

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 _____ перший (бакалаврський)

Розробка мобільного застосунку для класифікації побутових відходів та
навчання екологічному сортуванню на основі комп'ютерного зору
(тема)

Виконав:
здобувач _____ четвертого _____ року навчання,
групи _____ ІТШ-22-4

_____ Данііл ШКАРУПА
(власне ім'я, прізвище)

Спеціальність 122 Комп'ютерні науки

(код і повна назва спеціальності)

Тип програми _____ освітньо-професійна

Освітня програма _____ Штучний інтелект

(повна назва освітньої програми)

Керівник _____ ас. Максим ПОЛІТ
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ШІ

_____ Лариса ЧАЛА
(підпис) (власне ім'я, прізвище)

2026 р.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____

Кафедра _____ Штучного інтелекту _____

Рівень вищої освіти _____ перший (бакалаврський) _____

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

Тип програми _____ освітньо-професійна _____

Освітня програма _____ Штучний інтелект _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

« _____ » _____ 20 ____ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ Шкарупі Даніїлу Артемовичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Розробка мобільного застосунку для класифікації побутових відходів та навчання екологічному сортуванню на основі комп'ютерного зору _____

затверджена наказом університету від 5 червня 2026 р. № 488Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 10 липня 2026 р.

3. Вихідні дані до роботи _____ Документація мов: Python 3, Dart, документація фреймворків: Flutter, FastAPI, документація бібліотек: TensorFlow, Keras, SQLAlchemy, Fl_chart, Pandas, науково-технічні публікації та аналітичні звіти щодо стратегій управління твердими побутовими відходами, відкриті набори даних (datasets) зображень побутових відходів, технічна документація існуючих систем комп'ютерного зору. _____

4. Перелік питань, що потрібно опрацювати в роботі _____

1) Системний аналіз предметної галузі та визначення вимог до програмного продукту _____

2) Дослідження та вибір архітектури згорткових нейронних мереж для розпізнавання відходів _____

3) Проектування та розробка клієнт-серверної архітектури застосунку _____

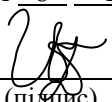
4) Експериментальна оцінка точності класифікації та оптимізація швидкодії системи _____

5) Фінальна верифікація функціональності та аналіз отриманих результатів _____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / термін виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	08.06.2026	виконано
2	Аналіз існуючих підходів до сортування відходів	10.06.2026	виконано
3	Порівняльний аналіз аналогічних мобільних рішень	12.06.2026	виконано
4	Формування технічних вимог та постановка задач	13.06.2026	виконано
5	Вибір алгоритмів Computer Vision для класифікації	15.06.2026	виконано
6	Підготовка датасету та попередня обробка зображень	17.06.2026	виконано
7	Експериментальне навчання нейронної мережі	20.06.2026	виконано
8	Вибір стеку технологій (Flutter/FastAPI)	22.06.2026	виконано
9	Проектування БД та API ендпоінтів	24.06.2026	виконано
10	Реалізація серверної логіки та моделі	28.06.2026	виконано
11	Розробка мобільного інтерфейсу та UI/UX	30.06.2026	виконано
12	Тестування та відлагодження системи	04.07.2026	виконано
13	Оформлення пояснювальної записки	07.07.2026	виконано
14	Захист перед ЕК	10.07.2026	

Дата видачі завдання 8 червня 20 26 р.

Здобувач 
(підпис)

Керівник роботи _____
(підпис)

ас. Максим ПОЛІТ
(посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 100 с., 58 рис., 3 дод., 26 джерел.

ЕКОЛОГІЯ, ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ, КЛАСИФІКАЦІЯ
ЗОБРАЖЕНЬ, КОМП'ЮТЕРНИЙ ЗІР, МОБІЛЬНИЙ ЗАСТОСУНОК,
СОРТУВАННЯ ВІДХОДІВ, ТРАНСФЕРНЕ НАВЧАННЯ.

Об'єкт дослідження – процес автоматизованої класифікації побутових відходів та інформаційної підтримки користувачів щодо правил їх екологічного сортування.

Предмет дослідження – алгоритми комп'ютерного зору для розпізнавання типів сміття та архітектура мобільних застосунків.

Мета роботи – розробити концепцію та реалізувати мобільний застосунок, який за допомогою технологій комп'ютерного зору забезпечить автоматичне розпізнавання побутових відходів за зображенням і надаватиме користувачам покрокові інструкції для їх правильного сортування.

Методи дослідження – системний аналіз предметної галузі та існуючих програмних рішень, методи машинного навчання (зокрема, використання згорткових нейронних мереж та алгоритмів трансферного навчання) для обробки та класифікації зображень.

У роботі проведено аналіз предметної галузі екологічного сортування відходів та існуючих програмних рішень у цій сфері. Досліджено сучасні підходи до класифікації зображень на основі згорткових нейронних мереж, виконано порівняльний аналіз декількох моделей машинного навчання та обґрунтовано вибір найефективнішої архітектури. Розроблено клієнт-серверну систему, що включає модель комп'ютерного зору, серверну частину та мобільний застосунок для автоматизованого розпізнавання побутових відходів.

ABSTRACT

Bachelor's thesis contains: 100 pp., 58 fig., 3 ann., 26 references.

COMPUTER VISION, CONVOLUTIONAL NEURAL NETWORKS, ECOLOGY, IMAGE CLASSIFICATION, MOBILE APPLICATION, TRANSFER LEARNING, WASTE SORTING.

The object of research is the process of automated classification of municipal solid waste and user information support regarding environmental sorting rules.

The subject of research is computer vision algorithms for waste recognition and mobile application architecture.

The goal of the work is to develop a concept and implement a mobile application that uses computer vision technologies to provide automatic recognition of domestic waste from images and provide users with step-by-step instructions for proper sorting.

Research methods – system analysis of the subject area and existing software solutions, machine learning methods (specifically, convolutional neural networks and transfer learning algorithms) for image processing and classification.

The thesis analyzes the subject area of environmental waste sorting and existing software solutions in this field. Modern approaches to image classification based on convolutional neural networks are investigated, a comparative analysis of several machine learning models is performed, and the most effective architecture is selected and justified. A client-server system has been developed, comprising a computer vision model, a backend component, and a mobile application for the automated recognition of household waste.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	8
Вступ.....	9
1 Аналіз предметної галузі та постановка задачі.....	11
1.1 Аналіз предметної галузі екологічного сортування відходів	11
1.2 Огляд існуючих програмних рішень та аналогів.....	16
1.3 Проблеми та особливості автоматизованої класифікації відходів	22
1.4 Постановка задачі дослідження.....	25
2 Теоретичні основи машинного навчання для класифікації відходів.....	28
2.1 Огляд методів машинного навчання у задачах комп'ютерного зору.	28
2.2 Аналіз архітектур згорткових нейронних мереж для мобільних платформ	29
2.3 Проблематика обробки аномалій та нецільових зображень	32
2.4 Метрики оцінювання ефективності інтелектуальних моделей	34
3 Проєктування та розробка програмної системи	36
3.1 Обґрунтування вибору та попереднє оброблення набору даних	36
3.2 Експериментальне дослідження та порівняльний аналіз моделей класифікації	38
3.3 Архітектурне проєктування системи	43
3.4 Обґрунтування вибору стеку технологій.....	45
3.5 Проєктування бази даних	47
3.6 Розробка серверної частини та інтеграція моделі машинного навчання	50
3.7 Розробка кросплатформного мобільного застосунку.....	53
3.8 Контейнеризація та розгортання системи	56
4 Демонстрація та аналіз роботи системи	59
4.1 Інтерфейс користувача та навігація	59
4.2 Еко-профіль та візуалізація статистики	62
4.3 Механізм обробки аномалій та зворотний зв'язок.....	64

4.4 Оцінка ефективності розпізнавання у реальних умовах	69
4.4.1 Тестування за умов достатнього природного освітлення	69
4.4.2 Тестування за умов штучного освітлення (складні умови)	77
4.4.3 Тестування механізму відхилення аномалій (Out-of-Distribution)	81
Висновки	84
Перелік джерел посилання	86
Додаток А Основний код підготовки даних та машинного навчання	90
Додаток Б Основний код клієнтського мобільного застосунку	95
Додаток В Відомість кваліфікаційної роботи	100

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

БД – база даних;

ІД – ідентифікатор;

ІІ – інтерфейс програмування додатків;

МН – машинне навчання;

ШІ – штучний інтелект;

AI – Artificial Intelligence – штучний інтелект;

API – Application Programming Interface – інтерфейс програмування додатків;

CNN – Convolutional Neural Network – згорткова нейронна мережа;

CSS – Cascading Style Sheets – каскадні таблиці стилів;

HTML – HyperText Markup Language – мова розмітки гіпертексту;

HTTP – HyperText Transfer Protocol – протокол передачі гіпертексту;

IP – Internet Protocol – протокол міжмережевої взаємодії;

JSON – JavaScript Object Notation – текстовий формат обміну даними;

REST – Representational State Transfer – архітектурний стиль взаємодії;

SQL – Structured Query Language – мова структурованих запитів.

ВСТУП

Проблема накопичення твердих побутових відходів (ТПВ) є одним із найгостріших екологічних викликів сучасності. Із постійним зростанням чисельності населення та розвитком культури споживання, обсяги генерації сміття невпинно збільшуються, що призводить до переповнення полігонів, забруднення ґрунтів, водойм та атмосфери [1]. Ефективним шляхом вирішення цієї проблеми є перехід до моделі циркулярної економіки, ключовим етапом якої є переробка вторинної сировини. Проте успішна переробка неможлива без якісного сортування відходів безпосередньо у місцях їх утворення – домогосподарствах [2].

Незважаючи на глобальні ініціативи та впровадження систем роздільного збору сміття, рівень залученості громадян до екологічного сортування залишається недостатнім. Головною перешкодою для багатьох людей є складність та заплутаність правил сортування: необхідність розрізняти різні види пластику, відокремлювати кришечки від пляшок, очищати тару від залишків їжі тощо [3]. Традиційні методи інформування, такі як текстові інструкції або статичні довідники, є малоефективними, оскільки вимагають від користувача значного часу на пошук необхідної інформації. Помилки при сортуванні (людський фактор) призводять до забруднення цілих партій вторинної сировини, що робить її непридатною для подальшої переробки [4].

На даний момент на ринку бракує зручних та доступних інформаційних інструментів, які б автоматизували процес визначення типу відходів. Більшість існуючих рішень обмежується функціоналом статичних баз даних та вимагає ручного пошуку об'єкта. У зв'язку з цим гостро постає потреба в зміні підходів до екологічної освіти та підтримки користувачів під час сортування сміття.

Зважаючи на стрімкий розвиток технологій штучного інтелекту, зокрема методів комп'ютерного зору (Computer Vision), особливо

актуальним є створення інтелектуальної системи, яка здатна в режимі реального часу ідентифікувати об'єкт за допомогою камери смартфона. Розробка мобільного застосунку, який класифікує відходи (наприклад, на 12 різних категорій) та миттєво надає конкретну інструкцію щодо їх підготовки та утилізації, значно знизить поріг входження для користувачів та зменшить кількість помилок при сортуванні [5]. Крім того, така система повинна вміти розпізнавати аномалії (наприклад, коли в кадрі немає сміття), забезпечуючи високу надійність роботи.

У рамках даної роботи досліджується предметна галузь, пов'язана з управлінням побутовими відходами, визначаються основні проблеми сортування сміття населенням. Особлива увага приділяється теоретичним основам методів глибокого навчання, зокрема застосуванню попередньо навчених згорткових нейронних мереж, які здатні з високою точністю класифікувати зображення.

Отже, дана робота має значний потенціал для практичного застосування. Розроблений мобільний застосунок може стати щоденним помічником для громадян, сприяючи формуванню екологічних звичок. Автоматизація процесу ідентифікації сміття дозволить не тільки покращити якість вторинної сировини, але й зробити вагомий внесок у збереження навколишнього середовища та екосистеми загалом.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз предметної галузі екологічного сортування відходів

За останні десятиліття проблема накопичення твердих побутових відходів (ТПВ) набула масштабів глобальної екологічної кризи. Згідно з фундаментальним звітом Світового банку «What a Waste 2.0» [1], щорічно у світі генерується понад 2,01 мільярда тонн муніципальних відходів, і, за прогнозами аналітиків, до 2050 року ця цифра може стрімко зрости до 3,4 мільярда тонн (рисунок 1.1). При цьому глобальний рівень переробки залишається критично низьким. В Україні ситуація також потребує негайного втручання: за даними Міністерства захисту довкілля та природних ресурсів [6], переважна більшість утворених побутових відходів (понад 90%) все ще потрапляє на полігони та несанкціоновані звалища, замість того, щоб використовуватися як цінна вторинна сировина.

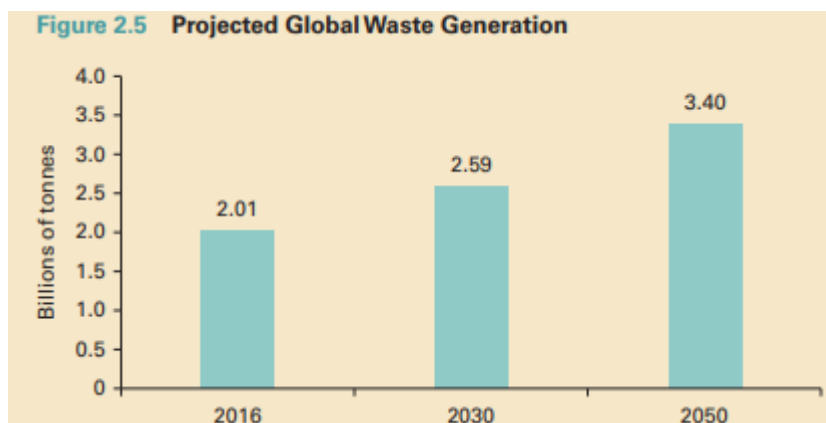


Рисунок 1.1 – Діаграма прогнозованого зростання обсягів генерації ТПВ у світі до 2050 року [1]

Окрім значних економічних втрат через невикористаний ресурсний потенціал, неконтрольоване накопичення сміття завдає непоправної шкоди екосистемам. Розпад пластику на мікропластик, виділення токсичних

фільтратів у ґрунтові та поверхневі води, а також емісія парникових газів (зокрема метану) з переповнених полігонів становлять пряму загрозу для біорізноманіття та кліматичної стабільності [7].

Для ефективного вирішення цієї проблеми необхідне розуміння морфологічного складу твердих побутових відходів. ТПВ є складною гетерогенною сумішшю, яку базово поділяють на кілька основних категорій (рисунок 1.2): органічні відходи (залишки їжі), папір та картон, різноманітні полімери (пластик), скло, метали, а також небезпечні відходи (батареї, електроніка) [8].

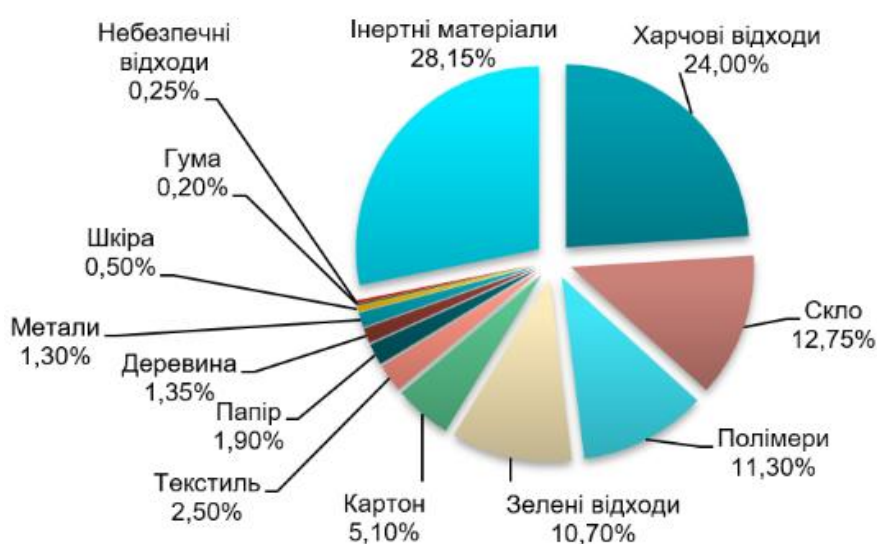


Рисунок 1.2 – Типова морфологічна структура твердих побутових відходів в Україні

Критичним етапом у ланцюжку циркулярної економіки є первинне сортування відходів (source separation) – розділення сміття безпосередньо у місці його утворення, тобто в домогосподарствах [3]. Важливість саме первинного сортування зумовлена тим, що змішування різних фракцій в одному контейнері призводить до незворотного псування вторинної сировини. Наприклад, органічні відходи (рідини, залишки їжі), потрапляючи на макулатуру, руйнують целюлозні волокна, роблячи папір

непридатним для переробки. Хоча існують сучасні автоматизовані сміттесортувальні лінії (MRF), їхня ефективність різко падає, а знос обладнання зростає, якщо на вхід подається сильно забруднене, змішане сміття [9].

Особливу складність у процесі сортування становить група полімерних матеріалів (пластик). З точки зору пересічного споживача пластик сприймається як єдиний матеріал, проте хімічно це велика група різних полімерів, які потребують абсолютно різних технологічних умов утилізації.

Згідно з міжнародною системою маркування (Resin Identification Code), пластикові вироби суворо поділяються на 7 основних категорій (рисунок 1.3) [10]:

- поліетилентерефталат (PET або PETE, маркування «1»): найпоширеніший пластик для виготовлення прозорих пляшок для води, соків та одноразових контейнерів для ягід. Він має високий потенціал до переробки, але дуже чутливий до забруднень іншими полімерами;

- поліетилен високої щільності (HDPE, маркування «2»): міцний, жорсткий та стійкий до хімікатів матеріал. З нього виготовляють тару для побутової хімії (пляшки для шампунів, миючих засобів), каністри, щільні пакети та кришечки для пляшок;

- полівінілхлорид (PVC або V, маркування «3»): використовується для виготовлення пластикових труб, віконних профілів, блістерного пакування для таблеток та деяких видів термоусадкової плівки. Це один із найнебезпечніших для екології видів пластику, який вкрай важко переробляти через високий вміст хлору та токсичних добавок [11].

Четверта та п'ята категорії також є дуже поширеними у побуті, проте вимагають інших підходів до збору та обробки [10]:

- поліетилен низької щільності (LDPE, маркування «4»): еластичний та м'який матеріал, що широко використовується для пакетів-майок, смітєвих пакетів, стрейч-плівок та м'яких тубиків. На автоматизованих

сортувальних лініях він часто намотується на вали механізмів, тому потребує специфічного роздільного збору;

– поліпропілен (PP, маркування «5»): термостійкий пластик, який витримує нагрівання. З нього роблять багаторазовий посуд, стаканчики для сметани та йогуртів, контейнери для доставки їжі, а також медичні шприци.

Останні дві категорії є найбільш проблемними для муніципального рециклінгу [10]:

– полістирол (PS, маркування «6»): буває звичайним (одноразові виделки, ложки, кришечки для кави) та спіненим (пінополістирольні лотки для м'яса, контейнери для яєць). Він є економічно не вигідним для переробки через великий об'єм при дуже малій вазі, а при нагріванні може виділяти токсичний стирол.

– інші види пластику (OTHER або «7»): збірна категорія, куди входять суміші різних полімерів, багатошарове пакування (композити), полікарбонат, а також інноваційні біопластики (наприклад, PLA). Такий пластик у більшості випадків не підлягає класичній механічній переробці і відправляється на високотемпературне спалювання або захоронення.



Рисунок 1.3 – Міжнародна система маркування пластику

Усі ці матеріали мають кардинально різні температури плавлення, в'язкість та хімічну структуру. Змішування навіть невеликої кількості одного типу пластику з іншим під час рециклінгу має катастрофічні технологічні наслідки. Наприклад, температура плавлення PET становить близько 260°C, тоді як PVC починає плавитися і руйнуватися вже при 160°C. Потрапляння навіть крихітної частки PVC до партії PET призводить до

утворення соляної кислоти при нагріванні. Це спричиняє хімічну деградацію всієї партії вторсировини, виділення токсичних газів, пошкодження заводського обладнання та значне погіршення міцності й прозорості вторинного продукту [11].

Ситуація значно ускладнюється тим, що візуально розрізнити ці матеріали в побутових умовах вкрай складно. Прозора пляшка з PET-пластику може виглядати ідентично прозорій пляшці з PVC. Крім того, сучасне пакування часто є багатошаровим композитом (наприклад, упаковка Tetra Pak містить шари картону, поліетилену та алюмінієвої фольги), що інтуїтивно збиває споживача з пантелику [12]. Екологічне маркування на товарах, яке мало б слугувати орієнтиром, часто наноситься занадто дрібним шрифтом, витискається на прозорому фоні, ховається під етикетками або взагалі відсутнє.

Через візуальну подібність матеріалів та брак зручної інформації виникає масове явище «wishcycling» (сліпого або бажаного сортування). Користувачі, керуючись добрими намірами, викидають до контейнерів для вторинної сировини предмети, які не підлягають переробці або вимагають попереднього очищення (наприклад, немиті лотки з-під їжі або стаканчики з-під кави, які мають внутрішній шар пластику). Це призводить до перехресного забруднення і відбраковування цілих партій потенційно цінної сировини на сортувальних станціях [13]. Традиційні методи екологічної просвіти – інформаційні буклети чи статичні вебсайти – виявилися недостатньо ефективними, оскільки вони вимагають від людини значних витрат часу на пошук відповіді безпосередньо в момент викидання сміття.

Таким чином, стрімке зростання обсягів побутових відходів, хімічна різноманітність матеріалів (зокрема пластиків), складність їх візуальної ідентифікації та критичний вплив людського фактора (помилки) на якість вторинної сировини зумовлюють гостру потребу в нових підходах. Стає очевидним, що для підвищення ефективності первинного сортування

необхідні інтелектуальні інформаційні системи. Застосування технологій комп'ютерного зору здатне подолати проблему візуальної ідентифікації матеріалів, автоматизувавши цей процес і надавши користувачу миттєву, локалізовану інструкцію з утилізації конкретного об'єкта.

1.2 Огляд існуючих програмних рішень та аналогів

Зі стрімким поширенням концепцій екологічної свідомості та розвитку ініціатив «розумного міста» (Smart City) у світі з'явилася потреба в інструментах, які б допомагали громадянам правильно поводитися з побутовими відходами. Враховуючи, що правила сортування можуть суттєво відрізнятися залежно від регіону, а кількість видів пакування постійно зростає, виникла нагальна необхідність в автоматизації процесу ідентифікації сміття. На сьогодні існує низка програмних продуктів та платформ, що тією чи іншою мірою вирішують завдання інформаційної підтримки процесу сортування. Далі наведено детальний огляд найпопулярніших рішень.

Одним із найвідоміших рішень на українському ринку є мобільний застосунок «Сортуй» [14], розроблений вітчизняними фахівцями для адаптації загальних екологічних правил до реалій українських сміттєпереробних станцій.

Цей застосунок працює як велика інтерактивна база даних. Користувач може вручну знайти потрібний тип відходу (наприклад, «тетрапак» або «ПЕТ-пляшка») у каталозі та отримати інструкцію щодо того, як його підготувати до переробки і до якого контейнера викинути. Система також містить вбудовану карту пунктів прийому вторинної сировини з графіками їхньої роботи.

На рисунку 1.4 наведено вигляд інтерфейсу застосунку «Сортуй».

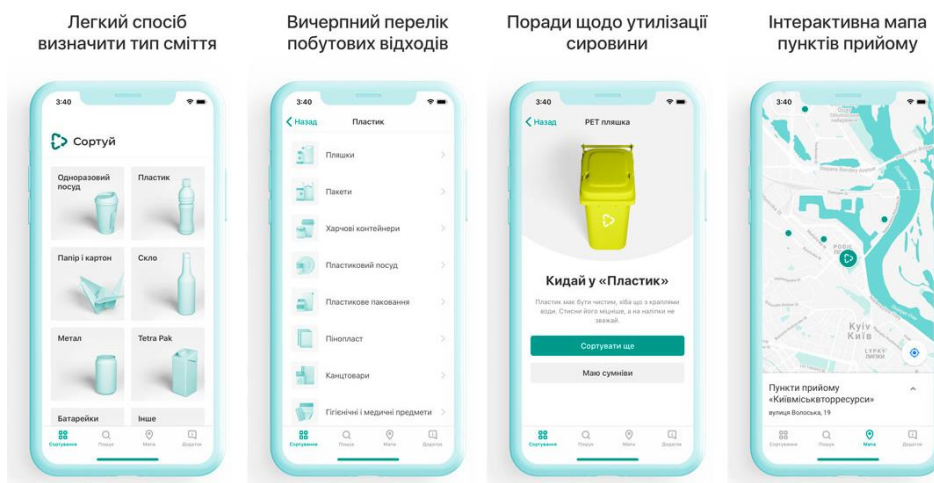


Рисунок 1.4 – Інтерфейс мобільного застосунку «Сортуй» з каталогом відходів

До переваг описаної системи варто віднести:

- високу якість локалізованих даних, адаптованих під українські реалії;
- наявність детальних текстових інструкцій щодо підготовки тари;
- можливість роботи застосунку в автономному режимі (офлайн).

До недоліків такої системи можна віднести:

- повну відсутність автоматизованого розпізнавання: користувач повинен самостійно знати назву матеріалу або шукати його вручну;
- значні витрати часу на пошук інформації, що знижує рівень залученості користувачів;
- нездатність системи допомогти у випадках, коли маркування на об'єкті відсутнє або незрозуміле.

Іншим глобальним підходом до проблеми управління відходами є краудсорсингова платформа «Litterati» [15]. Вона була створена з метою глобального картографування засміченості планети. Користувачі фотографують знайдене на вулиці сміття, після чого вбудовані алгоритми машинного навчання (ШІ) аналізують зображення та автоматично додають відповідні теги (наприклад, «пластик», «бренд: Coca-Cola»). Усі дані

прив'язуються до геолокації та формують глобальну теплову карту забруднень, яка допомагає муніципалітетам та екологічним організаціям планувати прибирання. На рисунках 1.5–1.7 представлено вигляд карти та процесу тегування в застосунку Litterati.

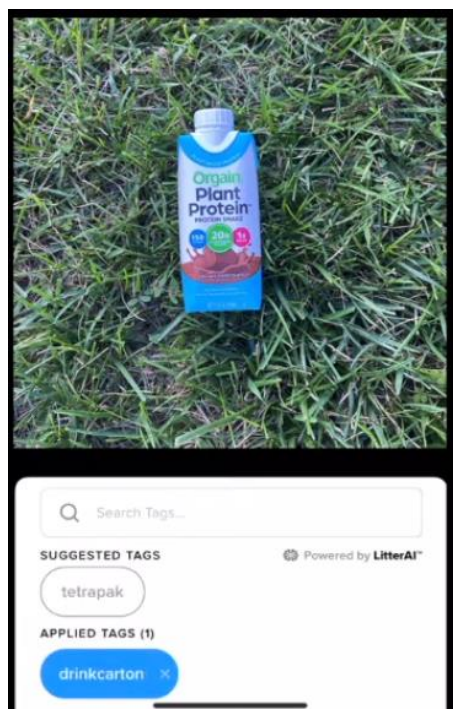


Рисунок 1.5 – Інтерфейс системи тегування платформи «Litterati»

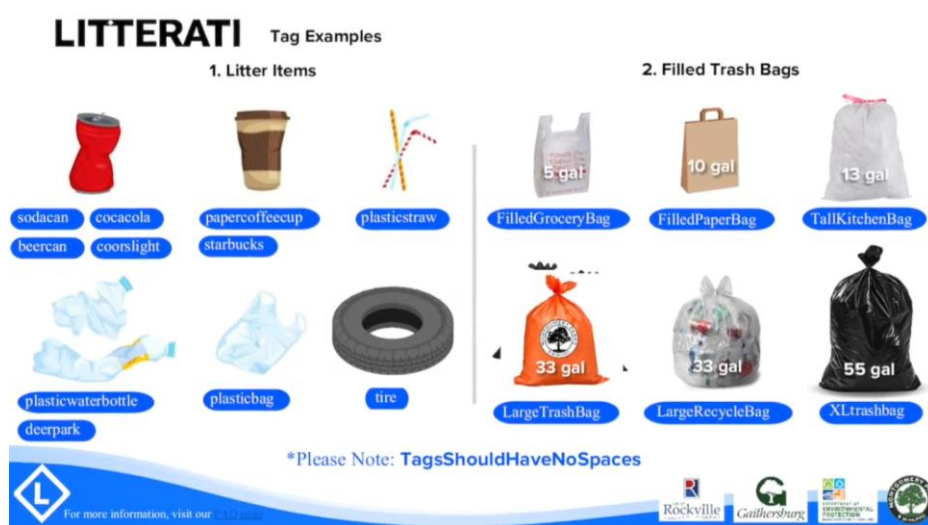


Рисунок 1.6 – Типові теги платформи «Litterati»

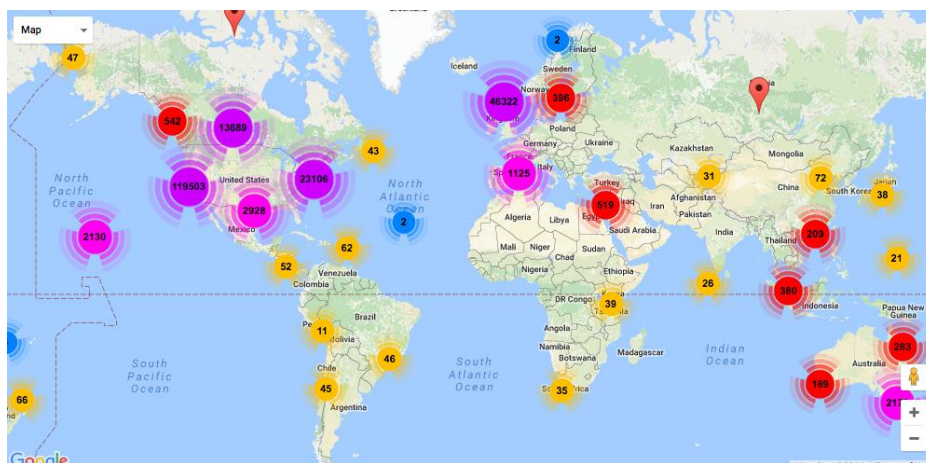


Рисунок 1.7 – Карта забруднень платформи «Litterati»

Система має наступні переваги:

- використання методів комп'ютерного зору для розпізнавання об'єктів;
- потужна гейміфікація та формування глобального екологічного ком'юніті;
- створення великих наборів даних (датасетів) для подальших досліджень.

До недоліків системи, що розглядається, належать:

- орієнтація платформи на збір та аналіз вуличного сміття, а не на допомогу домогосподарствам у сортуванні;
- відсутність алгоритмів зворотного зв'язку у вигляді покрокових інструкцій щодо переробки;
- затримки у розпізнаванні, оскільки обробка часто орієнтована на формування аналітичних звітів, а не на миттєву допомогу (Real-time inference).

Окрему категорію становлять універсальні інструменти візуального пошуку, такі як Google Lens або Apple Visual Look Up [16]. Завдяки використанню надпотужних згорткових нейронних мереж та доступу до гігантських хмарних баз даних, ці застосунки здатні з високою точністю ідентифікувати практично будь-який об'єкт у кадрі. Якщо навести камеру на

бляшанку з-під напою, система швидко визначить бренд, тип продукту та запропонує посилання на інтернет-магазини. На рисунку 1.8 наведено приклад роботи технології візуального пошуку.

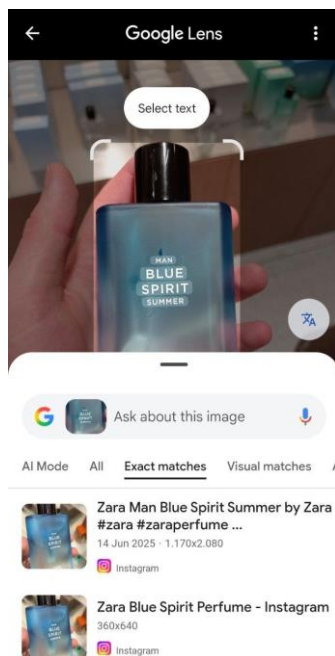


Рисунок 1.8 – Приклад розпізнавання об'єкта за допомогою Google Lens

До переваг універсальних систем можна віднести:

- надзвичайно високу точність розпізнавання та швидкість роботи;
- здатність ідентифікувати складні об'єкти в умовах поганого освітлення;
- доступність технології на більшості сучасних смартфонів «з коробки».

Проте для цілей сортування такі системи мають критичні недоліки:

- універсальні моделі розпізнають сутність об'єкта (наприклад, «пляшка з водою»), але не класифікують його за екологічним критерієм (наприклад, «ПЕТ-пластик, що потребує миття»);
- повна відсутність спеціалізованих інструкцій з утилізації;
- нездатність розрізняти придатні та непридатні до переробки матеріали зі схожим зовнішнім виглядом.

Також варто згадати численні академічні прототипи застосунків для класифікації сміття, які розробляються в рамках наукових досліджень [17]. Більшість із них базується на легких архітектурах, таких як MobileNet або ResNet. Зазвичай такі рішення демонструють високу точність розпізнавання, проте розробляються лише як Proof-of-Concept (доказ концепції). На рисунку 1.9 наведено типовий інтерфейс одного з таких дослідницьких прототипів.



Рисунок 1.9 – Інтерфейс MWaste застосунку для класифікації сміття

Перевагами таких прототипів є використання сучасних підходів трансферного навчання та висока швидкість обробки зображень. При цьому істотним недоліком є їхня обмеженість: як правило, вони класифікують сміття лише на 4–6 базових класів (скло, метал, папір, пластик), не розділяючи пластик на підвиди, що є критичним для переробки. Крім того, більшість таких рішень не має механізму обробки аномалій (клас «Not Trash»), що призводить до хибних спрацьовувань при скануванні об'єктів, які не є відходами.

Отже, аналіз існуючих рішень показує відсутність на ринку комплексного, спеціалізованого застосунку, який би поєднував

автоматизоване розпізнавання широкого спектра відходів (від 10 класів і більше) за допомогою комп'ютерного зору із миттєвим наданням чітких, локалізованих інструкцій із сортування. Це підтверджує доцільність розробки нової клієнт-серверної системи, яка б нівелювала виявлені недоліки аналогів.

1.3 Проблеми та особливості автоматизованої класифікації відходів

Під час проектування інтелектуальних систем для розпізнавання побутових відходів та надання рекомендацій щодо їх сортування інженери стикаються з комплексом специфічних технологічних та інформаційних перешкод.

Фундаментальною проблемою залишається дефіцит високоякісних, репрезентативних масивів даних для тренування моделей комп'ютерного зору. Наразі у відкритому доступі існує кілька відомих наборів, таких як TrashNet або Garbage Classification dataset [18]. Останній, наприклад, містить тисячі зображень, попередньо розподілених на 12 основних класів (картон, скло, метал, різні типи пластику тощо). На рисунках 1.10–1.12 наведено приклади фотографій об'єктів з цього набору даних.

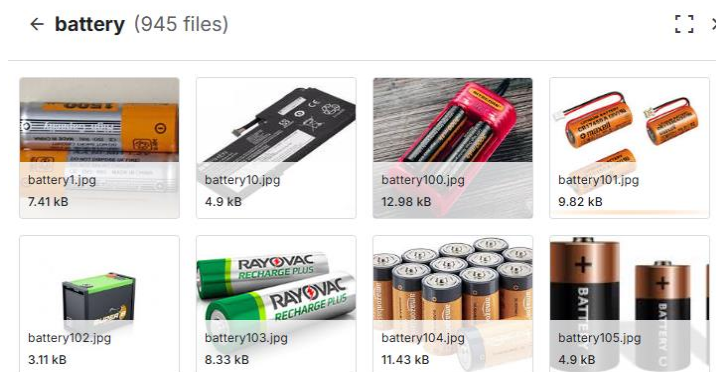


Рисунок 1.10 – Приклади зображень відходів класу «battery» з набору даних Garbage Classification (Kaggle)

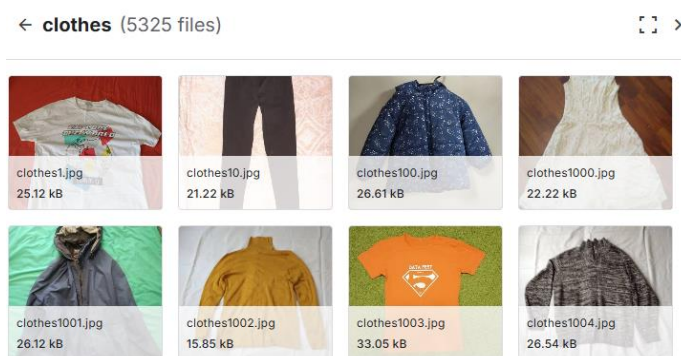


Рисунок 1.11 – Приклади зображень відходів класу «clothes» з набору даних Garbage Classification (Kaggle)

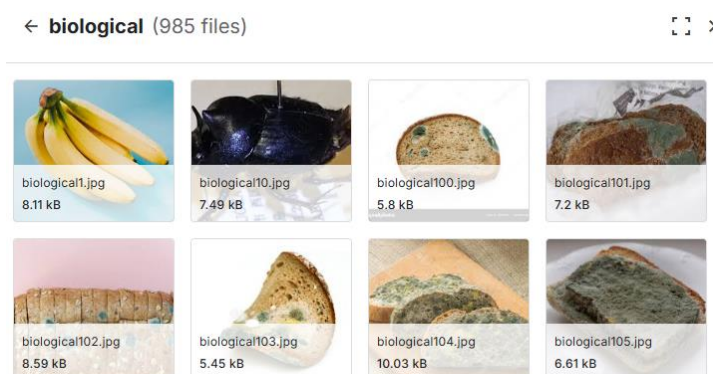


Рисунок 1.12 – Приклади зображень відходів класу «biological» з набору даних Garbage Classification (Kaggle)

Проте нейронні мережі, натреновані виключно на таких еталонних вибірках, часто демонструють різке падіння точності під час практичної експлуатації. Це зумовлено тим, що більшість існуючих датасетів формувалися в штучних «лабораторних» умовах: чисті предмети сфотографовані на ідеальному однотонному тлі [19]. Натомість у реальному житті користувач застосунку фотографує сміття в абсолютно неконтрольованому середовищі, що унеможливорює використання базових моделей без їх суттєвого донавчання.

Суттєвою перепорою для коректної класифікації є вплив фонового шуму та освітлення. Домашнє або вуличне оточення, наявність у кадрі

сторонніх предметів, рук людини або текстурних поверхонь (наприклад, стільниці чи килима) можуть відвертати увагу алгоритму від ключових ознак об'єкта. Крім того, відблиски від спалаху або штучного освітлення, особливо на глянцевиx чи прозорих поверхнях, значно спотворюють візуальні характеристики сміття [20].

Інший критичний виклик полягає у надзвичайній фізичній деформації об'єктів. На відміну від розпізнавання облич чи марок автомобілів, сміття практично ніколи не зберігає своєї початкової форми. Бляшанки зминаються, папір розривається, а пластикові пляшки скручуються. Через це виникає проблема високої внутрішньокласової варіативності: системі вкрай важко зрозуміти, що ідеально рівна пляшка та зім'ятий шматок пластику належать до однієї категорії [21].

Не менш складною є задача розрізнення матеріалів зі схожими візуальними характеристиками (проблема низької міжкласової відстані). Наприклад, прозорий PET-пластик та звичайне прозоре скло на цифровому зображенні виглядають майже ідентично. Для правильної класифікації таких матеріалів алгоритму необхідно аналізувати специфічні мікроознаки (характер заломлення світла, товщину країв), що потребує високої якості вхідного зображення.

У контексті розробки мобільних клієнт-серверних застосунків додатково виникає проблема балансу між обчислювальною складністю та швидкістю роботи [22]. Оскільки обробка зображень на сервері має займати лічені мілісекунди для забезпечення комфортного користувацького досвіду (User Experience), розробникам доводиться використовувати легкі згорткові архітектури. Це вимагає проведення тонкого налаштування для збереження високої точності при обмеженій кількості параметрів мережі.

Окремим, але вкрай важливим викликом є обробка аномалій – так званий клас «Not Trash» (не сміття). Якщо система не здатна розпізнавати випадки, коли в кадрі немає відходів, вона буде намагатися класифікувати будь-який відсканований предмет (наприклад, клавіатуру чи домашню

тварину) як один із видів сміття [23]. Відсутність надійного порогу впевненості для відсіювання таких хибних спрацьовувань призводить до надання абсурдних інструкцій із сортування та швидкої втрати довіри користувача до програмного продукту.

Отже, проведений аналіз предметної галузі дозволив визначити ключові особливості процесу екологічного сортування побутових відходів, а також основні фактори, що ускладнюють його ефективну реалізацію на побутовому рівні. Розглянуті програмні рішення та дослідницькі прототипи демонструють перспективність використання методів комп'ютерного зору для автоматизації процесу класифікації відходів, проте водночас підтверджують наявність низки технологічних і практичних обмежень, які необхідно враховувати під час розробки інтелектуальних систем даного типу.

1.4 Постановка задачі дослідження

Здійснений аналіз предметної галузі та огляд існуючих програмних рішень дозволяють стверджувати, що проблема створення ефективних та зручних інструментів для підтримки екологічного сортування відходів залишається гострою, незважаючи на появу різноманітних еко-ініціатив. Актуальність подальших розробок у цьому напрямі зумовлена кількома ключовими чинниками.

По-перше, глобальні обсяги генерації твердих побутових відходів продовжують зростати критичними темпами. Згідно зі звітами міжнародних екологічних організацій, динаміка накопичення муніципального сміття демонструє стійкий висхідний тренд, при цьому рівень його переробки залишається непропорційно низьким [1]. За таких умов традиційні підходи до екологічної просвіти, побудовані на поширенні друкованих матеріалів чи статичних веб-довідників, дедалі менше здатні кардинально змінити

ситуацію, оскільки вони не забезпечують миттєвої підтримки користувача безпосередньо в момент викидання сміття.

По-друге, сучасні користувачі звикли до швидкого, візуального та інтуїтивно зрозумілого отримання інформації. Необхідність самостійно шукати маркування на упаковці, а потім звіряти його з текстовими каталогами у додатках викликає у людей відторгнення та небажання витратити свій час. Користувачі очікують від технологій підходу «наведи камеру і отримай відповідь». Для задоволення цієї потреби необхідне використання методів комп'ютерного зору – здатності системи самостійно ідентифікувати візуальні ознаки матеріалу (текстуру, форму, характерні деформації) та класифікувати об'єкт без ручного введення даних [24].

По-третє, існуючі на ринку інформаційні системи здебільшого є фрагментованими. Популярні екологічні застосунки пропонують гарні бази знань та карти пунктів прийому, але не вміють автоматично розпізнавати об'єкти. З іншого боку, універсальні інструменти візуального пошуку (наприклад, Google Lens) чудово розпізнають предмети, але не надають жодних інструкцій щодо їх правильної утилізації [25]. Відсутність спеціалізованих систем, які б поєднували потужний модуль розпізнавання зображень із локалізованою екологічною експертизою, є суттєвою прогалиною на стику IT та екології.

По-четверте, академічні дослідження останніх років демонструють високий потенціал згорткових нейронних мереж та алгоритмів трансферного навчання для класифікації відходів. Проте перехід від «лабораторних» прототипів до повноцінних програмних продуктів залишається слабким місцем [26]. Більшість існуючих AI-моделей натреновані на ідеальних датасетах і не здатні адекватно працювати в реальних умовах мобільної зйомки, а також не містять механізмів обробки аномалій (відсіювання об'єктів, які не є сміттям). Розрив між теоретичними метриками точності моделей та їхньою практичною інтеграцією в зручний клієнт-серверний інтерфейс залишається значним.

Таким чином, сукупність наведених факторів визначає актуальність створення інтелектуальної клієнт-серверної системи класифікації відходів, орієнтованої на високу швидкість обробки зображень, стійкість до візуального шуму та надання персоналізованих екологічних інструкцій.

У зв'язку з цим, метою дослідження є розробка та реалізація мобільного застосунку для автоматизованої класифікації побутових відходів на основі методів комп'ютерного зору з функцією надання користувачам миттєвих інструкцій щодо правильного сортування.

Для досягнення поставленої мети необхідно виконати такі завдання:

- проаналізувати сучасні підходи до управління твердими побутовими відходами та існуючі програмні рішення у цій сфері;
- дослідити методи комп'ютерного зору, архітектури згорткових нейронних мереж та алгоритми трансферного навчання для задачі класифікації зображень;
- проаналізувати підходи до обробки аномалій (Out-of-Distribution) для забезпечення надійності розпізнавання в неконтрольованих умовах;
- сформулювати функціональні та нефункціональні вимоги до інтелектуального мобільного застосунку;
- розробити клієнт-серверну архітектуру програмного комплексу;
- підготувати набір даних (датасет) та реалізувати модуль штучного інтелекту для класифікації відходів на вибрану кількість категорій;
- реалізувати серверну частину системи для швидкої обробки запитів та маршрутизації результатів розпізнавання;
- розробити кросплатформний мобільний клієнт із інтеграцією камери та зручним інтерфейсом користувача (UI/UX);
- провести експериментальне тестування роботи ШІ-моделі на реальних зображеннях та оцінити загальну продуктивність і точність розробленої системи.

2 ТЕОРЕТИЧНІ ОСНОВИ МАШИННОГО НАВЧАННЯ ДЛЯ КЛАСИФІКАЦІЇ ВІДХОДІВ

2.1 Огляд методів машинного навчання у задачах комп'ютерного зору

Швидкий розвиток технологій штучного інтелекту фундаментально змінив підходи до автоматизованої обробки візуальної інформації. У контексті вирішення екологічних проблем, зокрема класифікації твердих побутових відходів, традиційні алгоритми комп'ютерного зору виявилися неефективними. Це пов'язано з тим, що об'єкти сміття зазвичай мають високу варіативність форм, зазнають значних деформацій, можуть бути забрудненими або частково перекритими іншими предметами. Тому сучасні підходи до розв'язання таких нелінійних задач базуються на використанні методів глибинного навчання (Deep Learning), які здатні самостійно виявляти та формувати ієрархію візуальних ознак без необхідності їх ручного програмування інженером [24].

Основним та найбільш ефективним інструментом для обробки зображень на сьогодні є згорткові нейронні мережі (Convolutional Neural Networks, CNN). На відміну від класичних повнозв'язних нейронних мереж, де кожен нейрон пов'язаний з усіма нейронами попереднього шару, згорткові мережі враховують просторову структуру зображення. Процес розпізнавання в таких мережах починається з операції згортки. Суть цієї операції полягає в тому, що спеціальна матриця (ядро або фільтр) послідовно «ковзає» по всьому зображенню, виконуючи математичні обчислення над групами пікселів. На початкових шарах ці фільтри вчать розпізнавати прості базові елементи: горизонтальні та вертикальні лінії, кути, градієнти кольору. На глибших шарах мережі ці прості ознаки комбінуються, дозволяючи системі розпізнавати складні текстури (наприклад, прозорість скла або текстуру зім'ятого картону) та цілісні об'єкти [25].

Після кожного згорткового шару в сучасних архітектурах застосовується нелінійна функція активації, найчастіше ReLU (Rectified Linear Unit). Її головне завдання – відкинути негативні значення, залишаючи лише корисний сигнал. Це дозволяє мережі вирішувати складні задачі та уникати проблеми загасання сигналу під час навчання. Для того, щоб модель не фокусувалася на дрібних, незначущих деталях і не залежала від точного розташування об'єкта в кадрі, використовується операція агрегації (Pooling). Вона зменшує розмірність отриманих карт ознак, вибираючи, наприклад, лише найяскравіші пікселі з певної ділянки. Завдяки цьому досягається властивість трансляційної інваріантності – здатність системи розпізнати пластикову пляшку незалежно від того, чи знаходиться вона в центрі фотографії, чи збоку.

Окремою та надзвичайно важливою концепцією у сучасному комп'ютерному зорі є трансферне навчання (Transfer Learning). Навчання глибокої нейронної мережі «з нуля» вимагає колосальних обчислювальних потужностей та мільйонів розмічених фотографій, що часто є недосяжним для вузькоспеціалізованих задач. Трансферне навчання вирішує цю проблему шляхом використання моделей, які вже були попередньо натреновані на велетенських загальних базах даних (наприклад, ImageNet). Оскільки перші шари таких моделей вже чудово розпізнають універсальні форми та текстури, залишається лише замінити останні класифікаційні шари мережі і «доучити» (Fine-Tuning) їх на спеціалізованому наборі даних зі сміттям [21]. Це багаторазово прискорює процес розробки та дозволяє досягати високої точності навіть на відносно невеликих вибірках.

2.2 Аналіз архітектур згорткових нейронних мереж для мобільних платформ

Проектування інтелектуальних систем, які передбачають використання штучного інтелекту в мобільних застосунках, накладає низку

жорстких технічних обмежень. Сучасні смартфони мають обмежений обсяг оперативної пам'яті, порівняно невисоку обчислювальну потужність процесорів і, що найголовніше, живляться від акумулятора. Використання класичних «важких» архітектур нейронних мереж, таких як VGG-16 або глибокі версії ResNet, є нераціональним для локального мобільного виконання. Хоча ці моделі демонструють еталонну точність, вони містять десятки мільйонів параметрів, а їхній розмір може сягати сотень мегабайтів. Запуск таких моделей на мобільному пристрої призводить до тривалих затримок у розпізнаванні, перегріву пристрою та швидкого розрядження батареї.

Для подолання цього бар'єру дослідницькими спільнотами було створено окремий клас оптимізованих, «легких» архітектур. Головною метою таких розробок є збереження високої точності розпізнавання при радикальному зменшенні кількості параметрів та математичних операцій. Найбільш успішним та широко використовуваним рішенням у цьому напрямку є архітектура MobileNetV2, яка була розроблена спеціально для мобільних та периферійних (Edge) пристроїв [22].

Висока ефективність MobileNetV2 досягається за рахунок інноваційної заміни стандартних згорткових шарів на так звані глибинно-сепарабельні згортки (Depthwise Separable Convolutions). Традиційна згортка виконує дві дії одночасно: вона фільтрує зображення у просторі (шукає форми) та змішує інформацію з різних кольорових каналів (наприклад, RGB). Це вимагає величезної кількості обчислень. Концепція сепарабельної згортки полягає в тому, щоб розділити цей процес на два незалежні кроки. На першому кроці (глибинна згортка) фільтри застосовуються до кожного кольорового каналу окремо, досліджуючи лише просторові форми. На другому кроці (точкова згортка) використовуються мікро-фільтри розміром 1x1 піксель, які просто об'єднують результати попереднього кроку. Такий розділений підхід дозволяє зменшити кількість математичних операцій у 8–9 разів порівняно зі стандартними

архітектурами, практично не втрачаючи при цьому здатності мережі «розуміти» зображення.

Ще одним критично важливим нововведенням MobileNetV2 є використання інвертованих залишкових блоків з лінійними вузькими місцями (Inverted Residuals with Linear Bottlenecks) (рисунок 2.1). В архітектурі ResNet залишкові зв'язки з'єднують «широкі» шари мережі, які споживають багато оперативної пам'яті. У MobileNetV2 застосовано зворотний підхід: інформація стискається у «вузькі» блоки перед передачею на наступний етап, а розширюється лише всередині самого блоку для виконання обчислень. Окрім того, відмова від нелінійних функцій активації у вузьких місцях запобігає втраті важливої інформації під час стиснення.

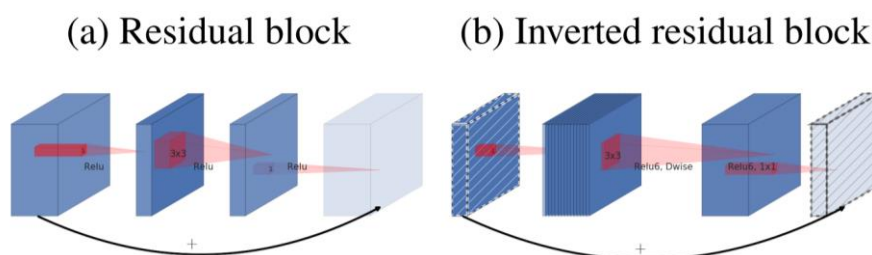


Рисунок 2.1 – Порівняння звичайного залишкового шару та інвертованого залишкового шару

Завдяки поєднанню глибинно-сепарабельних згорток та інвертованих блоків, модель MobileNetV2 забезпечує ідеальний баланс між швидкістю інференсу (розпізнавання), малим розміром файлу моделі та високою точністю, що робить її оптимальним вибором для інтеграції у клієнт-серверну систему класифікації відходів.

Окремо у межах проєкту було розглянуто доцільність використання наступного покоління цієї архітектури – MobileNetV3 (зокрема її конфігурації MobileNetV3-Large). Попри те, що MobileNetV3 пропонує ряд теоретичних переваг завдяки інтеграції блоків стиснення та збудження (Squeeze-and-Excitation) та автоматизованому підбору шарів за

допомогою AutoML, для розроблюваної системи було обрано саме MobileNetV2. Це рішення зумовлене значно вищою стабільністю MobileNetV2 під час оптимізації та конвертації у мобільні кросплатформні формати (такі як TensorFlow Lite), а також її незалежністю від наявності специфічних апаратних нейропроцесорів (DSP/NPU) у смартфонах користувачів. MobileNetV2 демонструє прогнозований час інференсу та стабільну збіжність на етапі навчання, що мінімізує ризики виникнення апаратних помилок серіалізації шарів під час бездротового тестування.

2.3 Проблематика обробки аномалій та нецільових зображень

Перехід від теоретичних розробок та тестування нейронних мереж у лабораторних умовах до їх впровадження у реальні програмні продукти виявляє низку специфічних проблем. Однією з найбільш критичних є проблема обробки аномалій, або даних поза розподілом (Out-of-Distribution, OoD). Набори даних, на яких навчаються моделі класифікації сміття, містять виключно зображення цільових класів: пластику, паперу, скла, металу тощо. Проте під час реальної експлуатації мобільного застосунку користувач може свідомо чи випадково сфотографувати об'єкти, які не мають жодного відношення до відходів – стіл, домашню тварину, руку або пейзаж [23].

Проблема полягає в архітектурних особливостях стандартних класифікаційних моделей. На фінальному етапі нейронна мережа використовує функцію активації Softmax, яка перетворює вихідні сигнали мережі на розподіл ймовірностей. Сума ймовірностей для всіх відомих моделі класів завжди дорівнює 100%. Це означає, що мережа змушена віднести будь-яке вхідне зображення до одного з відомих їй класів сміття. Якщо системі надати зображення, наприклад, комп'ютерної миші, математичний апарат нейронної мережі знайде в ньому певні візуальні збіги з пластиком і може видати результат «Пластик» із високим показником

впевненості. У контексті екологічного застосунку такі хибнопозитивні спрацьовування є неприпустимими, оскільки система надасть користувачу некоректні інструкції з утилізації нецільового предмета.

Для подолання цієї проблеми у системах комп'ютерного зору застосовують декілька підходів. Перший полягає у додаванні до навчального датасету штучного класу «Інше» (Background/Unknown), який наповнюється випадковими зображеннями предметів інтер'єру, природи та людей. Однак зібрати датасет, який би репрезентативно охоплював усе різноманіття об'єктів навколишнього світу, є практично неможливим завданням.

Другий, значно ефективніший та прагматичніший підхід, полягає у запровадженні порогу впевненості (Confidence Thresholding) на етапі постобробки результатів розпізнавання. Хоча функція Softmax і намагається максимізувати ймовірність одного з класів, для об'єктів, які кардинально відрізняються від навчальної вибірки, розподіл ймовірностей часто стає більш «розмитим» між кількома класами. На основі експериментальних досліджень встановлюється мінімально допустимий поріг впевненості моделі.

У межах розроблюваної системи класифікації сміття такий механізм реалізується на рівні серверної логіки. Після того, як нейронна мережа аналізує отримане від мобільного клієнта зображення, система перевіряє максимальне значення ймовірності. Якщо цей показник виявляється нижчим за емпірично встановлений поріг (у нашому випадку 60%), система класифікує вхідні дані як аномалію (Out-of-Distribution). Замість того, щоб вгадувати клас сміття, сервер генерує спеціальну відповідь про неможливість розпізнавання об'єкта. Отримавши таку відповідь, мобільний застосунок виводить попередження користувачеві з проханням змінити ракурс, покращити освітлення або переконатися, що об'єкт справді підлягає сортуванню. Такий підхід значно підвищує надійність системи та довіру користувачів до інформації, яку вона надає.

2.4 Метрики оцінювання ефективності інтелектуальних моделей

Для об'єктивного аналізу та порівняння архітектур згорткових нейронних мереж у межах дослідження виникла необхідність впровадження комплексної системи метрик оцінювання. Використання лише однієї метрики загальної точності (Accuracy), яка відображає загальне відношення правильно розпізнаних об'єктів до всієї кількості спроб, є недостатнім у задачах мультикласової класифікації. За умов наявності дванадцяти різних категорій, деякі з яких мають високу візуальну схожість (наприклад, папір та картон або різні кольори скла), високий показник загальної точності може приховувати критичні помилки моделі на конкретних класах небезпечних чи токсичних відходів [20].

З огляду на це, оцінювання моделей було побудовано на аналізі таких статистичних метрик, як точність (Precision), повнота (Recall) та інтегральний критерій F1-score для кожного окремого класу. Метрика Precision визначає частку об'єктів, які дійсно належать до заданого класу, серед усіх, які модель віднесла до цього класу:

$$Precision = \frac{TP}{TP+FP}, \quad (3.1)$$

де TP (True Positives) – кількість правильно розпізнаних об'єктів цільового класу;

FP (False Positives) – кількість об'єктів інших класів, які модель помилково віднесла до цільового.

У контексті екологічного застосування висока точність означає, що якщо система ідентифікувала предмет як «біологічні відходи», користувач з високою ймовірністю отримає правильну інструкцію з компостування, а не викине пластик в органіку.

Метрика Recall (повнота) відображає здатність системи виявляти максимальну кількість об'єктів заданого класу з усіх існуючих у вибірці:

$$Recall = \frac{TP}{TP+FN}, \quad (3.2)$$

де TP (True Positives) – кількість правильно розпізнаних об'єктів цільового класу;

FN (False Negatives) – кількість об'єктів цільового класу, які модель пропустила або помилково віднесла до інших категорій.

Для нашої системи повнота є критично важливою, наприклад, у категорії «батарейки», оскільки пропуск токсичного відходу та його випадкове потрапляння до загального смітника через помилку ІІІ завдає суттєвої шкоди навколишньому середовищу [3].

Оскільки метрики точність та повнота часто перебувають у зворотній залежності, для фінального порівняння нейромережових архітектур використовується збалансована метрика F1-score, яка є їхнім середнім гармонійним:

$$F1 - score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}, \quad (3.3)$$

Цей критерій дозволяє отримати єдину інтегральну оцінку ефективності моделі на кожному з дванадцяти класів побутових відходів [14]. Додатковим інструментом для глибокого аналізу помилок моделі було обрано матрицю плутанини (Confusion Matrix), яка візуалізує розподіл передбачень моделі відносно реальних міток класів. Графічне представлення цієї матриці є невіддільною частиною аналізу результатів експериментів, оскільки воно чітко вказує на вектори помилок системи, дозволяючи оптимізувати архітектуру перед деплоєм у мобільний застосунок.

3 ПРОЄКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ

3.1 Обґрунтування вибору та попереднє оброблення набору даних

Ефективність навчання будь-якої глибокої згорткової нейронної мережі та її здатність до узагальнення критично залежать від якості, обсягу та різноманітності вхідних даних. У межах цього дослідження було здійснено детальний аналіз існуючих відкритих наборів даних, орієнтованих на завдання розпізнавання відходів. Зокрема, було розглянуто такі відомі бази даних, як Stanford TrashNet [26] та набір даних TACO (Trash Annotations in Context) [19]. Незважаючи на високу деталізацію анотацій у наборі TACO, його специфікація більше орієнтована на завдання сегментації об'єктів у природному середовищі (виявлення сміття на вулицях, парках та водоймах), що не цілком відповідає концепції розроблюваного мобільного застосунку, призначеного для класифікації поодиноких побутових предметів безпосередньо користувачами під час щоденного сортування.

Для забезпечення максимальної практичної цінності та всебічного охоплення побутових відходів за базовий орієнтир було обрано мультикласовий набір даних Garbage Classification, який у первинному вигляді містить дванадцять повноцінних категорій відходів [18]. На відміну від стандартних звужених вибірок, цей набір включає повний спектр матеріалів, що утворюються в межах житлових домогосподарств: папір, картон, біологічні (органічні) відходи, метал, пластик, батарейки, одяг, взуття, загальне непереробне сміття (trash), а також три окремі підкатегорії скла, розділені за кольором (зелене, коричневе та біле скло). Таке глибоке структурування є надзвичайно важливим для сучасної інфраструктури переробки, оскільки технологічні процеси утилізації кольорового скла та небезпечних відходів, таких як хімічні джерела живлення, кардинально відрізняються [13]. Вибірка містить вичерпну кількість вхідних даних для

всіх класів: біологічні відходи (985 зображень), папір (1050), картон (891), метал (769), пластик (865), батарейки (945), непереробне сміття (697), три окремі підкатегорії скла (коричнєве – 607, зелене – 629, біле – 775), а також дві масові специфічні категорії – одяг (5325) та взуття (1977).

Однією з головних проблем під час роботи з великими реальними масивами візуальних даних є нерівномірний розподіл екземплярів, відомий як дисбаланс класів [20]. Аналіз структури первинного датасету виявив наявність критичного дисбалансу класів у обраному датасеті. Кількість зображень у категоріях одягу та взуття сукупно складає майже 47% від усього обсягу даних. Наявність такого дисбалансу під час навчання нейронної мережі призводить до зміщення класифікатора: модель оптимізує свої ваги для досягнення мінімальної помилки на великих класах, значно погіршуючи точність розпізнавання нечисленних категорій. Для детальної візуалізації та аналізу початкового стану даних було побудовано гістограму розподілу (рисунок 3.1). Програмний код динамічного підрахунку зображень та генерації графіка наведено у додатку А, лістинг А.1.

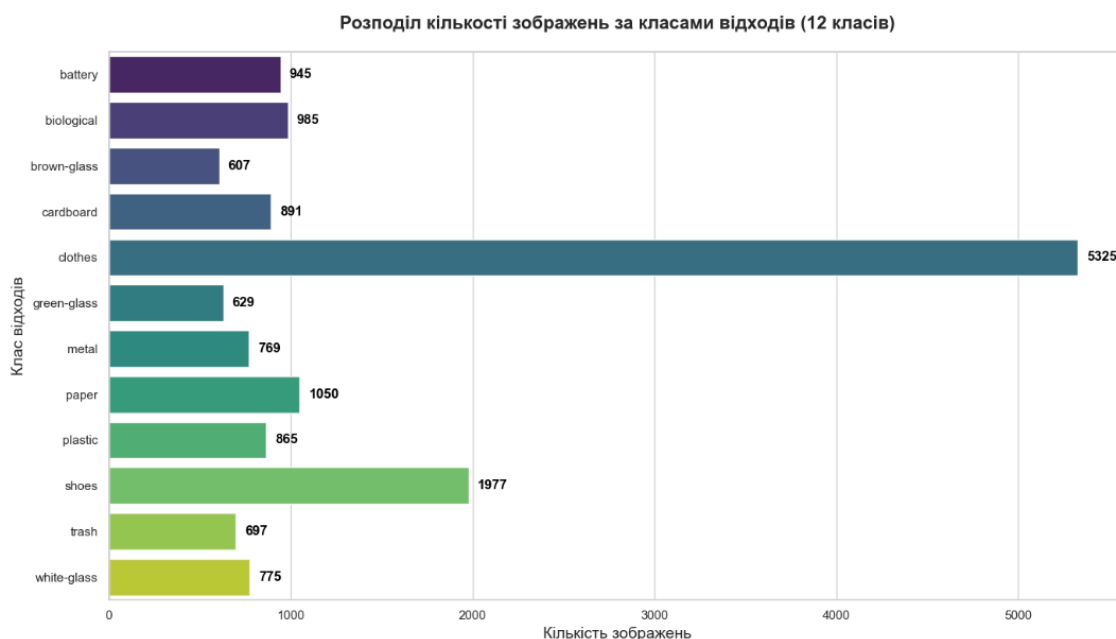


Рисунок 3.1 – Гістограма розподілу кількості зображень за дванадцятьма цільовими класами відходів

З огляду на виявлений дисбаланс, у межах дослідження було прийнято рішення провести оптимізацію структури даних для експериментів. Оскільки утилізація текстилю та взуття у реальному житті не належить до щоденного сортування і відбувається через спеціалізовані благодійні бокси, ці дві категорії були вилучені з вибірки. Очищений датасет охопив 10 суто екологічних класів загальним обсягом 8213 зображень, що дозволило збалансувати вибірку та сфокусувати „увагу моделі на виявленні тонких відмінностей між схожими матеріалами (наприклад, між картоном і папером).

Для підготовки даних було реалізовано комплексний конвеєр попередньої обробки, який включає розширення даних (Data Augmentation) та генерацію вибірок (відповідний програмний код наведено у додатку А, лістинг А.2.

Дані було розділено на дві незалежні вибірки: тренувальну (80%) та валідаційну (20%). Наступним критично важливим етапом підготовки стало уніфіковане приведення всіх графічних файлів до єдиного просторового розміру 224x224 пікселі із трьома кольоровими каналами (RGB). Цей крок є обов'язковим, оскільки архітектура MobileNetV2 має фіксовану розмірність вхідного тензора [22]. Також було проведено нормалізацію матриць пікселів зображень: цілі значення яскравості в діапазоні від 0 до 255 були масштабовані до плаваючого діапазону від 0.0 до 1.0. Нормалізація дозволяє значно прискорити процес градієнтного спуску та стабілізувати процес навчання глибокої мережі, запобігаючи занадто великим коливанням ваг на початкових епохах.

3.2 Експериментальне дослідження та порівняльний аналіз моделей класифікації

З метою визначення оптимальної архітектури нейронної мережі для мобільного застосунку було проведено серію експериментальних

досліджень. Як базову архітектуру було обрано MobileNetV2, оптимізовану для роботи на ресурсообмежених пристроях. Для об'єктивного оцінювання впливу структури даних на здатність моделі до узагальнення було сформовано два сценарії тестування:

– Експеримент А (базова модель): навчання на повному наборі даних із 12 категорій, що включає побутове сміття, а також категорії одягу та взуття;

– Експеримент Б (оптимізована модель): навчання на очищеному наборі даних із 10 категорій (вилучено класи «одяг» та «взуття» як такі, що не є релевантними для концепції швидкого сортування).

Навчання обох моделей проводилося з використанням алгоритму оптимізації Adam, функції втрат категоріальної кросентропії (Categorical Crossentropy) та механізму відсіювання (Dropout) з коефіцієнтом 0.2 для запобігання перенавчанню. Тривалість навчання становила 15 епох. Програмний код ініціалізації архітектури MobileNetV2, процесу навчання та автоматизованої побудови матриць плутанини наведено у додатку А, лістинг А.3–А.5.

Результати Експерименту А продемонстрували високий показник загальної точності (Ассигасу), який досяг 90% на валідаційній вибірці. Динаміка навчання базової моделі представлена на рисунках 3.2. та 3.3.

Проте, детальний аналіз метрик за класами (Classification Report) виявив явище «парадоксу точності», зумовлене значним дисбалансом у навчальній вибірці. Категорії «одяг» та «взуття» складала сукупно близько 47% валідаційної вибірки. З огляду на високу візуальну відмінність цих об'єктів від класичних відходів, модель класифікувала їх із безпрецедентно високою точністю (F1-score 0.98 для одягу та 0.95 для взуття). Це математично завищило загальний показник моделі, приховавши серйозні недоліки в розпізнаванні цільових екологічних категорій. Зокрема, базовій моделі було складно ідентифікувати пластик (F1-score 0.71) та біле скло (F1-score 0.66).

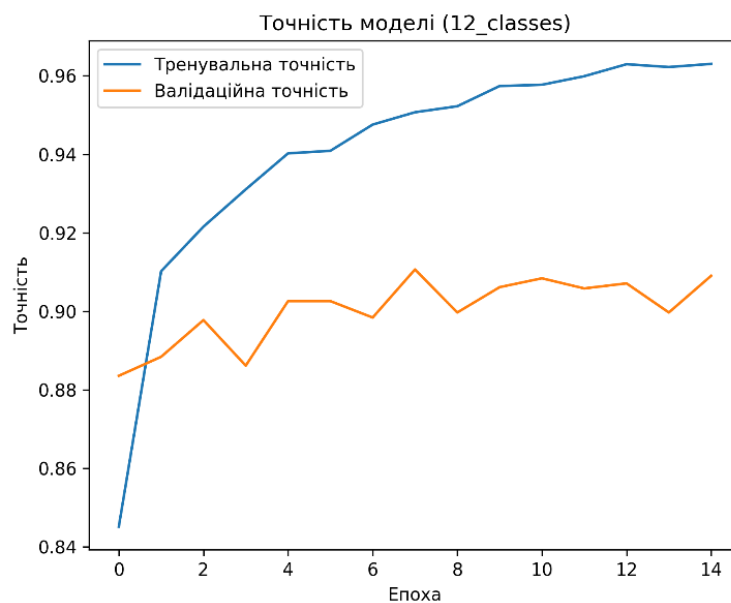


Рисунок 3.2 – Графік точності під час навчання базової моделі (12 класів)

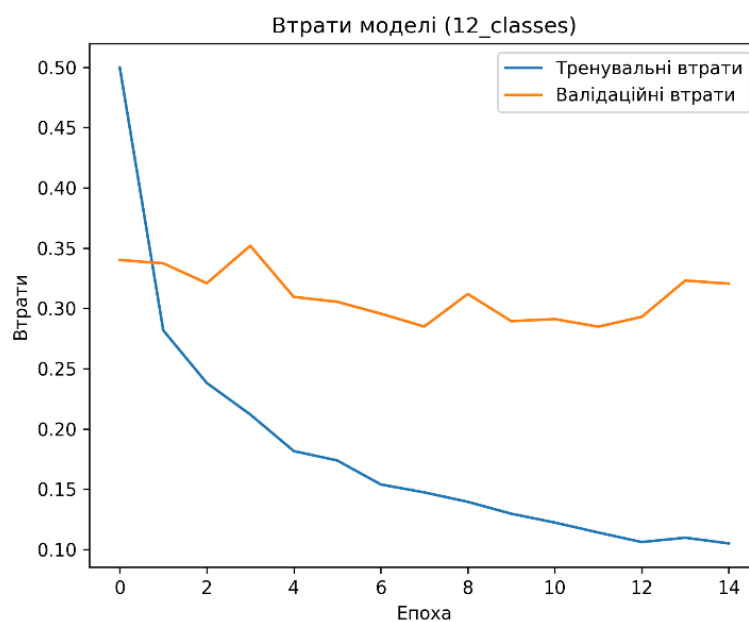


Рисунок 3.3 – Графік втрат під час навчання базової моделі (12 класів)

З метою покращення розрізняювальної здатності моделі щодо цільових відходів було проведено Експеримент Б. Вилучення нерелевантних класів дозволило оптимізувати розподіл ваг нейронної мережі, сфокусувавши її увагу на виявленні тонких візуальних відмінностей між схожими

матеріалами (наприклад, різними типами скла). Графік навчання оптимізованої моделі наведено на рисунках 3.4. та 3.5.

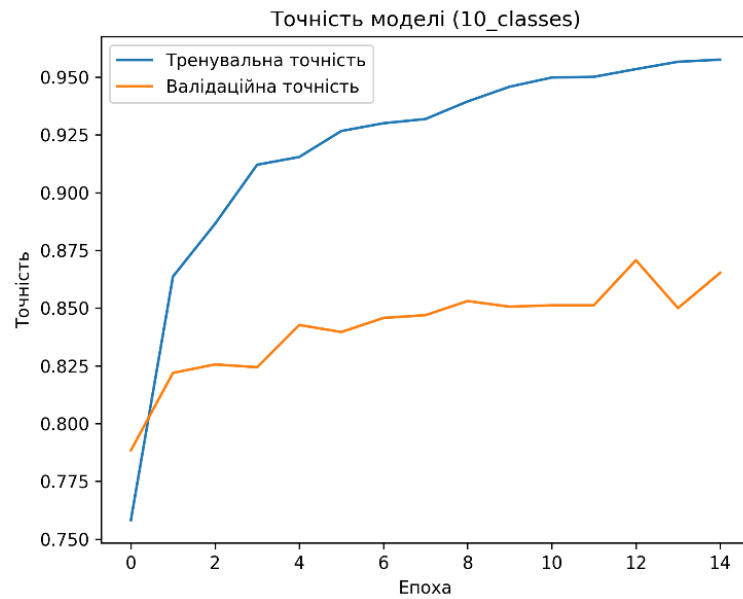


Рисунок 3.4 – Графік точності під час навчання оптимізованої моделі (10 класів)

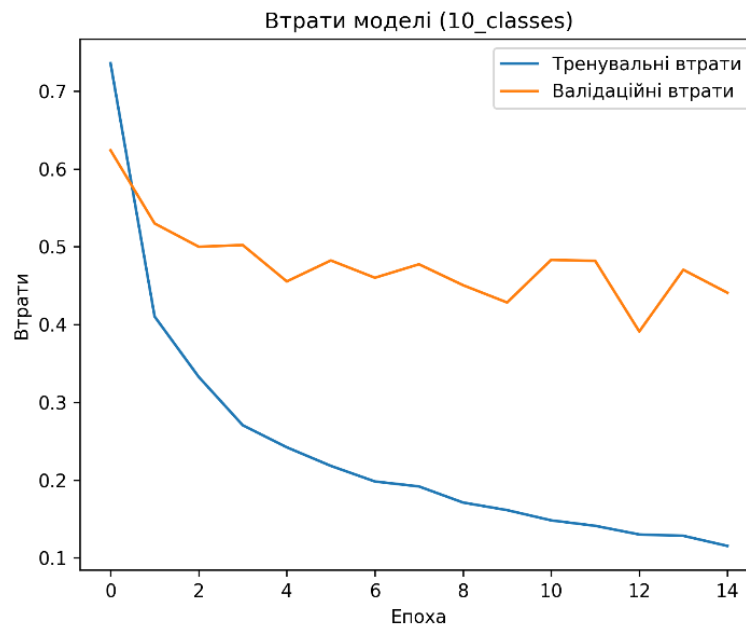


Рисунок 3.5 – Графік втрат під час навчання оптимізованої моделі (10 класів)

Після навчання оптимізованої моделі загальний показник точності (Accuracy) формально знизився до 85%. Однак це зниження є математичним наслідком вилучення найбільш легких масових класів, а не деградацією самої моделі. Аналіз матриць плутанини (Confusion Matrix) та метрики F1-score переконливо довів перевагу 10-класової архітектури.

У оптимізованій моделі спостерігається значне зростання точності для критично важливих типів відходів. Так, показник F1-score для коричневого скла зріс із 0.79 до 0.88, для білого скла – із 0.66 до 0.71, для пластику – із 0.71 до 0.75, а для органічних відходів досяг 0.96.

Для порівняння наведено матриці плутанини для обох експериментів (рисунки 3.6 та 3.7) Аналіз матриці плутанини оптимізованої моделі (рисунок 3.7) показує, що найвищі показники класифікації закріпилися за біологічними відходами, картоном та папером. Певна частка плутанини зберігається між пластиком та склом через їхню візуальну подібність (відблиски, прозорість), проте рівень хибнопозитивних спрацьовувань суттєво знизився порівняно з базовою моделлю.

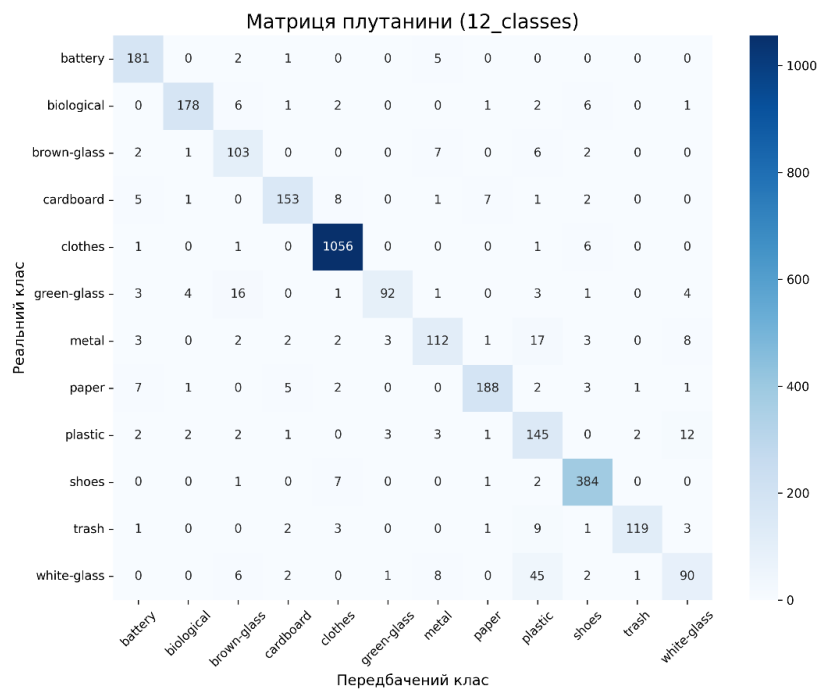


Рисунок 3.6 – Матриця плутанини для повної моделі (12 класів)

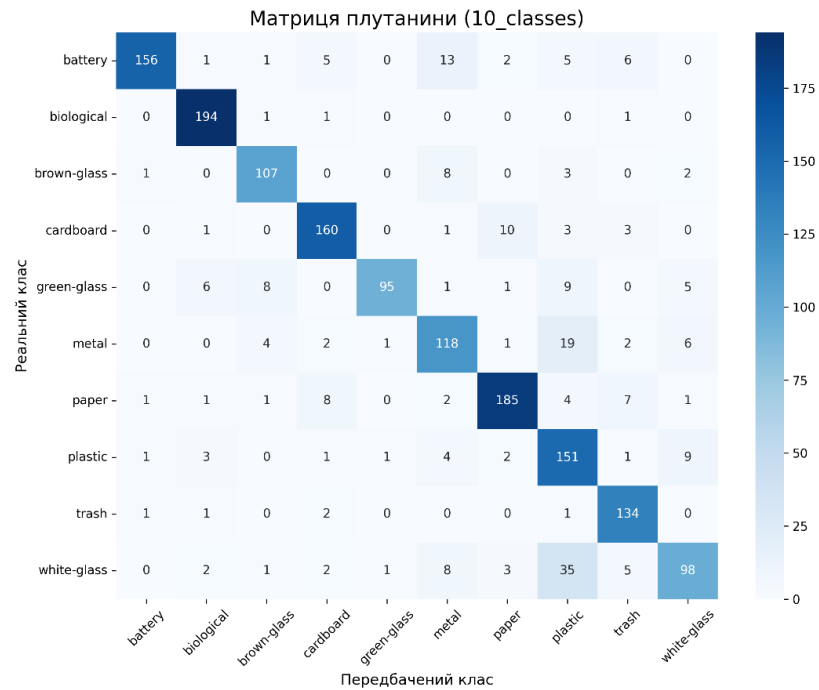


Рисунок 3.7 – Матриця плутанини для оптимізованої моделі (10 класів)

Підсумовуючи результати експериментального етапу, можна стверджувати, що оптимізована модель на 10 класів володіє значно кращою здатністю до генералізації в межах предметної галузі сортування відходів. Відмова від нерелевантних масових категорій дозволила підвищити чутливість системи до складних матеріалів. Саме цю натреновану модель MobileNetV2 було обрано для подальшого розгортання на серверній частині (FastAPI) та інтеграції з мобільним застосунком.

3.3 Архітектурне проєктування системи

Процес створення інтелектуальної інформаційної системи вимагає ретельного попереднього проєктування її компонентної бази та способів взаємодії між модулями. З огляду на специфіку поставленої задачі – забезпечення користувачів миттєвими інструкціями щодо сортування відходів на основі фотографій – архітектура системи повинна гарантувати

високу швидкість обробки візуальних даних, масштабованість та зручність клієнтського інтерфейсу.

Для реалізації цих вимог було обрано класичну клієнт-серверну архітектуру (Client-Server Architecture) з використанням RESTful API (Representational State Transfer Application Programming Interface) для обміну даними. Такий підхід дозволяє чітко розмежувати зону відповідальності: клієнтський застосунок відповідає виключно за збір даних (фотографування) та відображення результатів (UI/UX), тоді як сервер бере на себе ресурсомісткі завдання, пов'язані з обробкою зображень, виконанням математичного інференсу нейронної мережі та веденням статистичних записів у базі даних.

Центральною ланкою архітектури є розроблений серверний модуль (Backend), який містить попередньо натреновану модель згорткової нейронної мережі MobileNetV2 (описану в розділі 2). Сервер прослуховує вхідні HTTP-запити від мобільних клієнтів. Мобільний застосунок (Frontend) отримує зображення з камери смартфона або галереї та відправляє його на сервер у вигляді multipart-запиту. Отримавши зображення, сервер перетворює його у масив тензорів, нормалізує відповідно до вимог моделі, виконує класифікацію і повертає клієнту результат у форматі JSON, який містить ідентифікований клас, рівень впевненості штучного інтелекту (confidence) та відповідну екологічну інструкцію локалізованою мовою.

Для забезпечення простежуваності та збору аналітики, архітектура передбачає наявність реляційної бази даних на стороні сервера. Кожен запит на класифікацію реєструється в базі даних із прив'язкою до унікального ідентифікатора користувача, що дозволяє формувати персоналізовану статистику в мобільному застосунку.

Загальна структурна схема взаємодії компонентів розробленої системи наведена на рисунку 3.8.

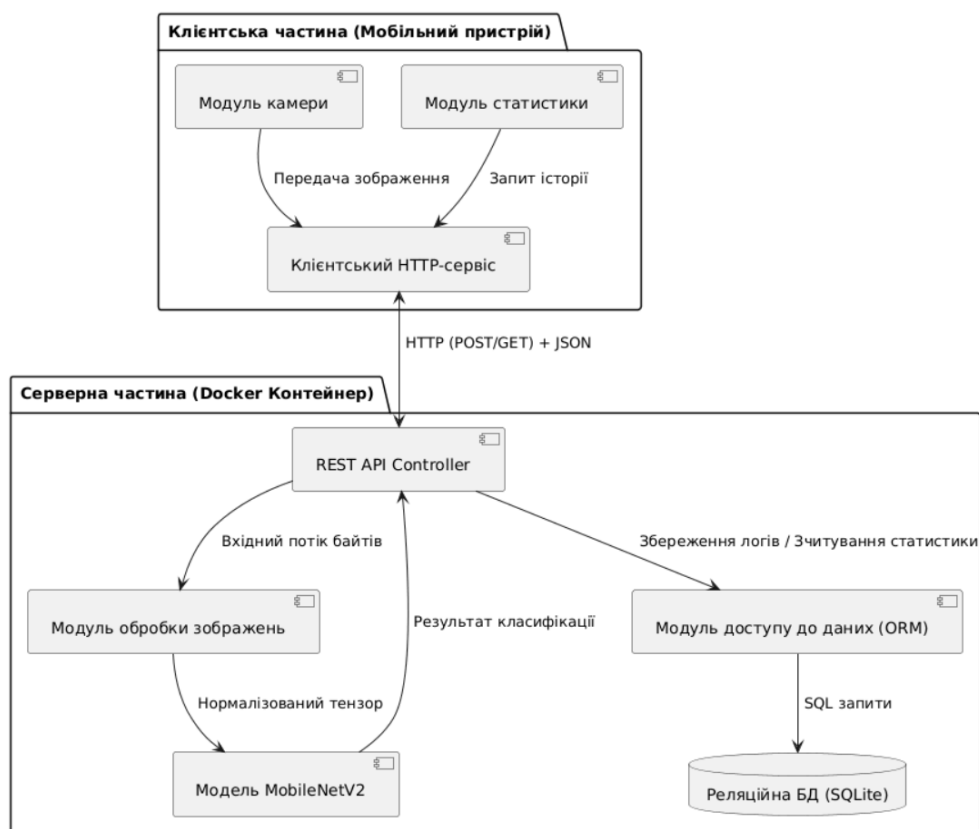


Рисунок 3.8 – Архітектура клієнт-серверної взаємодії системи класифікації відходів

Перенесення «важкої» моделі штучного інтелекту на сторону сервера звільняє мобільний пристрій від необхідності завантажувати великі файли ваг мережі, зберігає заряд акумулятора смартфона та дозволяє розробникам оновлювати алгоритми розпізнавання централізовано, без необхідності випускати оновлення самого мобільного застосунку у цифрових магазинах.

3.4 Обґрунтування вибору стеку технологій

Успішна реалізація спроектованої архітектури вимагає підбору релевантного стеку технологій (Technology Stack), який забезпечить високу продуктивність, надійність та швидкість розробки. Вибір інструментів здійснювався окремо для серверної та клієнтської частин.

Для розробки серверної частини (Backend) безальтернативним лідером у сфері машинного навчання є мова програмування Python. Ця мова володіє найпотужнішою екосистемою бібліотек для роботи зі штучним інтелектом, таких як TensorFlow, Keras, NumPy та Pandas. Використання Python дозволяє безшовно інтегрувати навчену модель у вебсервер без необхідності складних конвертацій форматів або використання проміжних мостів (наприклад, JNI для Java).

Серед розмаїття вебфреймворків на Python (Django, Flask, Tornado) для розробки REST API було обрано фреймворк FastAPI. На відміну від громіздкого Django, який більше підходить для монолітних застосунків із вбудованими HTML-шаблонами, FastAPI створений спеціально для швидкої побудови високонавантажених мікросервісів та API. Фреймворк базується на стандартах ASGI (Asynchronous Server Gateway Interface), що забезпечує асинхронну неблокуючу обробку запитів. Це критично важливо у контексті комп'ютерного зору, оскільки обробка зображення моделлю може займати певний час, і сервер не повинен блокуватися в очікуванні результату для одного користувача, відхиляючи паралельні запити інших клієнтів. Крім того, FastAPI автоматично генерує інтерактивну документацію (Swagger UI) на основі типів даних Pydantic, що суттєво спрощує тестування API та інтеграцію з мобільним клієнтом.

Для розробки мобільного застосунку (Frontend) було прийнято рішення відмовитися від нативної розробки окремо під платформи iOS (на Swift) та Android (на Kotlin) на користь кросплатформного підходу. Серед існуючих кросплатформних рішень (React Native, Xamarin, Ionic) найоптимальнішим вибором визнано фреймворк Flutter, розроблений компанією Google.

На відміну від гібридних фреймворків, які використовують WebView для рендерингу HTML/CSS-елементів (Ionic), або фреймворків, що трансклюють код через JavaScript-мости (React Native), Flutter компілює вихідний код мови Dart безпосередньо у машинний код цільової

платформи. Це забезпечує продуктивність на рівні нативних застосунків та плавність анімацій з частотою 60 кадрів на секунду. Крім того, Flutter має власний високопродуктивний графічний рушій Skia (у нових версіях – Impeller), що гарантує ідентичний вигляд користувацького інтерфейсу на екранах будь-якого розміру незалежно від версії операційної системи. Багатий набір готових віджетів Material Design дозволив швидко створити інтуїтивно зрозумілий та естетично привабливий еко-інтерфейс застосунку.

3.5 Проєктування бази даних

Будь-яка сучасна інтелектуальна клієнт-серверна система потребує надійного та структурованого механізму збереження персистентних (постійних) даних. У контексті розробленого застосунку для класифікації відходів, база даних виконує дві ключові функції. По-перше, вона забезпечує збереження історії взаємодій користувача з системою, що дозволяє формувати персоналізовану статистику, будувати графіки активності (еко-профіль) та стимулювати користувача до регулярного сортування через гейміфікацію процесу. По-друге, база даних акумулює звіти про помилки розпізнавання (аномалії), що є критично важливим матеріалом для подальшого донавчання та вдосконалення ШІ-моделі розробниками.

З огляду на архітектурні вимоги прототипу та необхідність забезпечення максимальної портативності серверного контейнера, як систему управління базами даних (СУБД) було обрано SQLite. На відміну від великих клієнт-серверних СУБД (таких як PostgreSQL або MySQL), які вимагають розгортання окремих демонів та виділення значних обсягів оперативної пам'яті, SQLite зберігає всю базу даних у єдиному локальному файлі безпосередньо у файловій системі контейнера. Це ідеально підходить для поточного етапу розвитку застосунку, оскільки забезпечує швидкий

доступ до даних (без мережових затримок між сервісами) та мінімізує складність конфігурації Docker-середовища.

Для взаємодії серверної частини (написаної на Python) з базою даних було застосовано технологію об'єктно-реляційного відображення (Object-Relational Mapping, ORM) за допомогою бібліотеки SQLAlchemy. Використання ORM-підходу дозволяє абстрагуватися від написання «сирих» SQL-запитів і працювати з таблицями бази даних як зі звичайними класами Python. Головною перевагою такого архітектурного рішення є те, що у разі майбутнього масштабування системи та переходу на потужнішу СУБД (наприклад, PostgreSQL), розробнику не доведеться переписувати логіку запитів – достатньо буде лише змінити рядок підключення (Connection String) в конфігурації сервера.

На етапі проєктування логічної структури бази даних було враховано принцип мінімізації збору персональних даних (Privacy by Design). Оскільки застосунок не вимагає від користувача введення електронної пошти, паролів чи номерів телефонів, класична сутність User (Користувач) не створюється. Натомість мобільний клієнт генерує унікальний криптографічний ідентифікатор (UUID v4) при першому запуску застосунку і зберігає його локально на пристрої (у SharedPreferences). Цей ідентифікатор (user_id) передається на сервер із кожним запитом, дозволяючи анонімно пов'язувати статистику з конкретним пристроєм.

База даних системи складається з двох основних сутностей (таблиць): «scan_logs» та «feedback_logs» (рисунок 3.9).

Сутність «scan_logs» (Журнал сканувань) – призначена для збереження результатів роботи нейронної мережі. Вона містить такі атрибути:

- id (Integer, Primary Key) – унікальний ідентифікатор запису;
- user_id (String) – анонімний ідентифікатор мобільного клієнта;
- item_name (String) – технічна назва розпізнаного класу сміття (наприклад, «brown-glass», «cardboard»);

- confidence (Float) – рівень впевненості нейронної мережі у відсотках (від 0.0 до 1.0);
- is_trash (Boolean) – логічний прапорець, що визначає, чи є об'єкт переробним (сміттям), чи це аномалія/невідомий об'єкт;
- timestamp (DateTime) – точний час і дата виконання сканування.

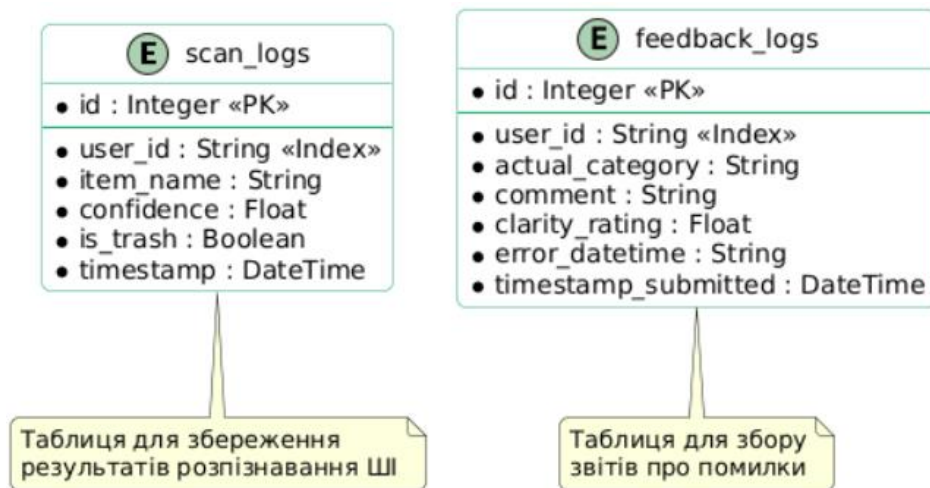


Рисунок 3.9 – ER-діаграма (Entity-Relationship) сутностей бази даних

Сутність «feedback_logs» (Журнал зворотного зв'язку) – таблиця для збору інформації від користувачів про випадки хибної класифікації. Її атрибути включають:

- id (Integer, Primary Key);
- user_id (String) – ідентифікатор користувача, який подав звіт;
- actual_category (String) – категорія, яку користувач вказав як правильну;
- comment (String) – текстовий опис проблеми або об'єкта;
- clarity_rating (Float) – суб'єктивна оцінка якості фотографії користувачем (від 1 до 10);
- error_datetime (String/DateTime) – час, коли виникла помилка;
- timestamp_submitted (DateTime) – час фактичного запису звіту в БД.

Спроектowana структура бази даних перебуває у третій нормальній формі (3NF), не містить надлишкових даних та забезпечує оптимальну швидкість виконання запитів на агрегацію (наприклад, підрахунок кількості відсканованих пластикових пляшок за останній місяць для формування кругових діаграм на клієнті).

3.6 Розробка серверної частини та інтеграція моделі машинного навчання

Серверна частина системи є її обчислювальним ядром, яке відповідає за реалізацію бізнес-логіки, взаємодію з мобільними клієнтами, роботу з базою даних та, що найважливіше, виконання математичного інференсу за допомогою попередньо навченої моделі згорткової нейронної мережі. Як було обґрунтовано раніше, розробка сервера здійснювалася мовою Python з використанням високопродуктивного асинхронного фреймворку FastAPI. Цей вибір дозволив створити легковагий, але надзвичайно потужний API, здатний ефективно обробляти мультипарт-запити із зображеннями.

Запуск серверного застосунку базується на ASGI-сервері Uvicorn, який ідеально підходить для розгортання FastAPI у виробничому середовищі. На етапі ініціалізації серверного застосунку відбувається кілька критично важливих процесів. По-перше, за допомогою методу `Base.metadata.create_all(bind=engine)` з бібліотеки SQLAlchemy відбувається автоматична перевірка наявності файлу бази даних. Якщо файл відсутній (наприклад, при першому запуску контейнера), система генерує таблиці відповідно до описаних моделей даних.

По-друге, реалізовано механізм глобального завантаження моделі штучного інтелекту у пам'ять сервера (Singleton Pattern). Модель MobileNetV2, збережена у форматі .h5 (результат Експерименту Б на 10 класів, описаного в підрозділі 3.2), завантажується за допомогою методу `tf.keras.models.load_model()` лише один раз під час старту сервера. Це

кардинально оптимізує продуктивність системи: якби модель завантажувалася з диска під час кожного вхідного запиту, час обробки однієї фотографії зростає би на 3–5 секунд через високі витрати на дисковий ввід-вивід (I/O). Попереднє завантаження моделі в оперативну пам'ять дозволяє виконувати класифікацію зображення за доли секунди. Також на етапі старту в оперативну пам'ять завантажується локалізований JSON-файл інструкцій (`sorting_instructions.json`), який містить поради щодо сортування кожної категорії українською та англійською мовами.

Основним інтерфейсом взаємодії клієнта із сервером є POST-ендпоінт «/predict». Процес обробки вхідного запиту на цьому ендпоінті є багатетапним (рисунок 3.10) і містить у собі весь життєвий цикл класифікації:

- прийом даних: сервер асинхронно приймає зображення від користувача (`UploadFile`), а також текстові параметри `user_id` та `lang` (мова);
- попередня обробка зображення (`Preprocessing`): отриманий потік байтів декодується за допомогою бібліотеки `PIL` (`Pillow`). Оскільки архітектура `MobileNetV2` вимагає чіткої розмірності вхідного тензора, зображення примусово масштабується (`resize`) до розміру 224x224 пікселі. Далі зображення перетворюється на матрицю `NumPy` (числовий масив). Оскільки модель навчалася на нормалізованих даних, значення яскравості кожного пікселя діляться на 255.0. На фінальному кроці масив розширюється на один вимір (додається `batch`-вимір), перетворюючись на тензор форми (1, 224, 224, 3), який готовий для подачі у нейронну мережу;
- виконання інференсу: сервер передає підготовлений тензор у завантажену модель викликом `model.predict()`. Модель повертає масив із 10 значень (відповідно до кількості класів оптимізованої моделі), де кожне значення представляє ймовірність належності зображення до певного класу відходів. За допомогою функції `np.argmax()` визначається індекс класу з найвищою ймовірністю, а відповідне значення зберігається як рівень впевненості (`confidence`);

– обробка аномалій та формування інструкцій: отриманий рівень впевненості порівнюється із заданим пороговим значенням (Threshold), яке було встановлено на рівні 60% (0.6). Якщо рівень впевненості нижчий за поріг, система ідентифікує об'єкт як такий, що не належить до відомих категорій сміття (Out-of-Distribution), і генерує повідомлення про неможливість класифікації, встановлюючи прапорець `is_trash = False`. Якщо об'єкт успішно розпізнано, прапорець `is_trash` встановлюється в `True`, а з підвантаженого JSON-файлу витягується відповідна екологічна інструкція обраною мовою;

– логування та відповідь: сформований результат (ідентифікований клас, рівень впевненості, прапорець та текст інструкції) асинхронно записується у базу даних (таблиця «scan_logs»), після чого повертається мобільному клієнту у форматі JSON.

Окрім ендпоінта класифікації, у серверній частині реалізовано аналітичний GET-ендпоінт «/stats/{user_id}». Він дозволяє мобільному застосунку отримувати агреговану статистику. Сервер отримує ідентифікованого користувача та часовий період (за день, місяць, рік або за весь час), формує SQL-запит до таблиці «scan_logs», відфільтровує записи, підраховує загальну кількість успішних класифікацій та генерує розбивку за категоріями у вигляді словника (Breakdown). Ці дані безпосередньо використовуються віджетом PieChart на стороні Flutter для малювання еко-профілю користувача. Третій ендпоінт, POST «/feedback», відповідає за прийом даних з екрану повідомлення про помилку та записує параметри, включаючи дату, час, рейтинг чіткості та опис користувача, у таблицю «feedback_logs».

Такий підхід до побудови серверної логіки дозволив створити високонадійний програмний інтерфейс, який забезпечує швидку роботу мобільного клієнта, збирає необхідні дані для покращення системи і є стійким до високих навантажень.

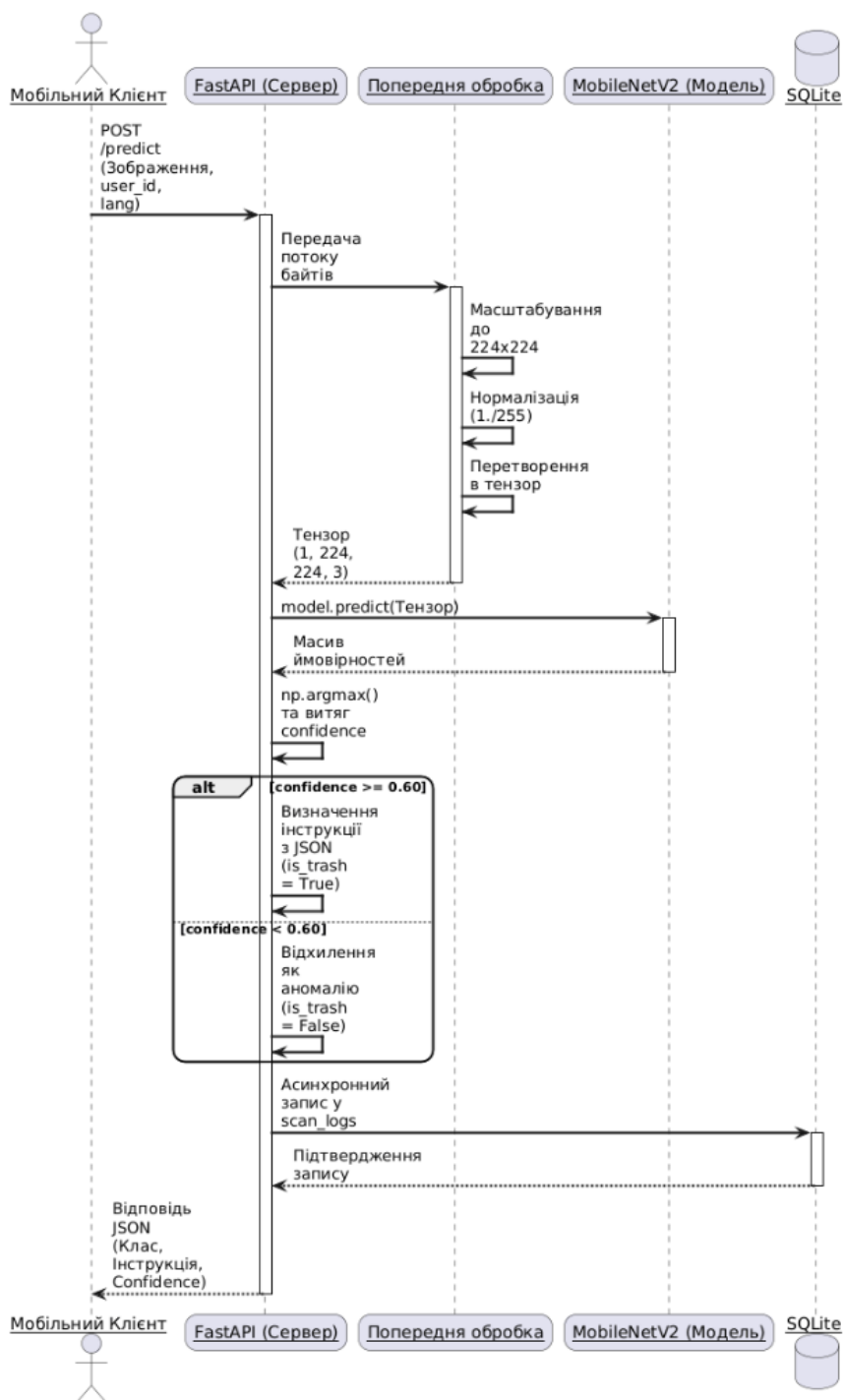


Рисунок 3.10 – Діаграма послідовності (Sequence Diagram) обробки запиту на розпізнавання відходів

3.7 Розробка кросплатформного мобільного застосунку

Клієнтська частина системи реалізована у вигляді мобільного застосунку, розробленого за допомогою фреймворку Flutter. Головним

завданням мобільного клієнта є забезпечення інтуїтивно зрозумілого інтерфейсу (UI) для взаємодії користувача із серверною моделлю машинного навчання, а також забезпечення безперебійного користувацького досвіду (UX) під час сортування відходів. Завдяки використанню кросплатформної технології, застосунок компілюється в нативний код як для операційної системи Android, так і для iOS з єдиної кодової бази.

Архітектура мобільного застосунку побудована на основі механізмів управління станом (State Management) віджетів StatefulWidget. Структурно навігація реалізована за допомогою нижньої панелі (BottomNavigationBar), яка дозволяє користувачеві миттєво перемикатися між трьома основними екранами: «Профіль» (статистика), «Сканер» (головний робочий екран) та «Помилка» (форма зворотного зв'язку). Для забезпечення локалізації застосунок підтримує динамічне перемикання між українською та англійською мовами. Вибрана мова, а також унікальний ідентифікатор пристрою (user_id), згенерований бібліотекою uuid, зберігаються у локальному сховищі смартфона за допомогою пакета «shared_preferences», що гарантує збереження сесії та налаштувань навіть після перезапуску програми.

Головний екран «Сканер» є ключовим модулем системи. Його логіка інтегрована з нативними компонентами пристрою через бібліотеку «image_picker», яка дозволяє отримувати фотографії безпосередньо з камери або обирати вже існуючі зображення з галереї. Після успішного отримання зображення застосунок ініціює асинхронний HTTP-запит до серверного ендпоінта «/predict» (програмний код функції відправлення фотографії та отримання результату наведено у додатку Б, лістинг Б.1). Для ефективної передачі бінарних даних використовується формат «multipart/form-data». На час очікування відповіді від сервера інтерфейс переходить у стан завантаження, блокуючи повторні запити та демонструючи індикатор процесу. Після отримання JSON-відповіді від

сервера екран оновлюється, відображаючи результати розпізнавання: назву класу, рівень впевненості ШІ, відповідну іконку (сміття або попередження) та локалізовану детальну інструкцію щодо правильної утилізації розпізнаного матеріалу (рисунок 3.11).

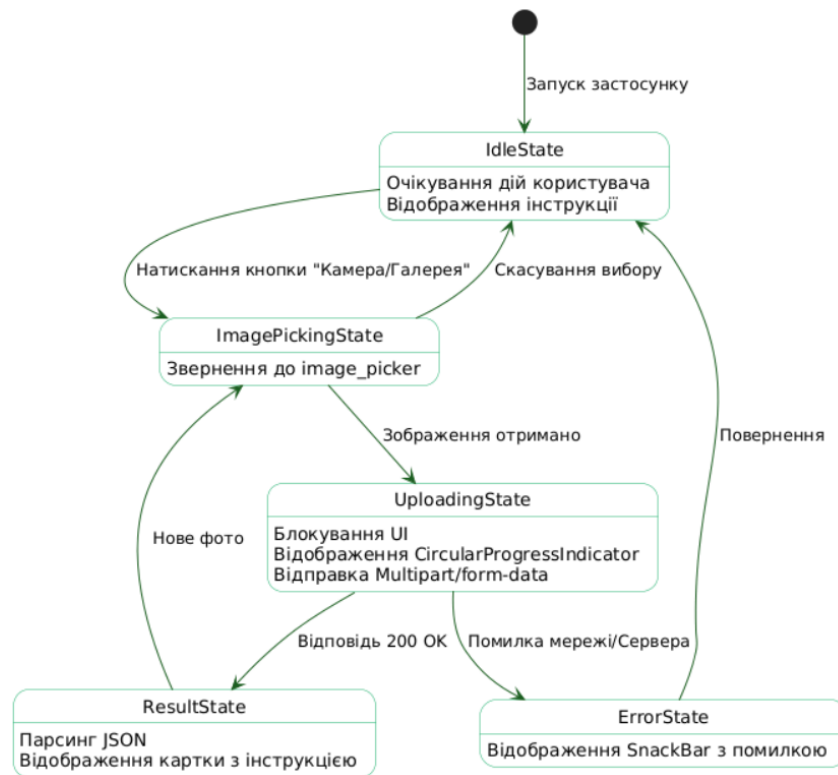


Рисунок 3.11 – Діаграма станів (State Machine) модуля сканування мобільного застосунку

Екран «Профіль» відповідає за відображення екологічної активності користувача. Цей модуль здійснює GET-запити до серверного ендпоінта статистики, дозволяючи фільтрувати дані за періодами (за день, місяць, рік або за весь час). Візуалізація накопичених даних реалізована за допомогою бібліотеки `fl_chart`, яка будує інтерактивну кругову діаграму (PieChart) на основі агрегованого розподілу розпізнаних категорій. Під діаграмою виводиться хронологічний список усіх сканувань (ListView), що дозволяє користувачу аналізувати власну поведінку та прогрес.

Модуль «Помилка» (AnomalyReportScreen) забезпечує механізм зворотного зв'язку (Feedback Loop), який є критично важливим для подальшого вдосконалення III-моделі. Екран містить складну форму, що включає текстові поля, випадаючі списки для вибору фактичної категорії відходів, слайдер для оцінки чіткості фотографії, а також нативні віджети вибору дати (showDatePicker) та часу (showTimePicker). Зібрані дані формуються у JSON-об'єкт та відправляються POST-запитом на сервер.

3.8 Контейнеризація та розгортання системи

Для забезпечення надійності, відтворюваності та масштабованості серверної частини розробленої системи було застосовано технологію контейнеризації на базі платформи Docker. У традиційній моделі розробки програмного забезпечення часто виникає проблема несумісності середовищ («works on my machine»), коли код, що успішно виконується на комп'ютері розробника, видає помилки на виробничому сервері через відмінності у версіях операційної системи, системних бібліотеках або залежностях Python. Контейнеризація повністю усуває цю проблему шляхом ізоляції застосунку та всіх його залежностей у єдиний стандартизований блок – Docker-контейнер.

Основою для збірки контейнера є декларативний файл конфігурації Dockerfile. Процес побудови образу (Image) розпочинається з використання мінімалістичного базового образу python:3.10-slim, який базується на урізаній версії дистрибутиву Debian Linux. Це дозволяє значно зменшити кінцевий розмір контейнера та скоротити час його завантаження. На наступному етапі відбувається оновлення системних пакетів та встановлення компіляторів (build-essential), необхідних для збирання специфічних C-розширень деяких Python-бібліотек.

Управління залежностями реалізовано через файл requirements.txt. Для оптимізації процесу розгортання встановлення пакетів за допомогою

утиліти `pip` виконується з параметром `--no-cache-dir`. Цей прапорець забороняє кешування завантажених архівів утилітою, що додатково економить десятки мегабайтів дискового простору всередині контейнера. Серед встановлюваних залежностей ключовими є фреймворк `fastapi`, сервер `uvicorn`, інструменти для роботи з базами даних `SQLAlchemy`, а також `tensorflow` версії, що відповідає формату збереженої моделі `MobileNetV2`.

Важливим архітектурним рішенням на етапі контейнеризації є правильне використання файлу `.dockerignore` (рисунок 3.12). До нього були включені всі файли та директорії, які не є критично необхідними для роботи сервера в режимі інференсу: віртуальне середовище (`.venv`), масиви навчальних даних (`dataset`), скрипти для тренування моделі (`train_model.py`), а також локальні лог-файли та інструменти прототипування.

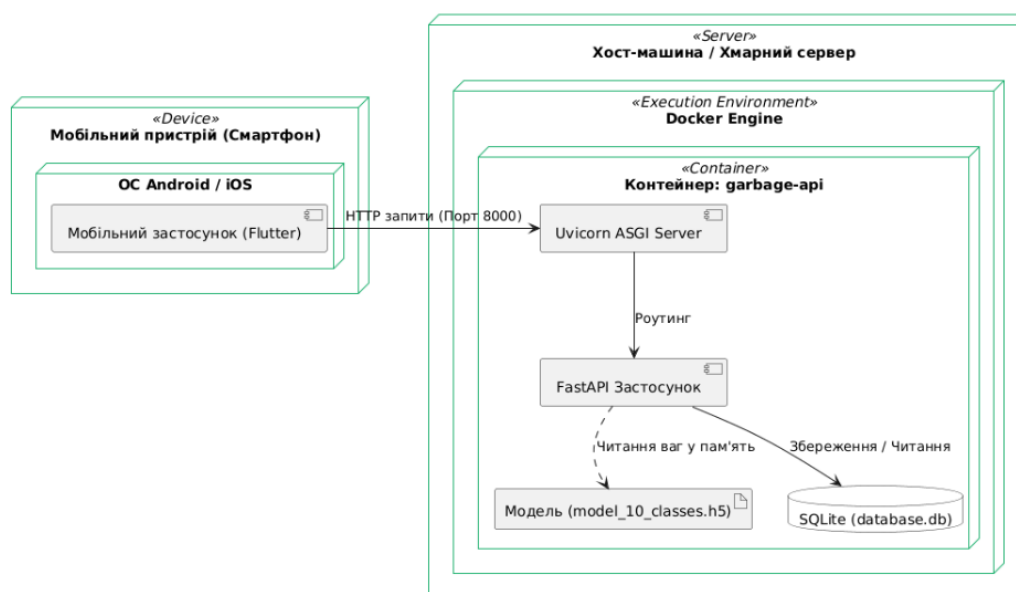


Рисунок 3.12 – Діаграма розгортання (Deployment Diagram) системи на базі Docker

Особливу увагу було приділено виключенню локального файлу бази даних (`database.db`). Це рішення гарантує, що Docker-контейнер завжди створюється без застарілих або тестових даних. Натомість, завдяки

механізмам ORM, нова база даних з актуальною схемою автоматично генерується всередині контейнера при першому запуску ASGI-сервера.

Оркестрація запуску контейнера передбачає відкриття порту 8000 для прослуховування зовнішніх HTTP-запитів. Запуск застосунку всередині контейнера виконується командою CMD ["uvicorn", "fastapi_server:app", "--host", "0.0.0.0", "--port", "8000"], де прапорець 0.0.0.0 є критично важливим для дозволу маршрутизації мережевого трафіку з хост-машини (фізичного комп'ютера або сервера) всередину ізольованої мережі Docker-контейнера.

Використання контейнеризації дозволило перетворити серверну частину системи на портативний мікросервіс. Завдяки цьому розроблене рішення може бути миттєво розгорнуте на будь-якій хмарній платформі (наприклад, AWS EC2, Google Cloud Run або DigitalOcean) за допомогою єдиної консольної команди, забезпечуючи високий рівень відмовостійкості та готовність системи до промислової експлуатації.

4 ДЕМОНСТРАЦІЯ ТА АНАЛІЗ РОБОТИ СИСТЕМИ

4.1 Інтерфейс користувача та навігація

Фінальним етапом розробки інтелектуальної системи сортування відходів стала імплементація та тестування мобільного клієнта, який безпосередньо взаємодіє з кінцевим користувачем. Дизайн застосунку, розробленого на базі фреймворку Flutter, спроєктовано з урахуванням сучасних стандартів мобільних інтерфейсів (Human Interface Guidelines) та орієнтований на максимальну простоту використання (Zero-Click підхід). Основна мета інтерфейсу – дозволити користувачеві отримати інструкцію з утилізації сміття менш ніж за три секунди після відкриття застосунку.

Загальна колірна палітра застосунку (ThemeData) побудована на відтінках зеленого кольору (Color(0xFF1CB06D) та Color(0xFF22C55E)), що психологічно асоціюється з екологією, природою та процесами переробки. Фоновий колір екранів має світло-зелений відтінок (0xFFF0FDF4), що знижує контрастне навантаження на очі користувача та створює естетично привабливе середовище. Важливою архітектурною особливістю інтерфейсу є його повна двомовність (українська та англійська мови), яка реалізована через внутрішній словник локалізації. Перемикання мови відбувається миттєво за допомогою випадаючого списку (DropDownButton) у верхній панелі керування (AppBar), без необхідності перезавантаження застосунку або виконання додаткових мережевих запитів.

Головним робочим простором системи є екран сканера. Візуальна ієрархія цього екрана зосереджена на великому центральному контейнері, який має градієнтну заливку та слугує видошукачем. У стані очікування (до моменту зйомки) контейнер містить тематичну ілюстрацію екологічного спрямування (рисунок 4.1). Під візуальним блоком розташовано дві великі круглі кнопки швидкої дії (Floating Action Buttons), які дозволяють миттєво

активувати камеру пристрою або вибрати існуюче зображення з галереї смартфона.



Рисунок 4.1 – Головний екран мобільного застосунку у стані очікування

Після того, як користувач робить знімок, інтерфейс динамічно змінюється. Зображення заповнює центральний контейнер, а поверх нього з'являється індикатор завантаження (CircularProgressIndicator), який сигналізує про виконання мережевого запиту до сервера FastAPI. Щойно сервер повертає результат класифікації, у нижній частині екрана плавно з'являється інформаційна картка (Result Card), код генерації цього елемента

інтерфейсу наведено у додатку Б, лістинг Б.2. Ця картка містить чітку та структуровану інформацію (рисунок 4.2): назву розпізнаного матеріалу (наприклад, «Пластик» або «Органіка»), відсоток впевненості нейронної мережі та, найголовніше, розгорнуту текстову інструкцію щодо правильної підготовки цього типу сміття до переробки (наприклад, «Зніміть пляшку для економії місця» або «Зніміть кришечку»).

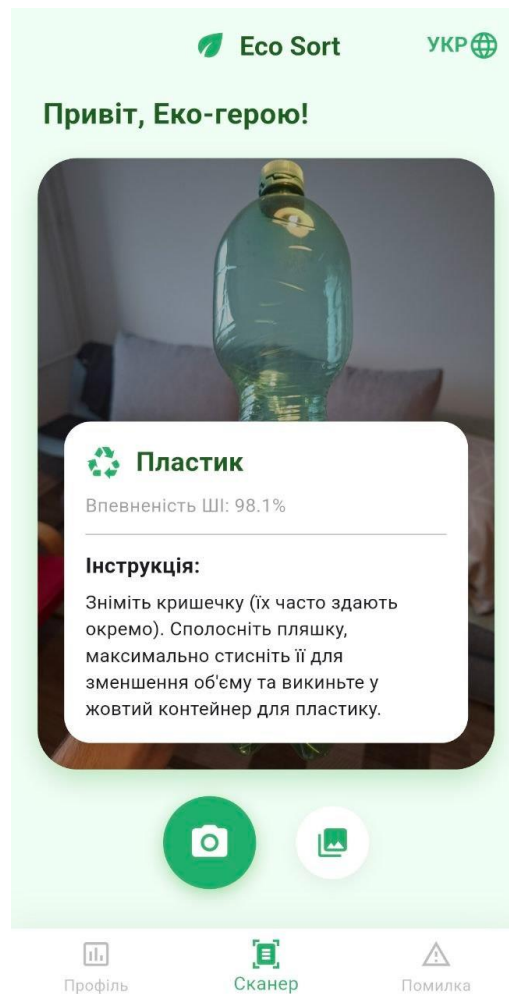


Рисунок 4.2 – Відображення результату розпізнавання та екологічної інструкції

Завдяки використанню асинхронного програмування у Dart, процес переходу від натискання кнопки затвора камери до відображення фінальної інструкції відбувається без блокування основного потоку інтерфейсу (UI

Thread), що забезпечує високу плавність роботи програми на пристроях різної потужності.

4.2 Еко-профіль та візуалізація статистики

Для формування екологічних звичок у користувачів, застосунок містить модуль персональної статистики – «Мій еко-профіль». Цей розділ виконує функцію гейміфікації та зворотного зв'язку, дозволяючи користувачеві візуально оцінити власний внесок у процес сортування відходів. Навігація до цього модуля здійснюється через нижню панель вкладок (BottomNavigationBar).

Екран статистики побудований за принципом інформаційного дашборду (рисунок 4.3). У верхній частині екрана користувач має можливість обрати часовий період для аналізу: за весь час використання, за поточний рік, місяць або день. Коли користувач змінює період, мобільний клієнт відправляє GET-запит на сервер (програмний код HTTP-запиту для агрегації статистики наведено у додатку Б, лістинг Б.3), передаючи свій унікальний `user_id` та параметр періоду. Сервер агрегує дані з таблиці «scan_logs» SQLite бази даних і повертає JSON зі статистикою.

Основою візуалізації на цьому екрані є кругова діаграма (PieChart), реалізована за допомогою пакета «fl_chart». Діаграма динамічно розраховує сектори відповідно до кількості відсканованих об'єктів у кожній категорії. Кожному типу матеріалу (скло, пластик, папір тощо) автоматично присвоюється унікальний колір з попередньо визначеної палітри, що робить діаграму інтуїтивно зрозумілою без необхідності тривалого вивчення легенди.

Під круговою діаграмою розташовано модуль «Історія сканувань» (ListView). Це хронологічний список-журнал, який відображає деталі кожної взаємодії з системою штучного інтелекту. Кожен елемент

списку оформлено у вигляді картки (Card), яка містить назву розпізнаного матеріалу та точний час сканування (timestamp).

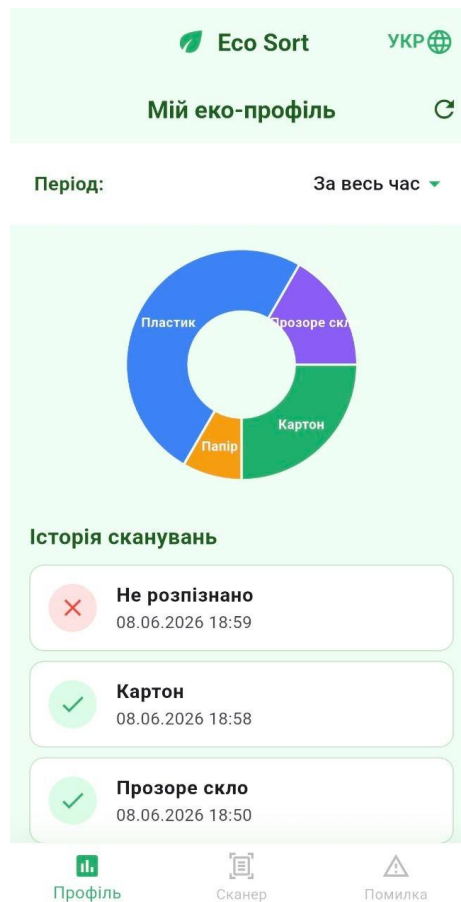


Рисунок 4.3 – Екран «Еко-профіль» з візуалізацією персональної статистики сортування

Важливою особливістю історії є візуальне маркування успішності. Якщо нейронна мережа розпізнала об'єкт як цільове сміття (рівень впевненості вищий за 60%), біля запису відображається зелена галочка на зеленому фоні. Якщо ж користувач намагався відсканувати предмет, що не підлягає сортуванню (або система не змогла його розпізнати), такий запис позначається червоним хрестиком. Таке розмежування допомагає користувачеві аналізувати власні помилки та краще розуміти логіку роботи системи.

4.3 Механізм обробки аномалій та зворотний зв'язок

Невіддільною складовою будь-якої інтелектуальної системи, що взаємодіє з реальними користувачами, є механізм обробки непередбачуваних ситуацій та збору зворотного зв'язку (Feedback Loop). У лабораторних умовах нейронні мережі тестуються на ідеальних датасетах, однак під час експлуатації мобільного застосунку система неминуче стикається з візуальним шумом, розмитими фотографіями або нецільовими об'єктами (Out-of-Distribution). Для вирішення цієї проблеми у мобільному клієнті розроблено окремий модуль – екран «Помилка» (Anomaly Report Screen), навігація до якого здійснюється через нижню панель (рисунок 4.4).



Рисунок 4.4 – Інтерфейс модуля зворотного зв'язку для повідомлення про помилки класифікації

Цей модуль виконує функцію краудсорсингу (збору даних від натовпу) для майбутнього донавчання та вдосконалення моделі MobileNetV2. Якщо користувач помітив, що система неправильно

класифікувала об'єкт (наприклад, переплутала пластик зі склом) або взагалі відмовилася його розпізнавати, він може надіслати деталізований звіт безпосередньо розробникам. Інтерфейс екрана спроектований у вигляді структурованої форми збору даних, що мінімізує ймовірність введення некоректної інформації.

Форма зворотного зв'язку складається з кількох логічних блоків. Перший блок – це випадаючий список (DropDownButtonFormField), у якому користувач має вказати фактичну (правильну) категорію об'єкта, що знаходився в кадрі (рисунок 4.5). Важливою архітектурною особливістю є те, що користувач бачить назви категорій своєю рідною мовою (наприклад, «Коричнєве скло» або «Органіка»), але «під капотом» застосунок формує JSON-запит із технічними англійськими ключами (brown-glass, biological). Це гарантує цілісність бази даних на сервері незалежно від мови інтерфейсу клієнта.

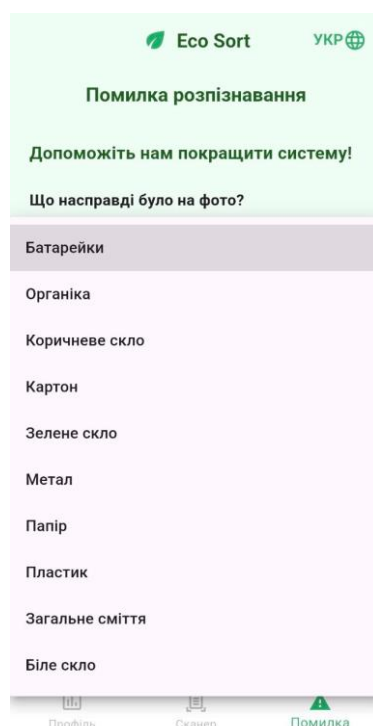


Рисунок 4.5 – Обрання категорії одиниці сміття на фото для повідомлення про помилки класифікації

Другий блок містить текстове поле (TextFormField) для введення довільного коментаря, де користувач може описати нестандартну форму пакування чи умови освітлення (рисунок 4.6).

Третій блок реалізовано у вигляді інтерактивного повзунка (Slider), за допомогою якого користувач за десятибальною шкалою оцінює чіткість зробленої фотографії (рисунок 4.6). Це критично важлива метрика для аналітиків даних: якщо неймережа помиляється на фотографіях з оцінкою чіткості 1–3 бали, це свідчить про проблему з апаратним фокусуванням, а якщо помилка виникає на знімках з оцінкою 9–10 балів – це прямий сигнал про необхідність донавчання моделі на конкретному типі об'єктів.

Рисунок 4.6 – Додавання коментаря для повідомлення про помилки класифікації

Останній блок дозволяє зафіксувати точний час виникнення помилки за допомогою нативних віджетів `showDatePicker` (рисунок 4.7) та `showTimePicker` (рисунок 4.8).

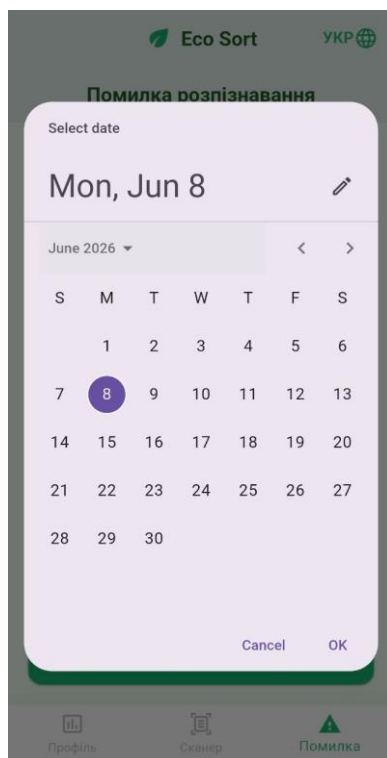


Рисунок 4.7 – Обрання дати для повідомлення про помилки класифікації

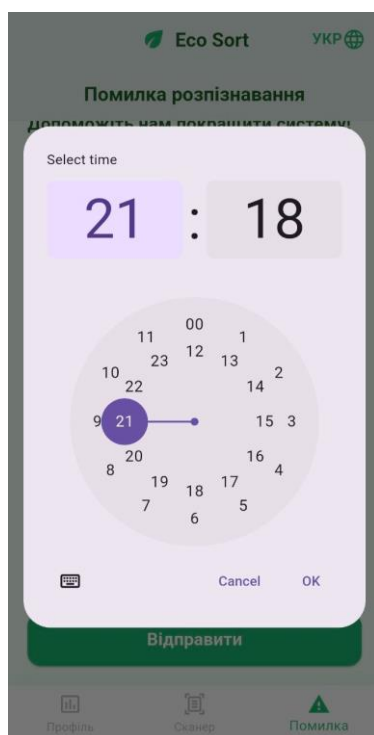


Рисунок 4.8 – Обрання часу для повідомлення про помилки класифікації

Після заповнення форми користувач натискає кнопку «Відправити». Застосунок блокує інтерфейс для запобігання дублюванню (відображаючи індикатор завантаження) та відправляє POST-запит на ендпоінт «/feedback» серверної частини (програмний код підготовки та відправлення звіту наведено у додатку Б, лістинг Б.4). Отримавши статус-код 200 OK, мобільний клієнт очищує форму та виводить повідомлення-подяку (SnackBar), підтверджуючи успішну транзакцію (рисунок 4.9). Дані надійно зберігаються у таблиці «feedback_logs» SQLite бази даних на сервері.

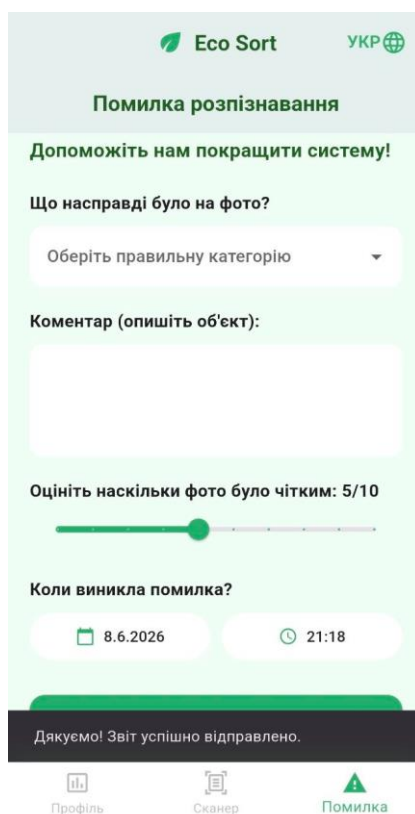


Рисунок 4.9 – Повідомлення про підтвердження відправлення форми
зворотного зв'язку

Впровадження підходу «людина в циклі» (Human-in-the-Loop) через даний модуль дозволяє перетворити мобільний застосунок із простого класифікатора на систему, що безперервно збирає розмічені дані для власної

еволюції, компенсуючи тим самим обмеження початкового навчального набору даних.

4.4 Оцінка ефективності розпізнавання у реальних умовах

Для підтвердження практичної цінності розробленої системи було проведено серію натурних експериментів (Field Testing). Метою цього етапу було дослідження поведінки натренованої ШІ-моделі (оптимізована версія MobileNetV2 на 10 класів) в умовах, які максимально наближені до повсякденного побуту користувача. Тестування здійснювалося безпосередньо через інтерфейс мобільного застосунку з використанням вбудованої камери смартфона. Оцінка ефективності проводилася з урахуванням трьох ключових змінних: типу освітлення (природне денне та штучне вечірнє), ракурсу зйомки та наявності деформацій об'єкта.

4.4.1 Тестування за умов достатнього природного освітлення

В умовах денного світла (біля вікна або на вулиці) система продемонструвала результати, які майже ідеально корелюють із метриками, отриманими на валідаційній вибірці (розділ 3). Завдяки правильному балансу білого та відсутності цифрового шуму на матриці камери, алгоритм впевнено виділяє текстурні ознаки матеріалів.

Макулатура (папір та картон) розпізнавалася системою з рівнем впевненості 87–98% незалежно від того, чи був це рівний аркуш, чи зім'ята картонна коробка (рисунки 4.10–4.13).

Органічні відходи також ідентифікувалися з невеликою часткою помилки (рисунки 4.14–4.15), що підтверджує високий F1-score (0.96) цієї категорії, отриманий під час матричного аналізу.



Рисунок 4.10 – Фото предмету макулатури (картон) за умов денного освітлення



Рисунок 4.11 – Класифікація предмету макулатури (картон) за умов денного освітлення

[illegible]

Рисунок 4.13 – Класифікація предмету макулатури (папір) за умов денного освітлення



Рисунок 4.14 – Фото предмету органіки (морква) за умов денного освітлення

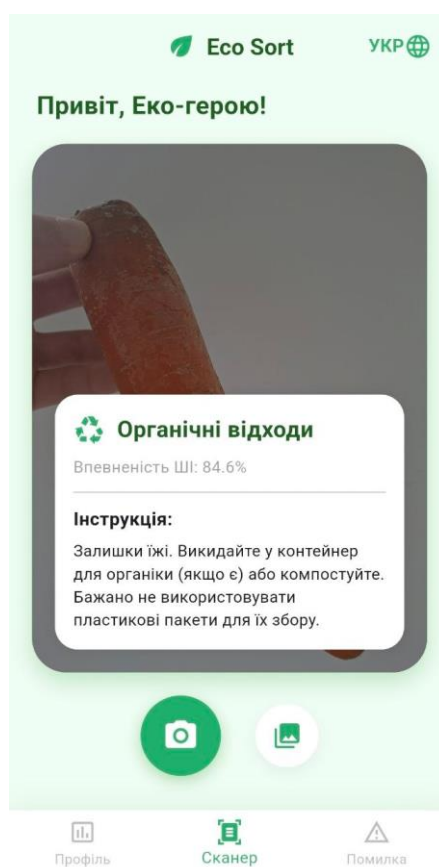


Рисунок 4.15 – Класифікація предмету органіки (морква) за умов денного освітлення

Батарейки та акумулятори розпізнавалися не дуже успішно. Встановлено, що через малий фізичний розмір батарейок, зйомка з відстані понад 20–25 см призводить до падіння впевненості моделі, оскільки об'єкт займає занадто малу площу тензора (224x224 пікселі), що передається у мережу (рисунки 4.16–4.17).



Рисунок 4.16 – Фото батарейки зроблено доволі далеко від камери

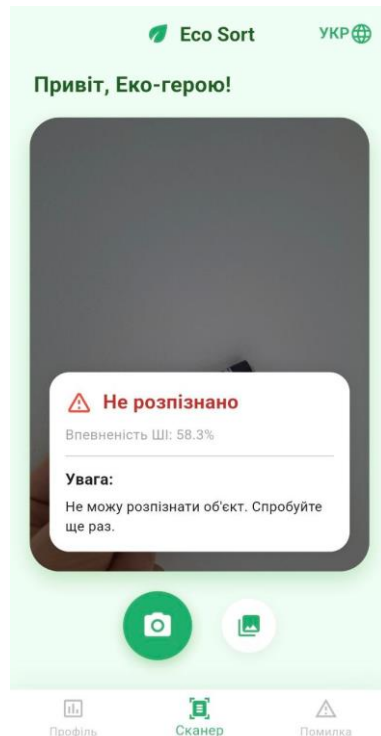


Рисунок 4.17 – Помилка класифікації батарейок через замалу частку пікселів, які відносяться саме до цільового об'єкту

Також батарейки часто розпізнавалися як метал, через схожість з металевими банками, або система, через замалу впевненість, видавала помилку (рисунки 4.18–4.19).



Рисунок 4.18 – Фото батарейки зроблене на дуже близькій відстані від камери



Рисунок 4.19 – Помилка класифікації батарейок через значну схожість з предметами класу металу

На рисунках 4.20–4.21 наведено правильну класифікацію батарейок.



Рисунок 4.20 – Фото батарейки зроблене на невеликій відстані від камери



Рисунок 4.21 – Успішна класифікація батарейок

Для успішного розпізнавання батарейок найкращою стратегією було фотографування кількох батарейок одразу (рисунки 4.22–4.23), бо у датасеті, який було використано для навчання моделі, була значна кількість саме таких зображень.



Рисунок 4.22 – Стратегія фотографування кількох батарейок одночасно



Рисунок 4.23 – Успішна класифікація батарейок за використання стратегії декількох предметів

4.4.2 Тестування за умов штучного освітлення (складні умови)

Більшість сценаріїв побутового сортування сміття відбувається ввечері при штучному освітленні (LED-лампи або лампи розжарювання). У таких умовах на поверхні глянцевого об'єктів з'являються різкі відблиски, а кольори спотворюються.

Експерименти виявили, що металеві банки (алюміній) та пластикове пакування (пляшки) є найбільш вразливими до зміни освітлення (рисунки 4.24–4.27). За наявності сильних відблисків від лампи, нейронна мережа іноді класифікує прозорий пластик як скло (через подібність спектру пропускання світла), знижуючи рівень впевненості до 65–70% (рисунки 4.28–4.29). Проте, завдяки відмові від 12-класової моделі на користь 10-класової, модель перестала помилково класифікувати пластикові пакети як одяг, що було типовою проблемою на попередніх етапах розробки. Скло різних кольорів (коричнєве та зелене) розпізнавалося коректно, якщо за ним знаходився нейтральний фон (рисунки 4.30–4.31).

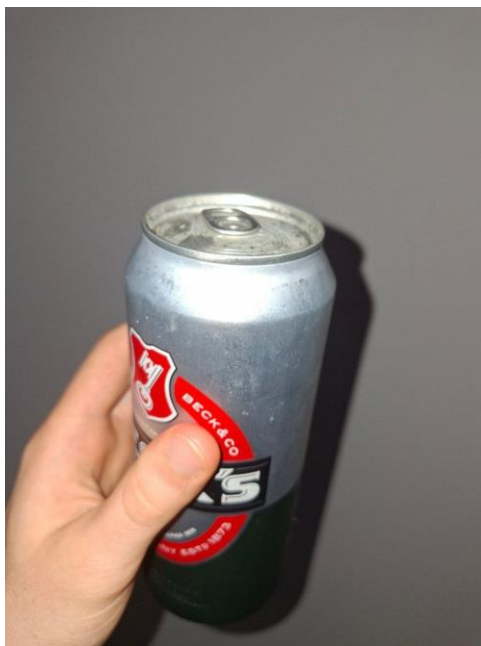


Рисунок 4.24 – Фото предмету з відбивною поверхнею (метал) за умов штучного освітлення

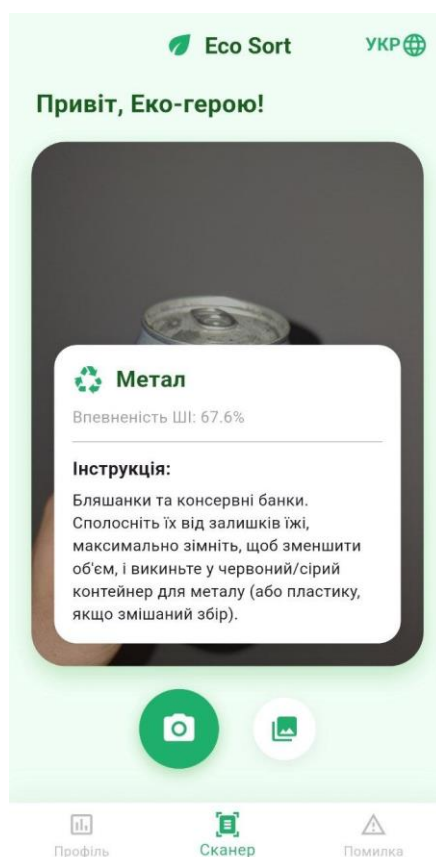


Рисунок 4.25 – Класифікація предмету з відбивною поверхнею (метал) за умов штучного освітлення



Рисунок 4.26 – Фото предмету з відбивною поверхнею (прозоре скло) за умов штучного освітлення

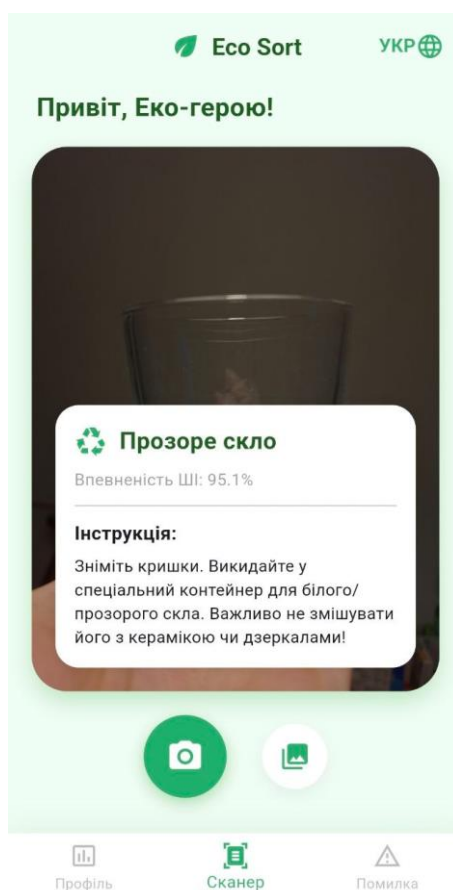


Рисунок 4.27 – Класифікація предмету з відбивною поверхнею (прозоре скло) за умов штучного освітлення



Рисунок 4.28 – Фото предмету з відбивною поверхнею (прозорий пластик) за умов штучного освітлення



Рисунок 4.29 – Невдала класифікація предмету з відбивною поверхнею (прозорий пластик) за умов штучного освітлення



Рисунок 4.30 – Фото предмету з відбивною поверхнею (зелене скло) за умов штучного освітлення



Рисунок 4.31 – Класифікація предмету з відбивною поверхнею (зелене скло) за умов штучного освітлення

4.4.3 Тестування механізму відхилення аномалій (Out-of-Distribution)

Критичним етапом натурного експерименту була перевірка роботи емпіричного порогу впевненості (Confidence Threshold), встановленого на сервері на рівні 60% (0.60). Для цього через мобільний застосунок було відскановано об'єкти, які не є відходами і не були представлені у навчальній вибірці: клавіатуру, чашку з кавою та власну руку (рисунок 4.32). В усіх випадках алгоритм `np.argmax()`, обробивши вихідний вектор Softmax-шару, видав максимальну ймовірність, що коливалася в межах 30–48%. Оскільки жоден клас не подолав бар'єр у 60%, серверна логіка коректно перехопила ці результати. Прапорець `is_trash` був переведений у стан `False`, а мобільному клієнту була повернута відповідь про помилку розпізнавання.



Рисунок 4.32 – Сканування предмету, що не є відходом

У результаті застосунок не видав користувачеві хибної інструкції з утилізації комп'ютерної миші, а натомість відобразив червону іконку попередження з текстом «Увага: Не вдалося розпізнати об'єкт. Спробуйте змінити ракурс або освітлення» (рисунок 4.33).



Рисунок 4.33 – Успішне спрацьовування механізму відхилення нецільових об'єктів (аномалій)

Підсумовуючи результати натурних випробувань, можна констатувати, що клієнт-серверна система класифікації відходів повністю виконує покладені на неї функції. Інтегрована архітектура MobileNetV2 забезпечує високу швидкість відклику (Latency), а розроблений інтерфейс користувача на базі Flutter ефективно компенсує апаратні та алгоритмічні недоліки через механізми зворотного зв'язку та інтуїтивні підказки. Впровадження порогу впевненості усунуло ризик надання екологічно шкідливих інструкцій у випадках фотографування нецільових об'єктів.

ВИСНОВКИ

Дана кваліфікаційна робота була присвячена розробці інтелектуальної клієнт-серверної системи класифікації твердих побутових відходів. На початку дослідження було проаналізовано предметну область та визначено ключові проблеми, з якими стикаються домогосподарства під час роздільного збору сміття. Встановлено, що відсутність швидкого доступу до локалізованих та візуально зрозумілих інструкцій є головним бар'єром для ефективного сортування, що обґрунтувало необхідність створення мобільного застосунку з інтеграцією технологій комп'ютерного зору.

У роботі було розглянуто сучасні архітектури згорткових нейронних мереж (CNN), оптимізовані для мобільних та периферійних пристроїв. Для реалізації системи було обрано архітектуру MobileNetV2 завдяки її балансу між високою точністю, низьким споживанням обчислювальних ресурсів та стабільністю при використанні у клієнт-серверних архітектурах. Під час роботи з набором даних було виявлено явище «парадоксу точності», зумовлене екстремальним дисбалансом класів (майже 47% вибірки складали класи одягу та взуття). Було проведено порівняльний експеримент між базовою моделлю на 12 класів та оптимізованою на 10 класів. Хоча 12-класова модель показала загальну точність 90%, вона мала низькі показники розпізнавання цільових відходів. Оптимізована 10-класова модель показала загальну точність 85%, проте продемонструвала значне зростання метрики F1-score для критично важливих екологічних категорій: від 0.88 до 0.96 для органіки, картону, паперу та скла. Саме оптимізовану 10-класову модель було інтегровано у фінальну систему.

Система складається з двох основних модулів: високопродуктивного REST API сервера, розробленого мовою Python з використанням фреймворку FastAPI, та кросплатформного мобільного застосунку, створеного за допомогою фреймворку Flutter. Серверна частина контейнеризована за допомогою Docker, що забезпечує її ізольованість та

готовність до розгортання на будь-якій хмарній платформі. Збереження історії сканувань та логування зворотного зв'язку реалізовано за допомогою бази даних SQLite та ORM-бібліотеки SQLAlchemy. Мобільний застосунок підтримує дві мови (українську та англійську), має модуль гейміфікації у вигляді еко-профілю з круговими діаграмами статистики та функцію відправлення звітів про помилки розпізнавання.

Під час тестування системи у реальних умовах було підтверджено високу ефективність розробленого механізму обробки аномалій (Out-of-Distribution). Завдяки встановленому порогу впевненості у 60%, система успішно відхиляє нецільові об'єкти, запобігаючи наданню хибних інструкцій. Дослідження впливу зовнішніх факторів показало, що модель ідеально працює в умовах природного денного освітлення (особливо для картону, паперу та органіки). При штучному освітленні спостерігається зниження впевненості для відбивних поверхонь (металу та пластику) через відблиски. Відповідно, для досягнення найкращих результатів користувачам рекомендується уникати різких тіней та сильних спалахів під час фотографування.

Очікується, що впровадження розробленого мобільного застосунку значно спростить процес щоденного сортування відходів для користувачів. Інтегрований механізм зворотного зв'язку дозволить безперервно збирати реальні дані для майбутнього донавчання моделі, що зробить розпізнавання ще більш точним та адаптованим до реальних побутових умов.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. What a waste 2.0: a global snapshot of solid waste management to 2050 / S. Kaza et al. Washington, DC: World Bank, 2018.
URL: <https://doi.org/10.1596/978-1-4648-1329-0> (date of access: 16.05.2026).
2. Towards the circular economy Vol. 1: an economic and business rationale for an accelerated transition. *Leading the Transition to a Circular Economy* | Ellen MacArthur Foundation.
URL: <https://www.ellenmacarthurfoundation.org/towards-the-circular-economy-vol-1-an-economic-and-business-rationale-for-an> (date of access: 16.05.2026).
3. Quantitative assessment of distance to collection point and improved sorting information on source separation of household waste / K. Rousta et al. *Waste management*. 2015. Vol. 40. P. 22–30.
URL: <https://doi.org/10.1016/j.wasman.2015.03.005> (date of access: 16.05.2026).
4. National recycling strategy | US EPA. *US EPA*.
URL: <https://www.epa.gov/circulareconomy/national-recycling-strategy> (date of access: 16.05.2026).
5. A distributed architecture for smart recycling using machine learning / D. Ziouzos et al. *Future internet*. 2020. Vol. 12, no. 9. P. 141.
URL: <https://doi.org/10.3390/fi12090141> (date of access: 16.05.2026).
6. Міністерство захисту довкілля та природних ресурсів України – офіційний сайт. *Міністерство захисту довкілля та природних ресурсів України*. URL: <https://mepr.gov.ua/> (дата звернення: 16.05.2026).
7. From pollution to solution: a global assessment of marine litter and plastic pollution. *United Nations Environment Programme*.
URL: <https://www.unep.org/resources/pollution-solution-global-assessment-marine-litter-and-plastic-pollution> (date of access: 16.05.2026).
8. Municipal solid waste composition: sampling methodology, statistical analyses, and case study evaluation / M. E. Edjabou et al. *Waste management*.

2015. Vol. 36. P. 12–23. URL: <https://doi.org/10.1016/j.wasman.2014.11.009> (date of access: 17.05.2026).

9. Waste recycling in Europe. *European Environment Agency*. URL: <https://www.eea.europa.eu/en/analysis/indicators/waste-recycling-in-europe> (date of access: 16.05.2026).

10. Payne J., McKeown P., Jones M. D. A circular economy approach to plastic waste. *Polymer degradation and stability*. 2019. Vol. 165. P. 170–181. URL: <https://doi.org/10.1016/j.polymdegradstab.2019.05.014> (date of access: 17.05.2026).

11. Hopewell J., Dvorak R., Kosior E. Plastics recycling: challenges and opportunities. *Philosophical transactions of the royal society B: biological sciences*. 2009. Vol. 364, no. 1526. P. 2115–2126. URL: <https://doi.org/10.1098/rstb.2008.0311> (date of access: 17.05.2026).

12. Knickmeyer D. Social factors influencing household waste separation: a literature review on good practices to improve the recycling performance of urban areas. *Journal of cleaner production*. 2020. Vol. 245. P. 118605. URL: <https://doi.org/10.1016/j.jclepro.2019.118605> (date of access: 16.05.2026).

13. Gundupalli S. P., Hait S., Thakur A. A review on automated sorting of source-separated municipal solid waste for recycling. *Waste management*. 2017. Vol. 60. P. 56–74. URL: <https://doi.org/10.1016/j.wasman.2016.09.015> (date of access: 16.05.2026).

14. Сортуй – мобільний застосунок для зручного сортування сміття. *Сортуй*. URL: <https://sortui.org.ua/> (дата звернення: 16.05.2026).

15. Litterati. The global community eradicating litter. *Litterati*. URL: <https://www.litterati.org/> (date of access: 16.05.2026).

16. Google lens - search what you see. *Google Lens*. URL: <https://lens.google/> (date of access: 16.05.2026).

17. Kunwar S. MWaste: an app that uses deep learning to manage household waste. *Clean technologies and recycling*. 2023. Vol. 3, no. 3. P. 119–133. URL: <https://doi.org/10.3934/ctr.2023008> (date of access: 16.05.2026).

18. Garbage classification (12 classes). *Kaggle*. URL: <https://www.kaggle.com/datasets/mostafaabla/garbage-classification> (date of access: 16.05.2026).
19. Proença P. F., Simões P. TACO: trash annotations in context for litter detection. *arXiv*. 2019. URL: <https://doi.org/10.48550/arXiv.2003.06975> (date of access: 16.05.2026).
20. Dodge S., Karam L. Understanding how image quality affects deep neural networks. *2016 eighth international conference on quality of multimedia experience (QoMEX)*, Lisbon, Portugal, 6–8 June 2016. 2016. URL: <https://doi.org/10.1109/qomex.2016.7498955> (date of access: 16.05.2026).
21. Automatic image-based waste classification / V. Ruiz et al. *From bioinspired systems and biomedical applications to machine learning*. Cham, 2019. P. 422–431. URL: https://doi.org/10.1007/978-3-030-19651-6_41 (date of access: 16.05.2026).
22. MobileNets: efficient convolutional neural networks for mobile vision applications / A. G. Howard et al. *arXiv*. 2017. URL: <https://doi.org/10.48550/arXiv.1704.04861> (date of access: 16.05.2026).
23. Hendrycks D., Gimpel K. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv*. 2018. URL: <https://doi.org/10.48550/arXiv.1610.02136> (date of access: 16.05.2026).
24. Bobulski J., Kubanek M. Deep learning for plastic waste classification system. *Applied computational intelligence and soft computing*. 2021. Vol. 2021. P. 1–7. URL: <https://doi.org/10.1155/2021/6626948> (date of access: 16.05.2026).
25. Lu W., Chen J. Computer vision for solid waste sorting: a critical review of academic research. *Waste management*. 2022. Vol. 142. P. 29–43. URL: <https://doi.org/10.1016/j.wasman.2022.02.009> (date of access: 16.05.2026).
26. Awe O., Mengistu R., Sreedhar V. Final report: smart trash net: waste localization and classification. *CS229: Machine Learning*. 2017.