Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____Комп'ютерна інженерія управління_____
(повна назва)
Кафедра _____Комп'ютерних інтелектуальних технологій та систем_____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти _____перший (бакалаврський)_____
Інтелектуальна система класифікації та
розпізнавання сміття для подальшого сортування
на основі глибокого навчання
(тема)

Виконав:
здобувач ___IV___ року навчання,
групи _____КІУКІ-21-10_____
_____Микола ГРИШИН_____
(власне ім'я, прізвище)

Спеціальність _____
_____123-Комп'ютерна Інженерія_____
(код і повна назва спеціальності)
Тип програми ____освітньо-професійна_____
Освітня програма _____
_____Комп'ютерна інженерія_____
(повна назва освітньої програми)

Керівник __ас. каф. КІТС Кирило ОЛІЙНИК_
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри _____ __Олег РУДЕНКО____
(підпис) (власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерна інженерія управління

Кафедра Комп'ютерних інтелектуальних технологій та систем

Рівень вищої освіти перший (бакалаврський)

Спеціальність 123 Комп'ютерна інженерія

Тип програми Освітньо-професійна

Освітня програма Комп'ютерна інженерія

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 2025 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

Здобувачу Гришину Миколі Миколайовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Інтелектуальна система класифікації та розпізнавання сміття для подальшого сортування на основі глибокого навчання

затверджена наказом університету від 21.05.2025 р. № 399 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 24 червня 2025 р.

3. Вихідні дані до роботи
Нейромережа YOLOv5
Мова програмування Python,
Telegram bot

4. Перелік питань, що потрібно опрацювати в роботі
Аналіз предметної області та постановка задачі класифікації відходів.
Підготовка та анотація датасету зображень для навчання моделі.
Розробка та навчання нейронної мережі для детекції та класифікації відходів.
Інтеграція моделі з Telegram-ботом для користувачів.
Оцінка якості роботи системи, аналіз результатів та перспективи розвитку.
Висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) 10 слайдів

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Вибір та затвердження теми проекту	26.05.2025-27.05.2025	Виконано
2	Аналіз предметної області, постановка задачі, вибір технологій	28.05.2025 -31.05.2025	Виконано
3	Підготовка та очищення датасету, анотація зображень	01.06.2025 -03.06.2025	Виконано
4	Розробка та навчання моделі YOLOv5	03.06.2025 -07.06.2025	Виконано
5	Інтеграція моделі з Telegram-ботом	08.06.2025 -10.06.2025	Виконано
6	Тестування та оцінка якості системи	11.06.2025 -13.06.2025	Виконано
7	Оформлення пояснювальної записки	14.06.2025 -17.06.2025	Виконано
8	Перевірка проекту керівником, доопрацювання	18.06.2025 -22.06.2025	Виконано

Студент _____
(підпис)

Керівник роботи _____ ас. Олійник К.О.
(підпис) (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи бакалавра: 69 с., 21 рис., 2 дод., 19 джерел.

КОМП'ЮТЕРНИЙ ЗІР, СОРТУВАННЯ ВІДХОДІВ, НЕЙРОННІ МЕРЕЖІ, YOLO, КЛАСИФІКАЦІЯ, ТЕЛЕГРАМ-БОТ, PYTORCH

Метою кваліфікаційної роботи є розробка системи автоматичного розпізнавання та класифікації побутових відходів на зображеннях із використанням методів комп'ютерного зору та глибокого навчання, а також інтеграція цієї системи з Telegram-ботом для спрощення доступу користувачів.

У ході виконання кваліфікаційної роботи було проведено аналіз сучасних підходів до сортування відходів, обґрунтовано вибір датасету та моделі YOLOv5, розроблено та реалізовано програмне забезпечення для автоматичної детекції шести категорій відходів (пластик, папір, метал, картон, скло, інше), виконано очистку та анотування даних, здійснено навчання нейронної мережі, проведено оцінку якості розробленої системи та порівняння результатів до і після очищення датасету. Окрему увагу приділено інтеграції моделі з Telegram-ботом, що дозволяє отримувати класифікацію об'єктів безпосередньо з мобільного пристрою. Робота завершується аналізом помилок, визначенням слабких класів і рекомендаціями щодо подальшого вдосконалення системи.

ABSTRACT

The explanatory note of the qualification work includes: 69 pages, 21 figures, and 19 referenices.

COMPUTER VISION, WASTE SORTING, NEURAL NETWORKS, YOLO, CLASSIFICATION, TELEGRAM BOT, PYTORCH

The purpose of the qualification work is the development of a system for automatic recognition and classification of household waste in images using computer vision and deep learning methods, as well as the integration of this system with a Telegram bot to simplify user access.

During the qualification work, an analysis of modern approaches to waste sorting was conducted, the choice of dataset and YOLOv5 model was justified, and software for automatic detection of six categories of waste (plastic, paper, metal, cardboard, glass, other) was developed and implemented. Data cleaning and annotation were performed, the neural network was trained, the quality of the developed system was evaluated, and the results before and after dataset cleaning were compared. Special attention was paid to integrating the model with a Telegram bot, which allows users to classify objects directly from a mobile device. The work concludes with an analysis of errors, identification of weak classes, and recommendations for further system improvement

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ ……..……8
ВСТУП..... 9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ..... 11
1.1 Сучасний стан проблеми сортування побутових відходів 11
1.2 Огляд традиційних методів сортування сміття..... 12
1.3 Переваги використання комп’ютерного зору та штучного інтелекту... 14
1.4 Постановка задачі..... 15
1.5 Вимоги до розроблюваної системи 16
2 АНАЛІЗ ВИКОРИСТОВУВАНИХ ТЕХНОЛОГІЙ……..……18
2.1 Методи комп’ютерного зору в задачах детекції та класифікації об’єктів. 18
2.2 Глибоке навчання і нейронні мережі 20
2.2.1 Згорткові нейронні мережі (CNN)..... 22
2.2.2 Алгоритми детекції об'єктів (YOLO)..... 24
2.2.3 Огляд моделей сімейства YOLO 26
2.3 Інструменти для реалізації проекту 28
2.3.1 Python як мова розробки..... 29
2.3.2 PyTorch як фреймворк для глибокого навчання..... 30
2.3.3 Бібліотека Python-Telegram-Bot..... 32
2.4 Аналіз вибраного датасету TACO..... 33
3 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ..... 36
3.1 Архітектура системи..... 36
3.2 Процес підготовки та анотації даних..... 37
3.3 Побудова та навчання моделі YOLOv5 40
3.4 Інтеграція моделі з Telegram-ботом..... 42
4 АНАЛІЗ РЕЗУЛЬТАТІВ ТА ЯКІСНА ОЦІНКА МОДЕЛІ..... 45
4.1 Оцінка якості розпізнавання на валідаційному датасеті..... 45
4.2 Метрики оцінки ефективності (Precision, Recall, mAP)..... 46

4.3 Аналіз впливу підготовки даних на результати моделі 49
4.4 Аналіз помилок та слабких класів..... 51
ВИСНОВКИ..... 56
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ..... 58
ДОДАТОК А..... 60
ДОДАТОК Б 66

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

YOLO — You Only Look Once (алгоритм одноетапної детекції об'єктів на зображенні)

CNN — Convolutional Neural Network (згорткова нейронна мережа)

R-CNN — Region-based Convolutional Neural Network (метод детекції об'єктів із виділенням регіонів інтересу)

Bounding Box — Обмежувальний прямокутник для локалізації об'єкта на зображенні

ReLU — Rectified Linear Unit (функція активації у нейронних мережах)

CPU — Central Processing Unit (центральний процесор)

GPU — Graphics Processing Unit (графічний процесор, використовується для пришвидшення обчислень у нейромережах)

TACO — Trash Annotations in Context (відкритий датасет із розміченими зображеннями сміття)

API — Application Programming Interface (інтерфейс прикладного програмування)

mAP — mean Average Precision (середнє значення точності для задачі багатокласової детекції об'єктів)

IoU — Intersection over Union (метрика схожості між передбаченим та реальним боксом)

ВСТУП

Сортування побутових відходів є одним з найважливіших завдань сучасного суспільства. З кожним роком людство стикається з усе більшими проблемами через накопичення сміття, яке негативно впливає на навколишнє середовище та здоров’я людей. Незважаючи на активне впровадження різних систем сортування, більшість процесів залишаються неефективними та потребують великих витрат часу і людських ресурсів.

На сьогоднішній день найпоширенішим методом сортування відходів є ручне сортування, що має багато недоліків. По-перше, це високі витрати на оплату праці працівників, а також ризики, пов’язані зі здоров’ям людини через постійний контакт з відходами. По-друге, ручне сортування має низьку ефективність через суб’єктивність людського фактора та велику кількість помилок.

Саме тому використання сучасних інтелектуальних технологій, зокрема комп'ютерного зору та штучного інтелекту, стає актуальним рішенням для автоматизації та покращення процесу сортування відходів. Такі системи дозволяють зменшити людський фактор, прискорити процес сортування та підвищити його точність.

Метою даного проекту є розробка інтелектуальної системи для автоматичного розпізнавання та класифікації сміття з використанням глибокого навчання, яка дозволить значно покращити та автоматизувати процес сортування побутових відходів. Основними завданнями проекту є:

- Аналіз існуючих методів сортування сміття;
- Вивчення та вибір відповідних технологій глибокого навчання;
- Створення нейронної мережі на основі моделі YOLOv5 для детекції та класифікації сміття;
- Реалізація зручного інтерфейсу користувача у вигляді Telegram-бота.

Очікується, що запропонована система зможе швидко та точно визначати тип сміття на зображеннях, що сприятиме більш ефективному сортуванню та покращенню екологічної ситуації.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Сучасний стан проблеми сортування побутових відходів

Проблема накопичення побутових відходів сьогодні є глобальною та актуальною для всього світу. Кількість сміття, яке генерують люди, постійно зростає, а ефективних рішень для його переробки та утилізації не вистачає. Основні труднощі виникають через низький рівень свідомості населення щодо сортування відходів, недостатню кількість та якість обладнання для сортування і переробки, а також через недосконаліть законодавчих механізмів.

У більшості країн, включаючи Україну, переважає традиційна модель поводження зі сміттям, коли значна частина відходів потрапляє на полігони, які швидко заповнюються та створюють екологічні загрози. Наприклад, в Україні лише близько 5% сміття переробляється, тоді як в країнах Європейського Союзу цей показник досягає 60-70% [1].

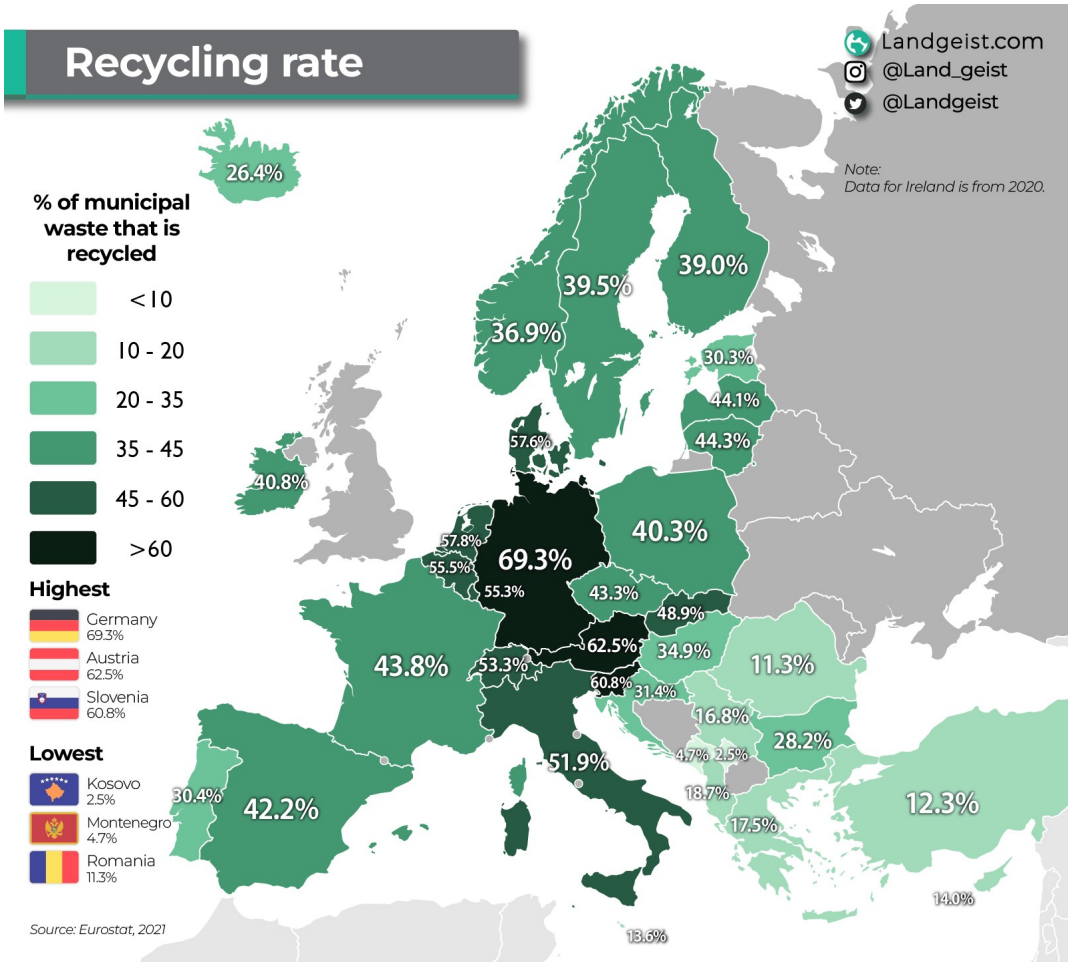

Рисунок 1.1 - Карта Європи з рівнями переробки сміття

Сучасні підходи до вирішення проблеми включають впровадження автоматизованих систем, які базуються на технологіях комп'ютерного зору та глибокого навчання. Такі технології дозволяють значно покращити точність сортування та зменшити негативний вплив на навколишнє середовище.

1.2 Огляд традиційних методів сортування сміття

Традиційні методи сортування сміття можна умовно поділити на ручні, механізовані та комбіновані.

Ручне сортування є найпростішим, але й найбільш поширеним методом. Його перевагою є можливість ретельного сортування різних типів відходів,

проте він характеризується високими затратами праці, низькою продуктивністю та ризиками для здоров’я працівників. Крім того, ручне сортування має значні похибки, оскільки багато типів відходів важко ідентифікувати лише за зовнішнім виглядом [1].

Рисунок 1.2 - Конвеєрна система для ручного сортування сміття

Механізовані методи включають використання спеціалізованих автоматичних систем та обладнання, такого як сортувальні конвеєри, магнітні сепаратори для металу, повітряні сепаратори для паперу та пластику, а також різноманітні сенсорні системи. Ці методи дозволяють значно підвищити швидкість та обсяг переробки, проте часто потребують значних інвестицій у обладнання, постійного технічного обслуговування та мають обмеження щодо точності визначення типу відходів.

Комбіновані методи поєднують переваги ручного та механізованого сортування. Зазвичай вони передбачають первинну автоматичну обробку відходів з подальшим ручним сортуванням тих типів сміття, які складно ідентифікувати автоматично. Хоча комбіновані методи мають кращу ефективність порівняно з іншими, вони також характеризуються високими затратами на реалізацію та підтримку.

Таким чином, традиційні методи сортування мають низку суттєвих недоліків, які можна подолати шляхом впровадження новітніх технологій, зокрема штучного інтелекту та комп'ютерного зору.

1.3 Переваги використання комп’ютерного зору та штучного інтелекту

Технології комп'ютерного зору та штучного інтелекту пропонують значні переваги для автоматизації процесу сортування сміття, порівняно з традиційними методами. Вони забезпечують значну точність розпізнавання різних типів відходів завдяки сучасним алгоритмам глибокого навчання, які здатні аналізувати великі масиви даних та навчатися на основі прикладів [2].

Застосування комп'ютерного зору дозволяє автоматично розпізнавати та класифікувати об'єкти за їхніми візуальними характеристиками, такими як форма, розмір, колір та текстура. Це робить процес сортування більш точним і надійним, знижуючи ймовірність помилок, які часто трапляються під час ручного сортування.

Рисунок 1.3 – Використання ШІ для детекції сміття

Крім того, автоматичні системи можуть працювати безперервно, що забезпечує значне підвищення продуктивності. Також вони мінімізують

контакт людини зі шкідливими речовинами, що суттєво знижує ризики для здоров'я працівників. Завдяки автоматизації зменшуються витрати на оплату праці та експлуатацію обладнання [3].

Інтелектуальні системи мають можливість постійно вдосконалюватися за рахунок накопичення нових даних і додаткового навчання нейронних мереж. Це дозволяє адаптувати систему до змін у складі відходів та покращувати її точність та ефективність з часом.

Таким чином, впровадження технологій комп'ютерного зору та штучного інтелекту є перспективним рішенням, яке може істотно покращити процес сортування сміття, зробивши його більш ефективним, економічно вигідним і безпечним для довкілля та людей.

1.4 Постановка задачі

Основною задачею цього проекту є створення інтелектуальної системи для автоматичного розпізнавання та класифікації сміття. Проект передбачає розробку нейронної мережі на основі моделі YOLOv5, яка буде здатна ефективно і точно детектувати та класифікувати різні типи відходів: пластик, скло, папір, картон, метал та інші відходи [4]. Важливою складовою проекту є підготовка якісного та репрезентативного набору даних, для чого планується використати відкритий датасет TACO, що містить значну кількість анотованих зображень сміття різних категорій.

Для навчання моделі будуть застосовані методи аугментації даних, що дозволить розширити набір навчальних прикладів та покращити узагальнюючу здатність нейронної мережі. Окрему увагу буде приділено вибору параметрів та налаштуванню нейромережі, аби досягти максимальної ефективності та точності її роботи.

Ще одним важливим завданням є створення зручного та доступного інтерфейсу користувача у вигляді Telegram-бота, який дозволить кінцевим

користувачам взаємодіяти з системою просто й ефективно. Telegram-бот матиме функціонал для завантаження зображень, їх автоматичної обробки та повернення результату класифікації у зрозумілій формі. Це значно спростить використання системи та сприятиме її широкому застосуванню.

Реалізація проекту дозволить автоматизувати процес сортування відходів, суттєво скоротити витрати на обслуговування сортувальних ліній, підвищити ефективність переробки сміття та позитивно вплинути на екологічну ситуацію в регіонах впровадження.

1.5 Вимоги до розроблюваної системи

Для успішної реалізації проекту необхідно чітко визначити та сформулювати вимоги до розроблюваної системи. Основними вимогами є:

– Точність класифікації: система повинна мати високу точність розпізнавання та класифікації різних типів сміття, а саме: пластик, скло, папір, картон, метал та інші відходи. Точність повинна складати не менше 90% для кожної категорії;

– Швидкість роботи: система має працювати швидко і стабільно, забезпечуючи обробку одного зображення за час не більше кількох секунд;

– Простота використання: інтерфейс користувача у вигляді Telegram-бота повинен бути інтуїтивно зрозумілим, зручним і доступним для широкого кола користувачів без спеціальної технічної підготовки;

– Надійність роботи: система повинна стабільно працювати в умовах реального часу, з мінімальною кількістю збоїв та помилок;

– Масштабованість: система має бути легко масштабованою та адаптивною до збільшення обсягів даних та кількості категорій відходів;

– Безпека та конфіденційність: зображення, що надсилаються користувачами, повинні оброблятися безпечним способом, що забезпечує конфіденційність даних;

– Зручність інтеграції: система повинна бути готова до інтеграції з існуючими процесами та обладнанням сортувальних станцій, забезпечуючи простоту впровадження та мінімальні витрати на інтеграцію.

Таким чином, сформульовані вимоги гарантуватимуть, що створена система буде ефективною, зручною та відповідатиме сучасним стандартам якості та продуктивності [4]

2. АНАЛІЗ ВИКОРИСТОВУВАНИХ ТЕХНОЛОГІЙ

2.1 Методи комп’ютерного зору в задачах детекції та класифікації об’єктів

Комп’ютерний зір - це одна з найбільш динамічно розвиваючихся галузей штучного інтелекту, яка дозволяє комп’ютерам отримувати, аналізувати та інтерпретувати інформацію з візуальних даних, таких як зображення та відео. Основна мета комп’ютерного зору - навчити комп’ютер розпізнавати, класифікувати та розуміти вміст зображень так само, як це робить людина. В сучасному світі технології комп’ютерного зору активно використовуються у багатьох сферах, включаючи медицину, автомобілебудування, безпеку, виробництво, сільське господарство та екологію.

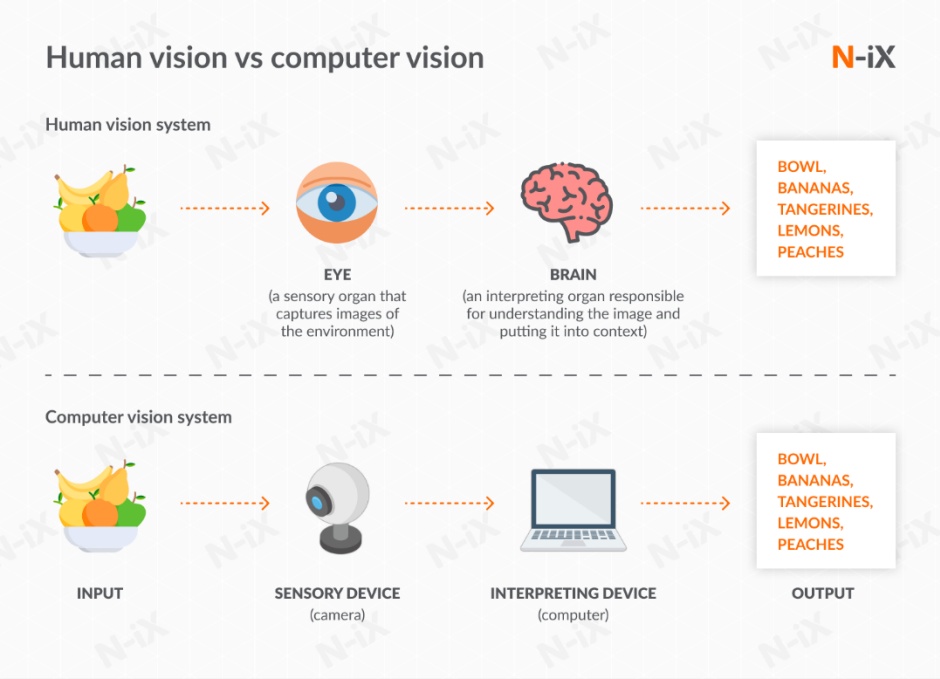

Рисунок 2.1 - Схема нейронної мережі

В задачах сортування сміття найбільшу цінність представляють такі напрями комп'ютерного зору, як детекція об'єктів і класифікація зображень. Розглянемо основні підходи, які використовуються у цих задачах.

Класифікація зображень - це процес, у якому система отримує зображення на вході та визначає, до якої категорії належить об'єкт, що зображений на фото. Для задач сортування сміття це може бути визначення типу матеріалу (наприклад, пластик чи метал). Традиційно для цього застосовувалися класичні алгоритми машинного навчання (наприклад, метод опорних векторів, дерева рішень), однак сучасні системи базуються переважно на нейронних мережах, особливо на згорткових нейронних мережах (CNN), які демонструють набагато вищу точність [5].

Детекція об'єктів - це складніший процес, який дозволяє не тільки визначити, що зображено на фото, а й вказати точне розташування об'єктів у вигляді прямокутної рамки (bounding box). Для задач сортування сміття це дозволяє визначати на одному зображенні відразу кілька різних типів сміття. Сучасні алгоритми детекції об'єктів (наприклад, YOLO, SSD, Faster R-CNN) можуть в режимі реального часу ідентифікувати та класифікувати об'єкти на зображенні з високою точністю.

Сегментація зображень - ще один важливий напрям, який дозволяє виділити межі кожного об'єкта на зображенні з точністю до кожного пікселя. Це особливо корисно в ситуаціях, коли об'єкти мають складну форму або перекривають один одного. У практиці сортування сміття сегментація допомагає точніше визначати різні типи матеріалів у складних композиціях.

Розвиток комп'ютерного зору став можливий завдяки використанню великих об'ємів даних (датасетів), вдосконаленню архітектур нейронних мереж, а також збільшенню обчислювальних потужностей. Важливою складовою успішної роботи систем комп'ютерного зору є попередня обробка даних - нормалізація, зміна розміру, аугментація зображень, що дозволяє зробити навчання більш якісним і запобігти перенавчанню.

Для вирішення задач детекції та класифікації сміття найчастіше використовують згорткові нейронні мережі (CNN) як основний інструмент для автоматичного вилучення ознак з візуальних даних. Сучасні архітектури CNN складаються з кількох шарів згортки, pooling-шарів і повнозв'язних шарів, що дозволяє поступово зменшувати розмірність даних і витягати найбільш важливі ознаки. Для задач детекції додатково використовуються спеціалізовані шари, які дозволяють визначати координати об'єктів на зображенні.

Окремо варто виділити алгоритми типу YOLO (You Only Look Once), які дозволяють здійснювати детекцію об'єктів надзвичайно швидко й ефективно навіть на слабких пристроях. Це робить такі підходи дуже актуальними для практичного застосування в екологічних проектах, таких як автоматичне сортування сміття. Серед інших популярних підходів можна відзначити Faster R-CNN (який має вищу точність, але нижчу швидкість) та SSD (Single Shot MultiBox Detector), який забезпечує баланс між швидкістю та точністю [6].

Загалом, використання методів комп'ютерного зору у задачах сортування сміття дозволяє суттєво підвищити рівень автоматизації, знизити витрати та досягти нової якості переробки відходів.

2.2 Глибоке навчання і нейронні мережі

Глибоке навчання (Deep Learning) - це сучасний напрям машинного навчання, який базується на використанні штучних нейронних мереж з багатьма шарами. Саме глибоке навчання зробило справжню революцію в галузі комп'ютерного зору, дозволивши досягти високої точності у задачах розпізнавання образів, детекції об'єктів, класифікації зображень та багатьох інших задачах.

У традиційному машинному навчанні ефективність моделей значною мірою залежала від якості ручного виділення ознак - спеціалісти самостійно

обирали, які характеристики (features) варто враховувати для побудови моделі. З появою глибокого навчання цей процес став автоматизованим: багатошарові нейронні мережі навчаються самостійно виділяти та комбінувати необхідні ознаки із сирих даних.

Глибокі нейронні мережі складаються з великої кількості шарів - зазвичай це десятки чи навіть сотні шарів, які виконують складні послідовні перетворення вхідних даних. Кожен шар навчається виділяти все більш абстрактні ознаки. На початкових шарах це можуть бути прості лінії та кути, а на більш глибоких - складні структури, текстури чи навіть цілі об’єкти [5].

Однією з найпопулярніших архітектур для роботи з візуальними даними є згорткові нейронні мережі (CNN). Вони чудово підходять для задач розпізнавання та класифікації об’єктів, оскільки здатні ефективно обробляти зображення завдяки використанню згорткових фільтрів.

Глибоке навчання дозволяє будувати системи, які:

– демонструють вищу точність і стабільність у порівнянні з класичними алгоритмами;

– легко адаптуються до нових типів даних і завдань;

– здатні працювати із великими наборами даних, що є ключовим для сучасних задач комп’ютерного зору.

У рамках цього проекту глибоке навчання використовується для побудови системи автоматичної детекції та класифікації сміття. Завдяки цьому підходу можна не лише ідентифікувати окремі види відходів, але й відрізняти схожі за зовнішнім виглядом об’єкти, що значно підвищує якість і ефективність сортування.

Детальніше далі будуть розглянуті основні типи нейронних мереж, їх структура та принцип роботи, а також особливості алгоритмів детекції об’єктів, які є найбільш актуальними для автоматизації процесу сортування сміття.

2.2.1 Згорткові нейронні мережі (CNN)

Згорткові нейронні мережі (Convolutional Neural Networks, CNN) - це клас нейронних мереж, який став стандартом для роботи із зображеннями та іншими двовимірними даними. CNN розроблені таким чином, щоб автоматично знаходити важливі ознаки на зображеннях та використовувати їх для класифікації, детекції чи сегментації об’єктів [7].

Основна ідея CNN полягає у використанні згорткових шарів (convolutional layers), які застосовують спеціальні фільтри (ядра згортки) до зображення. Кожен такий фільтр "сканує" зображення, виявляючи прості ознаки (лінії, кути) на перших шарах і все складніші (текстури, частини об'єктів) на глибших шарах мережі. Результати згорток потім передаються далі по мережі та комбінуються, щоб утворити загальну картину.

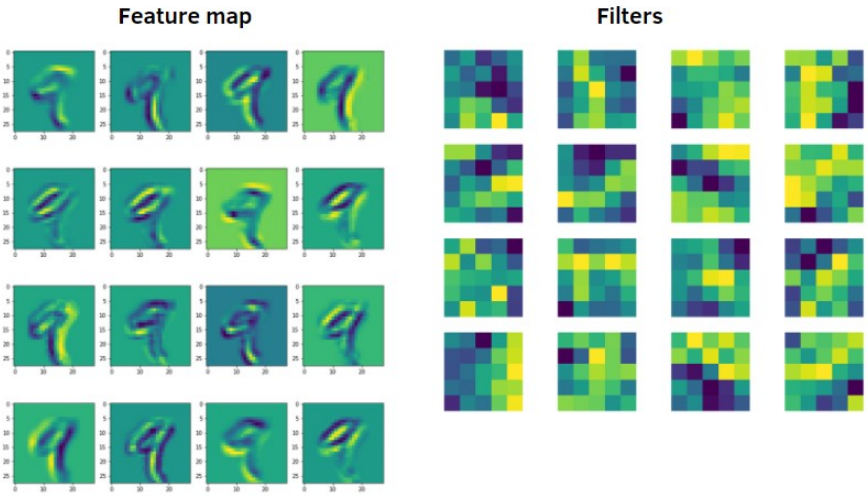

Рисунок 2.2 - Візуалізація фільтрів та feature maps.

Стандартна архітектура CNN складається з таких основних шарів:

– Згортковий шар (Convolutional Layer): виділяє ключові ознаки з вхідних даних за допомогою фільтрів;

– Шар підвибірки (Pooling Layer): зменшує розмірність даних, зберігаючи найбільш важливу інформацію і знижуючи обчислювальні витрати;

– Нелінійний шар активації (наприклад, ReLU): додає нелінійність, що дозволяє моделі краще апроксимувати складні функції;

– Повнозв’язний шар (Fully Connected Layer): використовується на фінальних етапах для прийняття рішення щодо класифікації об’єкта.

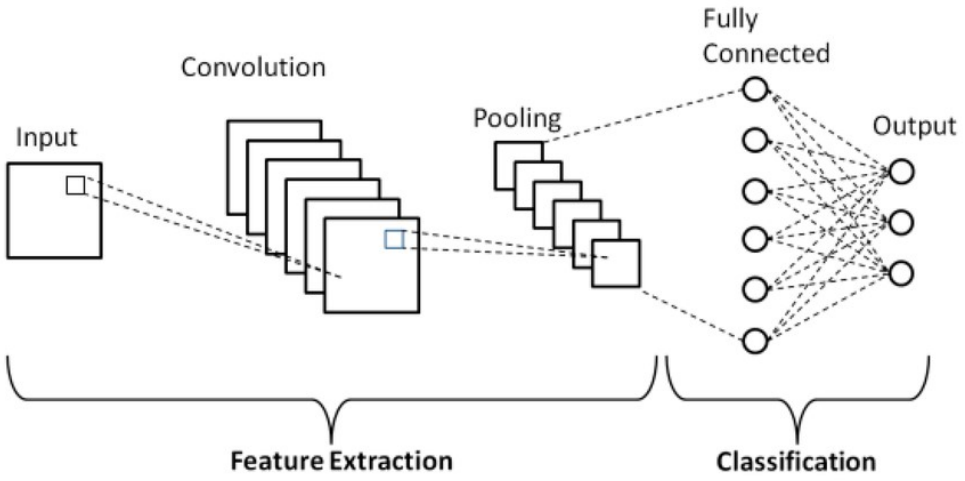

Рисунок 2.3 - Структурна схема згорткової нейронної мережі (CNN)

Навчання CNN відбувається за допомогою алгоритму зворотного поширення помилки (backpropagation), де мережа поступово коригує свої ваги на основі отриманих результатів. Перевагою CNN є здатність працювати без ручного виділення ознак - мережа "вчиться" самостійно, аналізуючи великі обсяги зображень.

Згорткові нейронні мережі є основою для багатьох сучасних алгоритмів комп’ютерного зору, таких як системи розпізнавання облич, класифікації медичних знімків, автоматичного керування автомобілями, а також детекції та сортування сміття. Їхня універсальність, ефективність та здатність працювати з великими даними робить CNN незамінними для сучасних екологічних проектів.

У нашому проекті саме CNN використовується як базова архітектура для побудови моделі детекції сміття, оскільки вони забезпечують високу точність

та швидкість обробки навіть при великій кількості різноманітних об’єктів на зображеннях [8].

2.2.2 Алгоритми детекції об'єктів (YOLO)

Детекція об'єктів - це задача комп’ютерного зору, яка полягає не лише у визначенні класу об'єкта на зображенні, а й у знаходженні його точного розташування. Серед різноманітних алгоритмів детекції найбільшою популярністю користуються методи сімейства YOLO (You Only Look Once).

YOLO - це сучасний підхід до детекції об’єктів, що відрізняється високою швидкістю роботи та здатністю обробляти зображення в реальному часі. Основна ідея YOLO полягає в тому, щоб «дивитися» на зображення лише один раз: вся картинка розбивається на сітку, і для кожної частини визначається, чи є в ній об’єкт, а також його клас і координати.

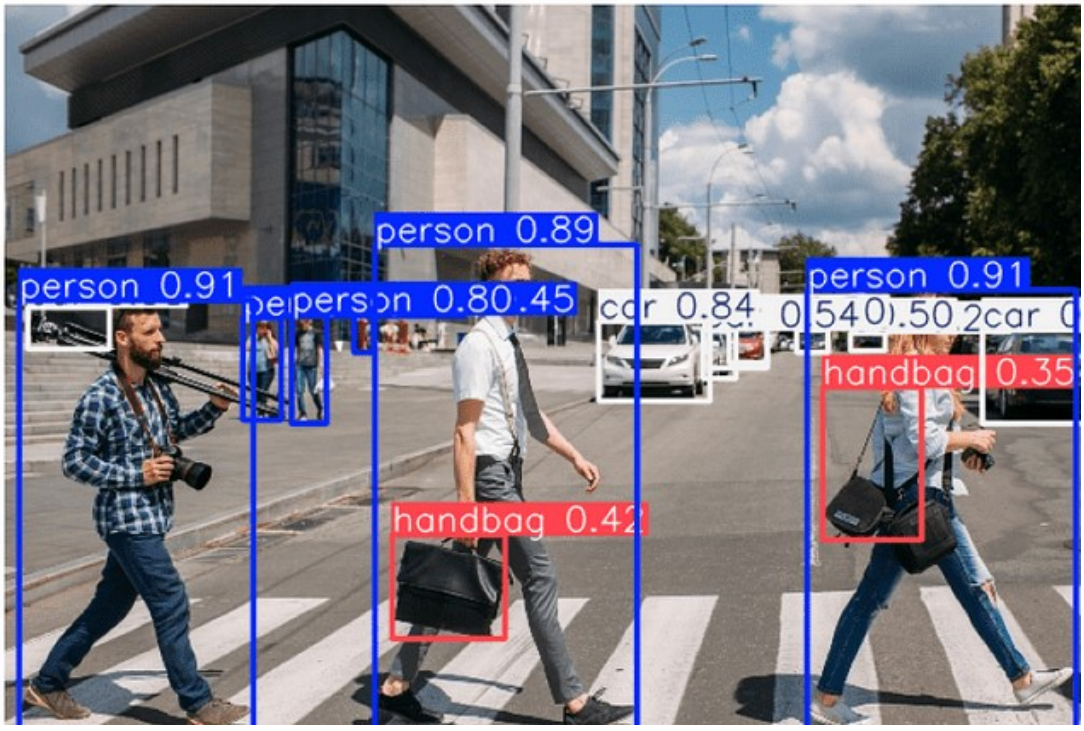

Рисунок 2.4 – Приклад роботи YOLO

Архітектура YOLO базується на згорткових нейронних мережах. Вона приймає зображення на вхід, обробляє його через кілька згорткових та інших шарів, після чого одразу видає координати рамки (bounding box) і клас об'єкта. Це забезпечує значне прискорення у порівнянні з традиційними підходами, які спочатку пропонували кандидати на об'єкти, а потім окремо класифікували їх [7].

Переваги YOLO:

- Висока швидкість обробки (до 45 і більше кадрів на секунду для сучасних моделей);
- Можливість використання на пристроях із обмеженими ресурсами;
- Відмінна точність для задач з багатьма об’єктами;
- Універсальність: YOLO застосовується для автомобільних систем безпеки, відеоспостереження, медичних задач, екології тощо.

Існує кілька версій YOLO: від перших експериментальних (YOLOv1) до сучасних вдосконалених моделей (YOLOv4, YOLOv5, YOLOv8). Кожна нова версія відрізняється оптимізацією архітектури, збільшенням точності та швидкості роботи.

У нашому проекті використовується YOLOv5, яка є однією з найпопулярніших і доступних для використання версій. Вона підтримується спільнотою, має відкритий код і добре документована, що дозволяє швидко запускати та тренувати власні моделі детекції об’єктів.

Загалом, використання YOLO у сфері сортування сміття дає змогу автоматично і швидко виявляти різні типи відходів навіть у складних сценах з багатьма об’єктами, підвищуючи точність і ефективність усього процесу [9].

2.2.3 Огляд моделей сімейства YOLO

Сімейство моделей YOLO (You Only Look Once) - це набір сучасних алгоритмів для детекції об’єктів, які з кожною новою версією стають все більш універсальними, швидкими та точними. Розвиток цих моделей суттєво вплинув на всю галузь комп’ютерного зору, адже вони дозволяють отримувати результати практично в реальному часі навіть на звичайних комп’ютерах і пристроях з обмеженими ресурсами. Зараз YOLO застосовується у відеоспостереженні, робототехніці, медицині, автомобільних системах, промисловості та навіть у мобільних додатках.

YOLOv1 - це перша реалізація, представлена у 2016 році дослідником Джозефом Редмоном. Основна ідея YOLOv1 полягала у тому, що мережа розбиває зображення на фіксовану сітку, і для кожної її клітинки одночасно прогнозує, чи є в ній об’єкт, координати його рамки та клас. Це забезпечило проривну швидкість (десятки кадрів на секунду), але водночас було й чимало обмежень - модель не дуже добре справлялася з дрібними об’єктами, перекриттями, складними сценами, оскільки для кожної клітинки сітки дозволялося розпізнати лише обмежену кількість об’єктів.

YOLOv2 (YOLO9000) - оновлена версія, яка з’явилася вже через рік після першої. Тут впроваджено використання попереднього навчання на великій кількості класів об’єктів (до 9000 класів завдяки комбінуванню різних датасетів), додано нові типи шарів, а найважливіше - використано механізм anchor boxes для ефективнішої кластеризації можливих розмірів об’єктів. Це дало змогу краще виявляти дрібні об’єкти і працювати з різноманітними сценами. Окрім того, YOLOv2 став швидшим, зберігаючи простоту впровадження.

YOLOv3 - це вже серйозний крок уперед: модель стала багаторівневою завдяки введенню multi-scale detection, що дозволяє визначати об’єкти різного розміру на кількох шарах мережі. Це особливо важливо у задачах, де на одному

зображенні є і великі, і маленькі об’єкти (наприклад, сортування сміття, відеоспостереження в натовпі, детекція дорожніх знаків тощо). YOLOv3 довгий час залишалася стандартом де-факто у галузі завдяки чудовому балансу швидкості та точності, а також відкритому коду і великій підтримці спільноти розробників.

YOLOv4 - ще більше удосконалень: тут з’явилися нові типи регуляризації (наприклад, DropBlock, CIoU loss), вдосконалені функції активації, оновлені стратегії навчання та оптимізації, а також механізми data augmentation. В результаті YOLOv4 отримала ще більшу точність і стала застосовуватись у найвідповідальніших системах. Попри появу новіших версій, YOLOv4 залишається популярною завдяки гарному співвідношенню швидкості, якості та зрілості коду.

YOLOv5 - неофіційне, але надзвичайно поширене продовження ідеї YOLO. На відміну від попередніх, YOLOv5 розроблена на PyTorch (а не на Darknet), має декілька варіантів розміру (YOLOv5s - найкомпактніша, YOLOv5m - середня, YOLOv5l та YOLOv5x - великі, з найвищою точністю). YOLOv5 вирізняється дуже швидким навчанням і тестуванням, доступністю для модифікації під різні задачі, можливістю легко інтегруватися у різноманітні пайплайни. Вона підтримує експортування моделей для мобільних пристроїв, має дружню документацію та активну спільноту. Саме YOLOv5 використовується в багатьох екологічних проектах, стартапах, медичних додатках і системах відеоаналітики завдяки своїй універсальності.

YOLOv6, YOLOv7, YOLOv8 - новітні покоління YOLO. Кожна наступна версія вводить додаткові архітектурні оптимізації, підвищення якості детекції, покращення підтримки сегментації та класифікації, а також розширення можливостей для промислових рішень. Наприклад, YOLOv7 оптимізована під різні типи обчислювального обладнання, а YOLOv8 вже містить гнучку модульну архітектуру та інструменти для роботи з багатьма задачами водночас

(наприклад, детекція+сегментація+класифікація), що відкриває ще більше можливостей для розробників.

Окрему увагу варто звернути на питання вибору версії для практичних задач. Не завжди найновіша версія є найкращою саме для вашої задачі - важливо враховувати складність сцен, кількість класів, потужність доступного обладнання, необхідний FPS та інші особливості. Для проєкту з автоматичного сортування сміття було обрано YOLOv5m як збалансований варіант, який дозволяє досягти відмінних результатів у співвідношенні швидкості, точності й ресурсоємності [10].

Загалом, сімейство YOLO - це універсальний набір інструментів для задач комп'ютерного зору будь-якої складності. Відкритий код, активна спільнота, постійні оновлення та безліч реальних прикладів впровадження роблять YOLO незамінним вибором для екологічних, промислових, дослідницьких та освітніх проектів.

2.3 Інструменти для реалізації проекту

Для створення інтелектуальної системи класифікації та розпізнавання сміття використовується сучасний стек програмних засобів і бібліотек, які забезпечують високу продуктивність, гнучкість та зручність у розробці. Грамотний вибір інструментів має ключове значення для успішної реалізації проекту, оскільки саме вони визначають швидкість розробки, можливість інтеграції з іншими системами, ефективність тренування та тестування моделей.

В процесі розробки проекту було вирішено використати Python як основну мову програмування, оскільки вона є найпоширенішою у сфері штучного інтелекту та має велику кількість бібліотек для роботи з нейронними

мережами, обробки даних, інтеграції з різними платформами та створення інтерфейсів.

Для реалізації глибокого навчання та побудови моделей комп'ютерного зору обрано фреймворк PyTorch - потужний і зручний інструмент для роботи з нейронними мережами, який дозволяє гнучко налаштовувати архітектуру, швидко тестувати гіпотези та працювати з різними форматами даних. Для організації та тренування моделі YOLOv5 використовуються відкриті реалізації, що підтримуються активною спільнотою [11].

Для створення користувацького інтерфейсу обрано бібліотеку python-telegram-bot, яка забезпечує просту інтеграцію з Telegram та дозволяє реалізувати зручний чат-бот для взаємодії із системою класифікації сміття.

Всі обрані інструменти мають відкритий код, широку документацію і активну спільноту, що дозволяє швидко знаходити відповіді на технічні питання та впроваджувати нові функції. Завдяки цьому розробка проекту є прозорою, масштабованою та придатною для майбутнього розвитку.

2.3.1 Python як мова розробки

Python - це одна з найпопулярніших мов програмування у світі, особливо у сфері штучного інтелекту, машинного навчання та обробки даних. Її обирають як професіонали, так і початківці завдяки простому й зрозумілому синтаксису, великій кількості готових бібліотек, зручності для прототипування та швидкій розробці.

Головними перевагами Python для задач глибокого навчання є:

- Велика кількість бібліотек: для нейронних мереж (PyTorch, TensorFlow, Keras), обробки зображень (Pillow, OpenCV), роботи з даними (NumPy, Pandas), а також для створення API, ботів та інтерфейсів;

- Відкритість і доступність: Python має відкритий код і безкоштовний для використання, працює на будь-якій операційній системі (Windows, Linux, MacOS);
- Активна спільнота: мільйони розробників у всьому світі діляться знаннями, підтримують нові проєкти, створюють корисні інструкції, відповідають на питання у форумах і соцмережах;
- Гнучкість: Python підходить як для швидкого створення прототипів, так і для побудови масштабованих систем;
- Інтеграція: легко взаємодіє з іншими мовами, базами даних, сторонніми сервісами та інструментами автоматизації.

В рамках нашого проекту Python використовується для:

- обробки зображень та підготовки датасету;
- створення, навчання та тестування нейронної мережі YOLOv5;
- реалізації Telegram-бота для взаємодії користувача із системою;
- аналізу результатів, побудови звітів, візуалізації роботи моделі.

Завдяки простоті коду та зручній структурі проекту Python дозволяє швидко тестувати ідеї, впроваджувати нові функції та адаптувати рішення під потреби користувачів. Саме тому ця мова є ідеальним вибором для сучасних проєктів у сфері штучного інтелекту та екології [9].

2.3.2 PyTorch як фреймворк для глибокого навчання

PyTorch - це сучасний відкритий фреймворк для створення, навчання та тестування штучних нейронних мереж, який був розроблений компанією Facebook AI Research. За останні роки PyTorch став одним з найпопулярніших інструментів у сфері глибокого навчання як серед дослідників, так і в індустріальних проєктах. Його основна перевага - простий та інтуїтивно

зрозумілий інтерфейс, який дозволяє швидко прототипувати ідеї, а також легко будувати складні архітектури мереж.

Головні переваги PyTorch:

– динамічні обчислювальні графи: на відміну від багатьох інших фреймворків, у PyTorch обчислювальний граф створюється “на льоту” під час виконання коду. Це спрощує налагодження, дозволяє легко змінювати структуру моделі під час навчання та пришвидшує дослідження нових архітектур;

– гнучкість та масштабованість: PyTorch підходить як для невеликих академічних проєктів, так і для створення великих систем із мільйонами параметрів. Фреймворк підтримує як CPU, так і GPU, що дозволяє суттєво пришвидшити навчання моделей;

– велика кількість бібліотек та модулів: навколо PyTorch створено багато додаткових бібліотек для комп’ютерного зору, обробки тексту, генерації даних та оптимізації моделей;

– підтримка спільноти: PyTorch має величезну міжнародну спільноту розробників, науковців та ентузіастів. Є детальна документація, навчальні курси, приклади коду та обговорення на форумах;

– інтеграція з іншими інструментами: PyTorch легко поєднується з популярними бібліотеками для роботи з даними (NumPy, Pandas), а також з фреймворками для розгортання моделей (ONNX, TorchServe).

У нашому проєкті PyTorch використовується для:

– створення і навчання нейронної мережі YOLOv5 для детекції та класифікації сміття;

– експериментів із різними архітектурами та гіперпараметрами моделі;

– оцінки точності, побудови валідаційних та тестових пайплайнів;

– збереження, завантаження та використання натренованих моделей для інтеграції з Telegram-ботом.

PyTorch забезпечує максимальну гнучкість, високу швидкість обробки та масштабованість системи, що дозволяє швидко переходити від ідеї до практичного впровадження, легко тестувати нові підходи та забезпечувати якісну роботу системи у реальних умовах [11].

2.3.3 Бібліотека Python-Telegram-Bot

Python-Telegram-Bot - це популярна бібліотека з відкритим кодом для створення чат-ботів у месенджері Telegram. Вона дозволяє швидко розробляти функціональні, масштабовані та безпечні Telegram-боти, які можуть інтегруватися з різноманітними сервісами та штучним інтелектом [12].

Основні переваги Python-Telegram-Bot:

- Простота використання: бібліотека має зрозумілий синтаксис, детальну документацію та приклади, що дозволяє створювати бота навіть з базовим рівнем знань Python;
- Гнучкість: підтримує всі функції Telegram Bot API: обробку повідомлень, фото, документів, інтерактивних кнопок, меню, inline-режиму тощо;
- Масштабованість: можна створювати як простих ботів для особистого використання, так і складні корпоративні рішення для автоматизації бізнес-процесів чи навчальних проєктів;
- Інтеграція: легко поєднується з іншими Python-бібліотеками (наприклад, для роботи з нейронними мережами, обробки даних чи візуалізації результатів);
- Безпека: підтримує різні рівні доступу, захист даних користувачів, обмеження на команди й функції.

У цьому проєкті бібліотека Python-Telegram-Bot використовується для створення зручного інтерфейсу користувача - Telegram-бота для взаємодії із системою класифікації сміття. Основні функції бота:

– Прийом фотографій від користувачів;

– Автоматична передача зображення в модель YOLOv5 для класифікації;

– Відправлення результату (назви категорії сміття) у відповідь користувачеві;

– Можливість розширення функціоналу (наприклад, статистика розпізнавань, зворотний зв’язок тощо).

Використання Python-Telegram-Bot дозволяє зробити систему максимально доступною для звичайних користувачів, спростити процес тестування моделі та швидко інтегрувати інтелектуальні функції у звичне цифрове середовище [13].

2.4 Аналіз вибраного датасету TACO

TACO (Trash Annotations in Context) - це відкритий датасет, створений спеціально для задач комп’ютерного зору, спрямованих на автоматичне розпізнавання та класифікацію сміття у реальних умовах. Даний датасет широко використовується дослідниками та розробниками для тренування та тестування моделей, оскільки містить велику30.06.2025В кількість різноманітних зображень відходів, анотованих по класах і об’єктах [14].

Головні особливості датасету TACO:

– Різноманіття зображень: датасет включає фотографії сміття у природному оточенні (на вулицях, пляжах, у парках тощо), що наближає навчальні дані до реальних сценаріїв використання;

– Багато класів: містить понад 60 різних категорій відходів, серед яких пластик, скло, папір, метал, картон, органіка та інші;

– Анотації: кожне зображення має точні розмітки (bounding boxes, segmentation masks), що дозволяє тренувати моделі як для задач детекції, так і сегментації об’єктів;

– Відкритість: дані, розмітки та супровідна документація є у відкритому доступі, що сприяє вільному використанню, обміну знаннями та швидкому старту проєктів.

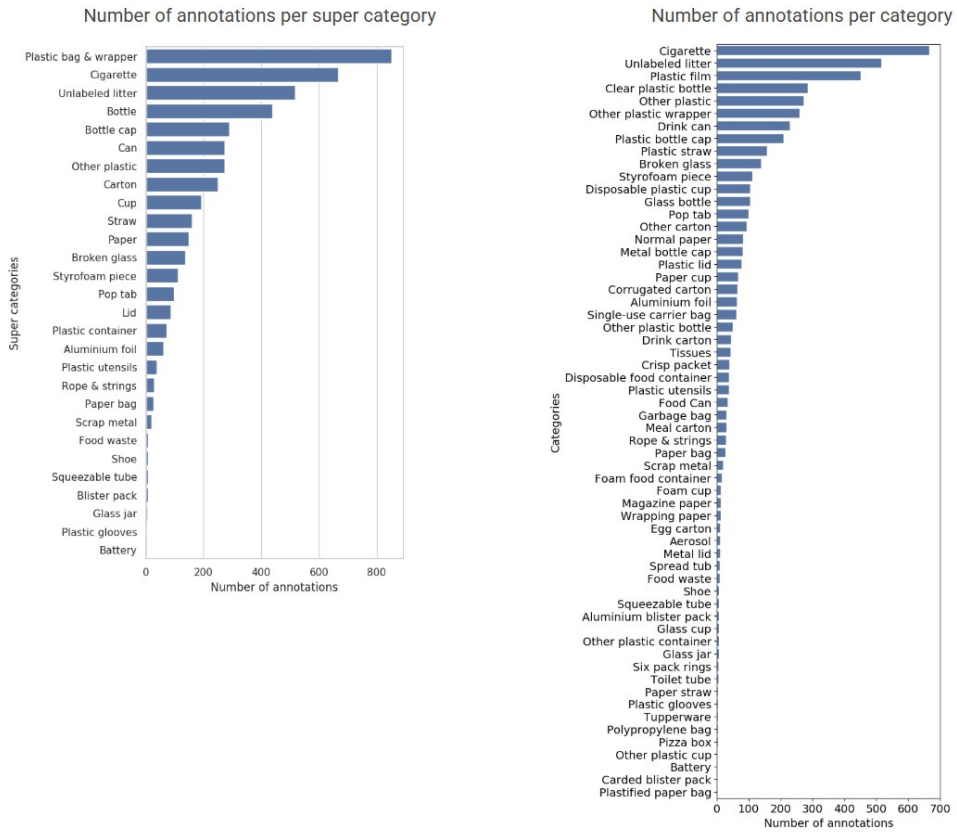

Рисунок 2.5 – Кількість фото для різних категорій

Використання датасету TACO в проекті має низку переваг:

– Можливість навчати моделі на максимально реалістичних даних із великою варіативністю фону, освітлення та розташування об’єктів;

– Достатньо велика вибірка для тестування якості моделі, проведення порівняльного аналізу та відбору оптимальних параметрів;

– Готові анотації дозволяють економити час на ручній розмітці, зосереджуючись на розробці та вдосконаленні архітектури нейронної мережі.

У межах нашого проекту з датасету TACO було відібрано ключові категорії сміття (пластик, скло, папір, картон, метал, інші відходи), а також проведено додаткову фільтрацію та підготовку даних для підвищення якості навчання моделі. Аналіз цього датасету допоміг краще зрозуміти типові помилки, труднощі та перспективи для подальшого розвитку автоматизованих систем сортування сміття.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ

3.1 Архітектура системи

Архітектура інтелектуальної системи для автоматичного сортування сміття побудована з урахуванням гнучкості, масштабованості та простоти інтеграції. В основі рішення лежить сучасна модель глибокого навчання для детекції та класифікації об’єктів, а також інтерфейс користувача, реалізований у вигляді Telegram-бота. Система складається з кількох основних модулів:

– модуль попередньої обробки даних: відповідає за підготовку вхідних зображень, масштабування, нормалізацію, аугментацію та формування навчальних і тестових вибірок;

– модуль нейронної мережі (YOLOv5): реалізує процес детекції та класифікації сміття на зображеннях. Модель приймає зображення на вхід, аналізує їх і повертає координати рамок для знайдених об’єктів та визначає їхню категорію (пластик, скло, папір, картон, метал, інші) [15];

– сервіс інтеграції з Telegram: через бібліотеку python-telegram-bot здійснюється прийом зображень від користувачів, обробка запитів і передача результатів розпізнавання назад у чат;

– модуль аналізу результатів: зберігає дані про оброблені запити, дозволяє формувати статистику розпізнавань, контролювати якість роботи моделі та відслідковувати типові помилки.

Рисунок 3.1 – Загальна блок схема проекту

Основна взаємодія виглядає так: користувач надсилає фото сміття Telegram-боту, бот приймає зображення і передає його у модель YOLOv5, яка здійснює детекцію й класифікацію. Після цього результат надсилається користувачу у вигляді текстового повідомлення з інформацією про тип сміття.

Архітектура дозволяє швидко масштабувати систему, підключати нові модулі або змінювати модель детекції під інші задачі. Всі компоненти побудовані так, щоб їх можна було розгортати як на одному сервері, так і розподілено - для обробки великої кількості запитів.

3.2 Процес підготовки та анотації даних

Підготовка якісного датасету та правильна анотація зображень - один із найважливіших етапів побудови системи автоматичної класифікації сміття. Від цього етапу залежить точність роботи моделі, її здатність узагальнювати,

розпізнавати нові об'єкти у різних умовах та уникати помилок у реальних сценаріях використання.

Перший крок - це збір і відбір зображень для навчання. Для проєкту було використано датасет TACO, який містить фото сміття з різних середовищ: міські вулиці, парки, пляжі, промислові зони. Датасет TACO має багато категорій сміття, але для нашого проекту було доцільнім спрощення класифікації до 6 типів: скло, пластик, метал, папір, картон та інші відходи. Для таких змін був створенний Python скрипт, який замінив старі класи новими - більш підходящими.

Наступний етап - анотація зображень. У TACO вже містяться розмітки для кожного об'єкта: координати прямокутної рамки (bounding box) та, для багатьох класів, маски сегментації. Однак для задачі детекції, яка стоїть у цьому проєкті, використовувалися саме bounding boxes, оскільки YOLOv5 працює із цим форматом анотацій. Частину зображень довелось вручну доанотувати або виправити через наявність помилок чи неповних розміток. Для цього використовували інструменти для редагування анотацій, наприклад, labelImg або онлайн-платформи.

Важливою складовою є також аугментація даних - штучне розширення датасету за рахунок модифікації вихідних зображень. Застосовували такі методи, як горизонтальне/вертикальне відзеркалення, зміна яскравості, повороти, масштабування, додавання шуму. Це дозволяє моделі вчитись розпізнавати об'єкти в різних умовах освітлення, під різними кутами та на різному фоні, що суттєво підвищує точність і стійкість до реальних ситуацій.

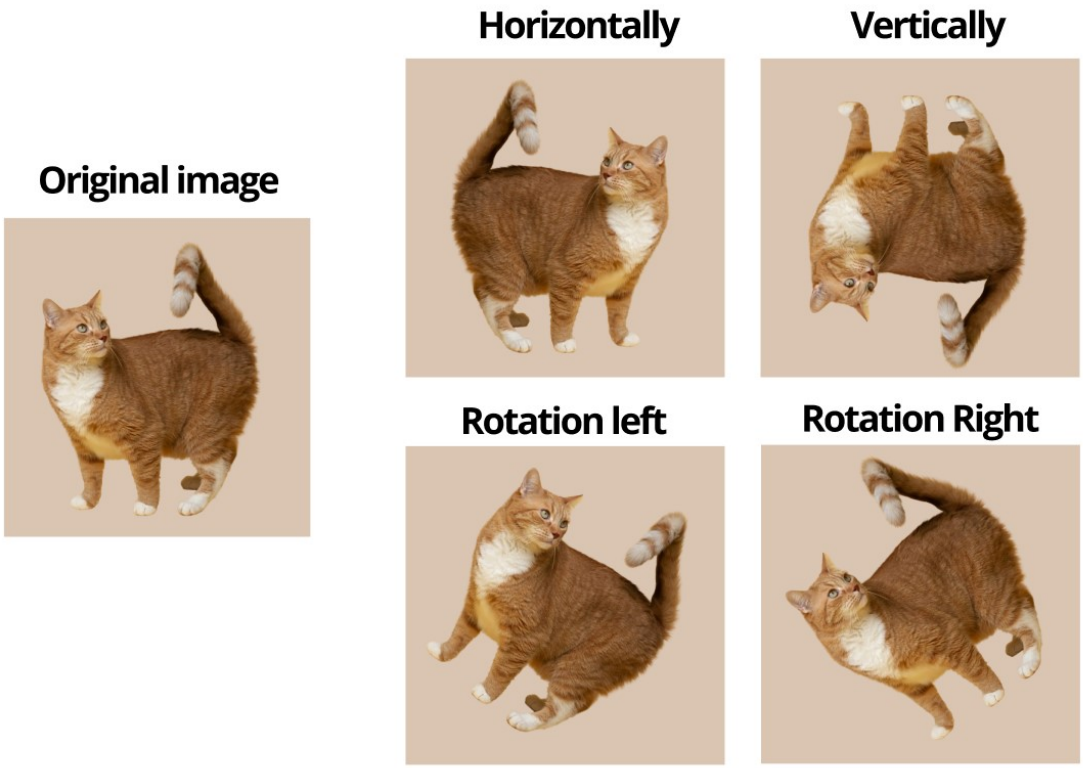

Рисунок 3.2 – Приклад аугментації данних

Після аугментації та перевірки якості анотацій усі зображення розподіляються на три підвибірки: навчальну (train), валідаційну (val) та тестову (test). Типове співвідношення - 70% train, 20% val, 10% test, але ці значення можуть змінюватися залежно від кількості даних та задачі.

Останній крок - підготовка анотацій у форматі, який підтримується YOLOv5 (файли txt із координатами для кожного об’єкта на зображенні). Після цього датасет готовий до завантаження у фреймворк та використання для навчання моделі.

В цілому, правильно організована підготовка та анотація даних - запорука високої якості майбутньої системи, її здатності точно та надійно працювати навіть із незнайомими зображеннями в умовах реального використання.

3.3 Побудова та навчання моделі YOLOv5

Побудова та навчання моделі YOLOv5 є одним з найважливіших і найскладніших етапів усього проєкту створення інтелектуальної системи для автоматичного сортування сміття. Від якості цього процесу залежить кінцева точність, надійність та ефективність роботи системи в реальних умовах. Саме на цьому етапі всі попередні кроки - підготовка даних, анотація, вибір інструментів - втілюються в конкретний практичний результат: робочу модель глибокого навчання, здатну аналізувати фотографії й розпізнавати категорії відходів [16].

Далі визначаються основні гіперпараметри навчання: кількість епох (iteration cycles), розмір пакету batch size, початкова швидкість навчання (learning rate), метод оптимізації, scheduler, розмірність вхідних зображень (наприклад, 640×640 пікселів). Підбір цих параметрів напряму впливає на якість навчання і час досягнення оптимального результату. В ході експериментів часто проводиться кілька спроб із різними значеннями гіперпараметрів для пошуку найкращої комбінації [17].

Навчання моделі, як правило, виконується на відеокарті (GPU), що дозволяє суттєво пришвидшити процес (у порівнянні з CPU час тренування може скоротитись у 10–20 разів). В ході тренування YOLOv5 автоматично зберігає найкращі ваги моделі (best.pt), веде детальне логування всіх метрик, таких як точність (precision), повнота (recall), середня точність по класах (mAP - mean Average Precision), F1-score, а також окремо по кожній категорії. Для моніторингу процесу навчання використовуються інструменти типу TensorBoard або вбудований веб-інтерфейс YOLOv5, які дозволяють відстежувати зміну метрик у реальному часі, аналізувати графіки збіжності, бачити типові приклади правильних і помилкових спрацювань.

Після завершення основного циклу навчання проводиться ретельне тестування моделі на відкладеній тестовій вибірці. Це необхідно для того, щоб

оцінити здатність нейромережі до узагальнення, її ефективність на нових, раніше невідомих зображеннях. Якщо результати тестування незадовільні (низька точність, багато помилкових позитивних чи негативних результатів), проводиться додаткова робота над підвибірками, аугментацією, змінюються гіперпараметри або навіть доопрацьовуються анотації. Часто тренування повторюють кілька разів, аналізуючи помилки і поступово покращуючи результат.

У цьому проєкті було проведено кілька експериментальних серій навчання з використанням різних підверсій YOLOv5 (наприклад, YOLOv5s, YOLOv5m, YOLOv5l), що дозволило підібрати найкращий баланс між швидкістю, точністю і ресурсозатратністю моделі. Також аналізувалися різні способи аугментації даних, що підвищило стійкість моделі до змін освітлення, фону, позиції об’єктів.

Натренована модель зберігається у вигляді спеціального файлу з вагами (наприклад, best.pt), який далі використовується для швидкої детекції сміття у реальних умовах - як на сервері, так і у взаємодії з Telegram-ботом. Фінальний результат - це працездатна система, здатна за секунди аналізувати зображення, знаходити й класифікувати різні категорії відходів, видавати результат користувачу в інтерфейсі бота або в автоматизованих рішеннях для сортувальних ліній.

Важливо, що відкритість коду, доступність архітектури YOLOv5 і прозорість усіх етапів навчання дозволяють швидко модифікувати систему під нові типи сміття, інтегрувати її в різні екологічні проекти та масштабувати для більших обсягі в даних у майбутньому. Саме тому правильна організація цього етапу - запорука успіху й практичної цінності розробленої системи.

3.4 Інтеграція моделі з Telegram-ботом

Інтеграція натренованої моделі YOLOv5 з Telegram-ботом - це фінальний етап практичної реалізації системи, який дозволяє зробити роботу з інтелектуальним сортуванням сміття максимально зручною та доступною для користувачів. Завдяки такому підходу можна швидко протестувати працездатність нейромережі у реальних умовах, отримати зворотний зв’язок від користувачів, а також популяризувати ідею розумного сортування серед широкої аудиторії.

Перший крок - розробка Telegram-бота за допомогою бібліотеки python-telegram-bot. Бот повинен уміти приймати фотографії від користувача, надсилати їх на сервер для обробки, отримувати результат від моделі та відповідати користувачу у зручному форматі. Для цього реалізується простий та інтуїтивний інтерфейс: користувач надсилає фото - бот приймає його, запускає детекцію, а потім надсилає відповідь із назвою категорії сміття або списком знайдених об’єктів.

Рисунок 3.3 – Приклад взаємодії з Telegram-ботом

Для інтеграції використовується API Telegram та механізми асинхронної обробки запитів. Коли бот отримує зображення, воно зберігається на сервері або передається у вигляді масиву даних у скрипт Python, де викликається

завантажена модель YOLOv5 (файл ваг best.pt). Зображення проходить ту ж попередню обробку, що й під час тренування (масштабування, нормалізація тощо), після чого передається в модель для розпізнавання.

Результати роботи моделі - список знайдених об’єктів, їхні координати та категорії - обробляються у скрипті. Для зручності користувача бот може накладати прямокутні рамки на об’єкти прямо на фото (наприклад, використовуючи бібліотеку OpenCV чи PIL), а також надсилати текстове повідомлення з описом: які види сміття знайдено та скільки кожного типу [17].

Особливу увагу слід приділити обробці типових помилок і винятків: якщо на фото немає жодного об’єкта або якість зображення надто низька, бот має повідомити про це користувача коректно та ввічливо.

Також до функціоналу Telegram-бота можна додати можливість формування статистики (скільки разів розпізнавалась кожна категорія), надання порад щодо сортування або навіть відправки фотографій до адміністратора для валідації складних випадків.

Інтеграція моделі з Telegram-ботом дозволяє перевести розроблену систему з лабораторного прототипу у реальний, зручний для користувача інструмент. Це відкриває перспективи для широкого застосування технології - як у побуті, так і у муніципальних або комерційних проєктах сортування сміття.

4 АНАЛІЗ РЕЗУЛЬТАТІВ ТА ЯКІСНА ОЦІНКА МОДЕЛІ

4.1 Оцінка якості розпізнавання на валідаційному датасеті

Після завершення навчання моделі було проведено оцінювання її якості на відкладеній валідаційній вибірці, що не використовувалася безпосередньо в процесі навчання. Це дозволяє об’єктивно перевірити здатність моделі до узагальнення та її ефективність у розпізнаванні об'єктів різних класів на реальних зображеннях побутових відходів [6].

Для валідації використовувалася окрема частина датасету, у якій були представлені всі шість класів: plastic, paper, metal, cardboard, glass, other. Зображення були розподілені так, щоб максимально відповідати розподілу у реальних умовах — із різними розмірами об'єктів, освітленням та ступенем складності сцени. В процесі оцінки розглядалися такі аспекти:

- коректність детекції об’єктів (наявність та точність побудови);
- якість класифікації кожного знайденого об’єкта;
- відсутність надмірної кількості хибнопозитивних та хибнонегативних спрацьовувань.

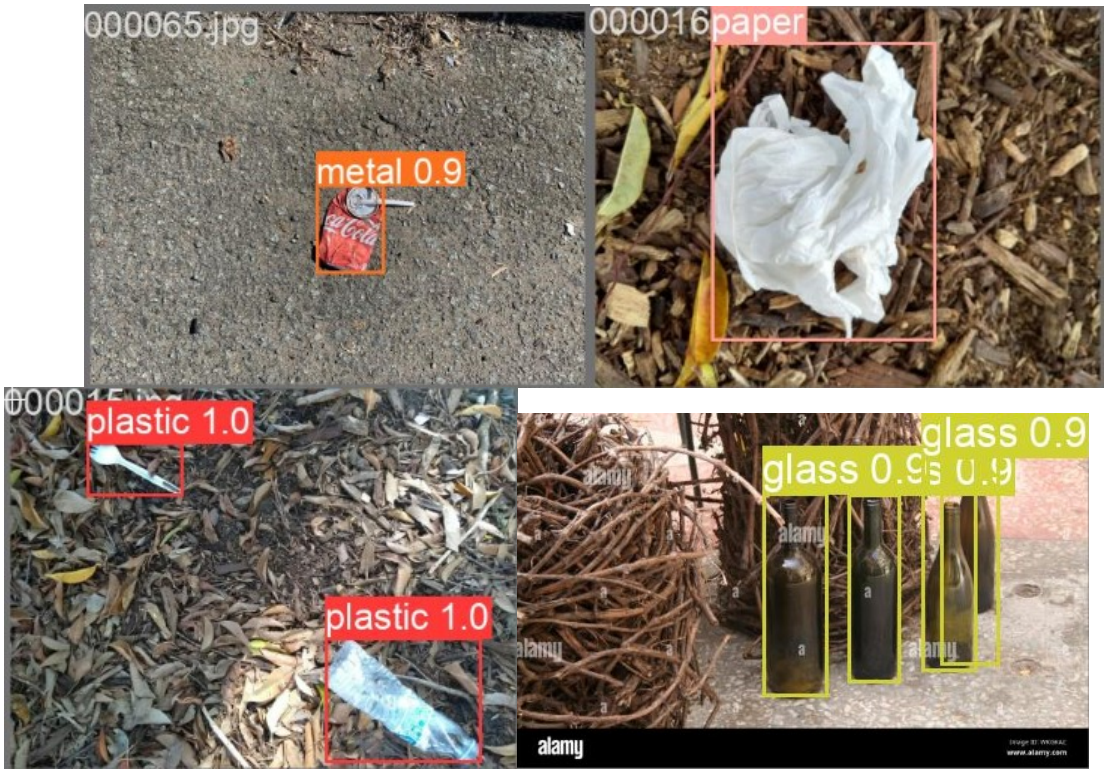

Рисунок 4.1 - Приклади роботи моделі на валідаційній вибірці

Ці приклади наочно демонструють, що модель впевнено розпізнає найбільш поширені категорії відходів (зокрема plastic, cardboard, glass) навіть у випадках, коли об'єкти частково перекриваються, мають неідеальну форму або розташовані у складних сценах.

Для кількісної оцінки якості використовувалися основні метрики — точність (precision), повнота (recall), середнє значення точності (mAP) для різних порогів IoU.

4.2 Метрики оцінки ефективності (Precision, Recall, mAP)

Для кількісної оцінки роботи моделі використовувалися загальноприйняті метрики якості для задач детекції об’єктів: точність (*precision*), повнота (*recall*) та середнє значення точності (*mean Average Precision, mAP*) [8].

Точність (precision). Показує, яку частку всіх передбачених об’єктів модель визначила правильно (тобто, наскільки мало хибнопозитивних спрацьовувань):

Повнота (recall). Відображає, яку частку реальних об’єктів модель змогла знайти серед усіх присутніх на зображенні:

Середнє значення точності (mAP). Це комплексна метрика, що враховує як точність, так і повноту при різних порогах перекриття (IoU) між передбаченими та реальними боксами. Найбільш популярними є:

- mAP@0.5 — середнє значення точності при IoU ≥ 0.5
- mAP@0.5:0.95 — усереднення mAP на діапазоні порогів IoU від 0.5 до 0.95

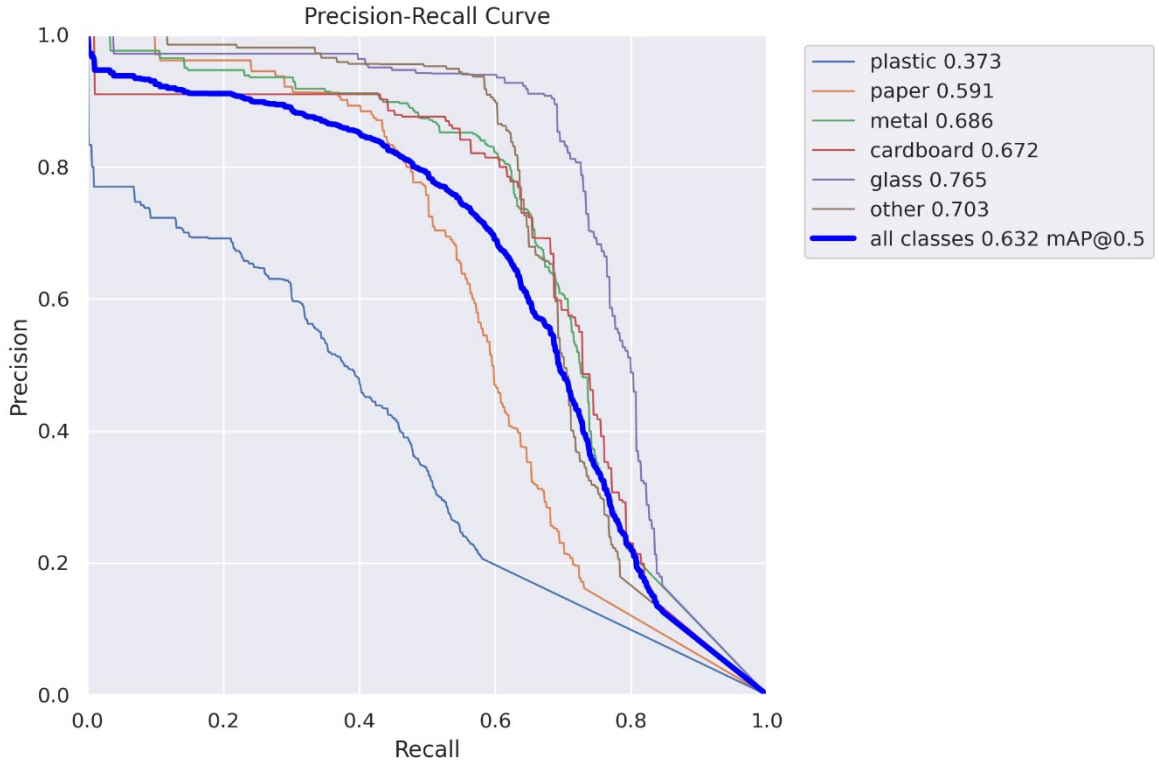

Рисунок 4.2 – Графік залежності precision та recall від порогу впевненості моделі для різних класів

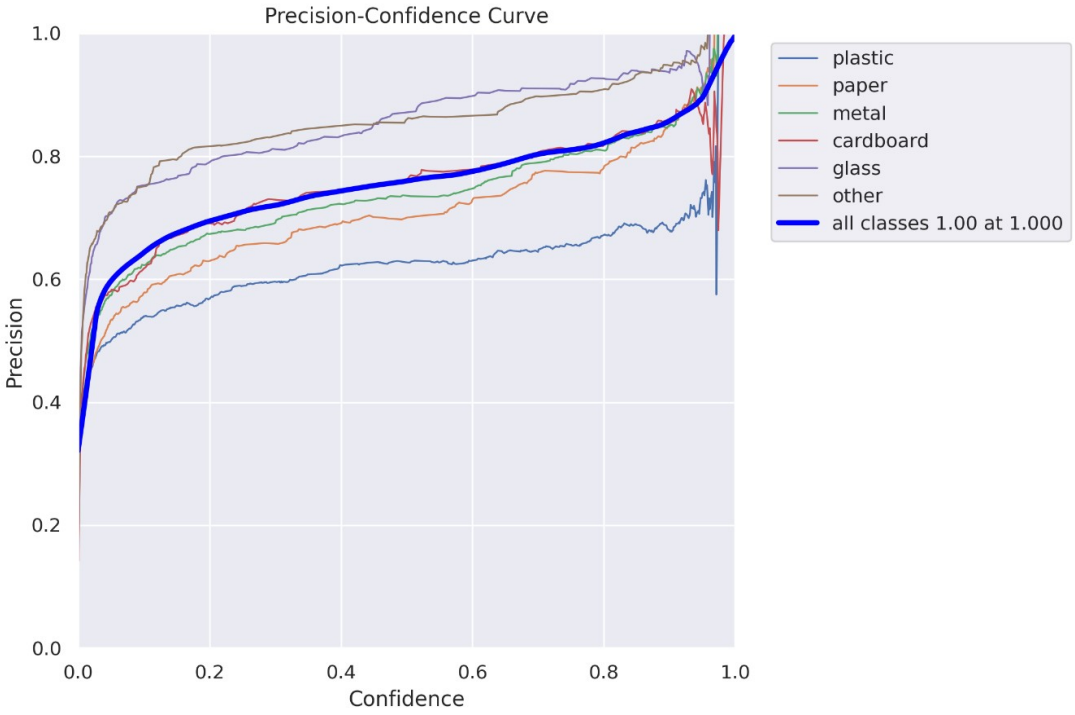

Рисунок 4.3 – Precision-Confidence крива для моделі YOLOv5m

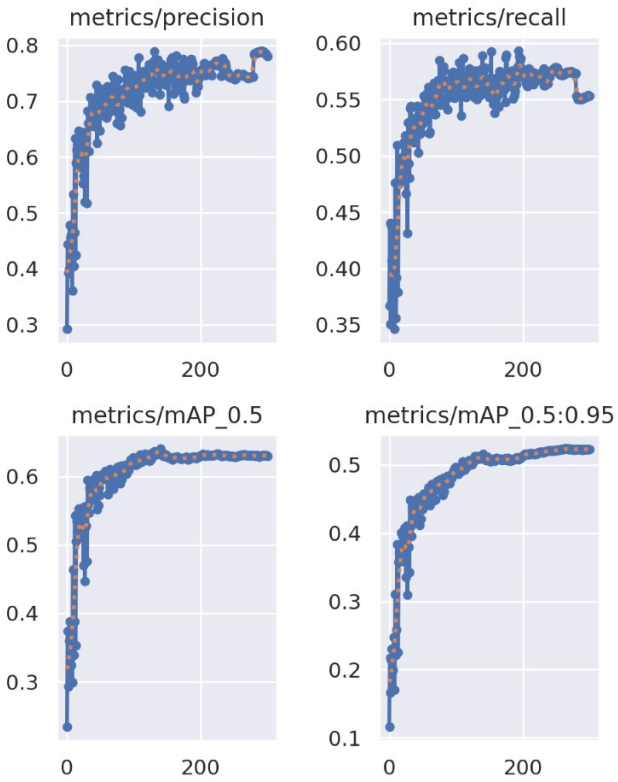

Рисунок 4.4 – Динаміка основних метрик якості (precision, recall, mAP@0.5, mAP@0.5:0.95) під час навчання моделі

Значення метрик дозволяють комплексно оцінити якість роботи моделі на різних класах відходів. Особливу увагу приділяють mAP@0.5:0.95 — саме ця метрика найкраще відображає реальну здатність моделі до точного виявлення та класифікації об'єктів у складних умовах.

4.3 Аналіз впливу підготовки даних на результати моделі

Якість роботи моделі глибокого навчання значною мірою залежить не лише від вибраної архітектури, а й від підготовки та попередньої обробки даних. У цьому проєкті особливу увагу було приділено аналізу, фільтрації та оптимізації датасету [4].

В процесі розробки системи була проведена глибока оптимізація датасету, що включала:

- Видалення помилкових, неінформативних або низькоякісних зображень. Під час попереднього перегляду датасету TACO було виявлено значну кількість зображень із поганою роздільною здатністю, нечіткими контурами об’єктів або некоректними анотаціями. Видалення понад тисячі таких прикладів дозволило підвищити чистоту вибірки.
- Фільтрація надто дрібних об’єктів. Було встановлено порогове значення мінімального розміру об’єкта (2–3% від сторони зображення), нижче якого анотації видалялися. Така фільтрація скоротила частку «нереалістичних» боксів, які не детектуються моделлю через обмеження архітектури YOLO.
- Балансування класів. Була проведена цілеспрямована робота з вирівнювання кількості прикладів для кожного класу шляхом видалення надлишкових анотацій (зокрема для класу “plastic”) та додавання нових зображень для “слабких” класів (наприклад, “cardboard”, “glass”).

Внаслідок чистки даних спостерігалося значне зменшення кількості помилкових детекцій на складних сценах, а також покращення загальної стабільності метрик: валідаційні втрати стали більш плавними, зникли аномальні стрибки під час навчання. Окрім того, відбувся суттєвий ріст основних показників якості — mAP@0.5 зріс приблизно з 0.33 до 0.65, а mAP@0.5:0.95 – з 0.24 до 0.54, що супроводжувалося підвищенням точності (precision) та повноти (recall). Загалом завдання для моделі стало простішим завдяки зменшенню кількості невидимих або нерелевантних прикладів у вибірці [18].

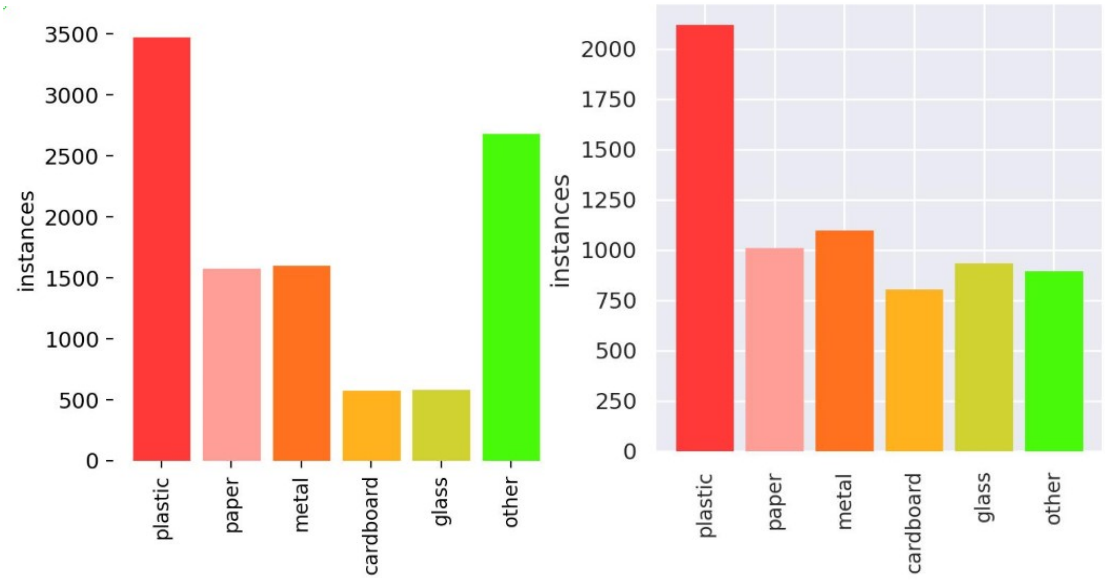

Рисунок 4.5 – Кількість об’єктів кожного класу до та після фільтрації датасету

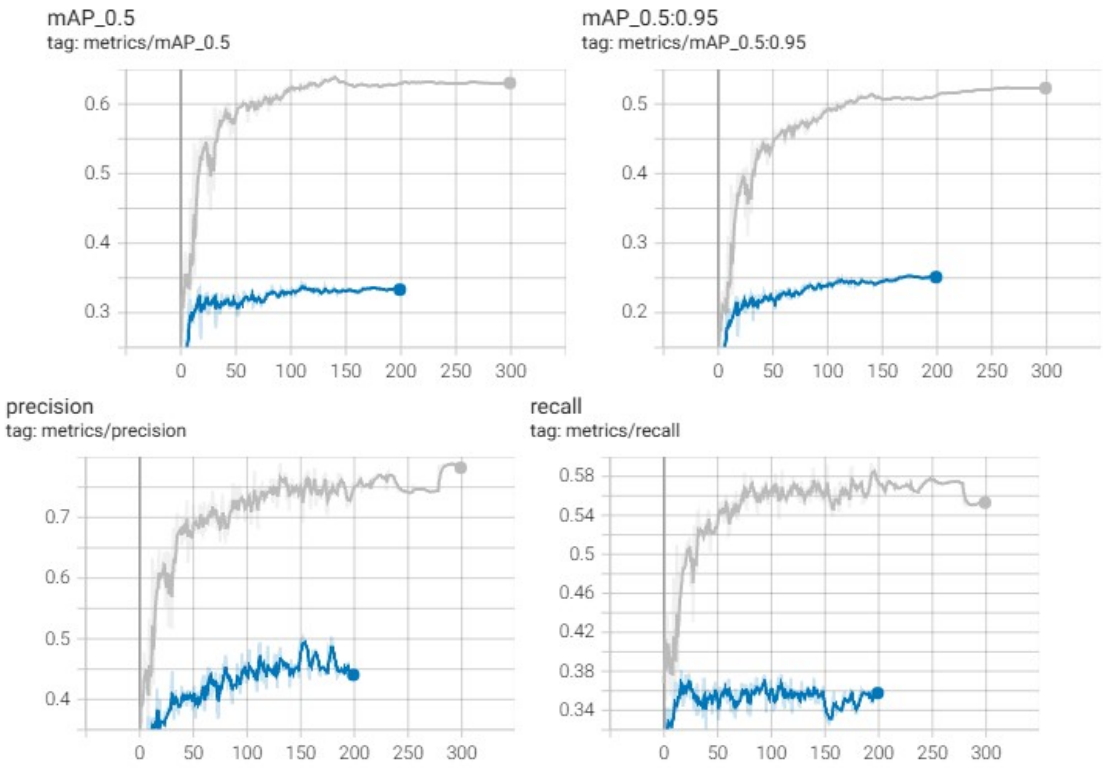

Рисунок 4.6 – Динаміка основних метрик якості моделі до і після чистки даних

4.4 Аналіз помилок та слабких класів

Для глибшого розуміння якості моделі проведемо аналіз типових помилок на основі матриці неточностей (confusion matrix). На рисунку 4.8 наведено матрицю для фінальної версії моделі, на рисунку 4.7 — для початкової версії.

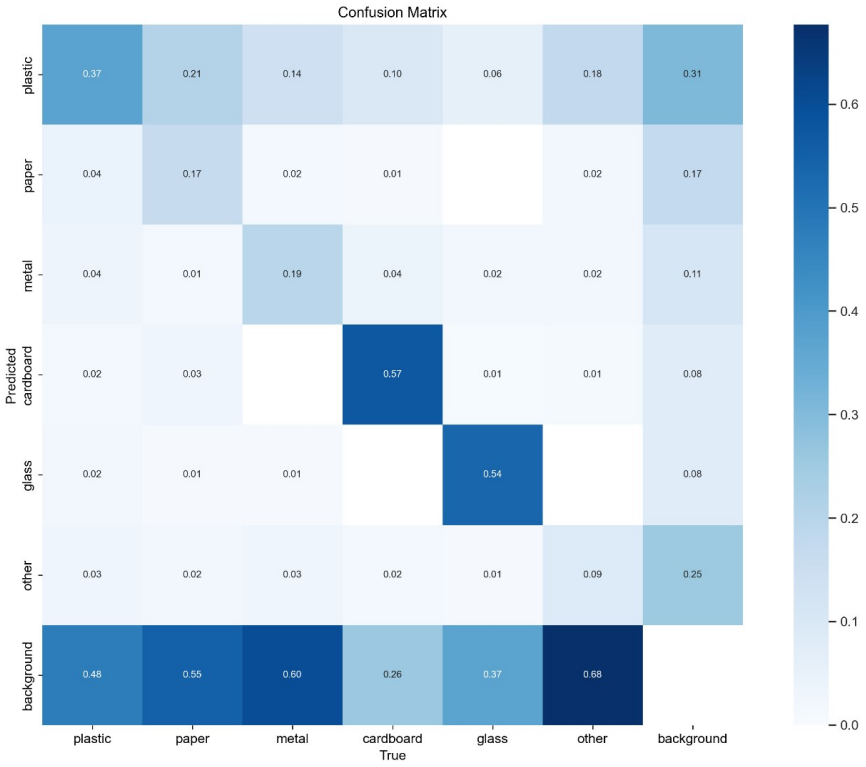

Рисунок 4.7 – Матриця неточностей для початкової версії моделі

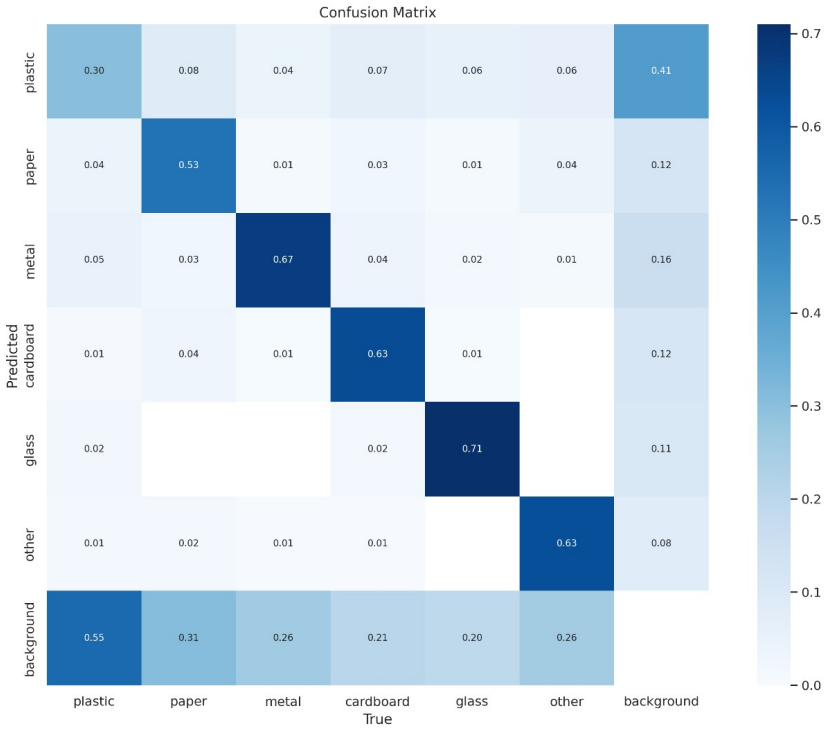

Рисунок 4.8 – Матриця неточностей для фінальної моделі

Порівнюючи обидві матриці, можна побачити, що якість класифікації значно покращилася для всіх класів, особливо для "paper", "metal", "glass" і "cardboard". У фінальній моделі, наприклад, "metal" та "glass" розпізнаються з часткою правильних відповідей 0.67 і 0.71 відповідно, що на 0.13–0.17 вище, ніж у початковій версії.

Водночас клас "plastic" залишається найпроблемнішим: лише 0.30 від об’єктів розпізнаються як "plastic", а 0.41 взагалі вважаються тлом (background), що свідчить про надмірну різноманітність зразків, дрібні об’єкти або невдалі приклади в датасеті. Інші класи, як-от "paper", "cardboard" і "other", також часто плутаються з тлом, але частка правильних відповідей тут вища — до 0.53 для "paper" і 0.63 для "cardboard".

Основні джерела помилок:

- Змішування з фоном: Найбільша частка помилок — це віднесення сміття до background. Особливо це стосується "plastic" (0.41), "paper" (0.31) та "other" і "metal" (0.26).
- Переплутування класів: У початковій версії моделі міжкласові плутанини були значно вищими — наприклад, "plastic" часто плутався з "paper" та "metal", а "cardboard" — з "glass". У фінальній версії ці помилки зменшилися.
- Слабкі класи: "plastic" залишається найбільш проблемним для моделі, ймовірно, через широку семантику класу, неоднорідність зразків та малий розмір об’єктів. Частка правильних відповідей тут лише 0.3.

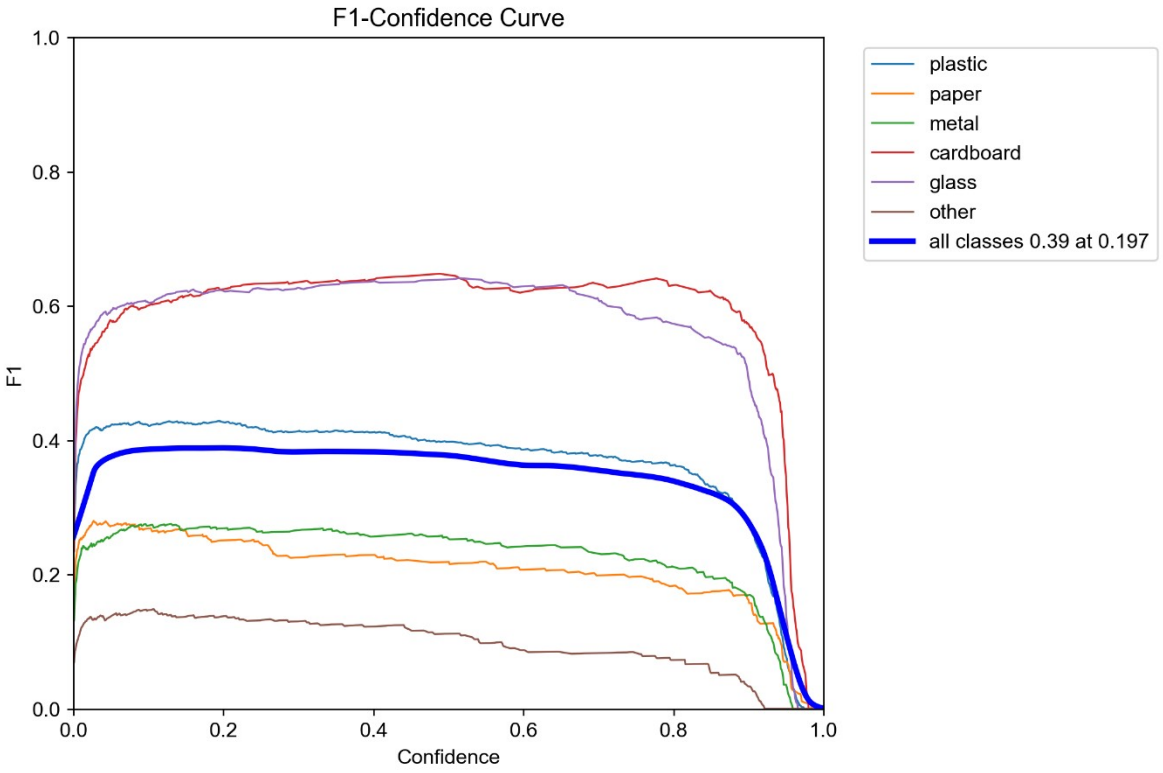

Рисунок 4.9 – F1-Confidence крива для різних классів для початкової версії моделі

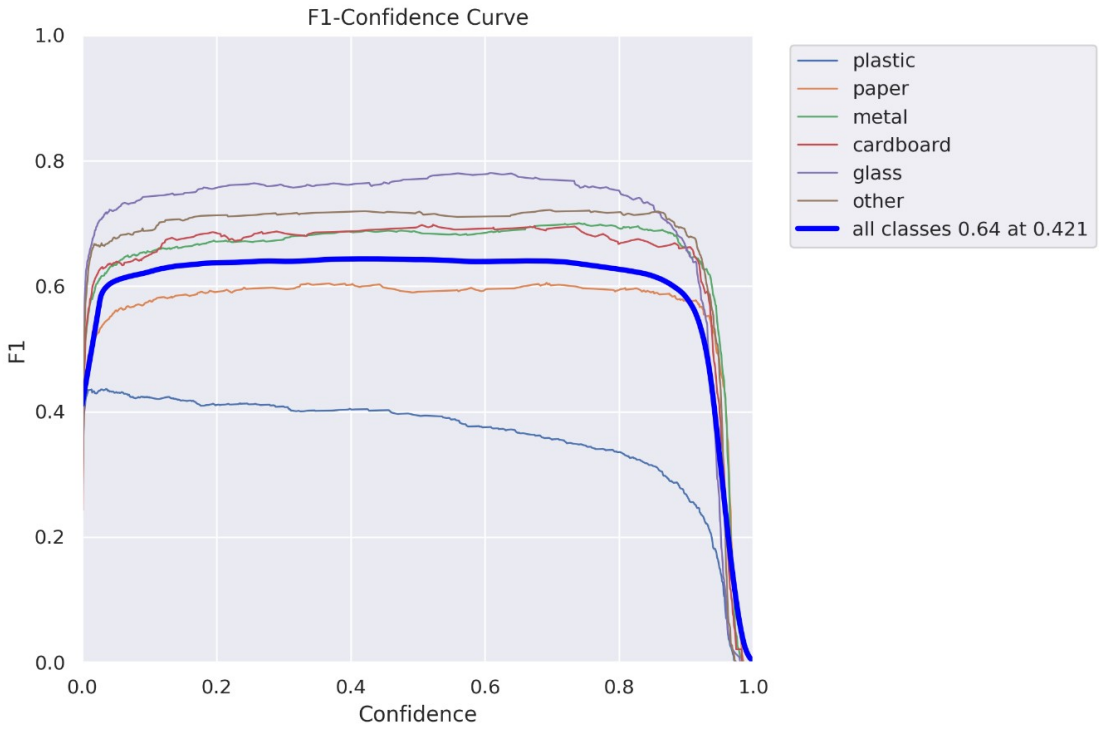

Рисунок 4.10 – F1-Confidence крива для різних классів для фінальної версії моделі

Очищення і балансування датасету дозволили суттєво підвищити якість детекції: модель не лише стала більш впевненою у складних класах, а й демонструє стабільну роботу без різких стрибків метрик на валідації. Це підтверджує, що якість початкових даних критично впливає на результативність глибоких моделей, і навіть проста фільтрація та додавання релевантних зразків можуть радикально покращити результат [6].

Проте для подальшого зростання якості рекомендується:

- Додатково поповнювати класи "plastic" і "paper" новими різноманітними прикладами.
- Видаляти занадто дрібні або нехарактерні об’єкти в цих класах.
- Використовувати складніші аугментації для кращого узагальнення моделі.

ВИСНОВКИ

За період практики виконано ключові завдання, пов’язані з використанням сучасних методів штучного інтелекту та комп’ютерного зору для автоматизації процесу сортування сміття. Робота була зосереджена на застосуванні простих і практичних інструментів, щоб побудувати систему, яка реально може використовуватись у житті.

У ході практики досягнуто наступного:

- проаналізовано існуючі підходи до розділення й сортування побутових відходів у різних умовах;
- розглянуто традиційні та новітні методи комп’ютерного зору, які використовуються для класифікації зображень і детекції об’єктів;
- обґрунтовано вибір основних інструментів: Python як мови розробки, PyTorch як фреймворку для навчання нейронної мережі, python-telegram-bot для інтеграції з Telegram і датасету TACO для отримання різноманітних зображень сміття;
- проведено підготовку, аугментацію й анотацію даних для якісного навчання моделі, звернено увагу на структуру датасету й коректність розмітки;
- здійснено налаштування та навчання моделі YOLOv5 для задачі детекції та класифікації сміття на фотографіях, підібрано гіперпараметри й оцінено якість роботи моделі;
- реалізовано Telegram-бот, який дозволяє взаємодіяти з моделлю, отримувати результати класифікації та тестувати роботу системи в реальному часі з різних пристроїв.

Під час виконання роботи вдалося поєднати теоретичні знання з реальними практичними навичками. Зібрано повний цикл: від аналізу проблеми й підготовки даних до побудови й тестування робочої системи. Особливу увагу приділено простоті й гнучкості використаних рішень, щоб можна було легко вносити зміни або розширювати систему під нові задачі та категорії.

Продемонстровано, що ідея, яка виникає на етапі обговорення, може бути втілена у вигляді робочого прототипу: інтеграція з Telegram-ботом дозволила побачити результати роботи моделі на практиці, що допомогло швидко виявити сильні сторони та обмеження підходу. Досвід використання відкритих бібліотек та готових інструментів дав змогу швидко рухатись вперед і експериментувати з різними варіантами.

Практика підтвердила, що сучасні технології штучного інтелекту дійсно можуть допомогти у вирішенні екологічних проблем — навіть на локальному рівні. Отримані результати й напрацювання можуть бути корисними для подальшого навчання, участі в командних проектах чи при розробці власних рішень для подібних задач. Також окрему увагу варто приділити питанням тестування системи в нових умовах і вдосконаленню підходів до підготовки даних, що стане корисним при роботі над дипломною роботою або розширенні існуючого проєкту.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Державне агентство з енергоефективності та енергозбереження України. “Статистика поводження з відходами.” URL: https://saee.gov.ua/ (дата звернення: 22.04.2025).

2. Waste Management World. “AI and Robotics in Waste Sorting.” URL: https://waste-management-world.com/ai-in-sorting/ (дата звернення: 22.04.2025).

3. “Artificial Intelligence in Waste Management: A Review” // Journal of Cleaner Production, 2023. (URL-адреса відсутня).

4. Brownlee J. “A Gentle Introduction to Object Detection with Deep Learning.” URL: https://machinelearningmastery.com/object-detection-with-deep-learning/ (дата звернення: 27.04.2025).

5. Zhang Z., et al. “A Vision-based Trash Classification System using Deep Learning.” // Procedia Computer Science, 2021. (URL-адреса відсутня).

6. Krizhevsky A., Sutskever I., Hinton G.E. “ImageNet Classification with Deep Convolutional Neural Networks.” // Communications of the ACM, 2017. (URL-адреса відсутня).

7. PyTorch: Офіційна документація. URL: https://pytorch.org/docs/stable/index.html (дата звернення: 30.04.2025).

8. TACO: Trash Annotations in Context. URL: https://tacodataset.org/ (дата звернення: 30.04.2025).

9. OpenCV: Бібліотека для роботи із зображеннями. URL: https://opencv.org/ (дата звернення: 10.05.2025).

10. Офіційна документація бібліотеки Pandas. URL: https://pandas.pydata.org/ (дата звернення: 13.05.2025).

11. “Open Data Science: Еволюція методів класифікації зображень.” URL: https://ods.ai/competitions (дата звернення: 13.05.2025).

12. “TensorBoard: Візуалізація навчання нейронних мереж.” URL: https://www.tensorflow.org/tensorboard (дата звернення: 20.05.2025).

13. Goodfellow I., Bengio Y., Courville A. “Deep Learning.” MIT Press, 2016.

14. Scikit-learn: Офіційна документація. URL: https://scikit-learn.org/stable/ (дата звернення: 20.05.2025).

15. Chollet F. “Deep Learning with Python.” 2nd edition. Manning, 2021.

16. Офіційна документація NumPy. URL: https://numpy.org/doc/ (дата звернення: 20.05.2025).

17. Szeliski R. “Computer Vision: Algorithms and Applications.” Springer, 2022.

18. Офіційний Telegram Bot API. URL: https://core.telegram.org/bots/api (дата звернення: 22.05.2025).

19. Python-Telegram-Bot: Документація бібліотеки. URL: https://python-telegram-bot.org/ (дата звернення: 22.05.2025).