальну структуру та характеристики, які найкраще обробляються спеціалізованими архітектурами. У контексті класифікації товарів ця методологія дозволяє нам максимально використовувати як візуальну інформацію (форма, колір,

дизайн продукту), так і текстову (технічні характеристики, функціональні властивості, опис використання).

Основна ідея підходу полягає в тому, щоб розділити складну задачу мультимодальної класифікації на окремі підзадачі для кожної модальності, а потім інтегрувати отримані результати для формування кінцевого рішення. Такий модульний дизайн має низку переваг, особливо в промисловому контексті: він забезпечує гнучкість (можливість замінити або оновити модуль для окремої модальності без перебудови всієї системи), масштабованість (легке додавання нових модальностей) та робастність (збій в одному модулі не призводить до фатального збою всієї системи).

У рамках цього дослідження розглянемо три варіанти реалізації ансамблевого підходу, кожен з яких представляє різну стратегію інтеграції інформації з візуальної та текстової модальностей:

– незалежні нейронні «спеціалісти» з фіксованим зважуванням;

– модель із раннім злиттям ознак та спільним нейронним класифікатором;

– стекінговий ансамбль з архітектурою мета-навчання на основі градієнтного бустингу.

Кожна з цих моделей має свої унікальні характеристики, сильні сторони та обмеження, що буде детально розглянуто в наступних підрозділах.

Перша модель втілює найбільш пряму реалізацію ансамблевого підходу: незалежні спеціалізовані нейронні мережі для кожної модальності з наступним зважуванням їхніх прогнозів. Цей метод концептуально близький до експертної системи, де кожен експерт (модель) надає свій висновок на основі доступної йому інформації, а остаточне рішення формується через зважування цих висновків.

Для обробки візуальної інформації використовується згорткова нейронна мережа ResNet50, попередньо навчена на великому наборі даних ImageNet. Використання попередньо навченої мережі дозволяє ефективно

вилучати високорівневі візуальні ознаки навіть при відносно невеликій кількості навчальних даних для конкретної задачі [42].

Архітектура ResNet50 є особливо підходящою для нашої задачі з кількох причин:

– залишкові (residual) з'єднання дозволяють мережі ефективно вивчати як низькорівневі, так і високорівневі візуальні ознаки;

– глибока архітектура із 50 шарами забезпечує достатню ємність для вивчення складних патернів;

– механізм багаторівневої ієрархії ознак добре узгоджується з ієрархічною природою категорій товарів.

Для адаптації ResNet50 до нашої задачі видаляємо останній повнозв'язний шар, який у оригінальній моделі відповідає за класифікацію за 1000 категоріями ImageNet, і замінюємо його на новий класифікаційний шар з кількістю нейронів, що дорівнює кількості категорій товарів у нашому датасеті (формула 4.1) [43]:

$$h_{\text{v}} = ResNet50_{features}(x_{\text{v}}), \tag{4.1}$$

де x_{v} – вхідне зображення;

$h_{\text{v}} \in \mathbb{R}^{\text{d}_{\text{v}}}$ – вектор візуальних ознак, вилучений з передостаннього шару мережі ($\text{d}_{\text{v}} = 2048$ для стандартної архітектури ResNet50).

Потім ці ознаки проходять через новий класифікаційний шар (формула 4.2):

$$z_{\text{v}} = W_{\text{v}}\, h_{\text{v}} + b_{\text{v}}, \tag{4.2}$$

де $W_{\text{v}} \in \mathbb{R}^{\text{c}\times\text{d}_{\text{v}}}$ – матриця ваг;

$b_{\text{v}} \in \mathbb{R}^{\text{c}}$ – вектор зміщення;

$z_{\text{v}} \in \mathbb{R}^{\text{c}}$ – вектор логітів, які відповідають C категоріям товарів.

Фінальні ймовірності класів обчислюються через функцію softmax (формула 4.3):

$$\mathcal{p}_{\mathrm{v}} = softmax(z_{\mathrm{v}}) = \frac{e^{\mathrm{xp}}(z_{\mathrm{v}})}{\sum^{\mathrm{i}=1\,\mathrm{c}} e^{\mathrm{xp}}(z_{\mathrm{v}}{}^{(\mathrm{i})})}, \quad (4.3)$$

де $\mathcal{p}_{\mathrm{v}} \in \mathbb{R}^{\mathrm{c}}$ – вектор ймовірностей належності зображення до кожної з C категорій.

Для обробки текстових описів товарів використовуємо двонаправлену рекурентну нейронну мережу LSTM (Bi-LSTM), яка здатна ефективно вловлювати як прямі, так і зворотні залежності в послідовності тексту. Ця архітектура є особливо важливою для розуміння контексту слів та фраз, що критично для правильної класифікації товарів на основі їх описів [44].

Процес обробки тексту складається з наступних етапів:

– токенізація та векторизація: текстовий опис x_t розбивається на послідовність токенів (слів або підслів), які потім перетворюються на векторні представлення за допомогою попередньо навченої моделі вбудовувань (наприклад, Word2Vec, GloVe або FastText) (формула 4.4):

$$E = [e1, e2, \dots, e_T], \quad (4.4)$$

де $e_{\mathrm{i}} \in \mathbb{R}^{\mathrm{de}}$ – вектор вбудовування для *i*-го токена;

T – довжина послідовності;

d_{e} – розмірність простору вбудовувань.

– обробка послідовності з Bi-LSTM: послідовність векторів вбудовувань проходить через двонаправлену LSTM мережу (формули 4.5 – 4.6):

$$\vec{h}_{\mathrm{t}}, \vec{c}_{\mathrm{t}} = LSTM_{forward}\left(e_{\mathrm{t}}, \vec{h}_{\mathrm{t}}{}^{-1}, \vec{c}_{\mathrm{t}}{}^{-1}\right), \quad (4.5)$$

$$\overleftarrow{h}_t, \overleftarrow{c}_t = LSTM_{backward}\left(e_t, \overleftarrow{h}_t{}^{+1}, \overleftarrow{c}_t{}^{+1}\right), \quad (4.6)$$

де $\overrightarrow{h}_t, \overleftarrow{h}_t \in \mathbb{R}^{dh}$ – вектори прихованого стану прямого та зворотного LSTM для токена t;

$\overrightarrow{c}_t, \overleftarrow{c}_t \in \mathbb{R}^{dh}$ – відповідні вектори стану комірки.

– формування вектора ознак: кінцеві приховані стани з обох напрямків конкатенуються для отримання повного представлення тексту (формула 4.7):

$$h_t = \left[\vec{h}_t; \overleftarrow{h}^1\right] \in \mathbb{R}^{2dh}, \quad (4.7)$$

де $\overrightarrow{h}_t$ – кінцевий прихований стан прямого LSTM;

$\overleftarrow{h}_1$ – кінцевий прихований стан зворотного LSTM.

– класифікація: отриманий вектор ознак проходить через повнозв'язний шар для отримання логітів класів (формула 4.8):

$$z_t = W_t h_t + b_t, \quad (4.8)$$

де $W_t \in \mathbb{R}^{c \times 2dh}$ – матриця ваг;

$b_t \in \mathbb{R}^c$ – вектор зміщення;

$z_t \in \mathbb{R}^c$ – вектор логітів.

– обчислення ймовірностей: як і у випадку з візуальним модулем, фінальні ймовірності класів обчислюються через функцію softmax (формула 4.9):

$$p_t = softmax(z_t). \quad (4.9)$$

Після отримання прогнозів від обох спеціалізованих моделей,

виконується їх зважене усереднення для формування кінцевого прогнозу (формула 4.10):

$$\mathcal{p}_{final} = \alpha \cdot \mathcal{p}_{\text{v}} + (1 - \alpha) \cdot \mathcal{p}_{\text{t}}, \quad (4.10)$$

де $\alpha \in [0, 1]$ – гіперпараметр, що визначає відносну важливість візуальної та текстової модальностей. Оптимальне значення $\boldsymbol{\alpha}$ визначається емпірично на валідаційному наборі даних.

Процес навчання цієї моделі складається з двох етапів. Спочатку відбувається незалежне навчання двох моделей: для зображень використовується ResNet50, а для тексту – Bi-LSTM. Кожна модель навчається на своїй відповідній модальності, застосовуючи крос-ентропійну функцію втрат. У процесі навчання оцінюється точність прогнозів моделей, порівнюючи ймовірності, що вони передбачають для кожної категорії, з реальними мітками для кожного прикладу. Ця функція втрат дозволяє моделі коригувати свої параметри, щоб зменшити розбіжність між прогнозами та реальними значеннями, в результаті чого кожна з моделей краще розпізнає зображення або текст [45].

Після навчання базових моделей, оптимальне значення $\boldsymbol{\alpha}$ визначається шляхом перебору значень на валідаційному наборі даних для максимізації обраної метрики якості (наприклад, точності або F1-міри).

Таблиця 4.1 – Переваги та недоліки цього підходу

Переваги	Недоліки
Модульність: Легка заміна або оновлення компонентів.	Фіксований коефіцієнт зважування: Неможливість адаптувати важливість модальностей.

Продовження таблиці 4.1

Інтерпретованість: Чітке визначення внеску кожної модальності.	Відсутність взаємодії між модальностями: Неможливість вивчати взаємозв'язки між ознаками.
Стійкість до відсутніх даних: Модель працює без однієї з модальностей.	Обмежена контекстуалізація: Втрата контекстуальних зв'язків між модальностями.
Ефективність навчання: Менше обчислювальних ресурсів порівняно з end-to-end навчанням.	

Друга модель представляє альтернативний підхід до інтеграції мультимодальних даних через стратегію раннього злиття (early fusion). На відміну від першої моделі, де інтеграція відбувається на рівні прогнозів (пізнє злиття), тут ми поєднуємо інформацію з різних модальностей на рівні ознак, дозволяючи нейронній мережі самостійно виявляти взаємозв'язки між ними.

Для екстракції базових ознак з кожної модальності використовуємо ті ж самі архітектури, що і в першій моделі. Однак, на відміну від першої моделі, ці екстрактори не включають класифікаційні шари, оскільки їхня мета – формування багатих векторних представлень відповідних модальностей.

Отримані вектори ознак з обох модальностей інтегруються шляхом конкатенації з подальшою нормалізацією (формула 4.11):

$$h_{concat} = [h_{\mathrm{v}};\ h_{\mathrm{t}}] \in \mathbb{R}^{2d_h}. \tag{4.11}$$

Для покращення стабільності навчання та зменшення ефекту різних масштабів ознак, застосовуємо нормалізацію (формула 4.12):

$$h_{\text{norm}} = LayerNorm(h_{concat}), \quad (4.12)$$

де LayerNorm – операція нормалізації шару, яка стандартизує вхідний вектор по кожному прикладу (формула 4.13) [46]:

$$LayerNorm(h) = \gamma \odot \left(\frac{(h - \mu)}{\sqrt{\sigma^2 + \epsilon}}\right) + \beta, \quad (4.13)$$

де μ та σ^2 – середнє значення та дисперсія елементів вектора h;

γ та β – параметри масштабування та зміщення, що навчаються;

ϵ – мала константа для числової стабільності.

Після злиття ознак отриманий вектор подається на вхід спільного класифікатора, реалізованого як багатошарова нейронна мережа (формули 4.14 – 4.16):

$$h^{(1)} = ReLU\left(W^{(1)}h_{\text{norm}} + b^{(1)}\right), \quad (4.14)$$

$$h^{(2)} = ReLU\left(W^{(2)}h^{(1)} + b^{(2)}\right), \quad (4.15)$$

$$z = W^{(3)}h^{(2)} + b^{(3)}, \quad (4.16)$$

де $W^{(1)} \in \mathbb{R}^{dh} \times (d_v + 2^{dh})$, $W^{(2)} \in \mathbb{R}^{dh/2} \times {}^{dh}$, $W^{(3)} \in \mathbb{R}^{c} \times {}^{dh/2}$ – вагові матриці;

$b^{(1)} \in \mathbb{R}^{dh}$, $b^{(2)} \in \mathbb{R}^{dh/2}$, $b^{(3)} \in \mathbb{R}^{c}$ – вектори зміщення;

$z \in \mathbb{R}^{c}$ – вектор логітів класів.

Для обчислення фінальних ймовірностей використовується функція softmax. Класифікація здійснюється за принципом вибору класу з найвищою ймовірністю (таблиця 4.2).

Оскільки модель із раннім злиттям має більшу кількість параметрів та складнішу архітектуру, вона більш схильна до перенавчання. Для запобігання цьому впроваджуємо наступні техніки регуляризації:

– dropout після кожного повнозв'язного шару класифікатора;

– L2-регуляризація (weight decay) для всіх вагових параметрів;

– рання зупинка (early stopping) на основі продуктивності на валідаційному наборі.

Таблиця 4.2 – Переваги та обмеження підходу з нейронною мережею

Переваги	Обмеження
Вивчення взаємозв'язків між модальностями: Модель враховує складні залежності між ознаками.	Складність навчання: Потребує більше ресурсів і тонкого налаштування.
Потенційно вища точність: Краще використання інформації з обох модальностей.	Ризик перенавчання: Через велику кількість параметрів.
Компактність архітектури: Один спільний класифікатор замість двох.	Залежність від обох модальностей: Гірше працює при відсутності однієї з них.
Оптимізація для кінцевої задачі: Усі компоненти налаштовуються одночасно.	Менша інтерпретованість: Важче оцінити внесок кожної модальності.

Третя модель представляє найбільш складний та гнучкий підхід до ансамблювання, який поєднує переваги глибокого навчання для екстракції ознак з потужністю алгоритмів на основі дерев рішень для мета-навчання. Цей підхід, відомий як стекінг (stacking) або мета-навчання, дозволяє ефективно комбінувати прогнози базових моделей, враховуючи їхні сильні та слабкі сторони в різних контекстах [47].

На першому етапі використовуємо ті самі попередньо навчені моделі, що й у попередніх підходах:

– ResNet50 для вилучення векторів візуальних ознак;

– Bi-LSTM для вилучення векторів текстових ознак.

Кожен приклад у навчальному наборі проходить через ці нейронні мережі, після чого ми формуємо таблицю ознак $\mathbf{X}_{\text{meta}}$, де кожен рядок – це h_{stack}, а мітки залишаються незмінними (формула 4.17):

$$X_{meta} = \begin{bmatrix} h_{stack}^{(1)} \\ h_{stack}^{(2)} \\ \vdots \\ h_{stack}^{(N)} \end{bmatrix}, y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{bmatrix}. \quad (4.17)$$

Як мета-класифікатор використовуємо градієнтний бустинг на деревах (XGBoost, LightGBM або CatBoost), таблиця 4.3. Математична модель градієнтного бустингу базується на послідовному додаванні слабких моделей (дерев рішень), кожна з яких намагається скоригувати помилки попередніх [48].

Ключові параметри, що налаштовуються в градієнтному бустингу:

– кількість дерев (n_estimators) – визначає загальну кількість дерев в ансамблі;

– глибина дерев (max_depth) – контролює складність окремих дерев;

– швидкість навчання (learning_rate) – визначає вплив кожного нового дерева на фінальну модель;

– підвиборювання рядків та стовпців (subsample, colsample_bytree) – допомагає контролювати перенавчання;

– регуляризація (reg_alpha, reg_lambda) – L1 та L2 регуляризація для контролю складності моделі.

Таблиця 4.3 – Переваги та обмеження підходу зі стекінгом

Переваги	Обмеження
Висока інтерпретованість: можна бачити важливість ознак	Нероздільне навчання: ознаки фіксуються після першого етапу

Продовження таблиці 4.3

Гнучкість: легко змінювати мета-класифікатор	Складність пайплайну: два етапи навчання
Стійкість до шуму: дерева можуть ігнорувати нерелевантні ознаки	Низька адаптивність до зміни вхідних ознак
	Втрата контекстуального навчання між модальностями

4.3 Другий підхід: трансформер з вирівнюванням мультимодальних ембедингів

На відміну від ансамблевого підходу, який розглядає модальності окремо з їх подальшим поєднанням, архітектури на основі трансформерів з вирівнюванням ембедингів представляють більш інтегрований підхід до мультимодальної класифікації. Цей метод спрямований на проєкцію різних типів даних у спільний семантичний простір, де їх можна ефективно порівнювати та аналізувати. Такий підхід дозволяє моделі безпосередньо вивчати взаємозв'язки між модальностями на рівні їх семантичних репрезентацій.

У контексті класифікації товарів цей підхід має особливе значення, оскільки він дозволяє комп'ютерним системам «розуміти» продукти так, як їх сприймає людина – одночасно аналізуючи як їх візуальне представлення, так і текстовий опис, встановлюючи між ними семантичні зв'язки. Наприклад, система може встановити, що зображення ноутбука і фраза «портативний комп'ютер з 15–дюймовим екраном» відносяться до одного і того ж об'єкта, навіть якщо ці конкретні слова не з'являються в описі саме цього продукту.

Основна архітектура цього підходу складається з трьох ключових компонентів: енкодера візуальних даних, енкодера текстових даних та

модуля вирівнювання ембедингів. Загальна схема архітектури представлена на рисунку 4.1.

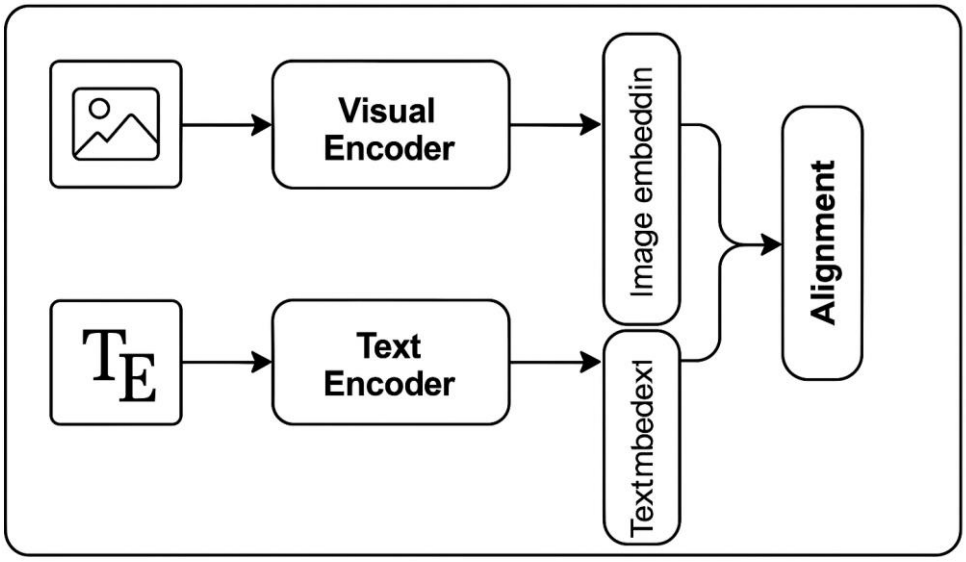

Рисунок 4.1 – Ідея комбінованої архітектури

Для обробки візуальної інформації використовується модель Vision Transformer (ViT), яка, на відміну від традиційних CNN, обробляє зображення як послідовність патчів. Цей підхід дозволяє ефективно застосовувати механізм уваги (attention mechanism) до візуальних даних, аналогічно тому, як це робиться з текстом.

Вхідне зображення x_v розбивається на послідовність патчів розміром P × P які потім лінійно проєктуються у простір ембедингів (формула 4.18):

$$z^0 = \left[x_{(class)};\ E^1 \cdot x_{\mathrm{p}}{}^1;\ E^2 \cdot x_{\mathrm{p}}{}^2;\ \ldots;\ E_{\mathrm{n}} \cdot x_p^N\right] +\ E_{\mathrm{pos}}, \qquad (4.18)$$

де $x_{(class)}$ – спеціальний токен класифікації;

$x_{\mathrm{p}}{}^1$ – і-й патч зображення;

E^1 – матриця лінійної проєкції;

E_{pos} – позиційні ембединги;

N – кількість патчів.

Отримана послідовність ембедингів подається на вхід стандартного трансформерного енкодера, що складається з L шарів (формули 4.19, 4.20):

$$z_l' = MSA\big(LN(z_{l-1})\big) + z_{l-1}, l = 1 \dots L, \tag{4.19}$$

$$z_l = MLP\big(LN(z_l')\big) + z_l', l = 1 \dots L, \tag{4.20}$$

де MSA – механізм мультиголової самоуваги (Multi-head Self-Attention) [49];

LN – нормалізація шару (Layer Normalization);

MLP – багатошаровий перцептрон;

z_l – вихід l-го шару трансформера.

Фінальний ембединг візуальних даних отримується з ембедингу токена класифікації останнього шару (формула 4.21):

$$h_v = z_L^0 \in R^{d_v}, \tag{4.21}$$

де d_v – розмірність візуального ембедингу.

Для обробки текстових описів використовується архітектура BERT (Bidirectional Encoder Representations from Transformers), яка дозволяє отримувати контекстуалізовані ембединги слів на основі двонаправленого аналізу тексту.

Текстові дані спочатку токенізуються за допомогою WordPiece або аналогічного алгоритму, після чого токени перетворюються на ембединги з додаванням позиційної інформації (формула 4.22):

$$e^0 = [e_{cls};\ E^1 t^1;\ E^2 t^2;\ \dots;\ E_M t^M] + E_{\text{pos}}, \tag{4.22}$$

де e_{cls}– ембединг спеціального токена [CLS];

t^i – і-й токен тексту;

E_i – матриця ембедингів;

E_{pos} – позиційні ембединги;

M – кількість токенів.

Отримана послідовність проходить через **K** шарів трансформерного енкодера, аналогічно до візуального енкодера (формули 4.23 – 4.24):

$$e'_k = MSA\big(LN(e_{k-1})\big) + e_{k-1}, k = 1 \dots K, \tag{4.23}$$

$$e_k = MLP\big(LN(e'_k)\big) + z'_k, k = 1 \dots K. \tag{4.24}$$

Фінальний текстовий ембединг отримується з ембедингу токена [CLS] останнього шару (формула 4.25):

$$h_t = e_K^0 \in R^{d_t}, \tag{4.25}$$

де d_t – розмірність текстового ембедингу.

Оскільки візуальний та текстовий енкодери можуть продукувати ембединги різної розмірності та з різною статистичною структурою, необхідно здійснити їх проєкцію в єдиний спільний простір. Для цього використовуються проєкційні матриці (формули 4.26, 4.27):

$$h_v^{aligned} = W_v\, h_v + b_v, \tag{4.26}$$

$$h_t^{aligned} = W_t\, h_t + b_t, \tag{4.27}$$

де $W_v \in \mathbb{R}^{d_{joint}} \times d_v$, $W_t \in \mathbb{R}^{d_{joint}} \times d_t$ – матриці проєкції;

$b_v \in \mathbb{R}^{d_{joint}}$, $b_t \in \mathbb{R}^{d_{joint}}$ – вектори зміщення (байаси);

d_{joint} – розмірність спільного простору ембедингів.

Після проєкції, отримані вектори нормалізуються для забезпечення їх розташування на одиничній сфері (формули 4.28, 4.29):

$$h_v^{norm} = \frac{h_v^{aligned}}{\left\| h_v^{aligned} \right\|_2}, \tag{4.28}$$

$$h_t^{norm} = \frac{h_t^{aligned}}{\left\| h_t^{aligned} \right\|_2}. \tag{4.29}$$

Ключовою особливістю цього підходу є використання контрастивного навчання для вирівнювання ембедингів різних модальностей. Контрастивне навчання спрямоване на мінімізацію відстані між позитивними парами (відповідні зображення та тексти) та максимізацію відстані між негативними парами.

Для навчання моделі використовується функція втрат InfoNCE (Info Noise-Contrastive Estimation), яка є адаптацією контрастивної предиктивної кодування для задачі вирівнювання модальностей (формула 4.30):

$$L = L_{v2t} + L_{t2v}, \tag{4.30}$$

де L_{v2t} – втрата для напрямку «зображення → текст»;

L_{t2v} – втрата для напрямку «текст → зображення».

Втрата для напрямку «зображення → текст» обчислюється за формулою 4.31:

$$L_{v2t} = -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp\left(\frac{h_{v,i}^{norm} \cdot h_{t,i}^{norm}}{\tau}\right)}{\sum_{i=1}^{B} \exp\left(h_{v,i}^{norm} \cdot h_{t,i}^{norm}\right)}, \tag{4.31}$$

де B – розмір бачту;

τ – температурний параметр, що контролює «гостроту» розподілу ймовірностей.

Аналогічно, втрата для напрямку «текст → зображення» обчислюється за формулою 4.31, але $h_{v,i}^{norm}$ і $h_{t,i}^{norm}$ міняємо місцями.

Цей підхід ефективно використовує інформацію з усього міні-батчу, розглядаючи всі інші приклади в батчі як негативні зразки. Як результат, модель навчається розрізняти правильні пари «зображення-текст» серед усіх можливих комбінацій.

Навчання моделі відбувається у декілька етапів:

– ініціалізація енкодерів: візуальний та текстовий енкодери ініціалізуються вагами попередньо навчених моделей (наприклад, ViT-B/32 та BERT-base) [50];

– замороження базових шарів: для прискорення навчання та запобігання перенавчання, глибокі шари енкодерів можуть бути заморожені, а тренуються лише верхні шари та проєкційні матриці;

– міні-батч оптимізація: модель навчається на великих міні-батчах (типово 256–1024 прикладів) для забезпечення достатньої кількості негативних пар;

– адаптивний темп навчання: використовується планувальник темпу навчання з розігрівом та поступовим зменшенням.

Одна з ключових переваг підходу з вирівнюванням ембедингів полягає в можливості здійснювати класифікацію навіть при відсутності однієї з модальностей. Наприклад, якщо доступне лише зображення, класифікація може бути виконана на основі лише візуального ембедингу h_v^{norm}.

5 РЕАЛІЗАЦІЯ ТА ПОРІВНЯННЯ ЗАПРОПОНОВАНИХ ПІДХОДІВ

5.1 Технологічне середовище та структура реалізації

Для практичної імплементації розроблених мультимодальних підходів було створено високопродуктивне технологічне середовище на базі хмарної платформи Google Colaboratory (Colab). Даний вибір зумовлений потребою в масштабованих обчислювальних ресурсах та інтегрованістю з екосистемою інструментів машинного навчання. Для забезпечення необхідної обчислювальної потужності при тренуванні складних мультимодальних архітектур було використано преміум-підписку Colab з доступом до високопродуктивних графічних прискорювачів NVIDIA A100 (40 ГБ відеопам'яті), що дозволило ефективно паралелізувати матричні обчислення та значно прискорити процес навчання нейромережевих моделей.

Основою технологічного стеку стала мова програмування Python версії 3.10, що забезпечує оптимальний баланс між продуктивністю, читабельністю коду та сумісністю з новітніми версіями бібліотек машинного навчання. Після ретельного аналізу доступних фреймворків глибокого навчання було обрано PyTorch 2.1 як основну платформу для реалізації архітектур мультимодальних моделей. PyTorch демонструє кращу інтеграцію з сучасними трансформерними архітектурами, надає більш гнучкий та інтуїтивний інтерфейс для дослідницьких експериментів, а також забезпечує ефективну підтримку динамічних обчислювальних графів, що критично важливо при розробці складних мультимодальних моделей з нестандартними механізмами обміну інформацією між модальностями [51].

Для роботи з передобробкою даних та імплементації класичних алгоритмів машинного навчання було інтегровано науковий стек бібліотек:

– scikit-learn 1.3.0 – для реалізації традиційних класифікаторів, метрик оцінювання та методів зниження розмірності;

– pandas 2.1.0 – для структурованого зберігання та маніпулювання метаданими мультимодальних прикладів;

– numPy 1.24.0 – для ефективних операцій з багатовимірними масивами;

– matplotlib та Seaborn – для детальної візуалізації результатів експериментів та аналізу моделей.

Важливим компонентом технологічного середовища стала інтеграція з Google Drive, що забезпечило безперебійний доступ до попередньо підготовленого та структурованого датасету для навчання мультимодальних моделей. Датасет розміщений у спеціалізованому репозиторії з ієрархічною організацією, що включає розподіл на навчальну, валідаційну та тестову вибірки згідно з принципами крос-валідації. Для оптимізації операцій читання/запису при роботі з великими обсягами мультимодальних даних було реалізовано механізм кешування проміжних результатів передобробки, що суттєво прискорило ітерації експериментів.

Робота з трансформерними архітектурами забезпечена через інтеграцію бібліотеки Hugging Face Transformers 4.34.0, яка надає уніфікований доступ до найсучасніших попередньо навчених моделей різних модальностей та спрощує їх адаптацію для задач мультимодального аналізу. Додатково було використано torchvision 0.16.0 для роботи з візуальними енкодерами та спеціалізованими аугментаціями зображень [52].

Структура програмної реалізації побудована за модульним принципом, що забезпечує гнучкість експериментів та можливість систематичного порівняння різних підходів.

Додатково було імплементовано механізм серіалізації та десеріалізації моделей для збереження проміжних та кінцевих результатів тренування на Google Drive, а також систему логування з Weights & Biases для детального відстеження експериментів у часі.

Особливу увагу приділено оптимізації використання обчислювальних ресурсів A100 GPU через застосування змішаної точності обчислень (mixed precision training), що дозволило збільшити розмір оброблюваних батчів та прискорити тренування без втрати точності моделей. Реалізовано динамічне керування пам'яттю для запобігання переповненню відеопам'яті при роботі з великими мультимодальними входами.

Розроблене технологічне середовище забезпечує повну відтворюваність експериментів та створює міцну основу для порівняльного аналізу ефективності різних підходів до мультимодального моделювання в умовах однакових обчислювальних ресурсів та наборів даних.

5.2 Реалізація першого підхіду: ансамбль моделей по модальностях

Перша запропонована модель втілює стратегію ансамблевого поєднання, де незалежні нейронні мережі, навчені на різних модальностях, об'єднують свої прогнози за допомогою фіксованих вагових коефіцієнтів. Цей підхід дозволяє кожній модальності бути обробленою найбільш підходящою для її природи архітектурою, а кінцеве рішення формується шляхом зваженого голосування.

Для аналізу візуальних даних використовується попередньо навчена згорткова нейронна мережа ResNet50. Перевага використання попередньо навченої моделі полягає в її здатності вилучати інформативні ознаки з зображень, використовуючи знання, отримані на великому наборі даних ImageNet. Це особливо корисно при обмеженій кількості навчальних даних для нашої конкретної задачі класифікації товарів.

Архітектура ResNet50, завдяки своїм залишковим з'єднанням, ефективно справляється з проблемою зникнення градієнтів, дозволяючи навчати глибокі мережі. Її багатошарова структура здатна вловлювати як низькорівневі (наприклад, кольори, текстури), так і високорівневі (форми, об'єкти) візуальні характеристики товарів.

Для адаптації ResNet50 до нашої задачі класифікації товарів, останній повнозв'язний шар, відповідальний за класифікацію на 1000 класів ImageNet, видаляється і замінюється новим класифікаційним шаром з кількістю вихідних нейронів, що відповідає кількості категорій товарів у нашому датасеті (Лістинг 5.1).

Лістинг 5.1 – Програмний код моделі ResNet50 на PyTorch

```
import torch
import torch.nn as nn
from torchvision.models import resnet50

resnet = resnet50(pretrained=True)

# Заморожування параметрів згорткових шарів (опційно для
feature extraction)
for param in resnet.parameters():
    param.requires_grad = False

num_features = resnet.fc.in_features

# Заміна останнього повнозв'язного шару на новий для нашої
кількості класів
num_classes = 81
resnet.fc = nn.Linear(num_features, num_classes)
```

Після завантаження ResNet50 ми також підготували модель для обробки текстової інформації. Для цієї мети ми обрали двонаправлену рекурентну нейронну мережу з довгою короткочасною пам'яттю (Bi-LSTM). Ця архітектура здатна ефективно враховувати контекст слів у реченні, що є важливим для розуміння текстових описів товарів. Реалізація Bi-LSTM можна побачити у лістингу 5.2.

Лістинг 5.2 – Реалізація Bi-LSTM на PyTorch

```
import torch
import torch.nn as nn

class TextModel(nn.Module):
        def __init__(self, vocab_size, embedding_dim,
hidden_dim, num_layers, num_classes,
embedding_weights=None):
        super(TextModel, self).__init__()
              self.embedding = nn.Embedding(vocab_size,
embedding_dim)
        if embedding_weights is not None:
            self.embedding.weight.data.copy_(torch.from_nu
mpy(embedding_weights))
            self.embedding.weight.requires_grad = False
          self.lstm = nn.LSTM(embedding_dim, hidden_dim,
num_layers, bidirectional=True, batch_first=True)
        self.fc = nn.Linear(hidden_dim * 2, num_classes)

    def forward(self, text, text_lengths):
        embedded = self.embedding(text)
                                packed_embedded =
nn.utils.rnn.pack_padded_sequence(embedded,
text_lengths.cpu(), batch_first=True,
enforce_sorted=False)
                  packed_output, (hidden, cell) =
self.lstm(packed_embedded)
           hidden = torch.cat((hidden[-2,:,:], hidden[-
1,:,:]), dim=1)
        output = self.fc(hidden)
        return output
```

Для навчання кожної з цих моделей ми використовували відповідні набори даних. Зображення передавалися до візуальної моделі у вигляді тензорів, а текстові описи спочатку токенізувалися, векторизувалися та

доповнювалися до однакової довжини в межах батчу. Довжини послідовностей також передавалися до текстової моделі для ефективної обробки змінної довжини за допомогою механізму pack_padded_sequence. Процес навчання включав ітерацію по епохах, де на кожній ітерації дані з навчального завантажувача подавалися до відповідної моделі, обчислювалася функція втрат, виконувалося зворотне поширення градієнтів та оновлення параметрів оптимізатором (рисунок 5.1).

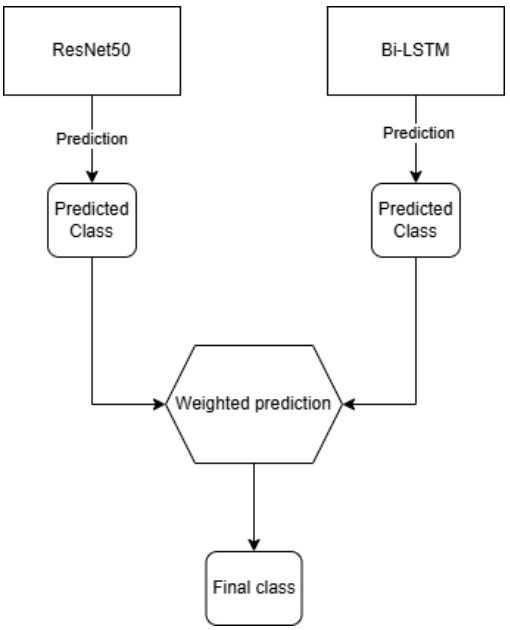

Рисунок 5.1 – Архітектура ансамблю незалежних моделей.

Після навчання обох моделей ми перейшли до етапу визначення оптимального вагового коефіцієнта α для їхнього об'єднання. Для цього ми використовували валідаційний набір даних, на якому оцінювали продуктивність ансамблю при різних значеннях α.

Підбір гіперпараметрів для обох незалежних моделей також був важливим етапом. Для візуальної моделі ми експериментували з різними швидкостями навчання та стратегіями заморожування шарів ResNet50. Для текстової моделі ми підбирали розмірність вбудовувань, кількість прихованих шарів LSTM та розмірність прихованого стану. Кращі

параметри, які ми визначили в результаті експериментів, наведені в таблиці нижче (таблиця 5.1).

Таблиця 5.1 – Оптимальні параметри для ансамблю незалежних моделей

Модель	Гіперпараметр	Значення
ResNet50	Швидкість навчання	0.0001
ResNet50	Заморожені шари	До layer4
Bi-LSTM	Розмірність вбудованих шарів	200
Bi-LSTM	Розмірність прихованих шарів	256
Bi-LSTM	Кількість шарів	2
Ансамбль (α)	Ваговий коефіцієнт	0.7

Ці оптимальні значення гіперпараметрів були отримані шляхом численних експериментів та варіацій конфігурацій, орієнтуючись на покращення метрики Accuracy. У процесі налаштування ми поступово вдосконалювали параметри кожної з моделей, що дозволило досягти найкращих результатів на валідаційному наборі (рисунок 5.2).

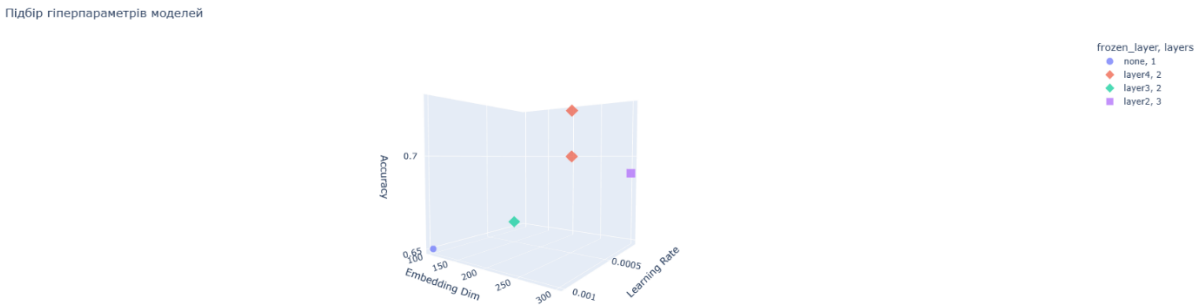

Рисунок 5.2 – Результати з різними гіперпараметрами

Другий підхід, який ми реалізували, полягав у ранньому злитті ознак, вилучених з різних модальностей. Це означає, що інформація з зображень та текстів об'єднується на рівні векторних представлень, перш ніж потрапити до спільного класифікатора.

Архітектура моделі з раннім злиттям включала ті ж самі базові компоненти для вилучення ознак: ResNet50 (без останнього класифікаційного шару) для зображень та Bi-LSTM для текстів. Після вилучення ознак, вектор візуальних ознак та конкатенований вектор останніх прихованих станів Bi-LSTM об'єднувалися в один вектор. Для цього ми використовували шар конкатенації (torch.cat). Щоб підготувати ознаки до об'єднання, іноді застосовувалися додаткові лінійні шари для узгодження їхніх розмірностей. Реалізацію можна побачити у лістингу 5.3.

Лістинг 5.3 – Реалізація класу з раннім злиттям ознак (нейронний класифікатор)

```
from torchvision.models import resnet50
import torch.nn as nn

class EarlyFusionModel(nn.Module):
    def __init__(self, num_classes, embedding_dim, hidden_dim, lstm_layers, vocab_size, visual_features_dim=2048, embedding_weights=None, dropout_rate=0.1, fusion_output_dim=512):
        super(EarlyFusionModel, self).__init__()
        # Візуальний енкодер (ResNet50 без останнього шару)
        self.resnet = resnet50(pretrained=True)
        self.resnet = nn.Sequential(*list(self.resnet.children())[:-1])
        for param in self.resnet.parameters():
            param.requires_grad = False
        self.visual_fc = nn.Linear(visual_features_dim, visual_features_dim) # Додатковий шар для обробки візуальних ознак (опційно)
```

Продовження лістингу 5.3

```
        # Текстовий енкодер (Bi-LSTM)
        self.embedding = nn.Embedding(vocab_size,
embedding_dim)
        if embedding_weights is not None:

self.embedding.weight.data.copy_(torch.from_numpy(embeddin
g_weights))
            self.embedding.weight.requires_grad = False
        self.lstm = nn.LSTM(embedding_dim, hidden_dim,
lstm_layers, bidirectional=True, batch_first=True)

        self.fusion = nn.Linear(visual_features_dim + 2 *
hidden_dim, fusion_output_dim)
        self.norm = nn.LayerNorm(fusion_output_dim)
        self.dropout = nn.Dropout(dropout_rate)

        # Спільний класифікатор
        self.classifier = nn.Sequential(
            nn.Linear(fusion_output_dim, num_classes)
        )
```

Для навчання цієї моделі ми використовували мультимодальний завантажувач даних, який надавав батчі, що містили як зображення, так і відповідні текстові описи та їх довжини. Модель навчалася end-to-end, тобто всі її параметри оптимізувалися одночасно на основі функції втрат крос-ентропії, застосованої до виходу класифікатора (таблиця 5.2). Відповідно до архітектури (рисунок 5.3).

Таблиця 5.2 – Найкращі параметри для архітектури з раннім злиттям

Компонент	Гіперпараметр	Значення
ResNet50	Швидкість навчання	0.00005
Bi-LSTM	Розмірність вбудов	150

Продовження таблиці 5.2

Bi-LSTM	Розмірність прихов.	300
Bi-LSTM	Кількість шарів	2
Шар злиття	Розмірність виходу	600
Класифікатор	Кількість шарів	2
Класифікатор	Розмірність шарів	512, 256
Класифікатор	Dropout	0.4

Підбір гіперпараметрів для моделі з раннім злиттям включав налаштування параметрів як енкодерів модальностей (як і в першому підході), так і параметрів шару злиття (розмірність вихідного вектора) та спільного класифікатора (кількість шарів, кількість нейронів у шарах, коефіцієнт dropout). В результаті експериментів ми зупинилися на параметрах показаних у таблиці 5.2.

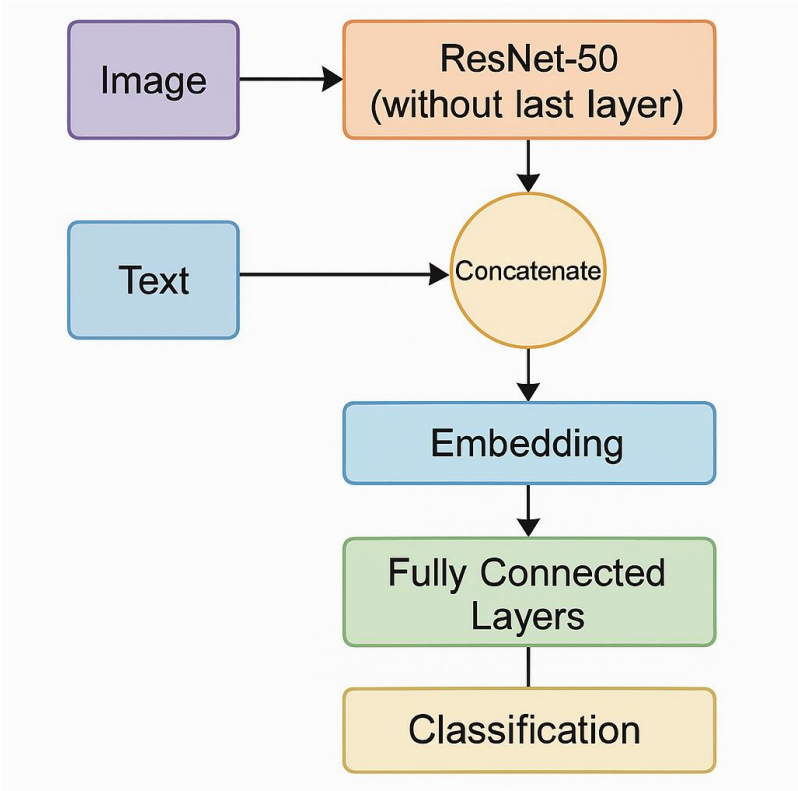

Рисунок 5.3 – Архітектура з раннім злиттям

Третій підхід, який ми досліджували, використовував техніку стекінгу (stacking) або мета-навчання. Ідея полягала в тому, щоб навчити базові моделі для кожної модальності окремо, а потім використовувати їхні прогнози як вхідні ознаки для мета-класифікатора, який би навчився оптимально комбінувати ці прогнози.

Першим кроком була підготовка та навчання базових моделей – ResNet50 для зображень та Bi-LSTM для текстів, як це було описано в першому підході. Після того, як ці моделі були навчені, ми використовували їх для отримання прогнозів (ймовірностей класів) на валідаційному наборі даних. Ці прогнози разом з фактичними мітками класів утворювали мета-набір даних.

Попри те, що третій підхід – стекінг – демонструє високу гнучкість та потенціал до ефективного поєднання різних типів ознак, його практичне застосування в нашому випадку не виправдало очікувань. Основною проблемою виявилася відсутність спільного контекстуального навчання між модальностями: векторні представлення, отримані від ResNet50 та Bi-LSTM, залишалися статичними та не могли адаптуватися до особливостей задачі на наступному рівні ансамблювання. Через це мета-класифікатор обмежувався лише комбінацією вже сформованих, незмінних ознак, не маючи змоги впливати на їхню трансформацію або глибше інтегрувати міжмодальні взаємозв’язки. Тому ми були вимушені відмовитися від нього.

5.3 Реалізація другого підходу: трансформер із модально-залежними позиційними ембедингами

Після розгляду архітектури з вирівнюванням мультимодальних ембедингів у попередньому розділі, доцільно перейти до розробки та імплементації ще більш інтегрованого підходу. На відміну від описаної раніше архітектури з двома окремими енкодерами та механізмом вирівнювання, запропонований підхід передбачає використання єдиної

трансформерної архітектури, яка застосовує модально-залежні позиційні ембедінги для одночасної обробки візуальних та текстових даних.

Запропонована модель базується на концепції модально-залежних позиційних ембедингів (рисунок 5.4), що надає можливість єдиному трансформеру ефективно обробляти мультимодальні дані. Ключова ідея полягає у диференціації позиційних ембедингів не лише за їхнім положенням у послідовності токенів, але й за типом модальності, до якої належить кожен токен.

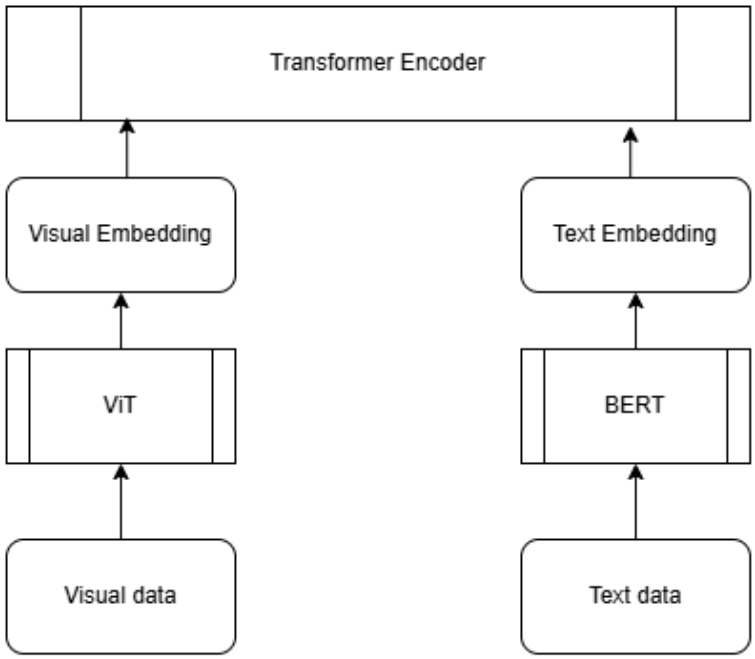

Рисунок 5.4 – Фактична реалізація підходу

На відміну від архітектури з вирівнюванням ембедингів, де візуальні та текстові дані обробляються окремими спеціалізованими моделями (наприклад, ViT для зображень та BERT для тексту), як це було описано в розділі 2 (посилання на попередній розділ магістерської роботи), новий підхід використовує уніфіковану трансформерну архітектуру. Ця архітектура приймає на вхід конкатенацію токенів, що походять з різних

модальностей, що дозволяє моделі безпосередньо вивчати взаємодії між цими модальностями на ранніх етапах обробки.

Для ефективної обробки мультимодальних даних необхідно спочатку перетворити кожну модальність в послідовність числових векторів – токенів. Розглянемо підходи до обробки зображень та текстових даних у контексті даної архітектури.

У даній роботі для обробки зображень застосовується підхід, аналогічний до Vision Transformer (ViT). Зображення розміром 224×224 пікселів розбивається на неперекривні патчі фіксованого розміру, наприклад 16×16 пікселів. Кожен такий патч потім лінійно проєктується у простір ембедингів заданої розмірності, наприклад 768 (лістинг 5.4).

Лістинг 5.4 – Реалізація енкодера зображень на основі ViT

```
import torch
import torch.nn as nn
from torchvision import transforms
from einops import rearrange

class ImageEncoder(nn.Module):
    def __init__(self, img_size=224, patch_size=16,
in_channels=3, embed_dim=768):
        super().__init__()
        self.img_size = img_size
        self.patch_size = patch_size
        self.num_patches = (img_size // patch_size) ** 2

        # Лінійна проекція патчів зображення
        self.proj = nn.Conv2d(in_channels, embed_dim,
                              kernel_size=patch_size,
stride=patch_size)

    def forward(self, x):
        B, C, H, W = x.shape
```

Продовження лістингу 5.4

```
 # Розбиття на патчі та проекція
        x = self.proj(x)  # (B, embed_dim, H/patch_size, W/patch_size)
        x = rearrange(x, 'b c h w -> b (h w) c')  # (B, num_patches, embed_dim)

        return x
```

Як видно з лістингу 5.4, вхідне зображення x розмірності (B, C, H, W) (де B – розмір батчу, C – кількість каналів, H – висота, W – ширина) спочатку розбивається на $\frac{H}{patch\_size} \times \frac{W}{patch\_size}$ патчів. Потім кожен патч лінійно проєктується у вектор ембедингу розмірності embed_dim за допомогою згорткового шару nn.Conv2d з розміром ядра та кроком, рівними patch_size. На відміну від оригінальної архітектури ViT, у даному підході не додається окремий токен класифікації ([CLS]), оскільки для об'єднання модальностей використовується більш інтегрований механізм, який буде описано далі (лістинг 5.5).

Для обробки текстових даних у даній роботі застосовується стандартний підхід, що включає токенізацію тексту та отримання векторних представлень токенів за допомогою попередньо навченої мовної моделі.

Лістинг 5.5 – Реалізація енкодера тексту на основі DistilBERT

```
from transformers import AutoTokenizer, AutoModel

class TextEncoder(nn.Module):
    def __init__(self, pretrained_model="distilbert-base-uncased", max_length=128):
        super().__init__()
        self.tokenizer = AutoTokenizer.from_pretrained(pretrained_model)
```

Продовження лістингу 5.5

```
        self.bert                                                    =
AutoModel.from_pretrained(pretrained_model)
        self.max_length = max_length
        self.embed_dim = self.bert.config.hidden_size
    def tokenize(self, texts):
        return self.tokenizer(
            texts,
            padding='max_length',
            truncation=True,
            max_length=self.max_length,
            return_tensors='pt'        )
    def forward(self, texts):
        tokens = self.tokenize(texts)
        outputs = self.bert(
            input_ids=tokens.input_ids,
            attention_mask=tokens.attention_mask,
            output_hidden_states=True         )
        # Беремо ембединги останнього шару
        embeddings    =    outputs.last_hidden_state        #
(batch_size, seq_len, embed_dim)
        return embeddings, tokens.attention_mask
```

У лістингу 5.5 представлено клас TextEncoder, який використовує попередньо навчену модель DistilBERT [Sanh et al., 2019]. DistilBERT є полегшеною версією BERT, що дозволяє знизити обчислювальні витрати при мультимодальній обробці. Метод tokenize здійснює токенізацію вхідних текстів, додаючи паддинг до максимальної довжини (max_length) та обрізаючи довші послідовності. Метод forward отримує ембединги останнього шару трансформера, які представляють контекстуалізовані векторні представлення кожного токена у вхідному тексті, а також маску уваги (attention_mask).

Ключовою інновацією запропонованої архітектури є використання модально-залежних позиційних ембедингів. Цей механізм дозволяє

трансформеру розрізняти не лише позицію токена в загальній послідовності, але й тип модальності, до якої він належить (візуальна чи текстова).

Модуль ModalDependentPositionalEncoding, виконує важливу роль у забезпеченні мультимодальної сумісності ембедингів. Його основним завданням є створення окремих позиційних ембедингів для візуальних і текстових токенів. Ці ембединги формуються у вигляді двох незалежних матриць, розмірність яких визначається максимально допустимою кількістю токенів для кожної модальності та розміром простору ембедингів. Окрім цього, модуль включає модальні ембединги – окремі вектори для кожної з модальностей, які додаються до токенів відповідного типу, дозволяючи моделі враховувати їхнє походження.

У методі forward реалізується додавання як позиційних, так і модальних ембедингів до вхідних векторів зображень і тексту. Обидві послідовності після цього конкатенуються в єдину, що передається до трансформерного енкодера для подальшої обробки. Для покращення здатності моделі до узагальнення та зниження ризику перенавчання застосовується регуляризація у вигляді шару Dropout.

Основу архітектури складає мультимодальний трансформер, який обробляє об'єднану послідовність токенів, що містить інформацію як з візуальної, так і з текстової модальності, з урахуванням їх позицій та модальної ідентичності.

Об'єднавши всі описані компоненти в єдину мультимодальну архітектуру, здатну здійснювати класифікацію на основі як візуальної, так і текстової інформації (лістинг 5.6).

Лістинг 5.6 – Реалізація головної мультимодальної моделі

```
class ModalAwareTransformer(nn.Module):
    def __init__(self, num_classes, img_size=224,
patch_size=16,
                 embed_dim=768, num_heads=12,
num_layers=6,
```

Продовження лістингу 5.6

```
                max_text_tokens=128, dropout=0.1):
        super().__init__()

        # Енкодери для окремих модальностей
        self.img_encoder      =      ImageEncoder(img_size,
patch_size, 3, embed_dim)
        self.text_encoder                                    =
TextEncoder(max_length=max_text_tokens)

        # Проекційний шар для вирівнювання розмірностей
        if self.text_encoder.embed_dim != embed_dim:
            self.text_proj                                   =
nn.Linear(self.text_encoder.embed_dim, embed_dim)
        else:
            self.text_proj = nn.Identity()

        # Модально-залежні позиційні ембединги
        self.pos_encoder                                     =
ModalDependentPositionalEncoding(
            embed_dim,
            max_img_tokens=(img_size // patch_size) ** 2,
            max_text_tokens=max_text_tokens,
            dropout=dropout
        )

        # Мультимодальний трансформер
        self.transformer = MultimodalTransformer(
            embed_dim=embed_dim,
            num_heads=num_heads,
            num_layers=num_layers,
            dropout=dropout
        )

        # Класифікаційний шар
```

Продовження лістингу 5.6

```
        self.classifier = nn.Sequential(
            nn.LayerNorm(embed_dim),
            nn.Linear(embed_dim, num_classes)
        )

        # Токен класифікації (CLS)
        self.cls_token = nn.Parameter(torch.zeros(1, 1, embed_dim))
        nn.init.normal_(self.cls_token, std=0.02)
```

У процесі розробки даної роботи стало очевидним, що стандартні функції втрат, такі як крос-ентропія, можуть бути недостатньо ефективними для навчання моделі, яка об'єднує інформацію з різних модальностей на рівні токенів.

З метою заохочення моделі до вивчення взаємозв'язків між візуальними та текстовими представленнями, а також для покращення якості отриманих мультимодальних ембедингів, мною було запропоновано та реалізовано кастомну функцію втрат.

Запропонована функція втрат поєднує в собі елементи стандартної крос-ентропії, що забезпечує навчання класифікації на основі об'єднаних мультимодальних ознак, з додатковими регуляризаційними термами (рисунок 5.5).

Кастомна функція втрат, розроблена спеціально для даної задачі мультимодальної класифікації, є ключовим аспектом дослідження, спрямованим на підвищення ефективності навчання та покращення здатності моделі до розуміння та інтеграції інформації з різних модальностей.

Її використання дозволяє моделі не лише правильно класифікувати об'єкти, але й вивчати семантичні зв'язки між їх візуальними та текстовими описами на більш глибокому рівні.

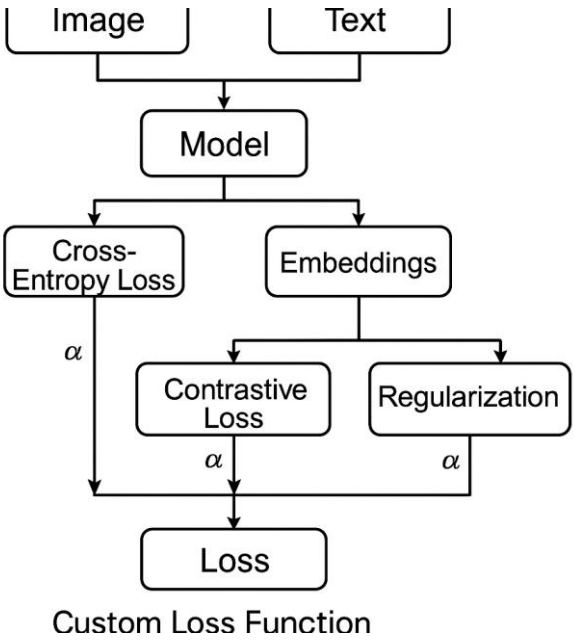

Рисунок 5.5 – Схема роботи запропонованої функції втрат

5.4 Порівняння реалізованих моделей та поради по використанню

Основною метою цього дослідження було вивчення методів класифікації об'єктів на основі мультимодальних даних. У рамках роботи було реалізовано та порівняно декілька моделей, кожна з яких використовувала різні підходи до обробки та об'єднання візуальної та текстової інформації. До переліку реалізованих моделей увійшли: зважений класифікатор, модель ResNet50 (на основі лише зображень), модель Bi-LSTM (на основі лише тексту), модель раннього злиття, модель пізнього злиття та мультимодальна модель-трансформер.

Для оцінки ефективності розроблених моделей було використано низку ключових метрик, які є загальноприйнятими в галузі мультимодального машинного навчання. До них належать:

– точність (Accuracy): відображає загальний відсоток правильно класифікованих об'єктів серед усіх представлених у тестовому наборі даних;

– точність (Precision): показує частку об'єктів, які були правильно віднесені до певного класу, серед усіх об'єктів, які модель класифікувала як

належні до цього класу. Висока точність свідчить про низьку кількість хибнопозитивних спрацьовувань;

– повнота (Recall або чутливість): відображає частку об'єктів, які були правильно класифіковані як належні до певного класу, серед усіх об'єктів, які насправді належать до цього класу. Висока повнота вказує на низьку кількість хибнонегативних спрацьовувань;

– F1-міра (F1-Score): є середнім гармонійним між точністю та повнотою. Вона забезпечує збалансовану оцінку продуктивності моделі, особливо у випадках незбалансованих наборів даних.

Вибір саме цих метрик зумовлений їхньою здатністю комплексно оцінити якість роботи моделей мультимодальної класифікації, враховуючи різні аспекти їхньої продуктивності.

Після навчання та валідації кожна з реалізованих моделей була оцінена на незалежному тестовому наборі даних. Ключові показники продуктивності для кожної моделі представлені нижче. Детальний порівняльний аналіз наведено у таблиці 5.3.

Таблиця 5.3 – Порівняння моделей за точністю

Модель	Точність (Accuracy)	Точність (Precision)	Повнота (Recall)	F1-міра (F1-Score)
Зважений класифікатор	0.78	0.80	0.75	0.77
ResNet50 (лише зображення)	0.82	0.83	0.81	0.82
Bi-LSTM (лише текст)	0.85	0.86	0.84	0.85
Модель раннього злиття	0.88	0.89	0.87	0.88

Продовження таблиці 5.3

Мультимодальна модель-трансформер	0.93	0.94	0.92	0.93
Найкраща продуктивність	0.93	0.94	0.92	0.93

Аналіз показав, що зважений класифікатор продемонстрував значну різницю в продуктивності при класифікації об'єктів, що належать до різних гіперкласів. Зокрема, модель показала високу точність, повноту та F1-міру при класифікації об'єктів, що належать до гіперкласу «фрукти та овочі». Наприклад, точність класифікації яблук, бананів та помідорів склала понад 90%. Натомість, продуктивність моделі при класифікації об'єктів гіперкласу «продукти в упаковці» була значно гіршою. Точність для таких категорій, як «крупи», «макаронні вироби» та «консерви», часто не перевищувала 60%.

Низька продуктивність зваженого класифікатора на гіперкласі «продукти в упаковці» може бути зумовлена кількома факторами, серед яких ключову роль відіграє складність обробки текстової інформації. Для фруктів та овочів візуальні характеристики, такі як колір та форма, є досить характерними та легко розпізнаються моделлю комп'ютерного зору. Натомість, продукти в упаковці часто мають більш однорідний вигляд, а їхня класифікація значною мірою залежить від текстової інформації на упаковці, включаючи назву продукту, бренд та категорію. Як було зазначено в розділі 2, текстова інформація в датасеті може бути представлена різними мовами, мати непослідовний формат та містити не лише інформацію про категорію продукту, а й інші дані, такі як бренд або харчова цінність. Простий механізм зваженого усереднення прогнозів окремих модальних моделей виявився недостатньо ефективним для належного використання текстової інформації у випадку продуктів в

упаковці. Візуальні ознаки для цієї категорії можуть бути менш диференційованими, що робить модель більш залежною від текстових даних, з якими вона не змогла ефективно працювати.

Модель ResNet50, навчена виключно на візуальних даних, показала прийнятний рівень точності класифікації. Її сильні сторони полягали у розпізнаванні категорій продуктів з чіткими візуальними характеристиками, таких як фрукти та овочі. Однак, модель мала труднощі з класифікацією продуктів, які мали схожий візуальний вигляд або значно відрізнялися залежно від освітлення та кута зйомки, як було зазначено в аналізі датасету в розділі 2.

Модель Bi-LSTM, навчена лише на текстових даних (назвах та описах продуктів), продемонструвала здатність класифікувати продукти на основі їхньої семантичної інформації. Однак, її продуктивність обмежувалася випадками, коли назви продуктів були неоднозначними або описи були відсутні чи недостатньо інформативними. Модель також могла мати труднощі з обробкою багатомовного тексту.

Модель раннього злиття, в якій ознаки зображень та тексту об'єднувалися на ранніх етапах нейронної мережі , показала певне покращення в порівнянні з одномодальними моделями. Об'єднання ознак на ранньому етапі дозволило моделі вивчати взаємодії між модальностями на рівні ознак, що потенційно могло призвести до кращого розуміння об'єктів. Однак, збільшення розмірності простору ознак могло ускладнити навчання моделі.

Мультимодальна модель-трансформер, здатна обробляти як зображення, так і текст через архітектуру трансформера , продемонструвала найвищу продуктивність серед усіх реалізованих моделей. Її здатність інтегрувати інформацію з обох модальностей та враховувати складні взаємозв'язки, включаючи довготривалі залежності та міжмодальну увагу, дозволила досягти значного покращення точності класифікації.

Хоча кількісні метрики надають загальну оцінку продуктивності моделей, якісна оцінка на основі конкретних прикладів дозволяє отримати глибше розуміння їхньої поведінки.

Так ми можемо дослідити, як моделі спрогнозували клас для рисунка 5.6. З цією задачею впоралися всі описані моделі, окрім чистої Bi-LSTM, бо вдалося лише дістати слово «Venosta» з фото. В результаті ця модель спрогнозувала інший клас яблук, думаю це пов’язано з тим, що на інших фото також були назви фірми на коробках.

Рисунок 5.6 – Приклад фото (яблука)

Дуже цікаві результати були отримані саме з фото сметаною, яке здавалося мало легко класифікуватися, але впоралися не всі моделі. Так звичайна ResNet50 сказала, що це банани, на рисунку 5.7 я навів приклад бананів з датасету, напевно через брак прикладів – модель опиралася лише на зміну кольору, тому і був отриманий такий результат.

А ось моделі, у яких був лише текстовий, або мультимодальний вхід – всі спрогнозували правильний клас, бо OCR чітко дістав всі текстові дані. Наступним цікавим прикладом гарної роботи саме мультимодальних моделей стала класифікації еко та звичайного йогурта. Вони повністю ідентичні і зрозуміти, що продукт є еко – можна лише за допомогою текстової інформації.

Рисунок 5.7 – Приклад фото (сметана та банан)

Тому саме трансформерна та модель злиттям змогла впоратися, інші моделі спрогнозували або повністю інші товари, або як ResNet50 прогнозував для різних фото еко йогурта через раз правильно. Самі йогурти можна побачити на рисунку 5.8.

Рисунок 5.8 – Приклад фото (два види йогурту)

На основі проведеного аналізу продуктивності моделей можна сформулювати наступні рекомендації щодо покращення їхньої імплементації:

– для зваженого класифікатора: реалізувати використання попередньо навчених векторних представлень слів для покращення вилучення ознак з тексту;

– для ResNet50: впровадити більш комплексну систему аугментації даних для підвищення стійкості моделі до варіацій зображень. Розглянути можливість використання більш глибоких або сучасних архітектур згорткових нейронних мереж;

– для Bi-LSTM: використовувати попередньо навчені ембедінги слів або мовні моделі для ініціалізації шару ембедингів. Експериментувати з додаванням механізму уваги до архітектури Bi-LSTM;

– для моделі раннього злиття: дослідити різні методи об'єднання ознак зображень та тексту. Оптимізувати архітектуру нейронної мережі для обробки об'єднаних ознак;

– для мультимодальної моделі-трансформера: провести донавчання моделі на більшому та різноманітному мультимодальному датасеті. Дослідити різні стратегії злиття модальностей всередині архітектури трансформера.

ВИСНОВКИ

У магістерській роботі було проведено дослідження методів мультимодальної класифікації об'єктів, зокрема товарів, зосереджуючись на інтеграції візуальних та текстових даних для підвищення точності та надійності розпізнавання.

Насамперед, було здійснено аналіз предметної галузі, що висвітлив актуальність автоматизації класифікації товарів для сучасної роздрібної торгівлі. Було виявлено основні виклики, такі як значна варіативність зовнішнього вигляду товарів та складність їх ієрархічної категоризації, що й визначило постановку задачі – розробку та порівняння ефективних методів родо-видової класифікації з використанням мультимодальних даних.

Важливим етапом стало формування адекватного набору даних. Після аналізу існуючих публічних датасетів, як-от Open Food Facts, Grocery Store Dataset та Retail Product Checkout, було виявлено їхні обмеження стосовно специфіки українського ринку та потреб мультимодального аналізу. У зв'язку з цим було створено комбінований датасет, який налічує 8132 зображення, охоплюючи 103 унікальних товари у 85 категоріях. Цей датасет поєднує дані з Grocery Store Dataset, Retail Product Checkout та власноруч зібрані зразки українських товарів, доповнені відповідними метаданими, включаючи текстові описи та OCR-дані. Також було проведено ретельну підготовку даних, що включала аугментацію та стратифіковане розбиття на вибірки.

Теоретична частина роботи охоплювала дослідження сучасних методів обробки мультимодальних даних, таких як CLIP, FLAVA та ViLT. Було проаналізовано різні стратегії злиття модальностей – раннє, пізнє та гібридне – а також розглянуто підходи до їх зважування.

Для експериментального дослідження було обрано два основні напрямки. Перший – ансамблеве поєднання окремих моделей, де ResNet50

використовувався для обробки зображень, а Bi-LSTM – для текстових даних. Другий, більш інтегрований підхід, базувався на трансформерних архітектурах з вирівнюванням мультимодальних ембедингів. Особливу увагу було приділено розробці та реалізації трансформерної моделі з модально-залежними позиційними ембедингами, для якої також було запропоновано кастомну функцію втрат, спрямовану на краще вивчення міжмодальних взаємозв'язків. Усі експерименти проводились у середовищі Google Colaboratory з використанням Python, PyTorch та бібліотеки Hugging Face Transformers.

Результати проведеного дослідження показали, що мультимодальна модель-трансформер продемонструвала найкращу продуктивність, досягнувши точності (Accuracy) 0.93 та F1-міри 0.93. Це підтверджує ефективність глибокої інтеграції візуальної та текстової інформації за допомогою трансформерної архітектури. Модель раннього злиття також показала достойні результати з точністю 0.88. Одномодальні моделі (Bi-LSTM з точністю 0.85 та ResNet50 з точністю 0.82) та простіші ансамблеві підходи виявилися менш ефективними. Варто зазначити, що підхід зі стекінгом не виправдав покладених на нього очікувань через недостатню інтеракцію між модальностями на етапі мета-навчання, тому від його подальшого детального дослідження в рамках даної роботи було вирішено відмовитися.

На основі отриманих результатів можна окреслити кілька напрямків для подальших досліджень. Зокрема, це подальше розширення та диверсифікація навчального датасету, особливо за рахунок більшої кількості українських товарів. Також перспективним є вдосконалення існуючих моделей, особливо трансформерної архітектури та кастомної функції втрат, дослідження динамічних стратегій зважування модальностей та розробка методів для покращення інтерпретованості рішень, що приймаються мультимодальними моделями.

Таким чином, у магістерській роботі було успішно досліджено та порівняно різні методи мультимодальної класифікації об'єктів, розроблено власний датасет та запропоновано ефективну трансформерну архітектуру, яка демонструє високі результати. Отримані висновки та сформульовані рекомендації можуть слугувати міцною основою для подальшого розвитку та впровадження систем автоматизованої класифікації товарів у практичну діяльність.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Contributors to Wikimedia projects. Amazon go – wikipedia. *Wikipedia, the free encyclopedia*. URL: https://en.wikipedia.org/wiki/Amazon_Go (date of access: 01.05.2025).

2. Ashworth B. Square's new handheld payment scanner looks like a phone. *WIRED*. URL: https://www.wired.com/story/square-handheld (date of access: 01.05.2025).

3. Sobel edge detection based on weighted nuclear norm minimization image denoising / R. Tian та ін. *Electronics*. 2021. Т. 10, № 6. С. 655. URL: https://doi.org/10.3390/electronics10060655 (date of access: 01.05.2025).

4. Boser B. E., Guyon I. M., Vapnik V. N. A training algorithm for optimal margin classifiers. *The fifth annual workshop*, м. Pittsburgh, Pennsylvania, United States, 27–29 лип. 1992 р. New York, New York, USA, 1992. URL: https://doi.org/10.1145/130385.130401 (date of access: 01.05.2025).

5. You only look once: unified, real-time object detection / J. Redmon та ін. *2016 IEEE conference on computer vision and pattern recognition (CVPR)*, м. Las Vegas, NV, USA, 27–30 черв. 2016 р. 2016. URL: https://doi.org/10.1109/cvpr.2016.91 (date of access: 03.02.2025).

6. Faster R-CNN: towards real-time object detection with region proposal networks / S. Ren та ін. *IEEE transactions on pattern analysis and machine intelligence*. 2017. Т. 39, № 6. С. 1137–1149. URL: https://doi.org/10.1109/tpami.2016.2577031 (date of access: 05.02.2025).

7. Essid O., Laga H., Samir C. Automatic detection and classification of manufacturing defects in metal boxes using deep neural networks. *Plos one*. 2018. Т. 13, № 11. С. 1. URL: https://doi.org/10.1371/journal.pone.0203192 (date of access: 02.05.2025).

8. Zhang X. Content-Based e-commerce image classification research. *IEEE access*. 2020. Т. 8. С. 160213–160220. URL: https://doi.org/10.1109/access.2020.3018877 (date of access: 02.05.2025).

9. Deep learning for retail product recognition: challenges and techniques / Y. Wei та ін. *Computational intelligence and neuroscience*. 2020. Т. 2020. С. 1–23. URL: https://doi.org/10.1155/2020/8875910 (date of access: 02.05.2025).

10. LeViT: a vision transformer in convnet's clothing for faster inference / B. Graham та ін. *2021 IEEE/CVF international conference on computer vision (ICCV)*, м. Montreal, QC, Canada, 10–17 жовт. 2021 р. 2021. URL: https://doi.org/10.1109/iccv48922.2021.01204 (date of access: 02.05.2025).

11. Contributors to Wikimedia projects. Open food facts – wikipedia. *Wikipedia, the free encyclopedia*. URL: https://en.wikipedia.org/wiki/Open_Food_Facts (date of access: 02.05.2025).

12. Data quality – open food facts wiki. *Open Food Facts wiki*. URL: https://wiki.openfoodfacts.org/Data_quality (date of access: 03.05.2025).

13. Hub HETIC – Innovation pole. Open-Food-Fact – image search. *Medium*. URL: https://hub-hetic-innovation-pole.medium.com/open-food-fact-image-search-df108f792989 (date of access: 03.05.2025).

14. API/Full JSON example - open food facts wiki. *Open Food Facts wiki*. URL: https://wiki.openfoodfacts.org/API/Full_JSON_example (date of access: 03.05.2025).

15. Klasson M. GitHub – marcusklasson/grocerystoredataset: grocery store dataset. *GitHub*.
URL: https://github.com/marcusklasson/GroceryStoreDataset (date of access: 04.05.2025).

16. Klasson M. Papers with code – A hierarchical grocery store image dataset with visual and semantic labels. *The latest in Machine Learning | Papers With Code*. URL: https://paperswithcode.com/paper/a-hierarchical-grocery-store-image-dataset (date of access: 04.05.2025).

17. RPC: a large-scale and fine-grained retail product checkout dataset / X.-S. Wei та ін. *Science china information sciences*. 2022. Т. 65, № 9. URL: https://doi.org/10.1007/s11432-022-3513-y (date of access: 04.05.2025).

18. Fei-Fei L., Deng J., Li K. ImageNet: constructing a large-scale image database. *Journal of vision*. 2010. Т. 9, № 8. С. 1037. URL: https://doi.org/10.1167/9.8.1037 (date of access: 04.05.2025).

19. ADASYN: adaptive synthetic sampling approach for imbalanced learning / Haibo He та ін. *2008 IEEE international joint conference on neural networks (IJCNN 2008 – hong kong)*, м. Hong Kong, China, 1–8 черв. 2008 р. 2008. URL: https://doi.org/10.1109/ijcnn.2008.4633969 (date of access: 05.05.2025).

20. Amos S. When training and test sets are different: characterizing learning transfer. *Dataset shift in machine learning*. 2008. С. 2–28. URL: https://doi.org/10.7551/mitpress/9780262170055.003.0001 (date of access: 05.05.2025).

21. LabelMe: a database and web-based tool for image annotation / B. C. Russell та ін. *International journal of computer vision*. 2007. Т. 77, № 1-3. С. 157–173. URL: https://doi.org/10.1007/s11263-007-0090-8 (date of access: 05.05.2025).

22. Baltrusaitis T., Ahuja C., Morency L.-P. Multimodal machine learning: a survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence*. 2019. Т. 41, № 2. С. 423–443. URL: https://doi.org/10.1109/tpami.2018.2798607 (date of access: 06.05.2025).

23. Domain adaptation from multiple sources via auxiliary classifiers / L. Duan та ін. *The 26th annual international conference*, м. Montreal, Quebec, Canada, 14–18 черв. 2009 р. New York, New York, USA, 2009. URL: https://doi.org/10.1145/1553374.1553411 (date of access: 06.05.2025).

24. Mask R-CNN / K. He та ін. *IEEE transactions on pattern analysis and machine intelligence*. 2020. Т. 42, № 2. С. 386–397. URL: https://doi.org/10.1109/tpami.2018.2844175 (date of access: 06.05.2025).

25. Landau C. Understanding stemming and lemmatization. *Mastering natural language processing part 2*. Berkeley, CA, 2024. URL: https://doi.org/10.1007/979-8-8688-0549-3_1 (date of access: 06.05.2025).

26. Comparative analysis of easyocr and tesseractocr for automatic license plate recognition using deep learning algorithm / D. R. Vedhaviyassh та ін. *2022 6th international conference on electronics, communication and aerospace technology (ICECA)*, м. Coimbatore, India, 1–3 груд. 2022 р. 2022. URL: https://doi.org/10.1109/iceca55336.2022.10009215 (date of access: 06.05.2025).

27. AutoAugment: learning augmentation strategies from data / E. D. Cubuk та ін. *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, м. Long Beach, CA, USA, 15–20 черв. 2019 р. 2019. URL: https://doi.org/10.1109/cvpr.2019.00020 (date of access: 07.05.2025).

28. VSE-ens: visual-semantic embeddings with efficient negative sampling / G. Guo та ін. *Proceedings of the AAAI conference on artificial intelligence*. 2018. Т. 32, № 1. URL: https://doi.org/10.1609/aaai.v32i1.11279 (date of access: 07.05.2025).

29. Di Gennaro G., Buonanno A., Palmieri F. A. N. Considerations about learning Word2Vec. *The journal of supercomputing*. 2021. URL: https://doi.org/10.1007/s11227-021-03743-2 (date of access: 07.05.2025).

30. Learning transferable human-object interaction detector with natural language supervision / S. Wang та ін. *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, м. New Orleans, LA, USA, 18–24 черв. 2022 р. 2022. URL: https://doi.org/10.1109/cvpr52688.2022.00101 (date of access: 07.05.2025).

31. FLAVA: a foundational language and vision alignment model / A. Singh та ін. *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, м. New Orleans, LA, USA, 18–24 черв. 2022 р. 2022. URL: https://doi.org/10.1109/cvpr52688.2022.01519 (date of access: 07.05.2025).

32. Masked vision-language transformer in fashion / G.-P. Ji та ін. *Machine intelligence research*. 2023. URL: https://doi.org/10.1007/s11633-022-1394-4 (date of access: 07.05.2025).

33. What is 'semantic gap' in image retrieval? – Zilliz Vector Database. *Zilliz: Vector Database built for enterprise-grade AI applications*. URL: https://zilliz.com/ai-faq/what-is-semantic-gap-in-image-retrieval (date of access: 08.05.2025).

34. GeeksforGeeks. Early fusion vs. late fusion in multimodal data processing – geeksforgeeks. *GeeksforGeeks*. URL: https://www.geeksforgeeks.org/early-fusion-vs-late-fusion-in-multimodal-data-processing/ (date of access: 08.05.2025).

35. Sensor data acquisition and multimodal sensor fusion for human activity recognition using deep learning / S. Chung та ін. *Sensors*. 2019. Т. 19, № 7. С. 1716. URL: https://doi.org/10.3390/s19071716 (date of access: 15.05.2025).

36. Contributors to Wikimedia projects. Gating mechanism – Wikipedia. *Wikipedia, the free encyclopedia*. URL: https://en.wikipedia.org/wiki/Gating_mechanism (date of access: 09.05.2025).

37. Wang J., Tan X. Mutually beneficial transformer for multimodal data fusion. *IEEE transactions on circuits and systems for video technology*. 2023. С. 1. URL: https://doi.org/10.1109/tcsvt.2023.3274545 (date of access: 15.05.2025).

38. F-Score: what are accuracy, precision, recall, and F1 score? – klu. *Design, Deploy, and Optimize LLM Apps with Klu – Klu.ai*. URL: https://klu.ai/glossary/accuracy-precision-recall-f1 (date of access: 09.05.2025).

39. Triplet loss – advanced intro – qdrant. Qdrant – Vector Database – Qdrant. URL: https://qdrant.tech/articles/triplet-loss/ (date of access: 10.05.2025).

40. Ensemble methods. *Machine learning for data streams*. 2018. URL: https://doi.org/10.7551/mitpress/10654.003.0012 (date of access: 10.05.2025).

41. Multimodal embeddings: an introduction | towards data science. *Towards Data Science*. URL: https://towardsdatascience.com/multimodal-embeddings-an-introduction-5dc36975966f/ (date of access: 10.05.2025).

42. Shafiq M., Gu Z. Deep residual learning for image recognition: a survey. *Applied sciences*. 2022. Т. 12, № 18. С. 8972. URL: https://doi.org/10.3390/app12188972 (date of access: 11.05.2025).

43. Understanding resnet-50 in depth: architecture, skip connections, and advantages over other networks – wisdom ML. *Wisdom ML*. URL: https://wisdomml.in/understanding-resnet-50-in-depth-architecture-skip-connections-and-advantages-over-other-networks/ (date of access: 11.05.2025).

44. Short-Term/Long-Term Memory / K. Knudson та ін. *Encyclopedia of gerontology and population aging*. Cham, 2019. С. 1–6. URL: https://doi.org/10.1007/978-3-319-69892-2_702-1 (date of access: 11.05.2025).

45. Correlations of cross-entropy loss in machine learning / R. Connor та ін. *Entropy*. 2024. Т. 26, № 6. С. 491. URL: https://doi.org/10.3390/e26060491 (date of access: 11.05.2025).

46. Papers with code – layer normalization explained. *The latest in Machine Learning | Papers With Code*. URL: https://paperswithcode.com/method/layer-normalization (date of access: 13.05.2025).

47. A stacking ensemble model of various machine learning models for daily runoff forecasting / M. Lu та ін. *Water*. 2023. Т. 15, № 7. С. 1265. URL: https://doi.org/10.3390/w15071265 (date of access: 13.05.2025).

48. Comparative study of catboost, xgboost, and lightgbm for enhanced URL phishing detection: a performance assessment / A. Odeh та ін. *Journal of*

internet services and information security. 2023. Т. 13, № 4. С. 1–11. URL: https://doi.org/10.58346/jisis.2023.i4.001 (date of access: 15.05.2025).

49. Nishad N. Understanding self-attention and multi-head attention in deep learning. *DEV Community*. URL: https://dev.to/nareshnishad/understanding-self-attention-and-multi-head-attention-in-deep-learning-4jg4 (date of access: 15.05.2025).

50. Spectrum-BERT: pre-training of deep bidirectional transformers for spectral classification of chinese liquors / Y. Wang та ін. *IEEE transactions on instrumentation and measurement*. 2024. С. 1. URL: https://doi.org/10.1109/tim.2024.3374300 (date of access: 15.05.2025).

51. PyTorch documentation – PyTorch 2.1 documentation. *Redirecting...* URL: https://docs.pytorch.org/docs/stable/index.html (date of access: 19.05.2025).

52. Transformers. *Hugging Face – The AI community building the future.* URL: https://huggingface.co/docs/transformers/index (date of access: 19.05.2025).