

Хмельницький національний університет  
Факультет програмування  
та комп'ютерних і телекомунікаційних систем  
Кафедра телекомунікацій, медійних та інтелектуальних технологій

ДИПЛОМНА РОБОТА МАГІСТРА

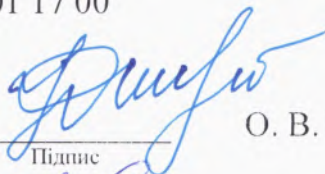
Телеграм-бот для класифікації зображень твердих побутових відходів  
Назва теми

Галузь знань 11 – Математика та статистика

Спеціальність 113 – Прикладна математика

ДРПМ 2019/084 01 17 00

Виконав: студент 2 курсу, група ПМм19-1

  
Підпис

О. В. Коломієць


Керівник: к.ф.-м.н., доцент

  
Підпис, дата

Н. В. Грипинська

До захисту допускаю:

Зав. кафедри ТМІТ, д.т.н., доцент

  
Підпис, дата

С. К. Підченко

2 грудня 2020 р.

Хмельницький, 2020

# ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ПРОГРАМУВАННЯ ТА КОМП'ЮТЕРНИХ І ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМ

Кафедра ТЕЛЕКОМУНІКАЦІЙ, МЕДІЙНИХ ТА ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ

Освітній рівень МАГІСТР

Галузь знань 11 МАТЕМАТИКА ТА СТАТИСТИКА

Спеціальність 113 ПРИКЛАДНА МАТЕМАТИКА

Освітня програма ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА ПІДГОТОВКИ МАГІСТРА

ЗАТВЕРДЖУЮ

Зав. кафедри

3 09

20 20 р.

## ЗАВДАННЯ НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ)

Коломієць Ольга Вікторівна

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Телеграм-бот для класифікації зображень твердих побутових відходів

Керівник проекту (роботи) Грипинська Надія Василівна, к.ф.-м.н., доцент

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 01.09.2020 р. № 118

2. Строк подання студентом проекту (роботи) на кафедру 01.12.2020 р.

3. Вихідні дані до проекту (роботи) Постановка завдання у вигляді блок-схеми системи класифікації зображень ТПВ, модель класифікації у вигляді архітектури згорткової нейронної мережі, архітектура клієнт-серверної системи, результати тренування згорткової нейронної мережі, Телеграм-блог для класифікації зображень твердих побутових відходів, результати верифікації та валідації інформаційної системи

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) Аналіз стану поводження з твердими побутовими відходами в Україні, огляд інформаційних систем збору та оброблення твердих побутових відходів, порівняльний аналіз методів та підходів до класифікації зображень, проектування класифікатора зображень, проектування інформаційної системи оброблення та класифікації зображень, розроблення Телеграм-боу, як користувачького інтерфейсу, оцінювання роботи Телеграм-боту

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) Блок-схема інформаційної системи класифікації зображень, UML-діаграма для системи класифікації зображень твердих побутових відходів, схематичне зображення топології згорткових нейронних мереж, представлення клієнт-серверної архітектури, зображення програмного коду реалізації класифікатора, графіки оцінок ефективності функції втрат для найкращого результату, результати класифікації зображень в інтерфейсі застосунку Телеграм

6. Консультанти розділів дипломного проекту (роботи)

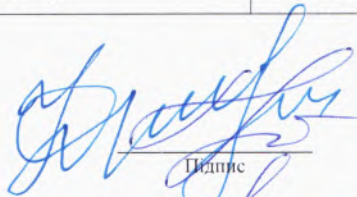
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
РОЗДІЛ 1	–	–	–
РОЗДІЛ 2	–	–	–
РОЗДІЛ 3	–	–	–

7. Дата видачі завдання «\_\_» \_\_\_\_\_ 20\_\_ р.

**КАЛЕНДАРНИЙ ПЛАН**

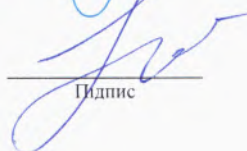
Назва етапів (розділів) дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1. Аналіз стану поводження з твердими побутовими відходами в Україні	20.07.2020 – 09.08.2020	Виконано
2. Огляд інформаційних систем збору та оброблення твердих побутових відходів	10.08.2020 – 28.08.2020	Виконано
3. Порівняльний аналіз методів та систем класифікації зображень	17.08.2020 – 04.09.2020	Виконано
4. Постановка задачі моделювання та реалізації класифікатора твердих побутових відходів	07.09.2020 – 11.09.2020	Виконано
5. Проектування класифікатора зображень	12.09.2020 – 20.10.2020	Виконано
6. Проектування архітектури клієнт-серверної системи	21.09.2020 – 28.10.2020	Виконано
7. Розроблення телеграм-боту для класифікації твердих побутових відходів	12.10.2020 – 01.11.2020	Виконано
8. Верифікація та валідація інформаційної системи	02.11.2020 – 08.11.2020	Виконано
9. Апробація результатів дослідження на двох конференціях	04.11.2020 – 10.11.2020	Виконано
10. Оформлення висновків і графічної частини дисертаційного дослідження	09.11.2020 – 15.11.2020	Виконано
11. Перевірка дипломної роботи на нормоконтроль та плагіат	16.11.2020 – 27.11.2020	Виконано
12. Представлення роботи до захисту	04.12.2020 – 15.12.2020	Виконано

Студент



О. В. Коломієць

Керівник проекту (роботи)



Н. В. Грипинська

## АНОТАЦІЯ

Тема дипломної роботи: Телеграм-бот для класифікації зображень твердих побутових відходів.

Автор роботи: Коломієць Ольга Вікторівна

Керівник роботи: Грипинська Надія Василівна

Загальний обсяг роботи: 107 сторінок, 41 рисунок, 6 таблиць, 4 додатки, 39 посилань

ТВЕРДІ ПОБУТОВІ ВІДХОДИ, СИСТЕМА КЛАСИФІКАЦІЯ, ЗГОТКОВА НЕЙРОННА МЕРЕЖА, АРХІТЕКТУРА КЛІЄНТ-СЕРВЕР, ТЕЛЕГРАМ-БОТ

Метою дипломної магістерської роботи є розроблення моделі класифікації та клієнт-серверної інформаційної системи для класифікації твердих побутових відходів.

Дипломна робота магістра присвячена розробленню інформаційної системи для класифікації твердих побутових відходів. У роботі спроектовано та розроблено класифікатор зображень твердих побутових відходів з використанням згорткових нейронних мереж. Вперше використано інтерфейс програмного забезпечення Телеграм для реалізації системи класифікації таких зображень. Телеграм-бот може бути використаний у місцях загального користування, де спеціалізовані системи збору сміття можуть захоплювати відходи всередину, використовувати камеру для виготовлення фотографії та перемістити його у відповідне класифіковане поле.

## ANNOTATION

a master's degree work of Olga Kolomiets  
entitled « Telegram-bot for the classification of solid household waste images».

Mentor: Nadiia Hrypynska

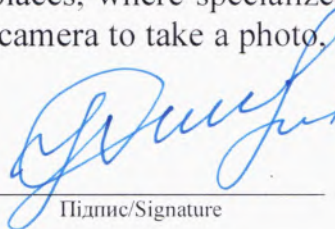
Total volume of work: 107 pages, 41 figures, 6 tables, 4 appendices, 39 references

SOLID WASTE, CLASSIFICATION SYSTEM, CONVOLUTIONAL NEURAL NETWORK, CLIENT-SERVER ARCHITECTURE, TELEGRAM-BOT

The master's thesis aims to develop a model of classification and client-server information system for the classification of solid waste.

The thesis is devoted to developing an information system for the classification of solid waste. The study describes the design and implementation of a solid waste images classifier based on convolutional neural networks. For the first time, the Telegram software interface was used to implement a system for classifying such images. The telegram-bot can be used in public places, where specialized garbage collection systems can capture waste inside, use a camera to take a photo, and move it to the appropriate classified field.

2 грудня 2020 р  
Дата/Date

  
Підпис/Signature

## ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ .....	6
ВСТУП .....	7
1 АНАЛІЗ СТАНУ ПОВОДЖЕННЯ З ТВЕРДИМИ ПОБУТОВИМИ ВІДХОДАМИ В УКРАЇНІ .....	11
1.1 Аналіз сучасного стану та тенденцій розвитку методів та підходів до управління відходами .....	11
1.1.1 Загальна характеристика побутових відходів.....	11
1.1.2 Сучасний стан управління побутовими відходами в Україні .....	13
1.1.3 Розвиток інтегрованих систем управління відходами .....	16
1.2 Огляд сучасних інформаційних систем та підходів до збору та оброблення твердих побутових відходів .....	20
1.3 Порівняльний аналіз методів та систем класифікації зображень .....	23
1.4 Проблема поводження з відходами на прикладі діяльності Філії «Старокостянтинівський завод залізобетонних шпал».....	30
1.5 Постановка задачі моделювання та реалізації класифікатора твердих побутових відходів.....	32
2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ КЛАСИФІКАЦІЇ ТВЕРДИХ ПОБУТОВИХ ВІДХОДІВ .....	35
2.1 Опис класифікатора на основі згорткової нейронної мережі .....	35
2.2 Проектування архітектури нейронної мережі типу Faster R-CNN.....	40
2.3 Сценарії використання користувачького інтерфейсу .....	47
2.4 Архітектура клієнт-серверної системи .....	49
2.5 Опис програмних інструментів .....	51
3 РОЗРОБЛЕННЯ ТЕЛЕГРАМ-БОТУ ДЛЯ КЛАСИФІКАЦІЇ ТВЕРДИХ ПОБУТОВИХ ВІДХОДІВ.....	56
3.1 Реалізація класифікатора.....	56
3.1.1 Підготовка даних для класифікатора.....	56

3.1.2	Попереднє оброблення даних.....	59
3.1.3	Налаштування параметрів згорткової нейронної мережі .....	63
3.2	Навчання класифікатора та аналіз результатів.....	65
3.3	Реалізація серверної частини.....	68
3.4	Реалізація клієнтської частини .....	69
3.5	Верифікація та валідація інформаційної системи .....	71
3.5.1	Тестування класифікатора .....	71
3.5.2	Інструкція користувача для впровадження системи класифікації	73
	ВИСНОВКИ.....	75
	ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....	77
	ДОДАТОК А АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ НА КОНФЕРЕНЦІЇ КМОСС-2020.....	82
	ДОДАТОК Б АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ НА КОНФЕРЕНЦІЇ АПКН-2020.....	86
	ДОДАТОК В ЛІСТИНГ ПРОГРАМНОГО КОДУ .....	93
	ДОДАТОК Г СЛАЙДИ ПРЕЗЕНТАЦІЇ.....	100
	ДОДАТОК Д АНТИПЛАГІАТ.....	106

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

- CNN – convolutional neural network
- GT – ground truth (істинне значення)
- БВ – будівельні відходи
- ВВ – великогабаритні відходи ВВ
- ЗНМ – згорткова нейронна мережа
- ІСУ – інтегрована система управління
- КЗ – комп'ютерний зір
- НМ – нейронна мережа
- ОВ – опаковальні відходи
- ОРС – оптичне розпізнавання символів
- ОМ – опаковальні матеріали
- ПВ – побутові відходи
- РВ – рідкі відходи
- ТПВ – тверді побутові відходи
- ШНМ – штучна нейронна мережа

## ВСТУП

Дипломна робота магістра присвячена розробленню інформаційної системи у вигляді Телеграм-боту для класифікації зображень твердих побутових відходів.

**Актуальність теми.** В ХХІ столітті питання збереження екології є одним із найбільш пріоритетних для людства. Наприклад в Україні щорічно накопичується сотні тонн сміття. Кожний українець генерує майже 300 кг відходів щороку; спільний доробок усього населення країни складає в середньому 35 млн відходів [1] і ці показники постійно зростають. Водночас більш як 50 % цих відходів можуть бути перероблені, хоча нині в країні переробляється менш як 30 % усього сміття [2]. На жаль, ситуація із відходами в Україні щороку погіршується.

Накопичення відходів та їхнє подальше утилізування стає все більш критичною проблемою. Водночас в регіонах України наразі немає стандартизованих інтегрованих систем управління побутовими відходами. Тому виникає потреба у розробленні та впровадженні подібних систем раціонального поводження з побутовими відходами на регіональному рівні. Отже, задача розроблення автоматизованої інформаційної системи для сортування різних типів відходів із зручним для людей користувацьким інтерфейсом нині є актуальним науково-технічним завданням.

**Аналіз останніх досліджень і публікацій.** Системи штучного інтелекту можуть стати ефективним рішенням для автоматизованого сортування та машинної класифікації твердих побутових відходів, що дасть змогу значно зменшити кількість сміття в Україні. Наприклад, деякі роботи пропонують використовувати оптимізовані глибокі нейронні мережі для класифікації вибраних класів об'єктів, що підлягають переробленню. На вхід такої системи поступає зображення з перемішаними типами відходів. Далі така система локалізує ознаки об'єктів на зображенні та класифікує їх на кілька категорій.

Загалом питання раціонального поводження з ПВ та підходи до їхнього утилізування розглядали у своїх працях такі вітчизняні та зарубіжні дослідники: С. А. Сафронов [1], О. С. Мельник [2], Ю. Б. Матвеев [4], Б. В. Уткіна [21], І. О. Хоменко [18], І. Л. Бондар [6], І. С. Сокірян [7], В. Г. Петрук [14], В. В. Погребенюк [16, 17], L. J. C. Brinez [22], S. Sudha [23], A. A. Rahmi [26], G. Mittal [27], M. Yang [39] та багато інших.

Отже, у даній роботі представлено автоматизовану інформаційну систему для виявлення твердих побутових відходів на зображеннях.

**Метою** дипломної магістерської роботи є клієнт-серверної інформаційної системи у вигляді Телеграм-боту для класифікації твердих побутових відходів.

Для досягнення поставленої мети необхідно виконати такі **завдання**:

1. Провести аналіз стану та тенденцій розвитку методів та систем управління твердими побутовими відходами.
2. Розробити класифікатор зображень твердих побутових відходів.
3. Побудувати модель клієнт-серверної системи управління відходами.
4. Реалізувати інформаційну систему з користувацьким інтерфейсом Телеграм-боту для класифікації зображень.

**Об'єктом** дослідження є процес сортування твердих побутових відходів.

**Предметом** дослідження є методи класифікації зображень твердих побутових відходів.

**Гіпотеза дослідження.** Застосування методів оброблення зображень є дієвим засобом для сортування твердих побутових відходів та контролю за його об'ємами. Проте невирішеною залишається проблема малої кількості зображень в навчальних наборах даних для налаштування автоматизованих систем сортування відходів. Ефективним рішенням може бути застосування нейронних мереж для класифікації зображень з використанням методу трансферного навчання. Тому дослідження полягає в тому, щоб створити клієнт-серверну систему на основі згорткової нейронної мережі з кількома шарами для розпізнавання шести основних класів відходів з малою кількістю зображень у кожному класі.

**Науково-практична новизна** дисертаційного дослідження полягає в наступному: вперше реалізовано веб-застосунок із зрозумілим користувацьким інтерфейсом Телеграм-боту для класифікації об'єктів твердих побутових відходів на зображеннях; розроблений Телеграм-бот апробовано до проблеми раціонального поводження з відходами на підприємстві ПАТ «Укрзалізниця» Філія «Старокостянтинівський завод залізобетонних шпал».

**Структура та обсяг роботи.** Кваліфікаційна робота магістра містить завдання, реферат, зміст, перелік скорочень та умовних позначків, вступ, 3 розділи, висновки, перелік джерел посилань із 39 найменувань та 4 додатки. Загальний обсяг дипломної роботи складає 107 сторінок, з яких 81 сторінка основного тексту та 26 сторінок додатків. У роботі наведено 41 рисунок та 6 таблиць.

Основна частина складається з трьох розділів.

У першому розділі проведено аналіз сучасного стану розвитку методів та систем управління відходами та виявлено, що стан поводження з побутовими відходами в Україні є незадовільним. На території нашої держави все ще немає єдиної інтегрованої системи управління відходами, що дала б змогу ефективно утилізувати як промислові, так і тверді побутові відходи. Здійснено огляд сучасних інформаційних технологій та підходів до збору та оброблення твердих побутових відходів. Проведено порівняльний аналіз комерційних сервісів, що є лідерами в задачах детектування та класифікації зображень. На основі аналізу сучасних методів та підходів до класифікації зображень обрано згорткову нейронну мережу, як класифікатор інформаційної системи.

У другому розділі описано процес проектування інформаційної системи для класифікації твердих побутових відходів, а саме: побудовано класифікатор зображень на основі згорткової нейронної мережі, розглянуто та описано сценарії використання користувацького інтерфейсу, розроблено основні модулі клієнт-серверної системи. Проведено огляд сучасних програмних інструментів та обрано середовище мови програмування Python для реалізації системи з користувацьким інтерфейсом Телеграм-боту.

У третьому розділі наведено реалізацію моделі згорткової нейронної мережі типу Faster R-CNN. Нейронну мережу навчено та протестовано на готовому наборі даних TrashNet, що містить 2527 зображень і поділений на шість класів. У результаті проведених експериментів визначено, що найвищу точність класифікації зображень досягнуто зі значенням коефіцієнта навчання в діапазоні та кількістю ітерацій, що дорівнює 2000. Навчена нейронна мережа Faster R-CNN досягла максимальної середньої точності класифікації в 84,5 %. Побудовану нейронну мережу використано, як класифікатор твердих побутових відходів користувачьким інтерфейсом у вигляді Телеграм-боту.

**Публікації та апробація результатів дослідження.** За темою роботи опубліковано:

1) тези (Додаток А) – Нрупунська Н. В., Коломійєць О. В. Convolutional neural network for image-based waste classification. VI International scientific and technical conference on the occasion of the 90th anniversary of USCTU “Computer Modeling and Optimization of Complex Systems”, (CMOCS-2020) : proceedings of IX inter. scien.-tech. conf (Dnipro, 4-6 Nov, 2020). Dnipro : Ukrainian State University of Chemical Technology, 2020. P. 112–113. doi:10.32434/CMOCS-2020

2) тези (Додаток Б) – Грипинська Н. В., Коломійєць О. В. Автоматизована система виявлення та класифікації твердих побутових відходів на зображеннях. XII Всеукраїнська науково-практична конференція “Актуальні проблеми комп’ютерних наук”, (АПКН-2020) : матеріали XII всеукр. наук.-практ. конф. (м. Хмельницький, 9-10 лис. 2020 р.). Хмельницький : Хмельницький національний університет, 2020. С. 86–90.

# 1 АНАЛІЗ СТАНУ ПОВОДЖЕННЯ З ТВЕРДИМИ ПОБУТОВИМИ ВІДХОДАМИ В УКРАЇНІ

1.1 Аналіз сучасного стану та тенденцій розвитку методів та підходів до управління відходами

## 1.1.1 Загальна характеристика побутових відходів

Термін «тверді побутові відходи» (ТПВ) уперше вказано в галузевому нормативному документі України – «Правила надання послуг зі збирання та вивезення твердих і рідких побутових відходів», що затверджені наказом № 54 від 21.03.2000 р. Держкомітету будівництва, архітектури та житлової політики України [3]. Відповідно до цих правил, ТПВ – це відходи, які утворюються в процесі життя й діяльності людини й накопичуються в житлових будинках, закладах соціально-культурного побуту, громадських, навчальних, лікувальних, торговельних та інших закладах; іншими словами, ТПВ – це харчові відходи, предмети домашнього вжитку, сміття, опале листя, відходи від прибирання й поточного ремонту квартир, макулатура, скло, метал, полімерні матеріали тощо, що не мають подальшого використання за місцем їхнього утворення [3]. Проте цей документ на підставі наказу № 391 від 18.12.2008 р. Міністерства з питань житлово-комунального господарства України втратив свою чинність [3]. З того часу повного офіційно затвердженого означення терміну ТПВ немає. Натомість цей термін у ст. 1 Закону України «Про відходи» доповнено від 21.01.2010 р. в абзаці про означення терміну «побутові відходи» (ПВ) [4]. Відповідно до цього доповнення [4], «побутові відходи – це відходи, що утворюються в процесі життя й діяльності людини в житлових та нежитлових будинках (тверді, великогабаритні, ремонтні, рідкі, крім відходів, пов'язаних із виробничою діяльністю підприємств) та не використовуються за місцем їхнього накопичення».

Відповідно до Державного класифікатора відходів ДК 005–96 [5], виділяють такі групи відходів.

а) Побутові відходи: комунальні (міські) змішані відходи: харчові відходи, кімнатне та дворове сміття, макулатура, тара, упаковальні матеріали, дерево, метал [6]; місцем утворення цих відходів є житлові будинки, адміністративні та суспільні (громадські) організації, підприємства торгівлі, культури, побуту тощо, прибудинкові території, зелені насадження.

б) Великогабаритні відходи (ВВ): старі меблі, холодильники, телевізори, сантехнічне обладнання, дерева, гілки, пеньки тощо; лікарняні відходи: перев'язувальні матеріали, бинти, вата, шприци, кімнатне сміття, харчові відходи, тара, упаковальні матеріали [7]; місцем утворення цих відходів є лікарні, поліклініки, медичні, кабінети, консультації тощо.

в) Будівельні відходи (БВ): відходи будівельних матеріалів та конструкцій, ґрунт, пісок, асфальт. Такі відходи утворюються від знесення старих будівель та будівництва нових, також від проведення капітальних ремонтів будинків і споруд.

г) Рідкі відходи (РВ): побутові: нечистоти, забруднені та стічні води від миття непромислових приміщень та прилеглої території. Місце їхнього утворення – неканалізовані житлові будинки та інші об'єкти.

На сьогодні відходи переважно вивозять на заміські полігони, що призначені для їхнього заховання, та частково потрапляють у місця неорганізованого зберігання (орієнтовано 10%). РВ сучасного міста становлять не тільки епідеміологічну, але й серйозну токсикологічну проблему, тому що на стадії збору приблизно 4% відходів є токсичними [8]. Дослідження свідчать [9], що звичайні РВ великого міста містять понад 100 найменувань токсичних сполук: барвники, пестициди, ртуть та її сполуки, свинець та його солі тощо. Особливе місце поміж РВ займають ртутні лампи, оскільки кожна з них містить від 80 до 120 міліграмів ртуті. Серйозну проблему становлять також пластмаси й синтетичні матеріали, оскільки вони не піддаються процесам біологічного руйнування й можуть тривалий час (сотні років) перебувати в об'єктах довкілля.

До складу ПВ також входять такі компоненти: харчові відходи, папір, картон, текстиль, скло, пластмаса (у тому числі ПЕТ- пляшки), метал чорний та кольоровий, дерево, шкіра тощо [9]. Встановлено, що в складі ПВ постійно збільшується вміст пластмас, фольги, різного роду тари, поліетиленових плівок та інших упакувань.

### 1.1.2 Сучасний стан управління побутовими відходами в Україні

Під поняттям «управління відходами» розуміють збирання, перевезення, оброблення (включно з відновленням та видаленням), нагляд за такими операціями й подальший контроль, догляд за об'єктами видалення відходів після їхнього закриття [10]. В Україні нараховується 6,5 тисячі законних і майже 35 тисяч незаконних сміттєзвалищ, загальною площею 7 % території, а це можна прирівняти до площі цілої Данії (понад 43 тисячі км<sup>2</sup>). Такий відсоток території завалений відходами перевищує площу природних заповідників, що становить 7 % проти 4,5 % [9]. Найбільші сміттєзвалища на території України наведено на рис. 1.1.



Рисунок 1.1 – Розподіл найбільших полігонів ПВ на карті України [9]

Ситуація з утилізуванням ПВ і з кожним роком гіршає. За підрахунками екологів [11], Україна накопичила 54 млн м<sup>3</sup> відходів; щороку сміттєві полігони поповнюються приблизно на 15–17 мільйонів тонн. На перероблення йде лише десята частина зібраного сміття.

За щорічними даними Мінрегіонбуду [12] обсяг утворення ПВ в Україні впродовж 2015–2019 рр. становив: у 2015 р. – приблизно 52 млн м<sup>3</sup>, у 2016 та 2017 рр. орієнтовано 59 млн м<sup>3</sup>, а у 2018 р. – майже 65 млн м<sup>3</sup>, у 2019 р. – приблизно 68 млн м<sup>3</sup>.

Однак, ці чи інші статистичні дані обсягів утворення ПВ в Україні не відображають реальну ситуацію, а тому не є точними. Однією з причин встановлення Мінрегіоном таких приблизних даних є малий відсоток (орієнтовано 20 %) населення України, які не охоплено послугою вивезення ПВ [12]. Тому таке явище, як створення несанкціонованих звалищ, самовільне засмічення територій тощо, набуло великих масштабів і тому реальні неофіційні показники можуть бути значно більшими. Другою причиною цих неточних даних є брак налагодженої практики зважування ПВ. Перерахунок в одиниці маси (т) здійснюється з огляду на густину ПВ, яка складає приблизно 0,2–0,3 т/м<sup>3</sup>.

На сьогодні в Україні панівними методами управління ПВ залишаються традиційні технології їхнього утилізування – спалювання, розміщення й заховання відходів на несанкціонованих звалищах і спеціальних полігонах, які перевантажені та не відповідають нормам екологічної безпеки [8]. Наприклад, у роботі [10] запропоновано інформаційну систему, яка дає змогу здійснювати моніторинг управління ПВ. Вона дає змогу відображати дані комунальних служб, зокрема, санітарних зон відходів, які їм належать, приватних підприємств та інших об'єктів господарювання про утворення, накопичення, рух та утилізування цих відходів. Інформаційна система забезпечує вибір ефективної схеми управління цими процесами. У роботі [13] наведено аналіз основних тенденцій розвитку методів і систем управління побутовими

відходами у Європі, зокрема, в Україні. Структуру відходів Хмельницької області, України та Європи показано в табл. 1.1.

Таблиця 1.1 – Структура відходів Хмельницької області, України та Європейського Союзу (ЄС), % [13]

Вид відходів	Хмельниччина	Україна	ЄС
Харчові відходи	35	45	35
Вторинні полімери	30	13	12
Папір, картон	8	10	33
Великогабаритні матеріали	–	–	10
Скло	12	8	3
Текстильні матеріали	2	6	–
Метали	4	3	5
Деревина	10	1	2
Відходи будівництва	6	5	–

Відповідно до цих даних харчові відходи переконливо переважають поміж інших у Європі, Україні та на Хмельниччині. З аналізу структури інших видів відходів Хмельницької області бачимо, що переважають відходи вторинних матеріалів, деревини та інших матеріалів. Саме така тенденція є характерною для України. Натомість у Європі переважають відходи з паперу, великогабаритних матеріалів та вторинних матеріалів [13, 14].

Упродовж останніх років кількість утворених побутових відходів у Хмельницькій області поступово зростає [15] через зміни способу життя (досягнення надмірної комфортності), та недооцінювання населенням природних реакцій на брак екологічної рівноваги. Динаміка утворення відходів на Хмельниччині продемонстрована на рис. 1.2.

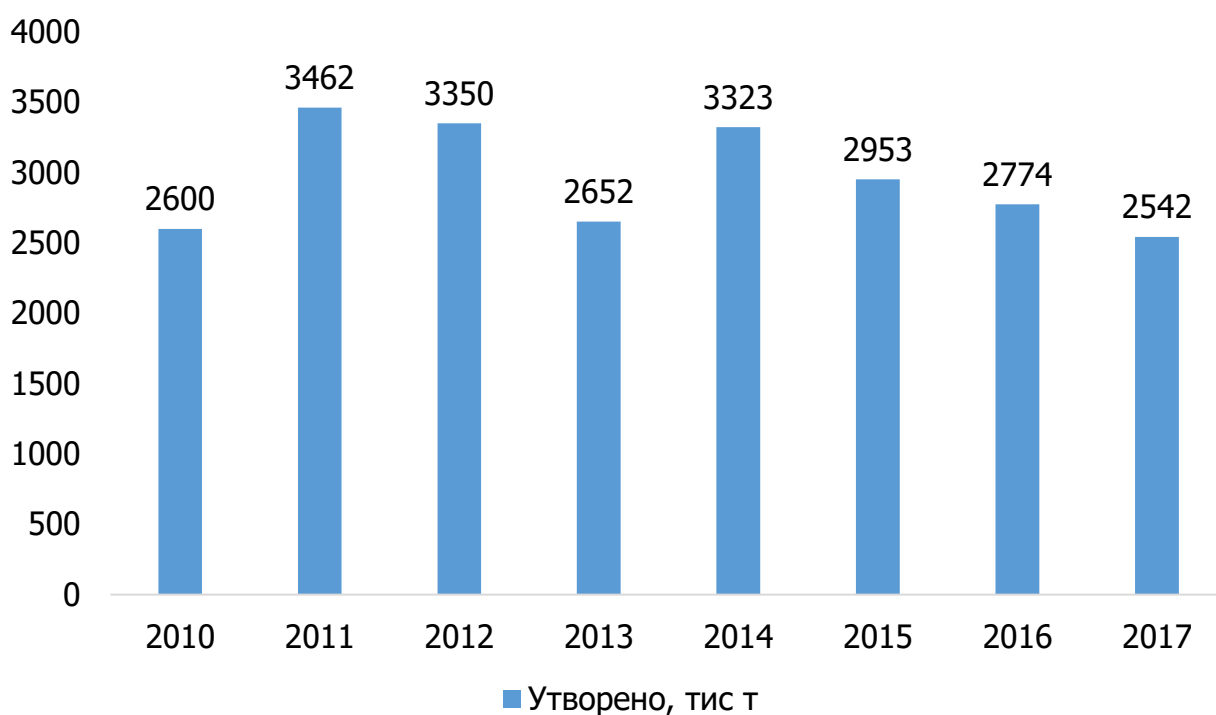


Рисунок 1.2 – Показники утворення відходів у Хмельницькій області з урахуванням обсягів відходів, зібраних підприємствами від домогосподарств

Отже, проаналізовано тенденції утворення та утилізуванню відходів у Хмельницькій області за 2015–2019 рр. Визначено зміни обсягу спалених відходів в Україні й на Хмельниччині та обґрунтовано важливість інтегрування інформаційних систем для поводження з ТПВ.

### 1.1.3 Розвиток інтегрованих систем управління відходами

Підвищення ефективності системи управління ПВ потребує включення різнорівневого поєднання методів і елементів систем, що передбачають не лише технічні, а і юридичні, інституційні та економічні зв'язки. Таке поєднання є виявом інтегрування елементів, процесів та технологій.

У різних наукових джерелах поняття «інтегрування» розглядається з різних підходів до застосування в тій чи іншій сфері [16, 17]. Водночас поняття

«інтегрування» зводиться до одного загального розуміння – це синтез суміжних елементів та процесів, які в поєднанні один з одним утворюють єдину систему, якій притаманні одні й ті ж цілі, завдання та зміст. Отже, інтегрований» – це комплексний; той, що ґрунтується на поєднанні відокремлених частин у єдине ціле. Наразі виділяють декілька визначень інтегрованої системи управління (ІСУ).

ІСУ – система, функції якої полягають в управлінні різними видами людської діяльності задля досягнення загальної глобальної цілі. Інтеграція полягає в об'єднанні відокремлених частин чи підсистем у межах однієї системи, яка охоплює інформаційні аспекти управління на основі загального програмно-технічного, інформаційного й організаційного забезпечення.

Поняття «інтегрований» у сфері управління відходами також має зміст різноманітних дій [18], які неодмінно передбачають запобігання утворенню відходів, повторне використання, компостування, спалювання, утилізування та управління логістикою відходів.

Методологія інтегрованого управління відходами має на увазі діяльність різних сторін від громадськості на місцях до різних структур державної виконавчої влади [19]. Система інтегрованого управління відходами зумовлена різними механізмами (процесами) функціонування, що сприяє її стійкості, а саме: технічний, екологічний, фінансово-економічний, соціальний, інституційний та політико-правовий.

Інтегрований підхід управління ПВ розроблено в 1980-х роках голландською недержавною організацією WASTE спільно з її партнерськими організаціями [15]. Цю методологію було успішно застосовано в багатьох країнах. Поняття «інтегрований» відображає принцип, який закладено в цей інструмент, – управління відходами складається з різноманітних дій, якими передбачено також і запобігання утворенню відходів та їхнє утилізування (рециклінг, компостування), а управління здійснюється різними зацікавленими сторонами на різних рівнях.

Розрізняють три компоненти в інтегрованому управлінні ПВ [19], а саме: зацікавлені сторони, елементи системи та аспекти стійкості (рис. 1.3).

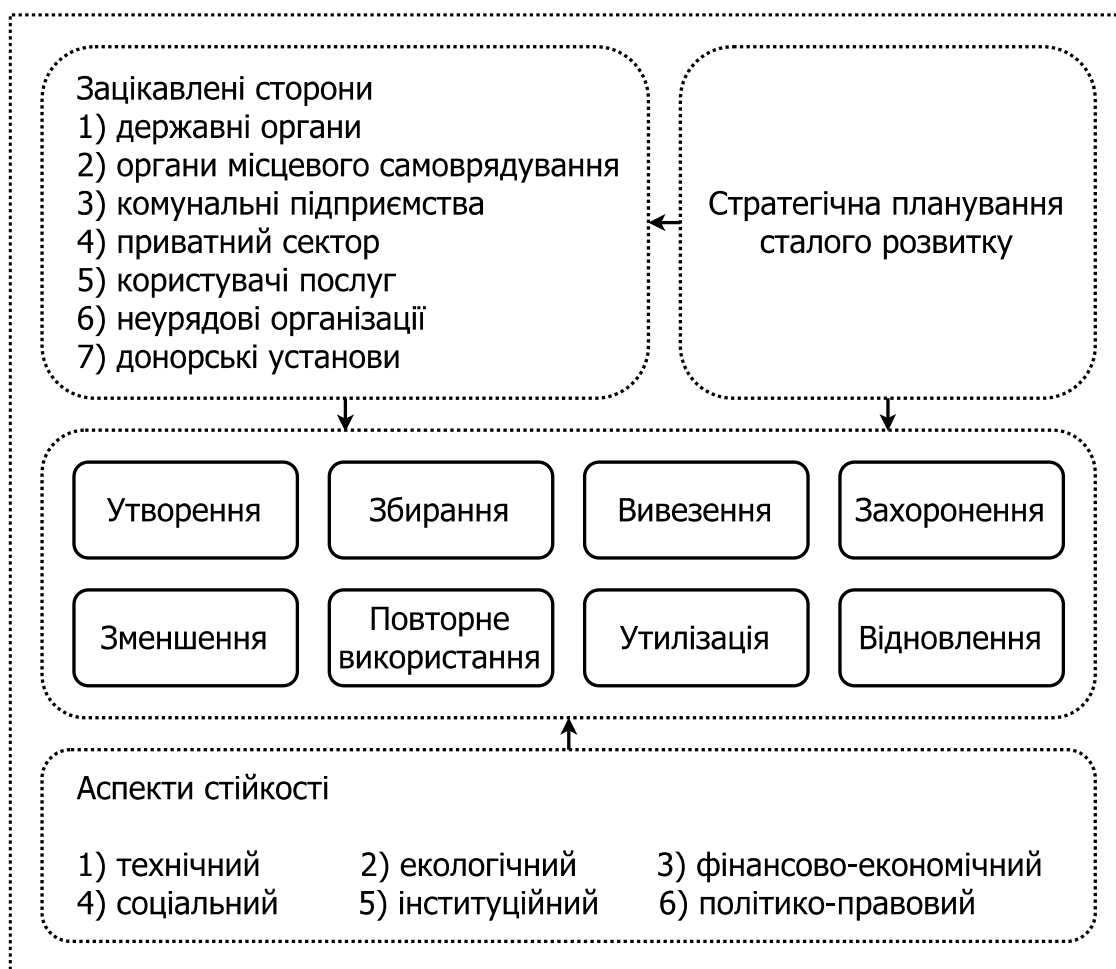


Рисунок 1.3 – Спрощена концептуальна модель інтегрованого управління побутовими відходами

Підкреслимо рівновагу трьох рівнів системи. Зазначимо, що ланцюг технічних елементів є лише одним із рівноцінних вимірів. Перший рівень під назвою “зацікавлені сторони” складається з різних учасників, що мають відношення до управління відходами. Виділимо окремо неформальний приватний сектор. Так, сьогодні в Україні в системі управління ПВ фактично присутні особи, що здійснюють вибір вторинної сировини з контейнерів та на звалищах, так само як і є підприємці, що неофіційно купують відібрану вторинну сировину.

Другий рівень (елементи системи управління ПВ) – це технічні компоненти системи управління відходами. На рис. 1.4 приведений ланцюг відображає компоненти традиційного життєвого циклу відходів.

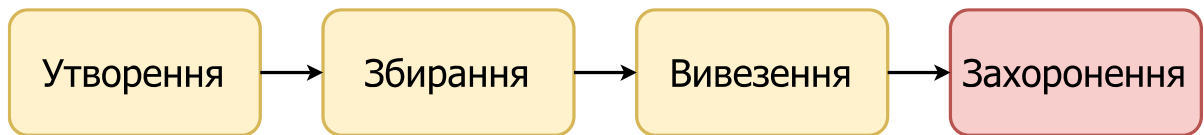


Рисунок 1.4 – Традиційний життєвий цикл управління побутовими відходами

Нижній третій рівень – це сукупність компонентів, що з’являються в процесі модернізації галузі, зокрема й перетворення відходів у товар. Третій рівень (рис. 1.5) відображає аспекти стійкості системи управління ПВ, її здатність стабільно функціонувати протягом невизначеного часу.

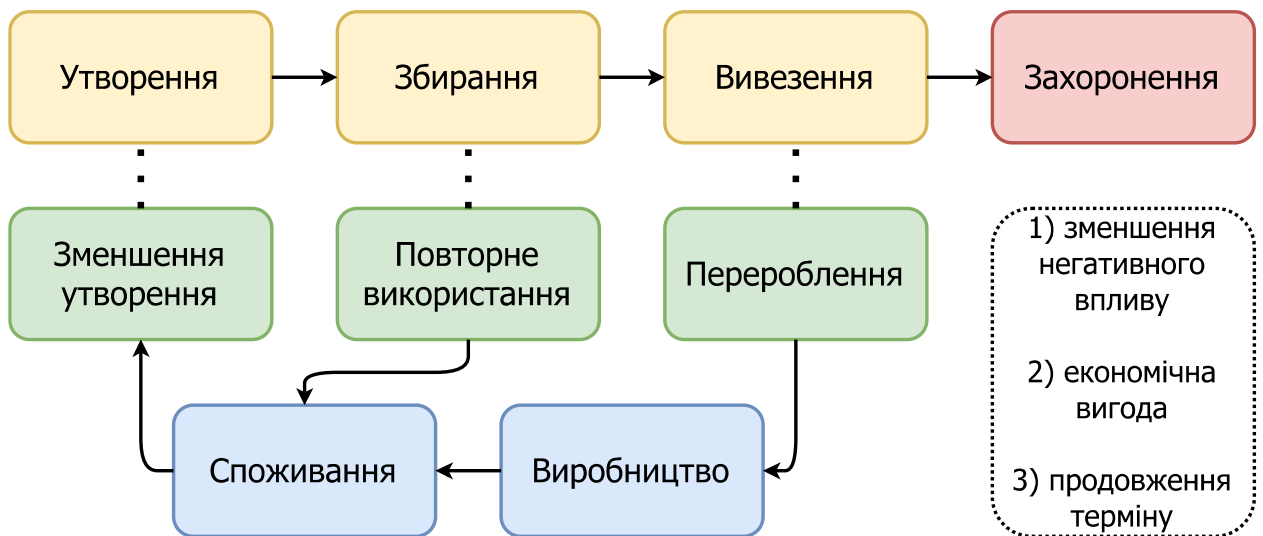


Рисунок 1.5 – Модернізований життєвий цикл управління побутовими відходами на основі принципів сталого розвитку

На сьогодні Хмельницька область усе ще активно використовує застарілий спосіб управління ПВ. Місцеву виконавчу владу регіону не зупиняє й той факт, що із 1 січня 2018 року в Україні запрацювали зміни до Закону «Про відходи» [4], які вказують на сортування сміття та заборону на поховання неперероблених ПВ. Причиною такої незмінної ситуації може бути чинний Закон України «Про відходи», який не має чіткого переліку та послідовності операцій поводження зі сміттям.

Отже, з огляду на накопичений світовий досвід і використовуючи сучасні теоретико-методичні досягнення наукової спільноти, виникає необхідність у

розробленні нових заходів щодо створення та втілення в життя інтегрованої системи поводження з ПВ. Водночас основними напрямками мають бути такі [20]: просвітницька робота поміж населення щодо раціонального поводження з ТПВ, забезпечення сортування та організація централізованого збору та вивезення до місць перероблення, створення виробниками спільних центрів із їхнього перероблення, відновлення й утилізуванню опаковальних відходів (ОВ) із паперу, скла, алюмінію і пластмаси, популяризація виробів з ОВ, стимулювання виробників зменшувати кількість відходів та знижувати їхній шкідливий вплив на довкілля, розроблення інформаційних систем для самообслуговування в процесі утилізуванню ТПВ. Для сталого розвитку регіонів та України загалом необхідно впроваджувати інформаційні системи раціонального поводження з відходами, зважаючи на закордонний досвід.

## 1.2 Огляд сучасних інформаційних систем та підходів до збору та оброблення твердих побутових відходів

На сьогодні потреба в класифікації сміття виникає не лише на спеціалізованих очисних спорудах, де зазвичай збирають відходи, але й у місцях масового скупчення людей. Водночас загальний процес класифікації включає сортування сміття залежно від його типу (класу), розміру, матеріалу тощо [21]. Для розв'язання цього питання на очисних спорудах в основному застосовуються такі підходи:

а) створення конвеєрних систем та підбір персоналу, який здійснює поводження з відходами за необхідними критеріями;

б) налагодження напіваавтоматизованих або автоматизованих сортувальних систем з огляду на певні якості сміття різного типу (температура плавлення, температура горіння, наявність магнетизму тощо);

в) поєднання цих підходів поряд із використанням інших методів.

Кожен із цих методів має певні сильні та слабкі сторони: наприклад, використання трубопроводів для ручного сортування дає змогу забезпечити досить хороший розподіл відходів за необхідними критеріями, але вимагає особливого підходу до персоналу, його підготовки та забезпечення, що вимагає безліч складних заходів і складних процедур. Що стосується систем автоматичного сортування, великою перевагою є їхня складність, автоматизація та продуктивність, але бувають ситуації, коли після взаємодії різних видів відходів їхні властивості змінюються, що ускладнює роботу таких систем і може потребувати додаткових витрат для запобігання зупинкам або порушенням процесів.

Наведемо огляд різних компаній, що пропонують подібні системи на ринку.

Компанія Leanpath є великим постачальником систем відстеження та утилізування ПВ, працює в 32 країнах світу. Упродовж останніх п'яти років із допомогою систем компанії Leanpath відстеження, орієнтований на споживачів бізнес у цих країнах запобіг викидам понад 40 мільйонів тонн різноманітних відходів на звалище [22]. Одна із таких систем, – Leanpath 360 – оснащена камерою, яка встановлена на підвісному екрані, та планшета Android для збору та відображення відповідної інформації, що пов'язана з утилізуванням відходів. Щоразу під час утилізування відходів, оператор має реєструвати тип відходів, причину утилізування та контейнер, у якому вони розташовані. Модель відстеження надсилає зображення з відходами оператору, автоматично класифікує їх, а далі об'єкти відходів зважуються.

Інші компанії, такі як Grace Organic та Winnow, пропонують подібні системи відстеження ТПВ [23]. Grace Organic пропонує різноманітний асортимент засобів – від настільної системи відстеження до великої системи компостування. Зазначимо систему відстеження під назвою «Відстежувач харчових відходів», яка схожа на Leanpath 360, але орієнтована на харчові відходи й не оснащена камерою для знімання їжі. Пропозиція Winnow називається системою Winnow [23]. Система Winnow відстежує ПВ у великому

смітнику. Ця інтегрована система відрізняється від Leapath 360 тим, що не відстежує поодинокі сміттєві контейнери. Коли відходи потрапляють до бака для сміття, система визначає, що саме змінилося в середині контейнера.

Загалом вище згадані системи поводження з ПВ ґрунтуються на методах комп'ютерного зору (КЗ). Спочатку зображення із відходами вибирається із загального набору даних. Може виконуватися певне попереднє оброблення зображення, наприклад, масштабування, обрізання, нормалізація або збільшення [24]. Потім оброблене зображення подається на модуль виявлення ознак об'єкта на зображенні, що може бути представлений, як метод основних компонентів, автодекодер, випадковий ліс або нейронна мережа (НМ). Далі виявлені ознаки об'єкта подаються до навченого класифікатора, як правило, керованого алгоритму навчання, наприклад, метод опорних векторів (МОВ), штучна нейронна мережа (ШНМ), логістична регресія, дискримінантний аналіз або дерево рішень [25]. Класифікатор визначає, до якого класу належить зображення. У випадку класифікації ПВ, клас може бути цілим числом, яке відповідає одній із категорій відходів. Зауважимо, що деякі простіші зображення, які містять менше непов'язаної інформації або фонових шумів, можуть пропустити крок вилучення ознак об'єкта.

На сьогодні одним із найкращих та найрезультативніших методів вилучення ознак на зображенні є згортова нейронна мережа (ЗНМ) [26]. ЗНМ демонструють найвищі результати класифікації на масштабних наборах даних, що часто складаються з різноманітних класів. Однією з причин цього є те, що вони поєднують переваги згортки ядра, що виконує різні типи виявлення особливостей, на багатьох глибоких шарах. На кожному рівні мережі може бути різна кількість ядер (фільтрів). Оскільки мережа навчається на певному наборі зображень, вивчаються ваги фільтрів. Ядра, засвоєні в цьому процесі, часто набагато менші за зображення, а це означає, що вони засвоюють розріджені подання об'єктів, таких, як краї, кути, лінії та навіть текстури [27]. Те саме ядро застосовується на усій мапі вхідних характеристик, що на відміну

від повністю підключеної нейронної мережі, яка має єдиний параметр ваги для кожної точки вхідних даних.

Така особливість ЗНМ називається спільним використанням параметрів [28]. Завдяки їй згорткові шари вимагають менше параметрів, ніж повністю підключені нейронні мережі, а отже, споживають менше пам'яті. Потім вилучені ознаки можна подати в класифікатор наприкінці. Класифікатор, як правило, є НМ прямої передачі, але також може бути іншим типом алгоритму, як згадувалося раніше. Часто класифікатор є просто кінцевим рівнем ЗНМ, тому вони є однією мережею.

Відстеження та класифікації об'єктів ПВ на зображеннях – це значний крок до розв'язання досліджуваної проблеми. Проте деякі сучасні технології вимагають від кінцевого користувача вводити тип об'єкта на зображенні вручну, і тип контейнера, в якому він знаходиться. В середньому пошук за допомогою інтерфейсу користувача для пошуку відповідної категорії товарів може тривати приблизно 3-5 секунд.

### 1.3 Порівняльний аналіз методів та систем класифікації зображень

Google Cloud AutoML Vision – це набір продуктів машинного навчання, який дозволяє розробникам з обмеженим досвідом машинного навчання навчати високоякісні моделі, специфічні для їхніх бізнес-потреб. Він спирається на найсучасніші технології навчання передачі даних та технологію пошуку нейронної архітектури. На рис. 1.6 зображено приклад використання системи Google Cloud AutoML Vision.

Основним принципом роботи AutoML є класифікація вмісту за категоріями та присвоєння різних міток усьому, що він «бачить». Google Cloud AutoML можна використовувати для взаємодії застосунків з камерами або AR. Іншими словами, ця послуга була розроблена таким чином, що вона дозволяє

тренувати моделі машинного навчання, щоб класифікувати ваші зображення відповідно до власних визначених міток. Через те, що для здійснення таких операцій використовується хмарна інфраструктура Google, користувачеві потрібно зареєструватися в системі та вибрати ціну, яка відповідає орієнтованому підрахунку зображень, що аналізуються.

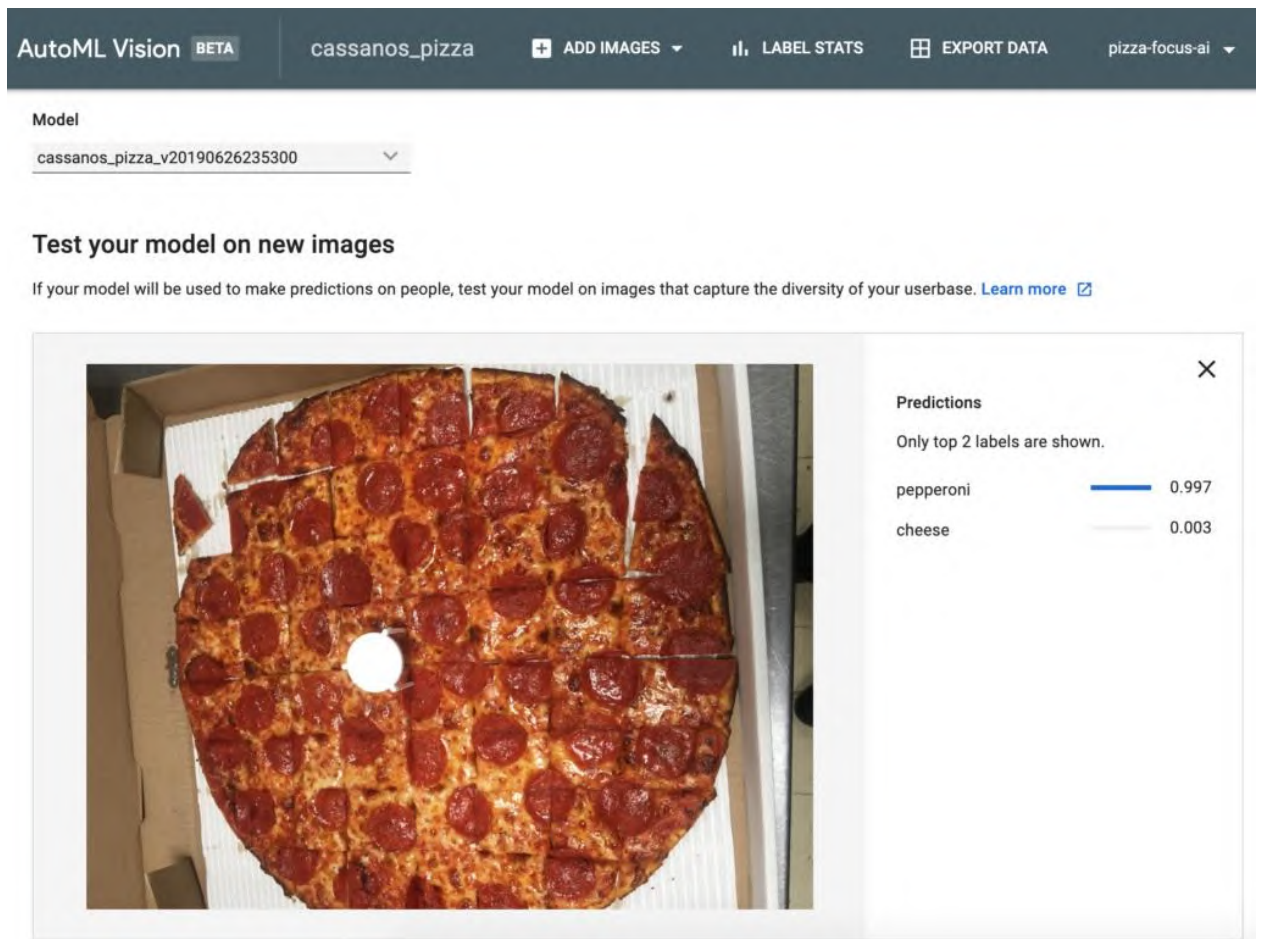


Рисунок 1.6 – Приклад класифікації їжі з допомогою системи Google Cloud AutoML Vision [29]

За своїм позиціонуванням API Google Cloud Vision досить близький до попереднього продукту, Google Cloud AutoML Vision. Головна відмінність між ними полягає в тому, що Vision API створений для пропонування попередньо навчених моделей машинного навчання за допомогою API REST та RPC.

Основним принципом Cloud Vision API вважається присвоєння міток вхідним зображенням та подальша класифікація за власними визначеними

категоріями. Це рішення дозволяє виявляти предмети та обличчя, читати друкований та рукописний текст та працювати з метаданими. Класифіковані елементи Google Cloud Vision наведено на рис. 1.7.

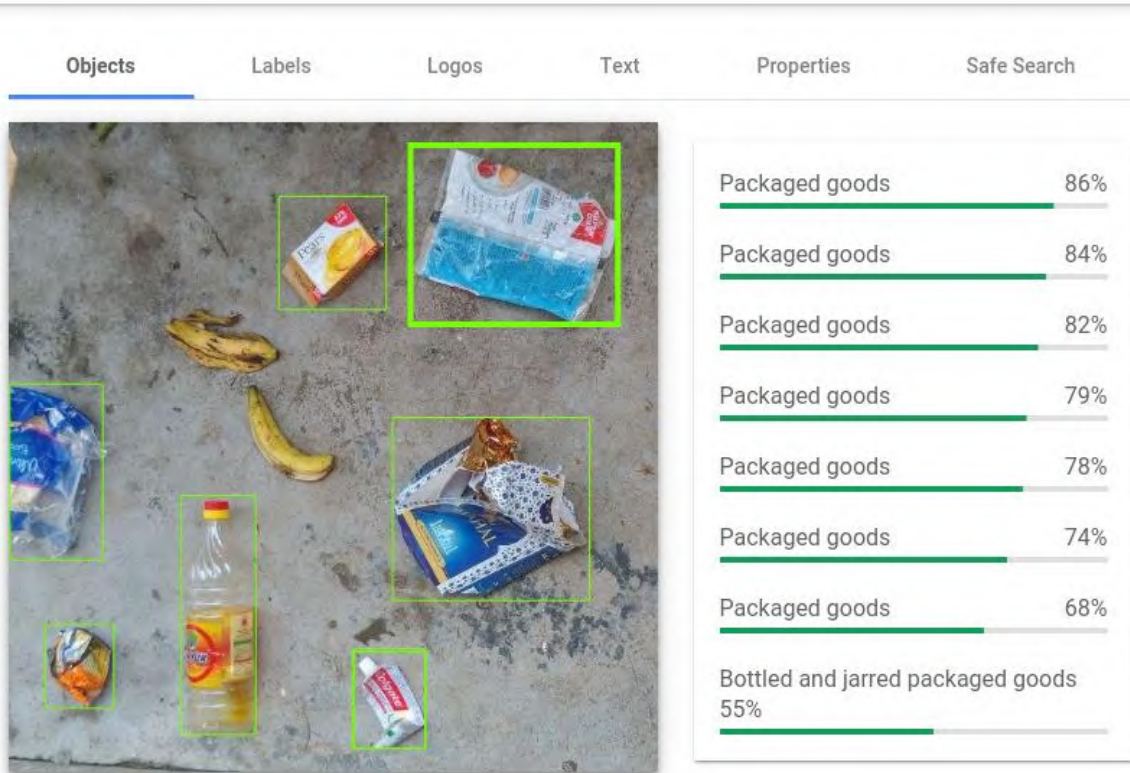


Рисунок 1.7 – Приклад ярликів класифікації, що виконані у системі Google Cloud Vision AI [30]

Amazon Rekognition базується на тій же самій перевірній, масштабованій, глибокій технології навчання, розробленій вченими комп'ютерного зору Amazon для аналізу мільярдів зображень та відео щодня, і не вимагає використання знань машинного навчання.

Amazon Rekognition – це простий і простий у використанні API, який дозволяє швидко проаналізувати будь-які зображення або відеофайли, що зберігаються в Amazon S3. Amazon Rekognition завжди вчиться на нових даних. Основне завдання служби Amazon Rekognition – додати аналіз зображень та відео до програм клієнтів, як Rekognition API. Переваги, про які Amazon згадує на своїй сторінці: проста інтеграція; постійне навчання; повністю керований;



На рис. 1.9 продемонстровано роботу IBM Watson Visual Recognition.

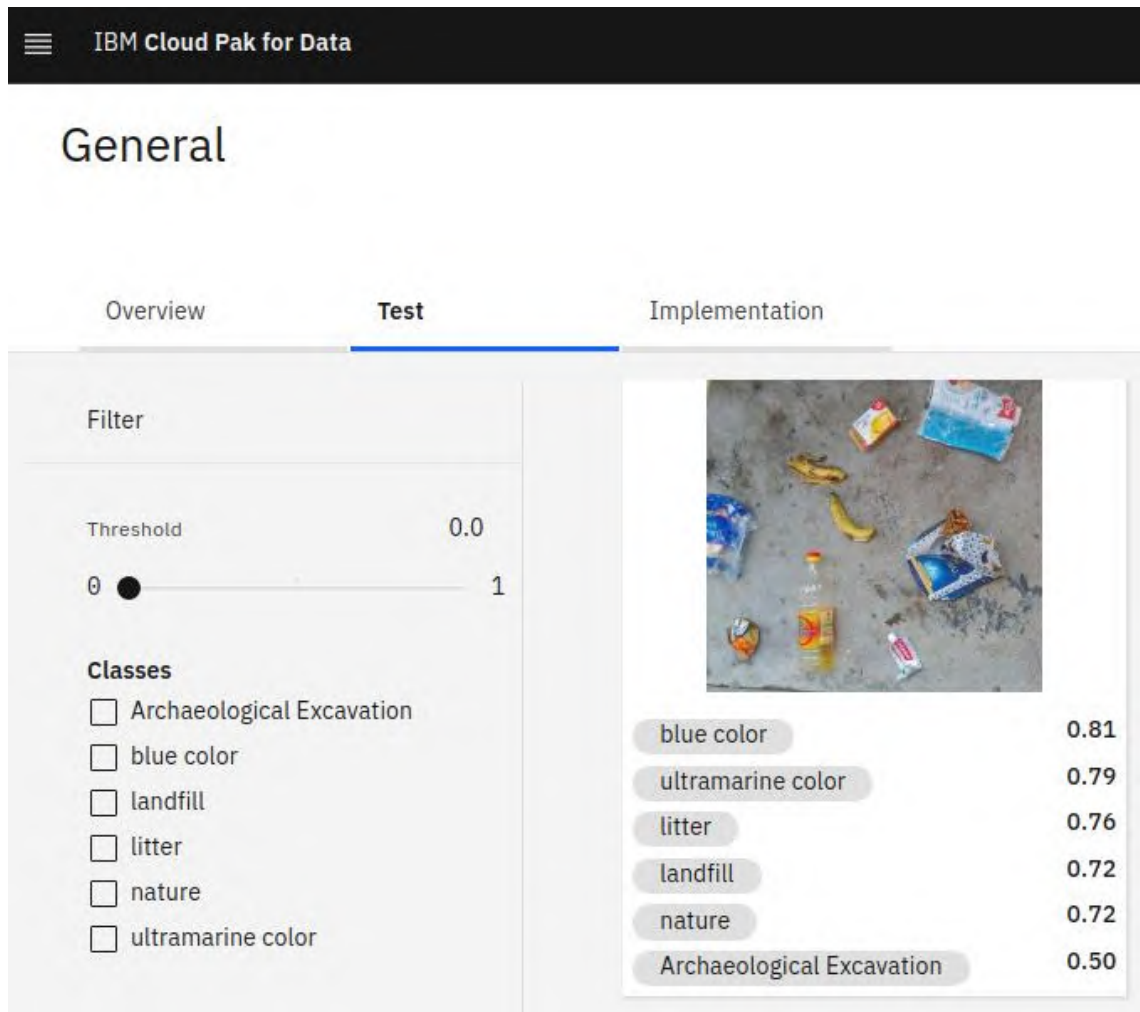


Рисунок 1.9 – Приклад системи класифікації зображень побутових відходів у системі IBM Watson Visual Recognition [32]

Imagga – це компанія, яка надає певні API для різних цілей оброблення зображень. Вони роблять акцент на своїй технології машинного навчання, що налаштовується. На своєму веб-сайті вони сформулювали свою місію забезпечити найпотужніший і найточніший набір технологій, що дозволяє різним розмірам та масштабам отримувати користь від вмісту зображень, яким він управляє. Вони також зосереджуються на покращенні внутрішніх робочих процесів та автоматизації, кращих статистичних даних, кращому взаємодії з користувачами для кінцевих користувачів послуг своїх клієнтів. Приклад виконання завдання класифікації в системі Imagga представлено на рис. 1.10.

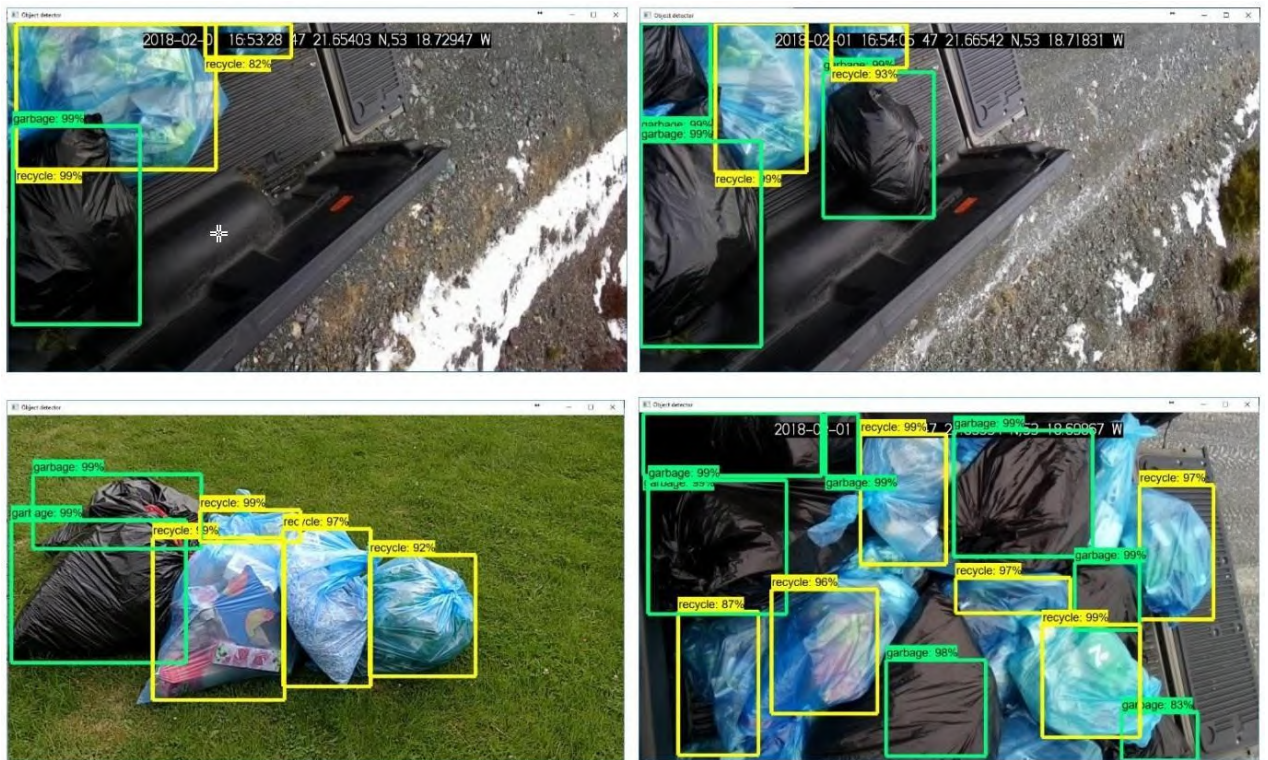


Рисунок 1.10 – Приклад детектування сміттєвих мішків у системі Imagga [33]

Завдяки вмісту веб-сайту вони можуть допомогти у наданні таких рішень:

- API тегування;
- API категоризації;
- API обрізання;
- API кольору;
- категоризація NSFW;
- спеціальна категоризація;

Також розглядувані вище системи використовують такі технології

- автоматичне позначення;
- автокатегоризація;
- автообрізання;
- вилучення ознак кольору.

Проведено порівняльний аналіз вище згаданих систем класифікації для визначення переваг та недоліків кожної із них. Результати порівняння наведено у табл. 1.2.

Таблиця 1.2 – Порівняльний аналіз систем класифікації зображень

Системи класифікації	Класифікація за попередньо визначеними мітками	Підтримка мульти-класовості	Офлайн режим	Легкість впровадження	Безоплатність
Google Cloud AutoML Vision	Ні	Так	Ні	Так	Ні
Google Cloud Vision API	Так	Ні	Ні	Так	Ні
IBM Watson Visual Recognition	Так	Так	Ні	Так	Ні
Amazon Rekognition	Так	Так	Ні	Так	Ні
Imagga	Так	Так	Ні	Так	Ні
Pre-trained neural networks	Так	Так	Так	Ні	Так

Програмне забезпечення в такому випадку є сервісом, реалізація якого відбувається на сервері компанії й використовуються після реєстрації. У деяких випадках сервіси забезпечують функціональність для класифікації зображень на основі заздалегідь визначених міток, в інших – створити власну систему специфікації на основі поточних вимог і змінити її пізніше. Такий підхід дає змогу абстрагуватися від таких речей, як процес розгортання, підтримування програмного забезпечення та виправлення помилок. Але в обох випадках користувача обмежує той факт, що для використання таких систем вам знадобляться дві речі:

- стабільне та швидке підключення до інтернету;
- оплата за перевищення ліміту даних або в разі потреби додаткових функцій оброблення зображень.

Водночас класифікація об'єктів на зображенні включає широкий спектр теоретичних підходів до прийняття рішень до ідентифікації зображень (або їхніх частин). Усі класифікаційні алгоритми базуються на припущенні, що розглянуте зображення відображає одну або кілька ознак (наприклад,

геометричні частини у випадку виробничої системи класифікації або спектральні області у випадку дистанційного зондування) і що кожна з цих функцій належить до одного з кількох ексклюзивних класів. Класи можуть бути призначені заздалегідь (як у контрольованій класифікації) або автоматично згенеровані (тобто, як за неконтрольованої класифікації) на набори класів-прототипів, де лише вказується кількість бажаних категорій.

Отже, під інформаційною системою класифікації ТПВ розуміємо модуль, програмний прототип або бібліотеку. Зокрема, виникає потреба в постійній адаптації наявних елементів для ефективного використання в системі; їхній модифікації та навчанні виконувати різні операції. З іншого боку, обмеження здебільшого пов'язані з архітектурою рішень, необхідним обладнанням та забезпеченням задовільної кількості даних про якість.

#### 1.4 Проблема поводження з відходами на прикладі діяльності Філії «Старокостянтинівський завод залізобетонних шпал»

На підприємстві Філія «Старокостянтинівський завод залізобетонних шпал» прийнята агрегатно поточна схема виробництва, за якої основним технологічним обладнанням на технологічній лінії є силова металева 10-гніздова форма, яка з допомогою роликів конвеєрів переміщається від одного технологічного посту на інший із різним циклом перебування на них.

Підприємство складається з таких підрозділів:

- об'єднаний формувально-бетонний цех;
- ремонтний паросиловий цех;
- транспортно-сировинний цех;
- ремонтно-будівельна діляниця;
- паро-котельна діляниця;
- відділ технічного контролю й лабораторія.

Виготовлення основної продукції складається з таких стадій:

- виготовлення бетонної суміші;
- встановлення пустотоутворювачів у форму;
- виготовлення та встановлення струнопакетів та їхня натяжка;
- виймання пустотоутворювачів і діафрагм на посту мийки діафрагм і деталей пустотоутворювачів;
- пропарювання виробів у камерах пропарювання;
- зняття напруження й передача його на вироби;
- розрізання плітів шпал та штабелювання шпал у штабелі.

Під час проходження науково-дослідної практики на ПАТ «Укрзалізниця» Філія «Старокостянтинівський завод залізобетонних шпал» визначено, що підприємство здійснює викиди різноманітних речовин у довкілля. Типи викидів наведено в табл. 1.3.

Таблиця 1.3 – Структура впливу Філії «Старокостянтинівський завод залізобетонних шпал» на навколишнє середовище

Постачання сировини, енергії, інших видів природних ресурсів		Виробництво продукції, надання послуг		Реалізація продукції, послуг	
Види сировини	Викиди забруднюючих речовин	Види виробленої продукції	Викиди, забруднюючих речовин	Види реалізованої продукції	Викиди забруднюючих речовин
– деревина; – пісок; – щебінь; – цемент; – природний газ; – вугілля; – вода; – бензин; – дизельне паливо.	– забруднення атмосферного повітря транспортними засобами при постачанні сировини на виробництво; – зменшення обсягів природних ресурсів, що використовуються у процесі виробництва (вода, деревина, пісок тощо.)	У процесі виробництва залізобетонних шпал і на всіх стадіях виробництва утворюється значна кількість відходів і забруднювачів навколишнього природного середовища	– газоподібні викиди (пил неорганічний, пил, бензин, сірчаний ангідрид, метан); – відходи (люмінесцентні лампи, ТПВ, відпрацьовані масла, шини, масляні фільтри)	Доставка продукції замовникам	– забруднення атмосферного повітря транспортними засобами, що доставляють продукцію; – шумове забруднення довкілля; – відходи, що утворюються при упакуванні продукції.

З проведеного діагностування підприємства впливає актуальність реалізації інформаційної системи для поводження з відходами та впровадження станції сортування відходів. Під час проходження практики на підприємстві ПАТ «Укрзалізниця» Філія «Старокостянтинівський завод залізобетонних шпал» були виявлені такі проблеми щодо поводження з відходами:

- на підприємстві немає єдиної стратегії поводження як з промисловими, так і з побутовими відходами;
- працівники підприємства не мають достатньо інформації щодо класів побутових відходів;
- наявні на підприємстві станції сортування не призначені для раціонального сортування ТПВ;
- система сортування відходів, що розробляється на підприємстві, не має зручного користувацького інтерфейсу.

Отже, метою дисертаційного дослідження є побудова моделі класифікації твердих побутових відходів та розроблення відповідної клієнт-серверної інформаційної системи управління побутовими відходами.

### 1.5 Постановка задачі моделювання та реалізації класифікатора твердих побутових відходів

Оброблення зображень є однією з популярних застосувань методології аналізу зображень. Однак побудувати автоматизовану систему, яка за своїми класифікаційними здібностями наближається до людських – не тривіальне завдання. У результаті проведеного аналізу джерел досліджень та сучасних систем перероблення ТПВ визначено та описано процес класифікації. Виявлено деталі сучасних широко використовуваних нейронних мереж, проаналізовано їхні типи та описано одну з найкращих моделей для розв'язання розглядуваної проблеми, а саме згорткові нейронні мережі.

Також встановлено, що різні наявні рішення забезпечують різну функціональність для класифікації даних, особливо розпізнавання зображень. Однак, жоден із розглянутих сервісів не забезпечує повний цикл оброблення зображень ТПВ, потребує постійного зв'язку з Інтернет-АРІ та використання хмарної інфраструктури. Крім того, пропріетарні сервіси ґрунтуються на заздалегідь визначених методах та алгоритмах КЗ і не забезпечують гнучкості налаштування моделей.

Для розв'язання задачі класифікації зображень ТПВ проведено такі роботи: виконано формальну постановку задачі; проведено порівняльний аналіз систем-аналогів; здійснено огляд нейронних мереж, як засіб класифікації зображень ТПВ. На рис. 1.11 вказано основні вимоги до клієнт-серверного програмного забезпечення для класифікації зображень ТПВ.

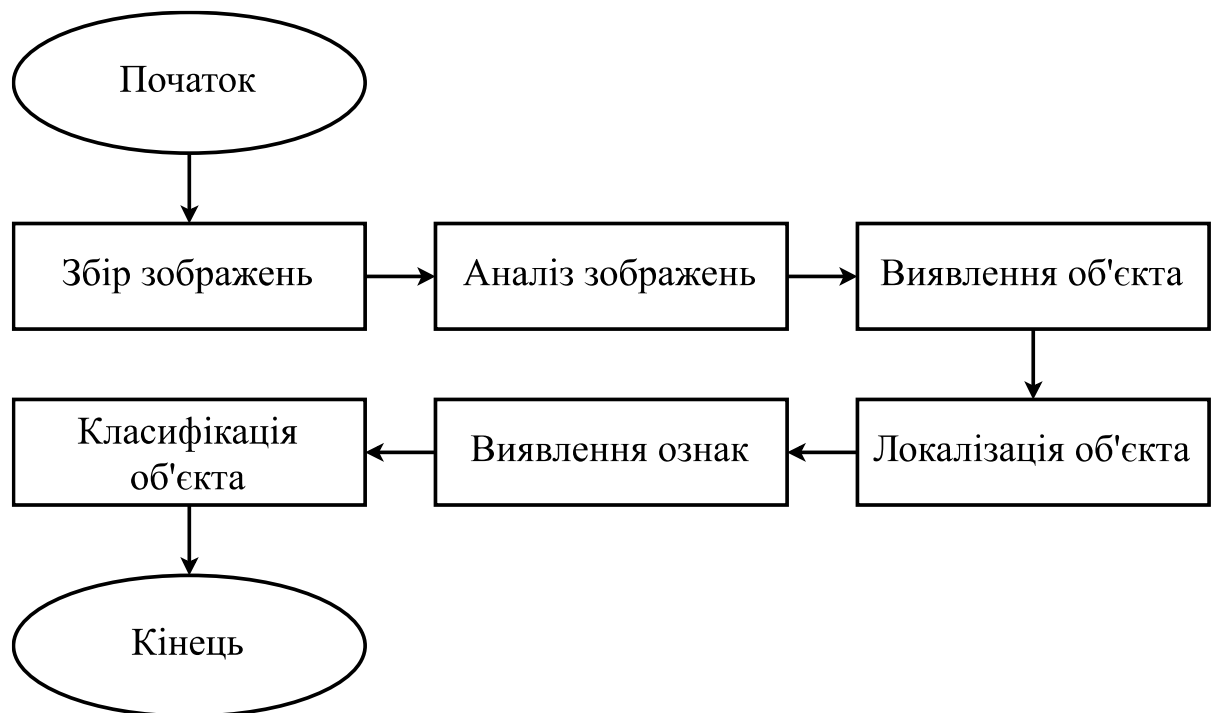


Рисунок 1.11 – Блок-схема інформаційної системи класифікації зображень твердих побутових відходів

Це програмне забезпечення повинно задовольняти такі етапи функціонування:

- користувач вибирає зображення, яке потрібно класифікувати;

- система завантажує зображення;
- зображення обробляється для проведення класифікації;
- система здійснює класифікацію та повертає результат.

Розв’язання цієї задачі можливе з використанням методів нейромережевого моделювання, які дають високу точність класифікації за невеликої кількості зображень у навчальних наборах даних.

## Висновки до розділу 1

Отже, у результаті проведеного аналізу сучасного стану поводження з відходами виявлено низький рівень сортування населенням сміття та низьку ефективність утилізування. Зокрема, на території нашої держави все ще немає єдиної інтегрованої системи управління відходами, що дала б змогу ефективно утилізувати як промислові, так і тверді побутові відходи.

Здійснено огляд сучасних інформаційних технологій та підходів до збору та оброблення твердих побутових відходів. Висунуто припущення, що інформаційна технологія на основі методів комп’ютерного зору може стати ефективним та дієвим рішенням до розв’язання проблеми поводження з відходами. Водночас під час проходження науково-дослідної практики на підприємстві ПАТ «Укрзалізниця» Філія «Старокостянтинівський завод залізобетонних шпал» виявлено низку проблем поводження з відходами, розв’язати які може інформаційна система із простим та зрозумілим користувацьким інтерфейсом.

Крім того, проаналізовано сучасні комерційні сервіси, що надають послуги з класифікації зображень. На основі проведеного порівняльного аналізу сучасних методів та підходів до класифікації зображень розроблено алгоритм реалізації системи класифікації зображень ТПВ, що наведено на рис. 1.11. Насамкінець, за можливу модель класифікації зображень обрано згорткову нейронну мережу.

## 2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ КЛАСИФІКАЦІЇ ТВЕРДИХ ПОБУТОВИХ ВІДХОДІВ

### 2.1 Опис класифікатора на основі згорткової нейронної мережі

На основі проведеного порівняльного аналізу методів та алгоритмів КЗ, а класифікатор зображень обрано ЗНМ. Цей метод забезпечує високий рівень точності класифікації та гнучкість проведення модифікацій. Крім того, у мережі наявні готові рішення з відкритим кодом.

Згорткові нейронні мережі – це глибокі штучні нейронні мережі, які в основному використовуються для розпізнавання, класифікації та сегментації об'єктів. Використовуючи зображення, як джерело даних, вони можуть виявити наявність заздалегідь визначеної інформації та працювати з нею (наприклад, знаходити на зображеннях визначені об'єкти та виявляти їхні параметри та поведінку), групувати їх подібність та знаходити кореляцію даних між іншими зображеннями (пошук фото) та виконувати безліч інших різних дій залежно від засобів, що використовує висновок згорткової нейронної мережі.

Мережа виконує оптичне розпізнавання символів (ОПС) для оцифрування тексту та дає змогу обробляти природну мову в аналогових та ручних документах, де зображення переписують символи. ЗНМ також можна застосовувати до звуку, коли він візуально представлений у вигляді спектрограми. Наразі такі мережі часто використовують для аналізу тексту, а також графіки даних за згодою мережевого графіка.

Ефективність згорткових нейронних мереж саме в розпізнаванні зображень є однією з головних причин теперішньої популярності та ефективності методів глибокого навчання. Вони показують суттєвий прогрес у різних галузях КЗ, як от впровадження автопілотів, відеореєстратори, робототехніка, медична діагностика, системи прийняття рішень тощо.

Застосування класифікації зображень передбачає мінімізацію цільової функції, яка в даному випадку називається функцією втрат. Ця функція використовується для вимірювання помилки під час прогнозування. У роботі використано L2-середньоквадратичну помилку, як функцію втрат виду

$$\mathcal{L}(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^N (\hat{f}(x_i) - y_i)^2 + \frac{\lambda}{2N} \sum_{i=1}^D \|w_i\|_2^2, \quad (2.1)$$

де  $w_i \in \mathbf{w} \in \mathbb{R}^D$  – множина ваг моделі,  $D$  – розмір зображень у пікселях,  $x_i \in \mathbf{X} \in \mathbb{R}^{N \times D}$  – множина зображень навчального набору даних;  $y_i \in \mathbf{y} \in \mathbb{R}^N$  – множина класів,  $N$  – кількість класів,  $\hat{f}(x_i)$  – цільова функції апроксимації,  $\lambda$  – параметр балансування.

Для виявлення ознак об'єктів на зображенні у даній роботі використано згорткову нейронну мережу з операцією згортки:

$$(K \times I)(u, v) = \sum_{k=-M}^M \sum_{q=-M}^M K(k, q) I(u + k, v + q), \quad (2.2)$$

де  $K(k, q)$  – ядро згорткового шару,  $M$  – кількість пікселів згорткового ядра,  $I(u + k, v + q)$  – мапа ознак на цільовому зображенні,  $u$  та  $v$  – позиція пікселя на мапі зображення,  $k$  та  $q$  – зміщення згорткового ядра по координатній площині зображення.

Нижче наведено перелік переваг ЗНМ, що є ключовими для задач розпізнавання зображень:

а) комплексний аналіз вигляду завдяки ядру згортки; масив чисел, що використовується ядром, змінюється в процесі навчання ЗНМ, що дає змогу такій мережі адаптуватися до різних моделей, які можна знайти під час згортки;

б) менша кількість ваг; через використання згорткової операції ЗНМ не потрібно з'єднувати кожен піксель один з одним;

в) зменшення розмірів; корисна перевага перед класичними нейронними мережами через складність операцій із величезними тензорами.

На додаток до згаданих раніше згорткових шарів, ЗНМ також використовує рівень вибірок. Субдискретизаційний шар чергується із згортковим шаром: замість того, щоби кілька разів переміщати ціле зображення із використанням деяких ядер, він просто ділить зображення на деякі квадрати даних (зазвичай розміром як  $2 \times 2$  або  $3 \times 3$ ) і порівнює кожен елемент всередині, щоби знайти найбільший. Як результат, розмірний розмір можна зменшити в кілька разів. Схематичне зображення описаного вище процесу показано на рис. 2.1.

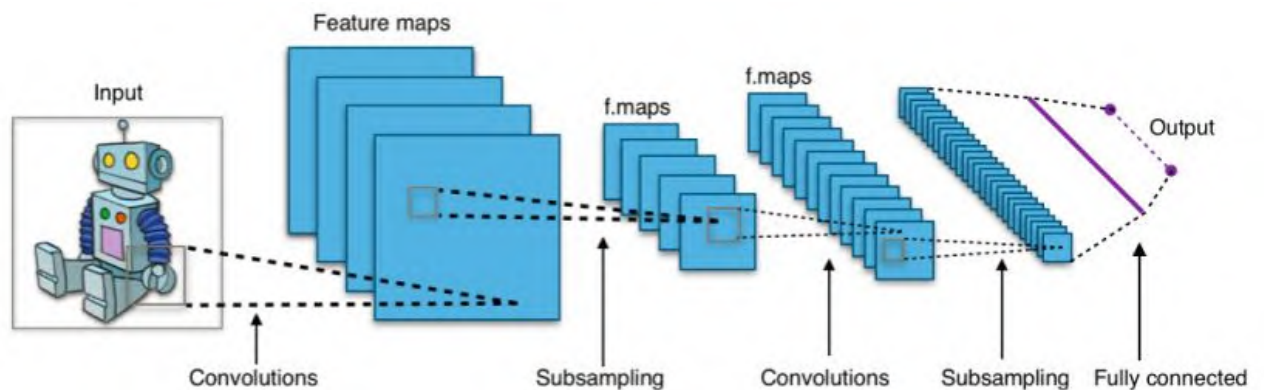


Рисунок 2.1 – Схематичне зображення топології ЗНМ [28]

Через те, що кожен етап виконання згортки зменшує розміри шару і вибирає лише цінні візуальні абстрактні ознаки, алгоритм виконання ЗНМ, як правило, менш залежить від таких речей, як конкретний стан зображення (кут, яскравість, контраст тощо).

На сьогодні модель ЗНМ під назвою Faster R-CNN демонструє найкращі результати виявлення об'єктів на зображеннях. Регіональна мережа з пропозиціями (RPN) [31] в складі Faster R-CNN розділяє повнорозмірні згорткові функції, що дає змогу виділяти контури зображень без додаткових втрат ресурсів. RPN – це повністю згорткова НМ, яка виділяє контури об'єктів та передбачає оцінки об'єктивності в кожній позиції. Підмережа RPN навчена наскрізно, щоби генерувати високоякісні контури, які далі використовуються

загальною мережею Faster R-CNN для виявлення ознак. На рис. 2.2 зображена топологія нейронної мережі типу Faster R-CNN з регіональними пропозиціями.

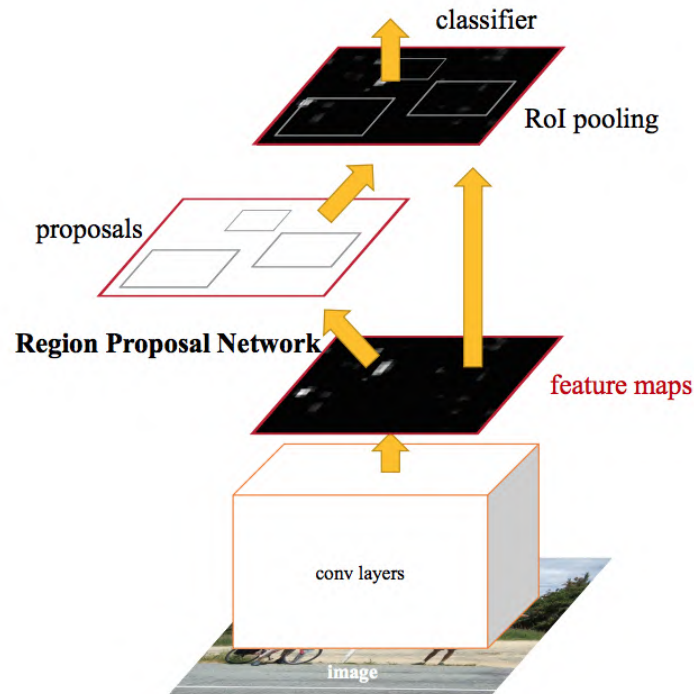


Рисунок 2.2 – Узагальнена схема архітектури Faster R-CNN [34]

Нейронна мережа Faster R-CNN складена в такий спосіб, що не потребує великого набору зображень для ефективного навчання. Використаний у роботі класифікатор отримує на вхід зображення розміром  $768 \times 1024$  із двома або більше об'єктами на звичайному фоні. Водночас на кожному зображенні стандартний фон видаляється, щоби виділити обмежувальне поле навколо кожного об'єкта. Далі на зображення накладається білий фон розміром  $768 \times 1024$ , що об'єднується з 2–6 об'єктами відходів у випадкових місцях. Ці об'єкти можуть перекриватися або перетинатися та мати різний розмір. Загалом модель класифікації налаштовано у такий спосіб, що вона виявляє регіональні пропозиції та класифікує об'єкти за трьома класами: звалище, вторинне перероблення та папір.

За базову архітектуру в роботі використано попередньо навчену модель Faster R-CNN зі зміненими останніми повнозв'язними шарами. Тобто були

використані попередньо навчені функції нижчого рівня. Модель навчено на наборі даних із 2000 зображень; кожне зображення із рівномірним поданням кожного класу на різних прикладах. У роботі використано середню точність (*avg\_prec*) для оцінювання продуктивності роботи моделі. Формалізований запис *avg\_prec* має вигляд:

$$avg\_prec = \int_0^1 p(x) dx. \quad (2.3)$$

Водночас крива точності та повноти визначається через обчислення IoU (перетину об'єднання) між передбачуваними та істинними значеннями класифікації. Також у роботі використано функцію втрат:

$$L(p_i, t_i) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \frac{\lambda}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*). \quad (2.4)$$

Функція втрат (2.1) є сумою двох нормованих втрат над регіонами зображення з індексами  $i$ , де  $p_i$  – ймовірність представлення регіоном об'єкта  $i$ ,  $p_i^* \in \{0,1\}$  – істинна інформація про те, чи є регіон насправді об'єктом чи ні,  $t_i$  – координати запропонованої моделі обмежувальної рамки,  $t_i^*$  – істинні координати об'єкта.

Перша частина класифікатора повинна виконати оброблення тензором зображень, включаючи згортку, об'єднання, фільтрацію тощо. Під час її роботи будуть відняті деякі загальні та корисні для моделей класифікації дані. Зрештою, він повинен містити шар, який буде вирівнювати дані для подальшого використання частини ЗНМ. Друга частина класифікатора використовує результати першої частини мережі та на їх основі виконувати передбачення класу зображення. Для даної задачі кількість класів відходів становить 6, тому вихідний шар класифікатора містить 6 відповідних вузлів, кожен для свого класу.

В результаті повинен бути отриманий масив з числовою ймовірністю зображення, що належить до певного класу. Цей результат пізніше повинен використовуватися іншими частинами програми для виконання інших операцій (ці результати потенційно можуть зберігатися та використовуватися для статистики або залежно від них можуть бути зроблені різні дії).

Типовим та ефективним підходом до глибокого вивчення зображень на малих наборах є використання заздалегідь підготовленої мережі. Попередньо підготовлена мережа – це мережа, що зберігається, яка, перш за все, попередньо підготовлена на великому наборі даних, зазвичай як частина великого завдання з класифікації зображень. Якщо цей початковий набір даних досить великий і високо узагальнений, то просторова ієрархія шаблонів може ефективно діяти як узагальнена модель видимого світу і бути корисною для багатьох різних завдань розпізнавання зображень, навіть якщо ці нові завдання пов'язані з дуже різними класами, відмінними від класів у початковому завданні.

Іншими словами, є можливість навчити мережу зображень із ImageNet (де переважна більшість класів – тварини та предмети домашнього вжитку), а потім використовувати цю навчену мережу для виявлення чогось іншого, наприклад, предметів меблів на рисунках. Ця портативність отриманих результатів між різними цілями є основною перевагою глибокого навчання перед багатьма старими хитрощами наземного навчання, що робить глибоке навчання дуже ефективним інструментом для вирішення завдань з низьким обсягом.

## 2.2 Проектування архітектури нейронної мережі типу Faster R-CNN

У цьому пункті описано процес проектування класифікатора зображень. Відповідно до мети дипломної роботи, розроблено блок-схему (рис. 2.3) реалізації інформаційної системи для класифікації зображень ТПВ.

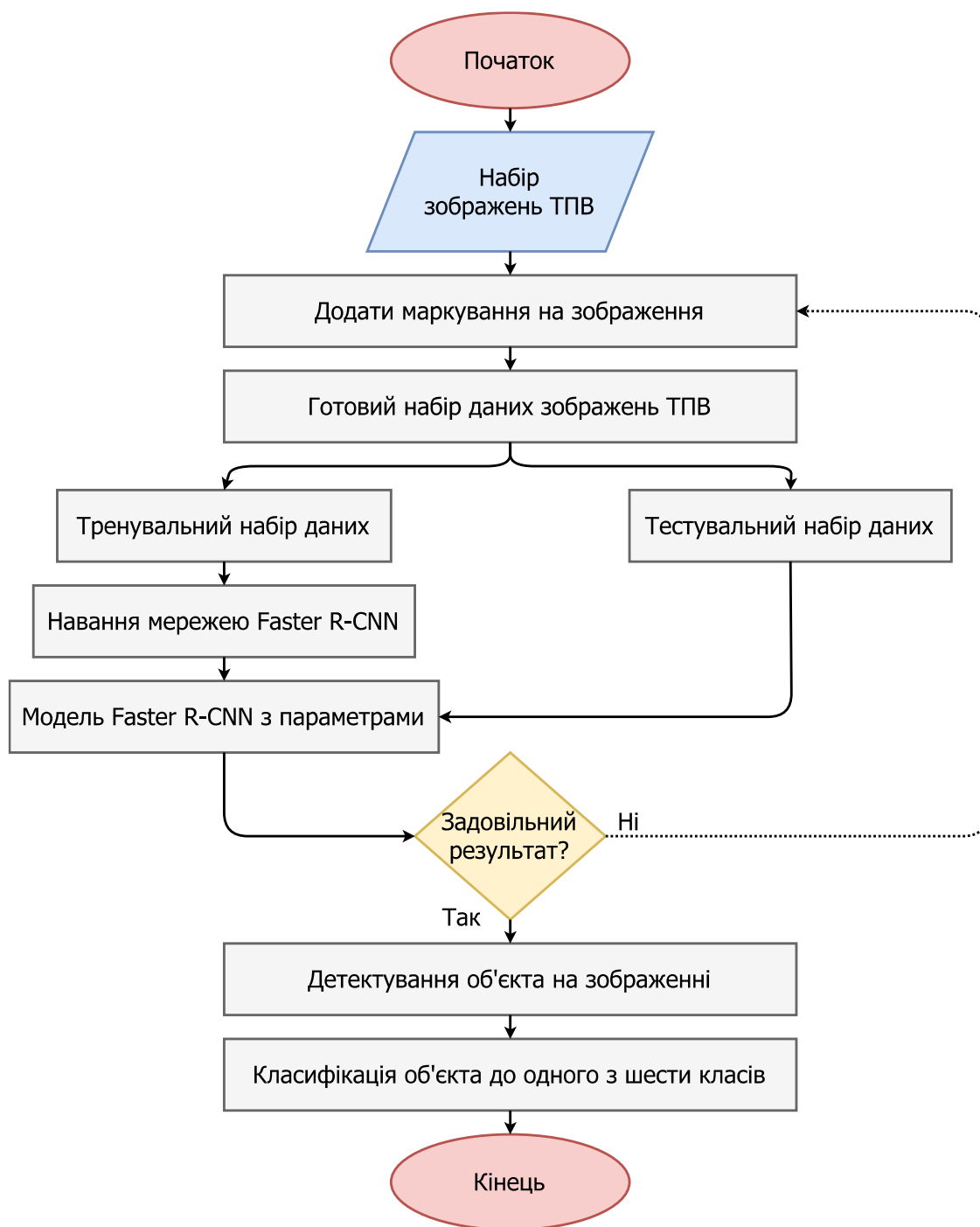


Рисунок 2.3 – Блок-схема оброблення, виявлення та класифікації зображень твердих побутових відходів

У роботі використано архітектура ЗНМ типу Faster R-CNN, як детектор об'єктів на зображенні та класифікатор. Архітектуру ЗНМ, що використана у даній роботі, зображено на рис. 2.4.

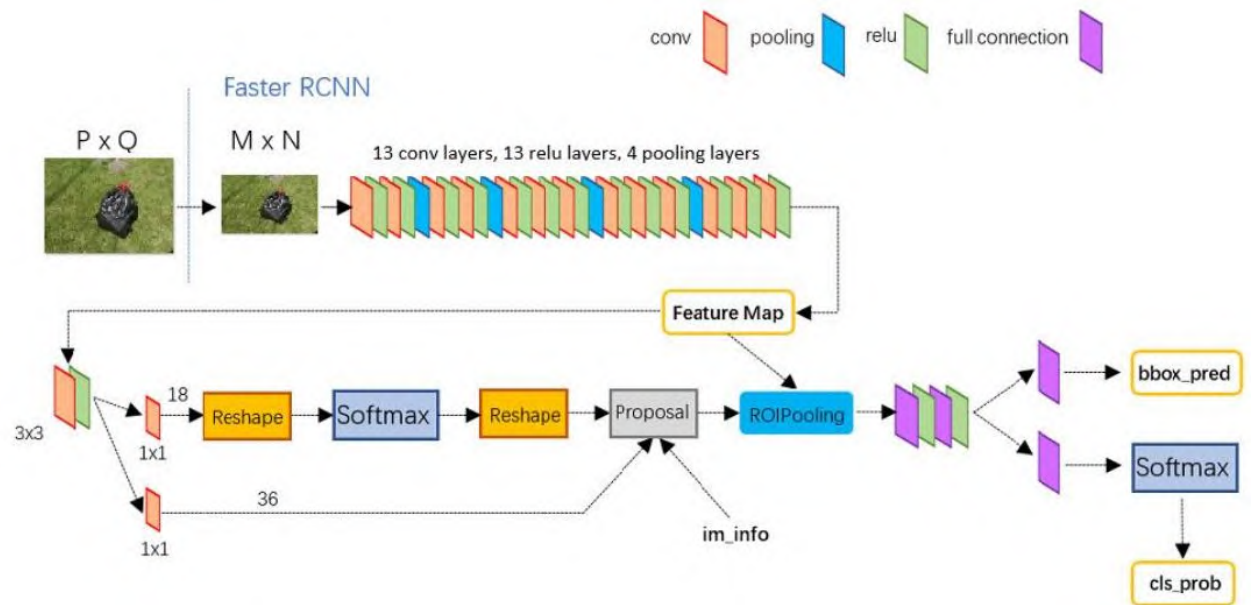


Рисунок 2.4 – Схематичне зображення архітектури ЗНМ типу Fast R-CNN, що використана для реалізації Телеграм-боту [34]

На вхід системи надходить зображення довільного розміру  $P \times Q$ . Система зжимає зображення до фіксованого розміру  $M \times N$  та надсилає його до мережі. У цій роботі ЗНМ містить 13 згорткових шарів, 13 активаційних шарів типу ReLU та 4 шари об'єднання, що розташовані відразу після шарів згортки. Мережа RPN спочатку зазнає згортки  $3 \times 3$ , а потім генерує якори переднього плану та зсуви регресії обмежувальної рамки відповідно, а потім обчислює пропозиції. Рівень регіону інтересу використовує пропозицію для вилучення функції пропозиції з карт функцій та надсилає її до наступної мережі повного зв'язку та вихідного шару softmax для класифікації.

Класифікатор на основі Faster R-CNN відкидає традиційні розсувні і безпосередньо використовує RPN для генерації кадру виявлення. Цей метод також є значною перевагою Faster R-CNN, який може значно покращити швидкість генерації кадру виявлення. ЗНМ типу Faster R-CNN складається з кількох частин.

Мережа RPN. Схематичне зображення RPN наведено на рис. 2.5.

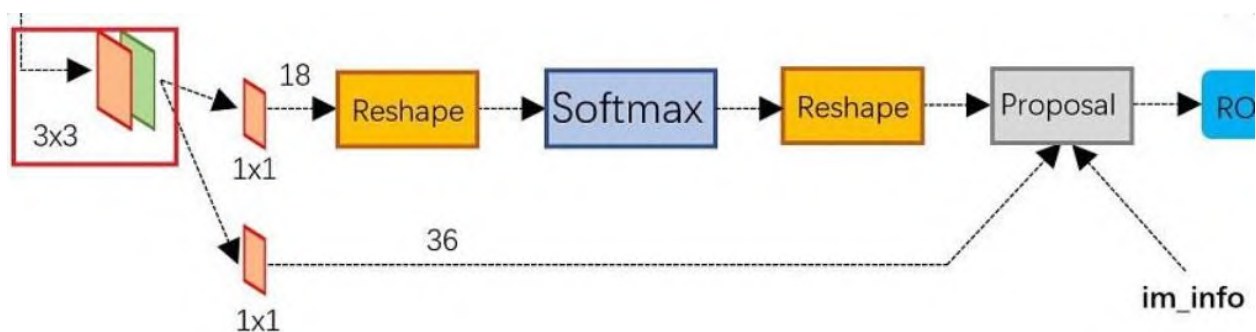


Рисунок 2.5 – Схематичне зображення мережі RPN [34]

Мережа RPN розділена на два шари. Вищий шар (рис. 2.5) отримує передній та задній план за допомогою якорів класифікації softmax (ціль виявлення – передній план). Нижній шар використовується для обчислення межі закріплення якорів для отримання точної пропозиції. Останній рівень пропозиції відповідає за синтез якорів переднього плану та придбання зсуву регресії обмежувальної рамки. У даній роботі вся мережа до рівня пропозиції є еквівалентною цільовій функції в роботі [27].

Для реалізації мережі RPN, у роботі також використано якорі, що є набором прямокутників, створених модулем `generate-anchors.py`. Масштабований метод RPN, який зазвичай використовується в процесі виявлення (детектування), введено з допомогою якорів [34]. Графічне зображення якорів представлено на рис. 2.6.

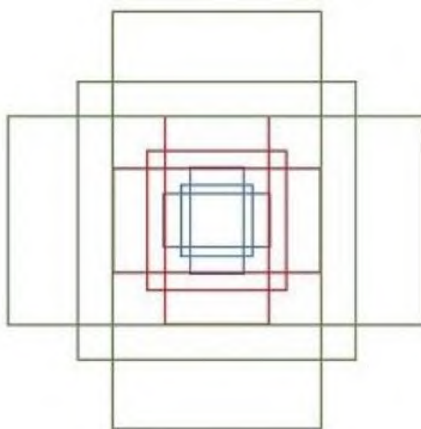


Рисунок 2.6 – Схематичне зображення якоря, що призначений для виявлення об'єктів на зображенні [28]

Розмір вищезазначених якорів встановлений, як розмір виявленого зображення.

Як показано на рис. 2.7, мапи об'єктів, що отримані під час обчислення згорткових шарів, обходять все зображення, і дев'ять якорів обладнані для кожної точки, як початковий кадр виявлення.

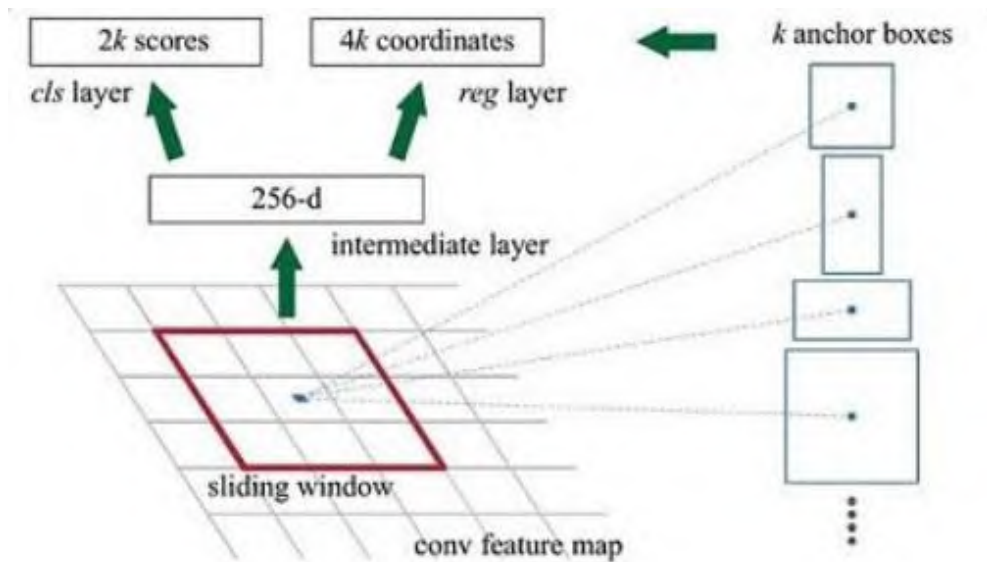


Рисунок 2.7 – Структура роботи якорів мережі RPN [34]

Крім того, у роботі використано дві регресійних обмежувальних рамки для виправлення положення зображення.

Для виконання поставленого завдання у роботі використовуватиметься невеликий набір даних. Невеликі зразки легко замінити, тому для візуалізації та розуміння згорткових мереж використано модель ZF-net. У такому разі, за останній згортковий шар взято conv5 з num-output рівним 256, що відповідає формуванню 256 мап ознак. Тому еквівалентом мапи ознак є 256-d.

Після останнього згорткового шару conv5 виконано згортку grp-conv  $3 \times 3$  і num-output = 256, що еквівалентно кожній точці, що об'єднує навколишню  $3 \times 3$  просторову інформацію, тоді як 256-d залишається незмінним (як показано в червоному полі на рис. 2.7 та 2.8). В кожній точці на карті знак conv5 є  $k$  якорів (за замовчуванням  $k = 9$ ), і кожен якорі поділяється на передній план та фон. Тому кожна точка перетворюється з 256-d ознаки на

оцінки  $cls = 2k$ . Якір має  $(x, y, w, h)$ , що відповідає 4 зміщенням, тому координати  $reg = 4k$ . Загалом під час навчання отримуємо 256 підходящих якорів.

Схема регресійної обмежувальної рамки наведена на рис. 2.8.



Рисунок 2.8 – Приклад калібрування розташування обмежувальної рамки об'єкта [28]

З рис. 2.9, зелена рамка – це Ground truth (GT) виявленого об'єкта, а червона – витягнуті якорі на передньому плані. Навіть якщо класифікатор розпізнає червону рамку, як об'єкт ТПВ, зображення є еквівалентним, оскільки червона рамка не калібрована. Відповідно, об'єкт ТПВ виявлено неправильно. Щоб уникнути подібних проблем, у роботі використано метод для точного налаштування червоної рамки, щоби якорі переднього плану та GT співпадали.

Для вікон чотиривимірний вектор  $(x, y, w, h)$  зазвичай використовується для представлення координат центральної точки та ширини й висоти вікна, як показано на рис. 2.9.

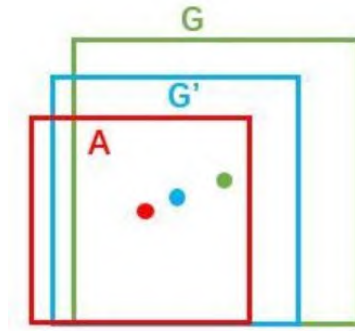


Рисунок 2.9 – Порівняння обмежувальної рамки з GT [28]

На рис. 2.10 червона рамка  $A$  є початковим якором переднього плану, зелене поле  $G$  представляє цільовий GT. Наша ціль – знайти взаємозв'язок, через який вхідний початковий якор  $A$  буде відображений для наближення регресії до реального вікна  $G$ .

Отже, для вектора  $A = (A_x, A_y, A_w, A_h)$  шукаємо таку мапу ознак  $f$ , що

$$f(A_x, A_y, A_w, A_h) = (G'_x, G'_y, G'_w, G'_h), \quad (2.5)$$

$$\text{де } (G'_x, G'_y, G'_w, G'_h) \approx (G_x, G_y, G_w, G_h).$$

Класифікаційна частина використовує набуті мапи ознак для обчислення кожної пропозиції, що стосується даної категорії. Схеми частини класифікації зображено на рис. 2.10.

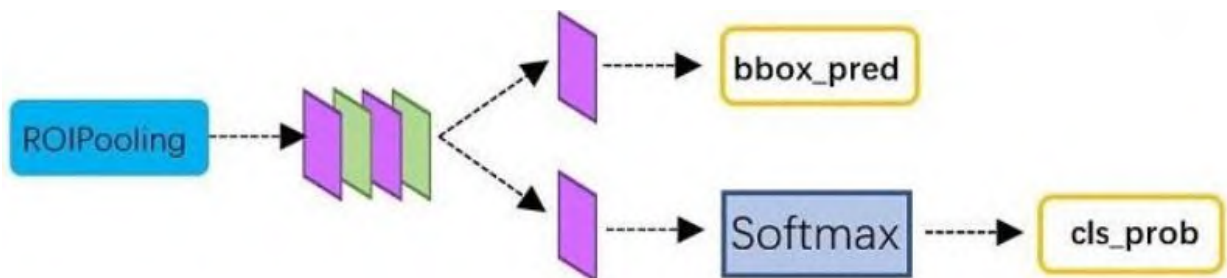


Рисунок 2.10 – Схематичне зображення процесу класифікації мережею Faster R-CNN [31]

Водночас регресійна обмежувальна рамка використовується знову для отримання зміщення положення кожної пропозиції. Vbox-pred використовується для повернення більш точного кадру виявлення об'єкта.

Загалом класифікатор реалізовано під платформою Windows 10 з використанням середовища розробки Anaconda, Tensorflow та мовами програмування Python і CUDA.

### 2.3 Сценарії використання користувацького інтерфейсу

Для візуального представлення вимог до інформаційної системи використано діаграму UML. На рис. 2.11 зображено дві частини програми: клієнтська та серверна.

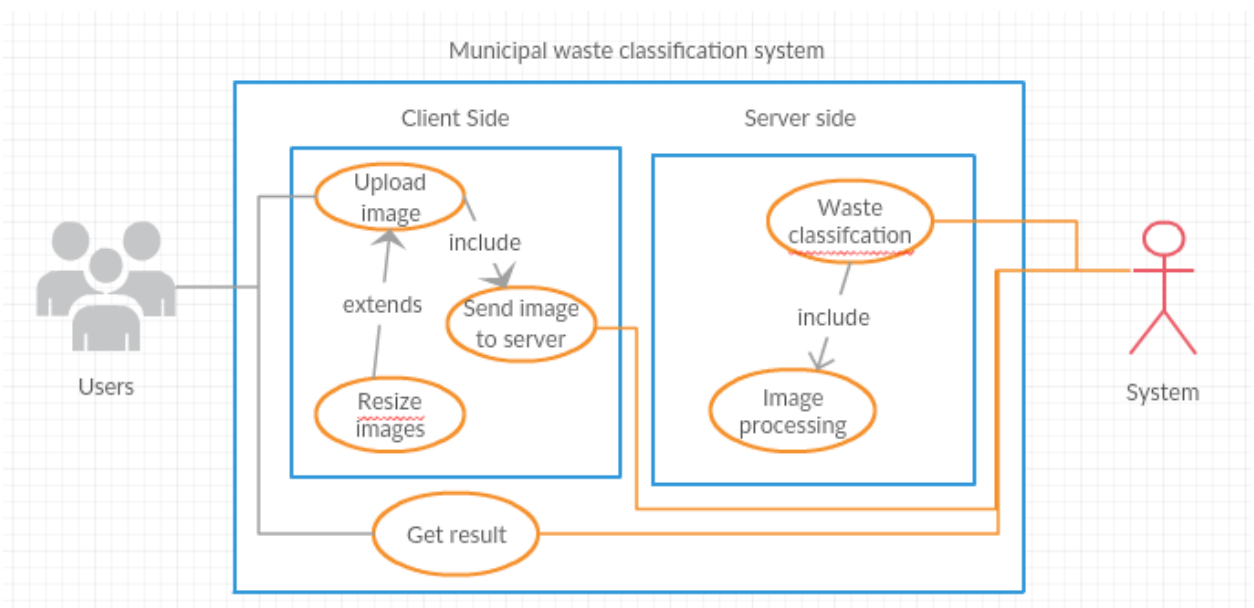


Рисунок 2.11 – UML-діаграма для системи класифікації зображень ТПВ

Детальний аналіз необхідного випадку використання визначений та представлений у табл. 2.1.

Таблиця 2.1 – Сценарії використання інформаційної системи класифікації зображень ТПВ у вигляді Телеграм-боту

Ідентифікація та назва	Одержати результат
Персона	Користувач
Опис	Після завантаження фотографії та обчислення користувач отримує результат про клас відходів на зображенні
Передумови	Користувач повинен завантажити зображення
Вихідні дані	Система виводить дані на інтерфейс користувача
Нормальна послідовність дій	Користувач завантажує зображення. Система отримує зображення та обрізає його. Система передає зображення модулю класифікації. Модуль класифікації досліджує ознаки із вхідного зображення та виконує внутрішні дії, пов'язані з діями згортки та субдискретизації. Модуль надсилає дані класифікації на вихід
Альтернативна послідовність дії	Не визначений
Виняток	Не завантажено дійсне зображення чи файл. Система повертає повідомлення про помилку

У табл. 2.2 наведено завдання, які виконуватиме інформаційна система для класифікації зображень у вигляді Телеграм-боту.

Таблиця 2.2 – Завдання користувачів Телеграм-боту

Завдання	Вхідні дані	Вихідні дані
Авторизація користувача. Додавання або зміна даних авторизації до бази даних боту.	Логін та пароль	Відповідь про успішний або неуспішний процес авторизації
Додавання або зміна мінімального та максимального інтервалу очікування між виконанням дій бота до його бази даних.	Діапазон часу	Відповідь про успішний або неуспішний процес виконання запиту

Продовження таблиці 2.2

Старт роботи боту	–	Відповідь про успішний або неуспішний процес виконання запиту або повідомлення про завершення роботи боту (при будь-яких помилках серверу або бази даних).
Завершення роботи боту	–	Відповідь про успішний або неуспішний процес виконання запиту
Перевірка статусу роботи боту	–	Відповідь про стан роботи боту у даний момент

#### 2.4 Архітектура клієнт-серверної системи

Для реалізаційної інформаційної системи класифікації ТПВ використано клієнт-серверну архітектуру.

Архітектура клієнт-серверної системи – це обчислювальна модель, у якій сервер розміщує, доставляє та керує ресурсами й послугами, які споживає користувач. Цей тип архітектури має один або кілька клієнтських комп'ютерів, підключених до центрального сервера через мережеве або інтернет-з'єднання. Архітектура клієнт-сервер також відома, як мережева обчислювальна модель або мережа клієнт-сервер, оскільки всі запити та послуги постачаються через мережу. Архітектуру програмного забезпечення поділено на дві частини:

- серверна частина, яка буде містити основну функціональність;
- клієнтська частина, яка дасть змогу користувачам взаємодіяти із сервером для проведення класифікації.

Клієнтська частина мусить бути реалізована в такий спосіб, щоби задовольняти таким вимогам:

- можливість завантажувати зображення для класифікації;

- відображення завантаженого зображення;
- можливість попереднього оброблення зображення, щоби мати можливість відправити його на сервер;
- можливість отримувати відповідь сервера з допомогою класифікації та показувати цю інформацію користувачеві.

Серверна частина має задовольняти таким потребам:

- отримувати HTTP-запити із зображенням, як вхідними даними;
- розміщення попередньо підготовленої згорткової нейронної мережі для класифікації зображень;
- перетворення отриманого зображення у форму, яке може бути завантажено до ЗНМ для проведення класифікації;
- підготувати відповідь, заповнену даними, спираючись на результати класифікації;
- відправити відповідь назад клієнту.

На рис. 2.12 відображено схема архітектури клієнт-сервер.

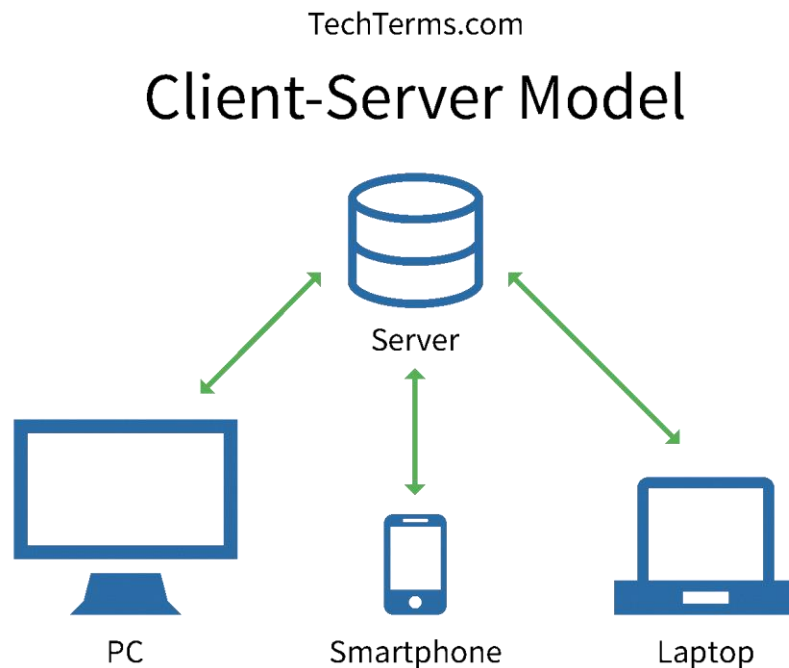


Рисунок 2.12 – Типове представлення клієнт-серверної архітектури

Однією з основних переваг архітектури клієнт-сервер є віддалені обчислення. Це означає, що велика кількість обчислень із великим навантаженням та даними буде проводитися на стороні сервера, тоді як із боку клієнта головною вимогою буде можливість завантажувати зображення та отримувати відповідь від сервера. Подібний підхід дає змогу використовувати різноманітні обчислювальні пристрої, наприклад, мобільні телефони чи розумні пристрої для будинків.

Як приклад можуть бути використані так звані «розумні будинки» – комплекси з вдосконаленими меблями, які зазвичай мають деяку кількість електроніки, що дає нагоду виконувати деякі заздалегідь задані функції та спілкуватися з іншими пристроями та власником будинку через конкретні програми. Крім того, застосування змін до оброблення зображень класифікації та загального функціонування вимагатиме лише змін на сервері, які можна виконати в будь-який момент, не вимагаючи очікування, поки кожен клієнт зможе оновити своє програмне забезпечення.

Для певних застосунків на стороні клієнта будуть створені два програмні агенти, і кожен із них продемонструє власний спосіб взаємодії із сервером. Перша буде зосереджена на обробленні повного циклу вимог, починаючи з завантаження зображень та фінішування, показуючи результати класифікації, а інша частина буде брати участь як «міст» між сервером та іншими системами, демонструючи інтегровану здатність рішення.

## 2.5 Опис програмних інструментів

За основу серверної частини та реалізації класифікатора обрано мову програмування з відкритим кодом Python версії 3.

Python – інтерпретована мова програмування високого рівня, загального призначення. Python підкреслює читабельність коду завдяки помітному

використанню значного пробілу. Її мовні конструкції та об'єктно-орієнтований підхід спрямовані на те, щоби допомогти програмістам написати чіткий логічний код для малих та легко масштабованих проектів.

Для навчання ЗНМ у даній роботі використано бібліотеку Keras, що є частиною фреймворку машинного навчання TensorFlow. Усе середовище TensorFlow програмним забезпеченням з відкритим кодом. На рис. 2.5 зображено лого бібліотеки Keras та фреймворку TensorFlow.

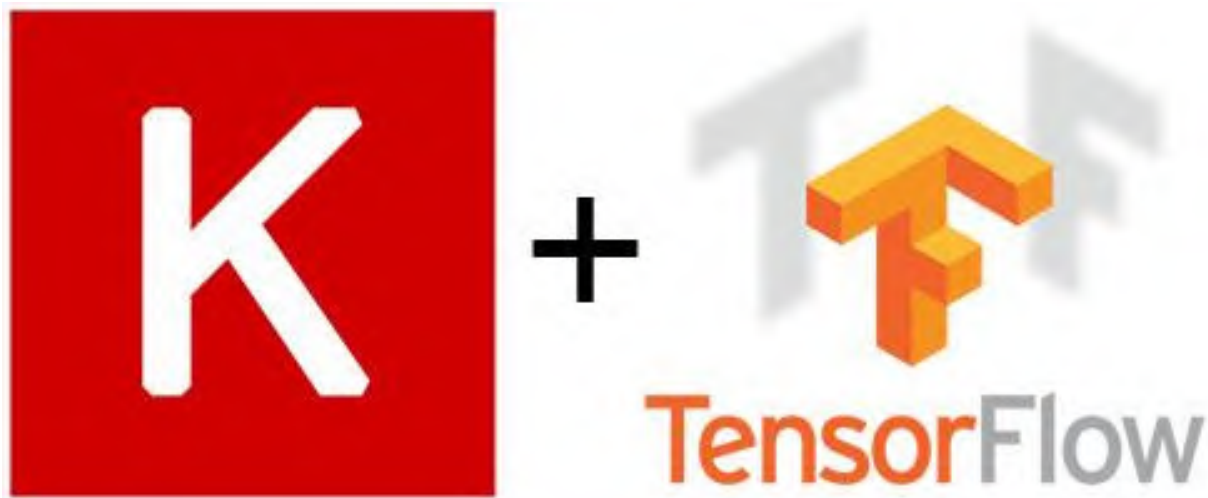


Рисунок 2.13 – Лого стеку програмування TensorFlow та Keras [35]

Середовище TensorFlow розробляється компанією Google і на сьогодні вважається флагманське рішенням для завдань, що пов'язані з навчанням нейронних мереж у будь-яких галузях людської діяльності, зокрема, детектування та класифікації зображень.

Для реалізації оброблення запитів на стороні сервера використано фремворк Flask. Цей інструментарій є частиною середовища Python і надає необхідну функціональність для оброблення HTTP-запитів. На рис. 2.6 представлено лого фреймворку Flask.



Рисунок 2.14 – Лого фреймворку Flask [36]

Flask пов'язаний з так званою категорією «мікрокадрів», та дає нагоду працювати без використання додаткових бібліотек. Цей інструмент є основним елементом серверної частини, що дає змогу обробляти запити клієнтів, передаючи дані зображення та надсилаючи відповіді даними класифікації.

Для обох клієнтської частини інформаційної системи використано поєднання мови програмування JavaScript та платформи NodeJS.

JavaScript – це мова програмування, що використовується, як вбудована мова в браузерах та основний елемент інтерактивних веб-сторінок. Node.js – це середовище запуску JavaScript з відкритим кодом, що дає нагоду виконувати код за межами браузера. На рис. 2.7 представлено лого середовища Node.js.



Рисунок 2.15 – Лого фреймворку Flask [37]

Node.js базується на парадигмі «JavaScript скрізь», об'єднуючи веб-застосунок з єдиною мовою програмування, а не поєднуючи кілька мов для скриптів на сервері та клієнтах.

Залежно від типу клієнтської програми було створено дві реалізації з різними бібліотеками: Electron.js та telegram-bot-api-вузлом.

ElectronJS – це набір інструментів з відкритим кодом, розроблена та підтримувана командою GitHub. ElectronJS дозволяє створювати графічні інтерфейси настільних комп'ютерів за допомогою веб-технологій. На рис. 2.8 зображено лого бібліотеки ElectronJS.



Рисунок 2.16 – Лого бібліотеки ElectronJS [38]

ElectronJS поєднує механізм візуалізації в браузері та середовище виконання Node.js. Ця бібліотека є основним графічним інтерфейсом для багатьох проектів з відкритим кодом.

Telegram-bot-api-вузол використовує API програмного забезпечення Телеграм. Цей пакет інструментів дає нагоду налаштувати сторонні програми, які функціонуватимуть у застосунку Телеграм. Контроль ботів здійснюється за допомогою HTTPS-запитів до відкритого API. Telegram-bot-api-вузол – це різновид програмної обгортки, з допомогою якого розробники використовують Telegram API в середовищі Node.js.

## Висновки до розділу 2

Отже, у другому розділі проведено проектування інформаційної системи для класифікації твердих побутових відходів. Побудовано класифікатор зображень, як згорткову нейронну мережу типу Faster R-CNN.

Впроваджено сценарії використання користувацького інтерфейсу для клієнт-серверної архітектури. Розроблено основні модулі клієнт-серверної інформаційної системи. Крім того, проведено огляд сучасних програмних інструментів та обрано середовище мови програмування Python для реалізації системи з користувацьким інтерфейсом Телеграм-боту.

### **3 РОЗРОБЛЕННЯ ТЕЛЕГРАМ-БОТУ ДЛЯ КЛАСИФІКАЦІЇ ТВЕРДИХ ПОБУТОВИХ ВІДХОДІВ**

#### **3.1 Реалізація класифікатора**

Як основний елемент всієї системи, а саме налаштування ЗНМ, виконано у першій частині реалізації. Ця частина була розділена на такі етапи:

- підготовлення та оброблення зображень;
- завантаження необхідних програмних інструментів;
- збільшення даних;
- завантаження та модифікація моделі ЗНМ;
- навчання класифікатора;
- налаштування гіперпараметрів навченої ЗНМ;
- збереження моделі на диску.

##### **3.1.1 Підготовка даних для класифікатора**

На першому кроці виконано підготовлення та оброблення візуальних даних. Знайдено та збережено необхідні дані для створення ЗНМ. Набір даних розділено на категорії та цілі (навчальні дані, дані валідації, дані тестування). У рамках даного дослідження система повинна класифікувати зображення, і ці зображення повинні бути розділені на такі категорії:

- картон;
- їжа;
- скло;
- метал;
- папір;

– пластик.

Навчання ЗНМ проводилось на поточному наборі даних TrashNet [39]. Набір даних містить 2527 зображень і розділений на 501 зображення зі скла, 594 паперу, 403 картону, 482 пластику, 410 металу та 137 загального сміття.

Оригінальна колекція зображень була випадковим чином розділена на три підмножини: навчання, перевірка та тест. Усі підмножини мають однакову швидкість класів. Через малу кількість зображень у навчальному наборі даних, 80% з них використано для навчання, 10% – для тестування, 10% – для валідації. Крім того, для надійності результатів прийнято 5-кратну стратегію перехресної перевірки, створивши довільно п'ять наборів даних про навчання, тестування та перевірку. Більше того, навчальна вибірка збільшується завдяки техніці збільшення даних. Результати, наведені в цьому підрозділі, відповідають середньому показнику п'яти запусків тестових наборів даних

Підготовлений набір даних розділено на три групи з шістьма підгрупами. Кожна група і підгрупа зображень були розміщені в окремі папки. На рис. 3.1 показано розділення зображень всередині тестової папки.

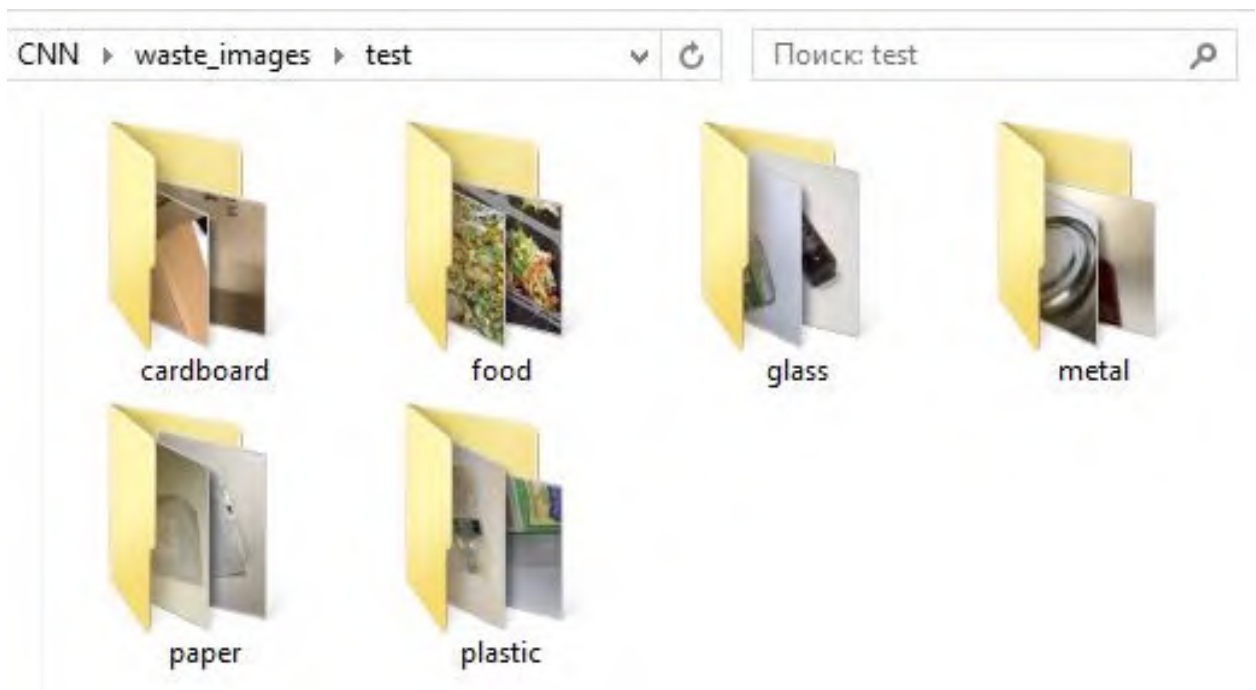


Рисунок 3.1 – Внутрішня організація зображень у папках

Перший рівень папок містив папки для розділення даних: одна папка під назвою «поїзд» містила зображення, які використовувались для тренувань моделі, друга папка під назвою «валідація» містила зображення, які використовувались для перевірки та виправлення моделі, а третя папка під назвою «тест» містили зображення, які використовувались для перевірки того, як модель реагуватиме на дані, яких вона раніше не бачила, тому результати класифікації будуть близькими до середнього рівня точності класифікації. Кожна папка містила шість підпапок, названих класами, які вона представляє, що дозволяє програмному забезпеченню автоматично приєднувати мітки до зображень під час завантаження зображення.

Для завантаження цих зображень для подальшого їх використання використовували визначення змінних з навчанням, валідацією та тестуванням імен папок. Після цього було створено ImageDataGenerator елементів, відображених у наступні папки. На рис. 3.2 показаний вихід.

```
In [2]: train_path = 'waste_images/train'
        validate_path = 'waste_images/validate'
        test_path = 'waste_images/test'

In [3]: train_batches = ImageDataGenerator(preprocessing_function=tensorflow.keras.applications.mobilenet.preprocess_input)
        .flow_from_directory(train_path, target_size=(224, 224), batch_size=12)
        validate_batches = ImageDataGenerator(preprocessing_function=tensorflow.keras.applications.mobilenet.preprocess_input)
        .flow_from_directory(validate_path, target_size=(224, 224), batch_size=12)
        test_batches = ImageDataGenerator(preprocessing_function=tensorflow.keras.applications.mobilenet.preprocess_input)
        .flow_from_directory(test_path, target_size=(224, 224), batch_size=12)

Found 2234 images belonging to 6 classes.
Found 557 images belonging to 6 classes.
Found 232 images belonging to 6 classes.
```

Рисунок 3.2 – Підготовлені партії зображень

Деталі відповідних класів можна побачити на рис. 3.3.

```
In [8]: train_batches.class_indices

Out[8]: {'cardboard': 0, 'food': 1, 'glass': 2, 'metal': 3, 'paper': 4, 'plastic': 5}
```

Рисунок 3.3 – Впорядкування класів

### 3.1.2 Попереднє оброблення даних

Для роботи з НМ в середовищі Jupyter Notebook виконано необхідний завантаження необхідних бібліотек. Під час цього кроку основні бібліотеки та їх елементи завантажуються в навколишнє середовище, щоб з них можна було отримати доступ і використовувати для цілей розвитку.

На рис. 3.4 відображено основний блок із імпортом бібліотеки, тобто ті бібліотеки та елементи, що завантажувались для оброблення даних, конфігурації нейронної мережі, навчання та тестування.

```
In [1]: import numpy as np
import tensorflow.keras
from tensorflow.keras import models
from tensorflow.keras import backend as K
from tensorflow.keras.layers import Dense
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.metrics import categorical_crossentropy
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import Model
from keras.applications import imagenet_utils
from sklearn.metrics import confusion_matrix
import itertools
import matplotlib.pyplot as plt
%matplotlib inline

Using TensorFlow backend.
```

Рисунок 3.4 – Виклик бібліотек

Після завершення імпорту бібліотеки було виконано етап збільшення даних. Збільшення даних – це процес, пов’язаний з модифікаціями та перетвореннями даних, що дозволяє розробникам збільшувати кількість навчальних даних шляхом незначної модифікації існуючих. Так, процес збільшення даних буде пов’язаний із виконанням перетворень зображень, які включатимуть такі операції:

- зміни орієнтації зображення (обертання та дзеркальне відображення);

- зміни масштабу зображення (масштабування та зсув);
- змінення положення зображення (зміщення).

Як результат, процедура дасть багато нових зображень на основі наявних даних, але які матимуть певні перетворення, які дозволять мережам субстрагувати необхідні візерунки та форми зображень із зменшенням ризику надмірного розміщення моделі.

Для виконання цієї операції був написаний та використаний фрагмент коду з рис. 3.5.

```
In [3]: from keras.preprocessing.image import ImageDataGenerator, array_to_img, img_to_array, load_img
        from PIL import Image
        import glob

        datagen = ImageDataGenerator(
            rotation_range=45,
            width_shift_range=0.15,
            height_shift_range=0.15,
            shear_range=0.25,
            zoom_range=0.25,
            horizontal_flip=True,
            fill_mode='reflect')

        image_list = []
        for filename in glob.glob('augment/*.jpg'):
            im=Image.open(filename)
            image_list.append(im)

            img = load_img(filename)
            x = img_to_array(img)
            x = x.reshape((1,) + x.shape)

        # the .flow() command below generates batches of randomly transformed images
        # and saves the results to the `preview/` directory
        i = 0
        for batch in datagen.flow(x, batch_size=1,
                                  save_to_dir='preview', save_prefix='food', save_format='jpeg'):
            i += 1
            if i > 7: # count of iterations before data augmentation stops
                break
```

Рисунок 3.5 – Код оброблення зображень

Під час його виконання він створює елемент Keras ImageDataGenerator з параметрами, що визначають параметри перетворення та їх значення. Потім для кожного файлу він виконує завантаження, перетворення формату та викликає метод змінного потоку ImageDataGenerator, який застосовує згадані вище перетворення випадковим чином, обмеженими межами параметра. Після

перетворень модифікований маг зберігається у папці «попередній перегляд» із заданим префіксом та внутрішнім номером ітерації.

### 3.1.3 Завантаження та модифікація згорткової нейронної мережі

ЗНМ побудована у такий спосіб: перша частина буде відповідати за згортання та розпізнавання візерунків, друга частина буде виконувати класифікацію за даними першої частини.

Для першої частини обрано модель ЗНМ під назвою MobileNetV2. Ця мережа спочатку була розроблена для мобільних пристроїв малої потужності для виконання різних наборів операцій, що пов'язані з обробленням зображень. Ця модель попередньо навчена на наборі даних ImageNet і має добре навчену частину розпізнавання, в результаті чого формуються абстрактні уявлення реальних об'єктів.

На рис. 3.6 продемонстровано модифікацію мережі.

```
In [11]: mobileV2 = tensorflow.keras.applications.mobilenet_v2.MobileNetV2()
x = mobileV2.layers[-2].output

In [12]: Prediction = Dense(6, activation='softmax')(x)
mobV2_model = Model(inputs = mobileV2.input, outputs = Prediction)

In [13]: mobV2_model.summary()
```

block_16_project (Conv2D)	(None, 7, 7, 320)	307200	block_16_depthwise_relu[0][0]
block_16_project_BN (BatchNorma	(None, 7, 7, 320)	1280	block_16_project[0][0]
Conv_1 (Conv2D)	(None, 7, 7, 1280)	409600	block_16_project_BN[0][0]
Conv_1_bn (BatchNormalizationV1	(None, 7, 7, 1280)	5120	Conv_1[0][0]
out_relu (ReLU)	(None, 7, 7, 1280)	0	Conv_1_bn[0][0]
global_average_pooling2d (Globa	(None, 1280)	0	out_relu[0][0]
dense (Dense)	(None, 6)	7686	global_average_pooling2d[0][0]

```
-----
Total params: 2,265,670
Trainable params: 2,231,558
Non-trainable params: 34,112
```

Рисунок 3.6 – Результат зміни структури моделі

Для класифікації мережі на шість категорій виконали такі дії:

- завантажено попередньо підготовлену модель MobileNetV2;
- видалено вихідний класифікаційний шар;
- додано спеціальний вихідний шар із шістьма виходами.

Поки нові шари в бібліотеці Keras додаються, їх ваги заповнюються випадковим чином. Так, якщо спробувати вдосконалити модель, не розморожуючи всі шари, крім останнього, є шанс втратити виявлені візуальні ознаки згорткової частини.

Тому було проведено заморожування шарів, а на рис. 3.7 показано результат цієї операції.

```
In [14]: for layer in mobV2_model.layers[:-1]:
         layer.trainable = False

In [15]: mobV2_model.summary()
```

block_16_project (Conv2D)	(None, 7, 7, 320)	307200	block_16_depthwise_relu[0][0]
block_16_project_BN (BatchNorma	(None, 7, 7, 320)	1280	block_16_project[0][0]
Conv_1 (Conv2D)	(None, 7, 7, 1280)	409600	block_16_project_BN[0][0]
Conv_1_bn (BatchNormalizationV1	(None, 7, 7, 1280)	5120	Conv_1[0][0]
out_relu (ReLU)	(None, 7, 7, 1280)	0	Conv_1_bn[0][0]
global_average_pooling2d (Globa	(None, 1280)	0	out_relu[0][0]
dense (Dense)	(None, 6)	7686	global_average_pooling2d[0][0]

```

Total params: 2,265,670
Trainable params: 7,686
Non-trainable params: 2,257,984

```

Рисунок 3.7 – Параметри моделі ЗНМ після «заморожування»

Модель була налаштована так, що єдині доступні параметри тренінгу належать останньому щільному шарі.

На рис. 3.8 показані результати навчання точності та втрат для наборів даних поїздів та валідації.

```

In [13]: mobV2_model.compile(Adam(.00025), loss='categorical_crossentropy', metrics=['accuracy'])

In [14]: mobV2_model.fit_generator(train_batches, steps_per_epoch=27, validation_data=validate_batches, validation_steps=3, epochs=9) #
Epoch 1/9
27/27 [=====] - 195s 7s/step - loss: 0.9886 - accuracy: 0.6204 - val_loss: 0.9551 - val_accuracy: 0.7500
Epoch 2/9
27/27 [=====] - 148s 5s/step - loss: 0.6843 - accuracy: 0.7516 - val_loss: 1.0346 - val_accuracy: 0.7222
Epoch 3/9
27/27 [=====] - 155s 6s/step - loss: 0.5804 - accuracy: 0.8025 - val_loss: 1.1945 - val_accuracy: 0.7222
Epoch 4/9
27/27 [=====] - 149s 6s/step - loss: 0.5398 - accuracy: 0.8117 - val_loss: 0.5711 - val_accuracy: 0.8056
Epoch 5/9
27/27 [=====] - 149s 6s/step - loss: 0.5161 - accuracy: 0.8395 - val_loss: 1.5074 - val_accuracy: 0.6944
Epoch 6/9
27/27 [=====] - 149s 6s/step - loss: 0.4191 - accuracy: 0.8673 - val_loss: 0.7609 - val_accuracy: 0.8056
Epoch 7/9
27/27 [=====] - 147s 5s/step - loss: 0.4421 - accuracy: 0.8241 - val_loss: 1.0176 - val_accuracy: 0.8056
Epoch 8/9
27/27 [=====] - 147s 5s/step - loss: 0.3364 - accuracy: 0.9076 - val_loss: 0.6864 - val_accuracy: 0.7778
Epoch 9/9
27/27 [=====] - 149s 6s/step - loss: 0.2914 - accuracy: 0.9198 - val_loss: 0.9977 - val_accuracy: 0.7222

```

Рисунок 3.8 – Результат моделі та вихідні дані навчального процесу

Після цього модель була складена за допомогою функції оптимізації Адам зі коефієнтом навчання у діапазоні  $2 \cdot 10^{-6} - 2 \cdot 10^{-4}$ . Далі проведено навчання моделі; у результаті навчання модифіковану модель збережено на локальному диску.

### 3.1.3 Налаштування параметрів згорткової нейронної мережі

Після отримання моделі, яка має правильне відображення між моделями згортки та відповідними класами, було виконано операцію під назвою «тонка настройка». Тонке налаштування – це процес навчання моделей, під час якого разом із вагою шару класифікації перекаваліфікується верхня частина попередньо підготовленої моделі ЗНМ із низькою швидкістю навчання.

Під час цього процесу найбільш абстрактні ознаки щодо згорткових шарів повільно змінюються відповідно до успіхів класифікації, що призводить до кращого розпізнавання шаблонів для необхідної класифікації та одночасно налаштування результатів класифікації. Зазвичай низький рівень навчання вимагається через те, що згорткова частина мережі вже була навчена великій кількості об'єктів реального світу, тому її базовий результат уже досить точний.

Водночас високі показники навчання зумовлюють значні зміни згорткових фільтрів, унаслідок чого є велика ймовірність того, що чинні фільтри втратять важливі елементи та замість них отримають інші шаблони, які належатимуть лише навчальних даним.

На рис. 3.9 зображено блоки коду, що відповідають налаштуванню ЗНМ.

```
In [5]:
for layer in mobV2_model.layers:
    layer.trainable = True

for layer in mobV2_model.layers[36:]:
    layer.trainable = False

In [6]: mobV2_model.compile(Adam(.00002), loss='categorical_crossentropy', metrics=['accuracy'])

In [7]: mobV2_model.summary()
```

block_16_project (Conv2D)	(None, 7, 7, 320)	307200	block_16_depthwise_relu[0][0]
block_16_project_BN (BatchNormali	(None, 7, 7, 320)	1280	block_16_project[0][0]
Conv_1 (Conv2D)	(None, 7, 7, 1280)	409600	block_16_project_BN[0][0]
Conv_1_bn (BatchNormalizationV1	(None, 7, 7, 1280)	5120	Conv_1[0][0]
out_relu (ReLU)	(None, 7, 7, 1280)	0	Conv_1_bn[0][0]
global_average_pooling2d (Globa	(None, 1280)	0	out_relu[0][0]
dense (Dense)	(None, 6)	7686	global_average_pooling2d[0][0]

```

Total params: 2,265,670
Trainable params: 1,632,582
Non-trainable params: 633,088

In [8]: mobV2_model.fit_generator(train_batches, validation_data=validate_batches, epochs=5)

Epoch 1/5
187/187 [=====] - 537s 3s/step - loss: 0.4583 - accuracy: 0.8384 - val_loss: 0.4893 - val_accuracy: 0.8097
Epoch 2/5
187/187 [=====] - 507s 3s/step - loss: 0.3596 - accuracy: 0.8850 - val_loss: 0.4628 - val_accuracy: 0.8276
Epoch 3/5
187/187 [=====] - 505s 3s/step - loss: 0.3053 - accuracy: 0.9064 - val_loss: 0.4871 - val_accuracy: 0.8151
Epoch 4/5
187/187 [=====] - 504s 3s/step - loss: 0.2528 - accuracy: 0.9266 - val_loss: 0.4466 - val_accuracy: 0.8348
Epoch 5/5
187/187 [=====] - 504s 3s/step - loss: 0.2021 - accuracy: 0.9445 - val_loss: 0.4179 - val_accuracy: 0.8492

Out[8]: <tensorflow.python.keras.callbacks.History at 0x217b4090ba8>

In [11]: mobV2_model.save('CNN_model Sakach.h5')
```

Рисунок 3.9 – Часткове розморожування шарів моделі, налаштування моделі та збереження шарів моделі

Міцно, останні тридцять п'ять шарів були розмороженими (це було виконано механізмом нарізки Python, у якому передача від'ємного індексу дає змогу виконувати необхідні операції від кінця до початку деякого масиву). Архітектура MobileNetV2 містить 88 прихованих шарів, і завдяки зміні лише одного шару з класифікацією ImageNet на інший для класифікації ТПВ, кількість шарів залишилася однаковою. Тому останні тридцять п'ять шарів

можна розглядати як «верхні шари» мережі, тому процес їхнього налаштування без зміни попередніх шарів можна вважати хорошим рішенням. Потім було виконано компіляцію моделі з функцією оптимізації Адама та швидкістю навчання 0,00002. Після цього було зроблено новий етап підгонки. Динаміка втрат і точності моделі показує, що операція тонкої настройки призвела до значного підвищення точності прогнозування моделі.

Після завершення процесу тонкої настройки поточна модель зберігається на диску з назвою «CNN\_model\_Param.h5». Метод «зберегти» – це збирання конфігурації та ваги поточної системи, стискання та збереження у форматі файлу.h5. Після цього модель готова до інтеграції з іншими застосунками для передачі даних.

### 3.2 Навчання класифікатора та аналіз результатів

Насамперед опишемо метрики для оцінювання результатів виявлення та класифікації об'єктів на зображенні.

Середня точність (mAP) є важливою оцінкою для виявлення об'єкта на зображенні. Коефіцієнт mAP призначений для вимірювання точності роботи детекторів, як Faster R-CNN. Це середнє значення максимальних точних значень за різних значень відкликання. Для обрахування mAP, попередньо обраховано такі метрики: precision, recal та IoU.

Позначимо кількість реальних позитивних випадків у наборі даних, як P, та кількість реальних негативних випадків, як N. Результати класифікації розподіляються, як істинно позитивні (TP), істинно негативні (TN), хибнопозитивні (FP) та хибнонегативні випадки (FN). Тоді precision вимірює, наскільки точними є прогнози, тобто обраховує відсоток правильних позитивних прогнозів. Recall обраховує наскільки добре всі позитивні випадки були виявлені. Математичні формули вищезгаданих метрик наведено нижче:

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

$$Precision = \frac{TP}{TP + FN}. \quad (3.2)$$

Водночас перетин над об'єднанням (IoU) вимірює наскільки перекриваються між собою 2 регіони. Ця метрика показує наскільки хорошим є прогнозування в детекторі об'єкта з GT (реальною межею об'єкта):

$$IoU = \frac{\text{area of overlap}}{\text{area of union}}. \quad (3.3)$$

У використуваному нашому наборі даних маємо загальну кількість зображень ТПВ. Збираємо всі прогнози, що зроблені моделлю для усіх зображень класифікуємо їх відповідно до передбачуваного рівня впевненості (від найвищої до нижчої). AP (average precision) обчислюється, як середнє значення максимальної точності на всіх рівнях відкликання:

$$AP = \frac{1}{n} \sum_{r \in \{P_r(i)\}} AP_r = \frac{1}{n} \sum_{r \in \{P_r(i)\}} P_{interp}(r), \quad (3.4)$$

де

$$P_{interp}(r) = \max_{\tilde{r} \geq r} \{p(\tilde{r})\}.$$

У результаті навчання Faster R-CNN отримали значення параметрів навчання, що також є значеннями функції втрат. Результат навчання зображено на рис. 3.10.

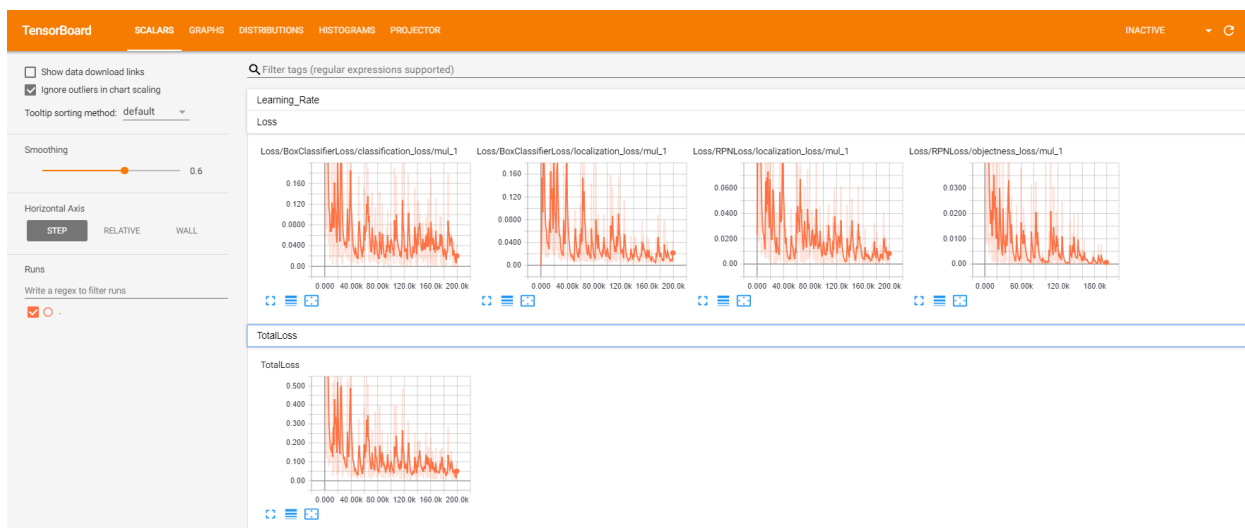


Рисунок 3.10 – Графіки оцінок ефективності функції втрат для найкращого результату

З рис. 3.10 випливає, упродовж усього процесу навчання, функція втрат має тенденцію до поступового зменшення. Величина втрат має великі коливання до 60 000 кроків. Після 120 000 кроків значення функції втрат поступово збігається, що свідчить про досягнення надійних результатів.

У табл. 3.1 наведено порівняння точності на основі різних параметрів.

Таблиця 3.1 – Порівняння результатів тренування мережі Faster R-CNN

Розмір зразків	К-сть ітерацій	Розмір партії зображень	Час тренування, год	Коефіцієнт навчання	Точність класифікації
10 000 (1:4)	10	10	2,15	$10^{-5} - 10^{-3}$	0,686
		50	1,37		0,701
		100	4,25		0,796
10 000 (1:4)	20	10	5,43	$10^{-6} - 10^{-4}$	0,736
		50	3,49		0,793
		100	7,04		0,741
10 000 (1:4)	30	10	8,14	$2 \cdot 10^{-6} - 2 \cdot 10^{-4}$	0,812
		50	9,67		0,823
		100	10,4		0,845

З наведеної вище табл. 3.1 дістали, що для час навчання є найбільшим для значення розміру партії рівним 100. Крім того, найвища точність класифікації

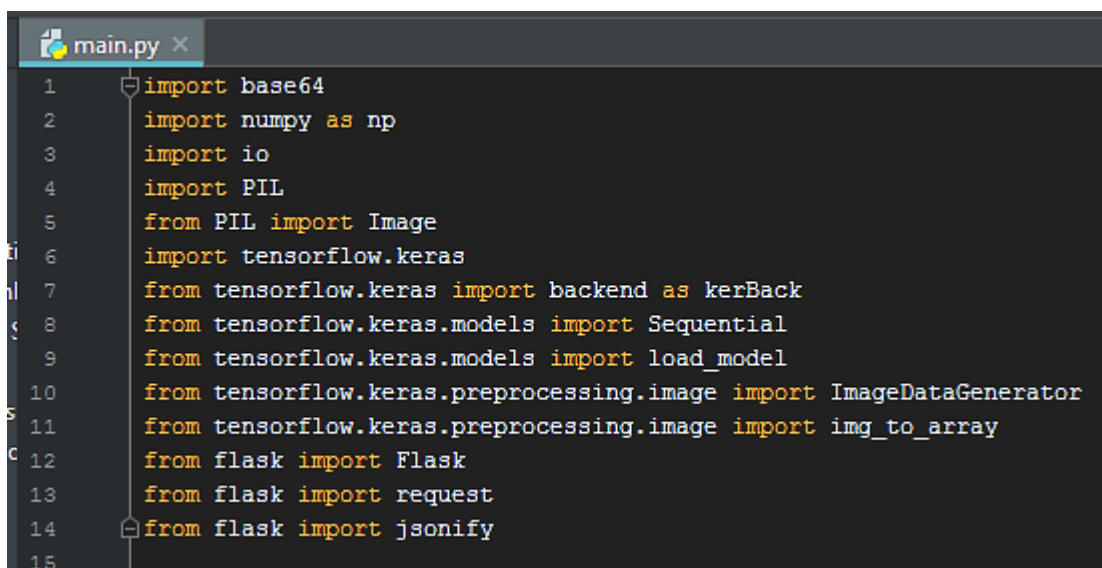
була отримана зі значенням коефіцієнта навчання в діапазоні  $2 \cdot 10^{-6} - 2 \cdot 10^{-4}$  та кількістю ітерацій, що дорівнює 30.

### 3.3 Реалізація серверної частини

Серверну частину інформаційної системи реалізовано з такою функціональністю:

- завантаження необхідних бібліотек (Flask, TensorFlow, PIL тощо);
- ініціалізація змінних та попереднє завантаження моделі;
- метод перетворення зображень Base64;
- функціональність класифікації об'єктів;

На рис. 3.11 показані необхідні залежності.



```
main.py x
1 import base64
2 import numpy as np
3 import io
4 import PIL
5 from PIL import Image
6 import tensorflow.keras
7 from tensorflow.keras import backend as kerBack
8 from tensorflow.keras.models import Sequential
9 from tensorflow.keras.models import load_model
10 from tensorflow.keras.preprocessing.image import ImageDataGenerator
11 from tensorflow.keras.preprocessing.image import img_to_array
12 from flask import Flask
13 from flask import request
14 from flask import jsonify
15
```

Рисунок 3.11 – Виклик бібліотек залежності сервера

Сервер постачається з двома внутрішніми методами та двома заздалегідь визначеними маршрутами. Перший маршрут обробляє чистий запит (наприклад, «localhost: 5000 /») і повертає тег <h1> як знак того, що сервер працює зараз. Другий маршрут викликається, коли запит POST надсилається до

маршруту «/ класифікувати». Цей метод приймає дані «зображення» із запити, декодує їх у форматі Base64, відкриває їх та викликає «`load_image`» метод, який перетворює його у формат RGB, змінює розмір зображення та завантажує його у масив NumPy. Метод повертає тензор із методу «`keras.applications.mobilenet.preprocess_input`». Повернений тензор передається моделі з допомогою методу «прогнозування», а результати прогнозування передаються у форматі відповіді після повернення результатів клієнту.

### 3.4 Реалізація клієнтської частини

Для обох клієнтів були створені програми, що ґрунтуються на NodeJS. Перша програма використовує ElectronJS, як основу й розроблена для роботи на пристроях кінцевих користувачів. У цій програмі є головне вікно, яке містить логіку вибору зображення, перетворення його у формат Base64 та відправлення його через HTTP-запит на сервер. Після отримання відповіді відображаються результати класифікації. По-друге, додаткове вікно, містить текстове поле, у якому посилання на сервер можна вказати.

Друга програма використовує згадану перед бібліотекою telegram-bot-арі-вузол. З допомогою цієї бібліотеки створюється внутрішній примірник бота та додається кілька обробників подій. Основний називається під час надсилання фото боту. Цей обробник приймає ID чату, ідентифікатор фотографії з найвищою роздільною здатністю та завантажує цю фотографію із серверів Телеграм. Пізніше фото завантажується в застосунок, перетворюється у формат Base64 і після сервера відправляється HTTP-запит із зображенням. Коли відповідь приходить, бот заповнює текст повідомлення відповіді за отриманим результатом класифікації та надсилає його в чат Телеграм. Програмне забезпечення мусить бути розгорнута на пристрої з доступом до

застосунку Телеграм і Flask-сервера. Найкращим вибором може вважатися інсталювання на тому ж сервері, що й застосунок класифікації на базі Flask.

Щоби скористатися послугами Телеграм-боту, необхідно попередньо зареєструватися. Для реєстрації бота буде використовуватися BotFather. На рис. 3.12 продемонстровано результат реєстрації Телеграм-боту.

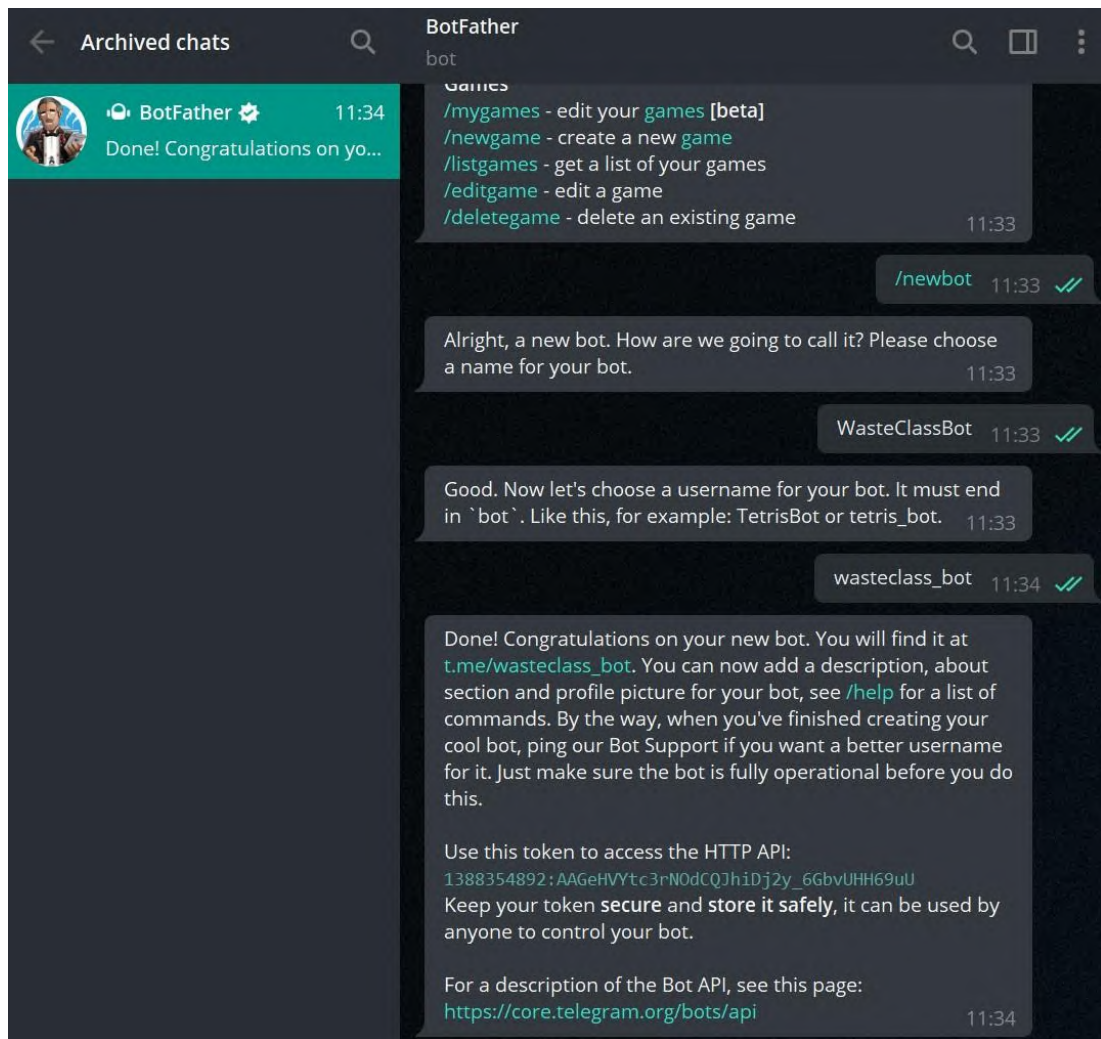


Рисунок 3.12 – Зображення процесу реєстрації Телеграм-бота

Загалом для реєстрації користувацького бота необхідно виконати наступні кроки:

- розпочати текстовий діалог з контактом BotFather через Телеграм;
- ввести команду /newbot для створення нового бота;
- вказати ім'я користувача бот-програми;
- дістати маркер авторизації.

### 3.5 Верифікація та валідація інформаційної системи

#### 3.5.1 Тестування класифікатора

Після запуску всіх описаних раніше програмних компонентів вимоги до програмного забезпечення задовольняються, і система виконує необхідні операції, відповідно до шаблону клієнт-серверної архітектури.

Для тестування обом клієнтам було надіслано зображення металевої банки. На рис. 3.13 наведено результати класифікації, здобуті класифікатором у настільній версії програмного забезпечення Телеграм.

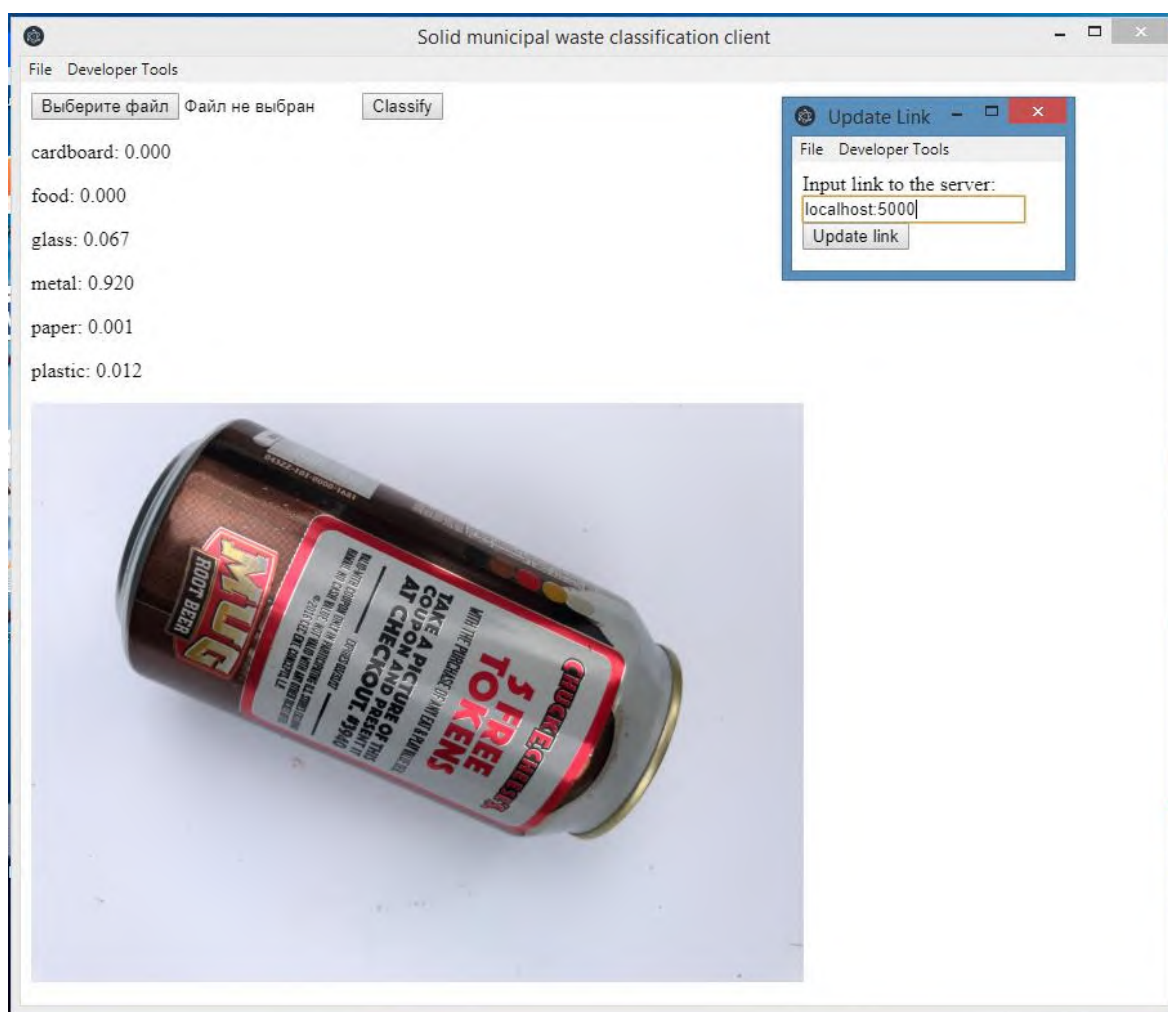


Рисунок 3.13 – Результат класифікації зображення ТПВ у клієнтській програмі для настільних ПК

Клієнтські програми дістають зображення у вигляді даних від користувачів, обробляють їх і відправляють на сервер. Сервер виконує класифікацію та повертає відповіді клієнтам, які форматують їх заздалегідь визначеним чином і показує цю інформацію користувачеві (настільний застосунок) або відправляє повідомлення на Телеграм (бот-сервер).

Те ж саме зображення надіслано Телеграм-боту. На рис. 3.14 наведено відповідь боту з деталями класифікації.

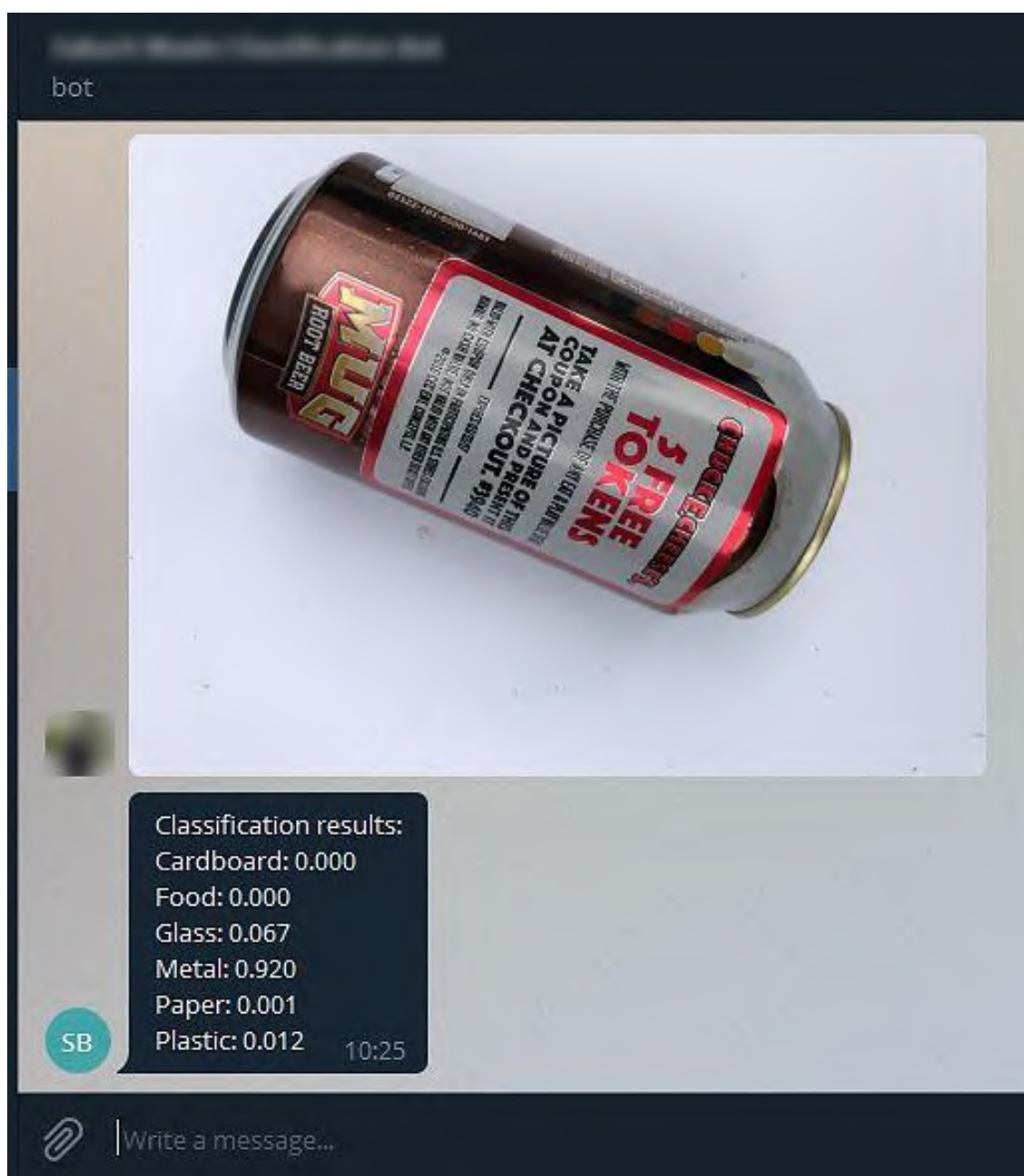


Рисунок 3.14 – Результати класифікації ТПВ в інтерфейсі настільної версії застосунку Телеграм

### 3.5.2 Інструкція користувача для впровадження системи класифікації

Щоби використовувати клієнтський застосунок в інтерфейсі Телеграм, користувачу інформаційної системи потрібно виконати такі кроки:

- увійти в Телеграм, викликати бота класифікації та надіслати боту зображення ТПВ;
- встановити застосунок на основі фреймворку Electron, вказати шлях до сервера та завантажити зображення.

Для розгортання на стороні сервера необхідно виконувати такі дії:

- налаштувати середовище Anaconda;
- інсталювати останні версії бібліотек Keras і TensorFlow v1.15;
- інсталювати бібліотеку Flask та налаштувати її для використання файлу main.js;
- налаштувати середовище Node.JS та інсталювати залежності NPM.

Усі вище вказані кроки можна виконати через інтерфейс командного рядка після встановлення необхідних середовищ. Далі потрібно запустити main.py для сервера і main.js для бота Телеграм.

### Висновки до розділу 3

На основі нейромережевого моделювання реалізовано модель згорткової нейронної мережі. Розроблену нейронну мережу навчено та протестовано на готовому наборі даних TrashNet, що містить 2527 зображень і поділений на шість класів. У результаті проведених експериментів визначено, що найвищу точність класифікації зображень досягнуто зі значенням коефіцієнта навчання в діапазоні  $2 \cdot 10^{-6} - 2 \cdot 10^{-4}$  та кількістю ітерацій, що дорівнює 30. Максимальна середня точність класифікації, яку досягла нейронна мережа Faster R-CNN, склала 84,5 %.

Впроваджено клієнт-серверну архітектуру до програмного забезпечення з внутрішнім класифікатором нейронної мережі. Реалізовано користувацький інтерфейс у вигляді Телеграм-боту для розв'язання поставленої задачі. Проведено валідування програмного забезпечення на реальних прикладах та наведено інструкцію користувача для подальшого розгортання.

Отже, створена інформаційна система здатна класифікувати зображення твердих побутових відходів як із мобільних застосунків, так і настільної версії Телеграму. Розроблений Телеграм-бот може бути використаний у побуті для особистісного сортування, як часткова заміна спеціалізованим системам збору та сортування сміття.

## ВИСНОВКИ

Кваліфікаційна робота магістра присвячена розробленню інформаційної системи класифікації твердих побутових відходів із використанням інтерфейсу програмного забезпечення Телеграм.

У роботі проведено аналіз сучасного стану розвитку методів та систем управління відходами. У результаті аналізу виявлено, що стан поводження з побутовими відходами в Україні є незадовільним. На території нашої держави все ще немає єдиної інтегрованої системи управління відходами, що дала б змогу ефективно утилізувати як промислові, так і тверді побутові відходи.

Здійснено огляд сучасних інформаційних технологій та підходів до збору та оброблення твердих побутових відходів. Висунуто припущення, що інформаційна технологія на основі методів комп'ютерного зору може стати ефективним рішенням до розв'язання проблеми поводження з відходами. Водночас під час проходження науково-дослідної практики на підприємстві ПАТ «Укрзалізниця» Філія «Старокостянтинівський завод залізобетонних шпал» виявлено низку проблем поводження з відходами, розв'язати які може інформаційна система зі зрозумілим користувацьким інтерфейсом.

Проведено порівняльний аналіз комерційних сервісів, що є лідерами в задачах детектування та класифікації зображень. Визначено переваги (підтримання багато класовості та просте впровадження) та недоліки (необхідність постійного підключення до інтернету та висока ціна) кожного сервісу. На основі аналізу сучасних методів та підходів до класифікації зображень обрано згорткову нейронну мережу, як класифікатор інформаційної системи.

Здійснено проектування інформаційної системи для класифікації твердих побутових відходів, а саме: побудовано класифікатор зображень на основі згорткової нейронної мережі типу Faster R-CNN, розглянуто та описано сценарії використання користувацького інтерфейсу, розроблено основні модулі клієнт-серверної системи. Проведено огляд сучасних програмних інструментів

та обрано середовище мови програмування Python для реалізації системи з користувацьким інтерфейсом Телеграм-боту.

На основі використання найкращих практик нейромережевого моделювання реалізовано модель згорткової нейронної мережі. Нейронну мережу навчено та протестовано на готовому наборі даних TrashNet, що містить 2527 зображень і поділених на шість класів. У результаті проведених експериментів визначено, що найвищу точність класифікації зображень досягнуто зі значенням коефіцієнта навчання в діапазоні  $2 \cdot 10^{-6} - 2 \cdot 10^{-4}$  та кількістю ітерацій, що дорівнює 2000. Навчена нейронна мережа Faster R-CNN досягла максимальної середньої точності класифікації в 84,5 %.

Впроваджено дизайн програмного забезпечення з клієнт-серверною архітектурою та внутрішнім класифікатором нейронної мережі. Реалізовано користувацький інтерфейс у вигляді Телеграм-боту для розв'язання поставленої задачі. Проведено валідування програмного забезпечення на реальних прикладах та наведено інструкцію користувача для подальшого розгортання.

Отже, створена інформаційна система здатна класифікувати зображення твердих побутових відходів як із мобільних застосунків, так і настільної версії Телеграму. Розроблений Телеграм-бот може бути використаний у побуті для особистісного сортування, як часткова заміна спеціалізованим системам збору та сортування сміття.

Результати дисертаційного дослідження апробовано на двох конференціях. Матеріали доповідей подано в додатках А та Б.

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Сафранов Т. А. Сучасний стан та можливості поводження з великогабаритними муніципальними відходами в регіонах України / Т. А. Сафранов, Т.П. Шаніна, В. Ю. Приходько // Український гідрометеорологічний журнал. – 2020. – № 25. – С. 115–126.

2. Мельник О. С. Переробка та утилізація побутових відходів : європейські практики для України [Електронний ресурс] / О. С. Мельник // Український щорічник європейських інтеграційних студій. – К. : APREI, 2020. – Вип. IV. – С. 97–111.

3. Про затвердження Правил надання послуг із збирання та вивезення твердих і рідких побутових відходів: документ z0457-00, втратив чинність. – Втрата чинності від 26.01.2009, підстава – z0023-09 [Електронний ресурс]. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/z0457-00#Text>

4. Про відходи: Закон України № 187/98-ВР від 05.03.98, ВВР, 1998, № 36–37, ст. 242 [Електронний ресурс]. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/187/98-%D0%B2%D1%80#Text>

5. Класифікатор відходів ДК 005-96 [Електронний ресурс] Державний класифікатор України. – Режим доступу: [http://konf.amsfo.com.ua/mizhnarodni-standarti-finansovo%D1%97-zvitnostiupravlinskij-aspekt/?fdx\\_switcher=true](http://konf.amsfo.com.ua/mizhnarodni-standarti-finansovo%D1%97-zvitnostiupravlinskij-aspekt/?fdx_switcher=true)

6. Бондар І. Л. Системи поводження з твердими побутовими відходами в українських містах, роль міського населення в роздільному збиранні сміття та рекомендації для органів місцевого самоврядування. Аналітичне дослідження / І. Л. Бондан, І. Л. Полтораченко. – К.: ПРООН / МПВСР, 2014. – 231 с.

7. Сокіран М. В. Класифікація відходів: порівняльний аналіз / М. В. Сокіран // “Порівняльно-аналітичне право”. Електронне наукове фахове видання юридичного факультету. – ДВНЗ “Ужгородський національний університет”, 2018. – № 2. – С. 270–272.

8. Міхно І. С. Методи утилізації відходів. Світовий досвід / І. С. Міхно // Економіка. Фінанси. Менеджмент: актуальні питання науки і практики. – 2015. – № 2. – С. 68–78.

9. Науково-дослідна робота: Дослідження впливу полігонів ТПВ на навколишнє середовище [Електронний ресурс]. Режим доступу:

<https://smida.gov.ua/db/participant/0552339>

10. Міщенко В. С. Удосконалення системи класифікації відходів і засади українського List of Wastes / В. С. Міщенко // Екологія і промисловість. – 2014. – № 2. – С. 107–111.

11. Міністерство екології та природних ресурсів [Електронний ресурс]. – Режим доступу: <http://www.meng.gov.ua>

12. Аналіз стану поводження з побутовими відходами в Україні за 2014-2019 роки. Міністерство регіонального розвитку, будівництва та житлово-комунального господарства України. [Електронний ресурс]. – Режим доступу: <http://www.minregion.gov.ua>.

13. Омеляненко Т. Напрями вдосконалення екологічної політики у сфері поводження з відходами в Україні на основі відповідного досвіду Європейського Союзу / Т. Омеляненко, Ю. Маковецька // Економіка природокористування і охорони довкілля. – 2015. – С. 35–43.

14. Petruk V. G. Household waste management. The European experiences / V. G. Petruk, F. Stalder, V. A. Ishchenko, I. V. Vasylykivskyi, R. V. Petruk, P. M. Turchyk, S. M. Kvatemyuk, M. I. Shymin, V. V. Volovodiuk. – Vinnytsia: “Nilan-Ltd.”, 2016. – 184 p.

15. Плаксіна А. О. Вплив полігонів ТПВ на навколишнє середовище / А. О. Плаксіна, В. В. Березуцький // Безпека людини в сучасних умовах : матер. доп. 7-ї Міжнар. наук.-метод. конф. та 105 Міжнар. конф. Європейської асоціації безпеки (EA8), 3-4 грудня 2015 р. – Харків : НТУ “ХПІ”, 2015. – С. 246–251.

16. Pohrebennyk V. The efficiency of wastewater treatment Landfill / V. Pohrebennyk, I. Podolchak // Proceedings of the International scientific conference

“New Trends in Ecological and Biological Research”, September 9th – 11th, 2015, University of Presov in Presov, Slovak Republic. – Presov, 2015. – P. 103–111.

17. Pohrebennyk V. Approaches to Integrated Waste Management System Modeling / V. Pohrebennyk, I. Koval, E. Dzhumelia // Proceedings of the 8-th International Joint Youth Science Forum “Litteris Et Artibus” & 13-th International Conference “Young Scientists Towards the Challenges of Modern Technology”, Lviv, November 22-24, 2018. – The Materials. – Lviv, Ukraine: Lviv Polytechnic National University, 2018. – P. 313–314.

18. Хоменко І. О. Проблеми утилізації сміття в Україні / Г. М. Гапоненко, І. О. Хоменко // Збірник тез Міжнародної науково-практичної конференції студентів, аспірантів і молодих вчених «Юність науки – 2017 : соціально-економічні та гуманітарні аспекти розвитку суспільства» (м. Чернігів, 26-27 квітня 2017 р.): ЧНТУ. – Чернігів: Черніг. нац. технол. ун-т, 2017. – С. 17–18.

19. Коваль І. І. Підвищення рівня екологічної безпеки регіону інтегрованою системою управління побутовими відходами : дис. ... канд. техн. наук : 26.06.01. Львів, 2019. – 198 с.

20. Матвєєв Ю. Б. Перспективи енергетичної утилізації твердих побутових відходів в Україні. Аналітична записка / Ю. Б. Матвєєв, Г. Г. Гелетуха // Біоенергетична асоціація України. – К. : БАУ, 2019. – № 22. – 48 с.

21. Уткіна К. Б. Впровадження комплексного управління відходами в Україні: Сучасний стан та перспективи / К. Б. Уткіна // Екологічна безпека. – 2013. – Вип. 2. – С. 23–27.

22. Brinez L. J. C. Automatic waste classification using computer vision as an application in Colombian high schools / L. J. C. Brinez, A. Rengifo, M. Escobar // Proceedings of the 6th Latin-American Conference on Networked and Electronic Media (LACNEM-2015). – IET Inc., 2015. – P. 1–5.

23. Sudha S. An automatic classification method for the environment / S. Sudha, M. Vidhyalakshmi, K. Pavithra // Proceedings of 2016 IEEE TIAR. – IEEE, 2016. – P. 26–34.

24. Torres-Garcia A. Intelligent waste separator / A. Torres-Garcia, O. Rodea-Aragion, O. Longoria-Gandara, F. Sanchez-Garcia, L. E. I. Gonzalez-Jimenez // *Computacion y Sistemas*. – № 19(3) – 2015. – P. 487–500.

25. Sakr G. E. Comparing deep learning and support vector machines for autonomous waste sorting / G. E. Sakr, M. Mokbel, A. Darwich, M. N. Khneisser, A. Hadi // *Proceedings of 2016 IEEE IMCET*. – IEEE, 2016. – P. 207–212.

26. Rahmi A. A. Classification of TrashNet dataset based on deep learning models / A. A. Rahmi, S. R. Keskin, M. Kaya, M. Hacıromeroglu // *Proceedings of the 2018 IEEE International Conference on Big Data (Big Data)*. – IEEE Inc., 2018. – P. 2058–2062.

27. Mittal G. SpotGarbage: Smartphone app to detect garbage using deep learning / G. Mittal, K. B. Yagnik, M. Garg, N. C. Krishnan. – ACM Press, 2016. – Vol. 14. – P. 940–945.

28. Yun Y.-S. Detection of GUI elements on sketch images using object detector based on deep neural networks / Y.-S. Yun, J. Jung, S. Eun, S.-S. So, J. Heo // *Proceedings of the 2018 International Conference on Green and Human Information Technology (ICGHIT-2020)*. – Springer, 2018. – Vol. 2, Issue 5. – P. 86–90.

29. AutoML Vision [Электронный ресурс] // Google Cloud. – Режим доступа: <https://cloud.google.com/vision/automl/docs>

30. Google Cloud Vision AI [Электронный ресурс] // Google Cloud. – Режим доступа: <https://cloud.google.com/vision>

31. Amazon Rekognition. Automate your image and video analysis with machine learning. [Электронный ресурс] // Amazon AWS. – Режим доступа: <https://aws.amazon.com/rekognition/?blog-cards.sort-by=item.additionalFields.createdDate&blog-cards.sort-order=desc>

32. IBM Visual Recognition [Электронный ресурс] // IBM Cloud. – Режим доступа: <https://cloud.ibm.com/catalog/services/visual-recognition>

33. Image Recognition Applications with Imagga's API [Электронный ресурс] // IBM Cloud. – Режим доступа: <https://imagga.com/>

34. Ren S. Faster R-CNN: Towards real-time object detection with region proposal networks / S. Ren, K. He, R. B. Girshick, J. Sun // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2015. – Vol. 39. – P. 1137–1149.
35. An end-to-end open source machine learning platform [Электронный ресурс] // TensorFlow. – Режим доступа: <https://www.tensorflow.org/>
36. Welcome to Flask [Электронный ресурс] // Flask, web development, one drop at a time. – Режим доступа: <https://flask.palletsprojects.com/en/1.1.x/>
37. Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine. [Электронный ресурс] // Node.js. – Режим доступа: <https://nodejs.org/en/>
38. Build cross-platform desktop apps with JavaScript, HTML, and CSS. [Электронный ресурс] // ElectronJS. – Режим доступа: <https://www.electronjs.org/>
39. Yang M. Classification of trash for recyclability status. / M. Yang, G. Thung // CS229 Project Report. – Stanford Inc., 2016. – 46 p.

**ДОДАТОК А**  
**(ОБОВ'ЯЗКОВИЙ)**  
**АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ**  
**НА КОНФЕРЕНЦІЇ КМОСС-2020**



Рисунок А.1 – Титульний лист збірника конференції КМОСС-2020 із тезами  
автора дипломної роботи

	7
<b>СЕКЦІЯ 3. ІНТЕЛЕКТУАЛЬНІ КОМП'ЮТЕРНІ СИСТЕМИ</b>	
<b>ПРОГРАМНИЙ ЗАСІБ ІДЕНТИФІКАЦІЇ ПЛАСТИКОВИХ ВІДХОДІВ НА ЗОБРАЖЕННІ</b>	108
Бацінко М.І., Парамонов А.І.	
<b>КРИТЕРІЇ РОЗБИТТЯ ІНФОРМАЦІЙНИХ СИСТЕМ НА СЕРВІСИ</b>	110
Горбенко О.Ю., Аушева Н.М.	
<b>CONVOLUTIONAL NEURAL NETWORK FOR IMAGE-BASED WASTE CLASSIFICATION</b>	112
Нгунунська Н.В., Колонієць О.В., Радіук Р.М.	
<b>МЕТОДИ АПРОКСИМАЦІЇ ФУНКЦІЙ БАГАТЬОХ ЗМІННИХ В ЗАДАЧАХ ОПТИМІЗАЦІЇ ЕНЕРГОКОНДЕНСОВАНИХ СИСТЕМ</b>	114
Єфремов Д.С., Коротка Л.І.	
<b>РОЗРОБКА ХМАРНОЇ БАЗИ ДАНИХ ГЕНЕТИЧНИХ ЕКСПЕРМЕНТІВ НА ОСНОВІ GOOGLE CLOUD PLATFORM</b>	116
Желізняк В. О., Олевський В. І.	
<b>ДОСЛІДЖЕННЯ АЛГОРИТМІВ РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ ЗАСОБАМИ PYTHON</b>	118
Животкевич О. В., Гнатушенко Вік. В, Фененко Т. В.	
<b>ВИКОРИСТАННЯ ТЕХНОЛОГІЇ ДОПОВНЕНОЇ РЕАЛЬНОСТІ ДЛЯ ВІЗУАЛІЗАЦІЇ ВІРТУАЛЬНОГО ОБ'ЄКТА В РЕАЛЬНОМУ ІНТЕР'ЄСІ</b>	120
Завадський К.В., Кодола Г.М.	
<b>ПРОБЛЕМИ ВИБОРУ ФУНКЦІЇ АКТИВАЦІЇ ПРИ НАВЧАННІ НЕЙРОННИХ МЕРЕЖ</b>	122
Зеленцов Д.Г., Шаптала Т.М.	
<b>ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ НАВЧАННЯ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ В ЗАДАЧАХ РЕГРЕСІЇ</b>	124
Кульбердиев Б. А., Хорошилов С. В.	

Рисунок А.2 – Зміст збірника конференції КМОСС-2020 із тезами автора  
дипломної роботи

## CONVOLUTIONAL NEURAL NETWORK FOR IMAGE-BASED WASTE CLASSIFICATION

**Hrypynska N.V., Kolomiets O.V., Radiuk P.M.**  
Khmelnyskyi National University, Khmelnytskyi, Ukraine

In recent decades, solid waste management in large cities has become a challenging issue because of the increasing amount of waste daily generated by citizens and enterprises. Modern computer vision and deep learning methods may facilitate the automatic detection and classification of waste for further recycling. This study proposes designing a deep learning architecture based on convolutional neural networks for solid waste classification tasks. We trained our model and compared it with other deep architectures that are recognized in the classification tasks. As a result of experiments, the combined Inception-ResNet model achieved the best classification accuracy.

Nowadays, city dwellers encounter a decline in available natural resources in urban areas and experience various environmental problems. The cities' ecology suffers from the amount of garbage that is generated daily [1]. The average European makes 517 kilos of waste per year, of which just a small percentage is recycled [2]. Most garbage segregation is performed manually, which leads to diverse health problems. The manual process of recycling is also time-consuming and requires huge taxes from citizens [3]. Therefore, there is an urgent task to establish effective recycling of accumulated waste. These solutions might reduce environmental pollution and solve the problem of improving citizens' lives.

The presented in this study approach adopts a supervised method to classify several types of waste depicted on images: glass, paper, cardboard, and plastic. Fig. 1 shows the general scheme of the waste image classification process.

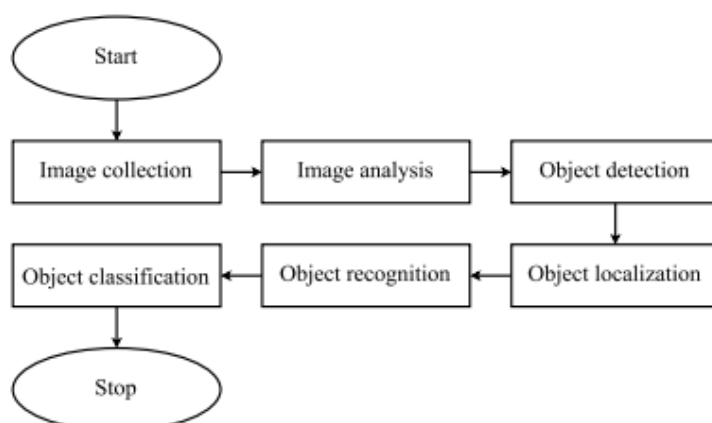


Figure 1 – Flow chart of the system for the waste image classification

Рисунок А.3 – Перша сторінка тез конференції КМОСС-2020 автора  
дипломної роботи

The training of the neural networks was conducted on the current TrashNet dataset [4]. The benchmark dataset comprises 2,527 images and split into 501 images of glass, 594 of paper, 403 of cardboard, 482 of plastic, 410 of metal, and 137 general trash.

The original collection of images was randomly partitioned into three subsets: training, validation, and test. All the subsets have the same rate of classes. As the number of images is small, we used 80% of them for training, 10% for validation, and 10% for tests. We adopted a 5-fold cross-validation strategy by creating randomly five training, test, and validation datasets. Moreover, the training sample is increased through the data augmentation technique. The results given in this subsection correspond to the average of the five runs of the test datasets. Table 1 compares our best results, achieved by the ResNet model, with other deep learning models applied to waste classification.

Table 1 – The comparison of convolutional neural networks for waste classification

Architecture	Accuracy
GarbeNet (CNN)	0.871
AlexNet	0.645
GoogleNet	0.783
The proposed architecture	0.875

According to table 1, the proposed model surpasses all the other architectures. Eventually, we evaluated several CNN architectures for the automatic classification of waste. In our experiments on the TrashNet dataset, the best classification results were achieved with a proposed ResNet architecture with 87.5% of average accuracy.

Our further work will be devoted to generating realistic synthetic images with multiple types of garbage, which can be used to train our models and test them with real images that combine several waste classes.

#### List of reference

1. Brinez L.J.C. Automatic waste classification using computer vision as an application in Colombian high schools / L.J.C. Brinez, A. Rengifo, M. Escobar // Proceedings of the 6th Latin-American Conference on Networked and Electronic Media (LACNEM-2015). – IET, 2015. P. 1–5.
2. Sudha S. An automatic classification method for the environment / S. Sudha, M. Vidhyalakshmi, K. Pavithra // Proceedings of 2016 IEEE TIAR. – IEEE, 2016. – P. 26–34. <https://doi.org/10.1109/TIAR.2016.7801215>
3. Sakr G.E. Comparing deep learning and support vector machines for autonomous waste sorting / G.E. Sakr, M. Mokbel, A. Darwich, M.N. Khneisser, A. Hadi // Proceedings of 2016 IEEE IMCET. – IEEE, 2016. – P. 207–212.
4. Yang M. Classification of trash for recyclability status. / M. Yang, G. Thung // CS229 Project Report. – Stanford Inc., 2016.

**ДОДАТОК Б**  
**(ОБОВ'ЯЗКОВИЙ)**  
**АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ**  
**НА КОНФЕРЕНЦІЇ АПКН-2020**

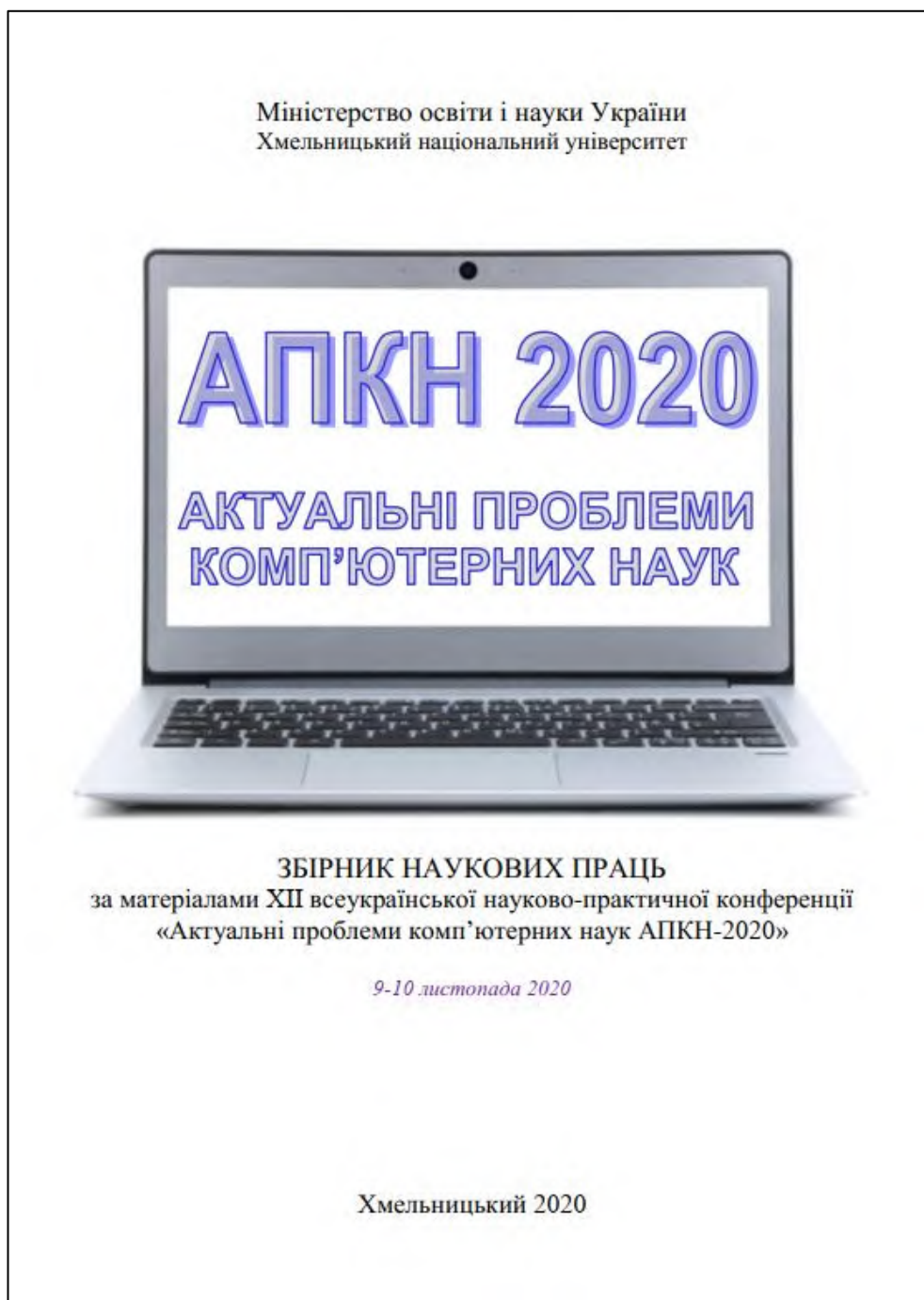


Рисунок Б.1 – Титульний лист збірника конференції АПКН-2020 із тезами автора дипломної роботи

<b>Гордійчук Б. Г., Манзюк Е. А., Скрипник Т. К.</b> Виявлення аномалій в даних.....	72
<b>Городний М. С., Тітова В. Ю.</b> Розробка архітектури додатку на основі технологій «розумний будинок» та «інтернет речей» .....	75
<b>Гребінчук А. Д., Поліщук В. Ю., Форкун І. В.</b> Модель багаторівневої автоматизованої системи керування будівельним виробництвом.....	78
<b>Грипинська Н. В., Дяблов Б. В.</b> Автоматизована система планування рекламної кампанії для малого та середнього бізнесу .....	82
<b>Грипинська Н. В., Коломісць О. В.</b> Автоматизована система виявлення та класифікації твердих побутових відходів на зображеннях.....	86
<b>Демчук Б. Р.</b> Динамічна модель перебігу вірусного захворювання .....	91
<b>Долгополов С. Ю., Цюцюра М. І.</b> Інноваційність використання технології глибокого навчання у контрольно-вимірювальному приладі будівельного спрямування «Builder of the Future» .....	97
<b>Драпатий О. В., Драч І. В.</b> Методи мережевого моделювання. Сучасні напрямки.....	102
<b>Євдокімов О. В., Татаревська О. Г., Радельчук Г. І.</b> Автоматизація і комп'ютерно-інтегровані технології моніторингу сонячних панелей у реальному масштабі часу.....	113
<b>Живага В. В., Шевченко Д. О.</b> Інтегрована Internet of Things система на основі одноплатного комп'ютеру .....	115
<b>Жовнір М. Ю., Кисіль Т. М.</b> Неформальне пояснення ДСМ-методу автоматичного породження гіпотез в задачах адаптивної поведінки ІС.....	120
<b>Злотаренчук О. І., Кучерук О. Я.</b> Сучасні підходи до організації маршрутів комплектації замовлень на складі.....	123
<b>Казлаускайте А. С., Шендрік С. О.</b> Інформаційна технологія визначення впливу погодних умов на продуктивність альтернативних джерел енергії.....	127

Рисунок Б.2 – Зміст збірника конференції АПКН-2020 із тезами автора  
дипломної роботи

УДК 004.932.72'1

Грипинська Н. В., Коломієць О. В.

Хмельницький національний університет

**АВТОМАТИЗОВАНА СИСТЕМА ВИЯВЛЕННЯ ТА КЛАСИФІКАЦІЇ  
ТВЕРДИХ ПОБУТОВИХ ВІДХОДІВ НА ЗОБРАЖЕННЯХ**

*В XXI столітті питання збереження екології є одним із найбільш пріоритетних для людства. Проблема накопичення відходів та їхня подальша утилізація стає все більш критичною. Тому нині актуальною постає задача розроблення автоматизованих технологій сортування різних типів відходів. У даній роботі представлено автоматизовану інформаційну систему для виявлення твердих побутових відходів на зображеннях. Система може виявляти об'єкти на зображеннях та класифікувати їх на три категорії: звалище, перероблення, папір. За класифікатор взято згорткову нейронну мережу Fast R-CNN. У роботі наведено схему автоматизованої системи та проведено експерименти щодо її ефективності. Загальна точність класифікації склала 69,1%.*

*In the XXI century, environmental protection is becoming one of the highest priorities for humanity. The problem of waste accumulation and their further utilization is becoming increasingly critical. Therefore, today the task of developing automated technologies for sorting different types of waste is urgent. This paper presents an automated information system for detecting solid waste in images. The system can classify images into three categories: landfill, recycling, paper. The convolutional neural network Fast R-CNN was used as a classifier. In this study, the scheme of the automated system and relevant experiments are presented. The proposed model achieved the final classification accuracy of 69.1%.*

Щорічно в Україні накопичується сотні тонн сміття. Кожний українець генерує майже 300 кг відходів щороку; спільний доробок усього населення країни складає в середньому 35 млн відходів [1] і ці показники постійно зростають. Водночас більш як 50 % цих відходів можуть бути перероблені, хоча нині в країні переробляється менш як 30 % усього сміття [2]. На жаль, ситуація із відходами в Україні гіршає з року в рік.

Системи штучного інтелекту можуть стати ефективним рішенням для автоматизованого сортування та машинної класифікації твердих побутових відходів, що дасть змогу значно зменшити кількість сміття в Україні. Наприклад, у роботі [3] пропонують використовувати оптимізовані глибокі нейронні мережі для класифікації вибраних класів об'єктів, що підлягають переробленню. На вхід такої системи поступає зображення з перемішаними типами відходів. Далі система локалізує ознаки об'єктів на зображенні та класифікує їх на кілька категорій. Отже, метою даної роботи є розроблення автоматизованої системи сортування та класифікації твердих побутових відходів на три категорії: ті відходи, що

Рисунок Б.3 – Перша сторінка тез конференції АПКН-2020  
автора дипломної роботи

відправляються на звалище, ті відходи, що готові до вторинного перероблення та папір.

На сьогодні згорткова нейронна мережа (ЗНМ) під назвою Fast R-CNN демонструє найкращі результати виявлення об'єктів на зображеннях. Регіональна мережа з пропозиціями (RPN) [4] в складі Fast R-CNN розділяє повнорозмірні згорткові функції, що дає змогу виділяти контури зображень без додаткових втрат ресурсів. RPN – це повністю згорткова нейронна мережа, яка виділяє контури об'єктів та передбачає оцінки об'єктивності в кожній позиції. Підмережа RPN навчена наскрізно, щоби генерувати високоякісні контури, які далі використовуються загальною мережею Fast R-CNN для виявлення ознак. Рисунок 1 зображає топологію нейронної мережі типу Fast R-CNN з регіональними пропозиціями.

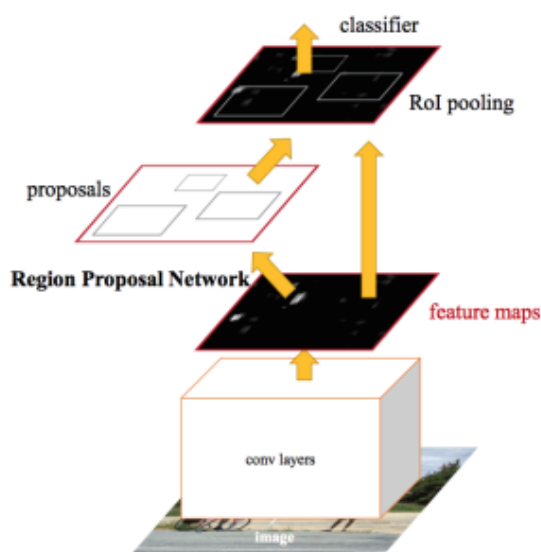


Рисунок 1 – Схема архітектури Fast R-CNN [4]

Нейронна мережа Fast R-CNN складена в такий спосіб, що не потребує великого набору зображень для ефективного навчання. У представленій роботі для навчання мережі використано готовий набір даних із зображеннями твердих побутових відходів [5]. Набір даних загалом містить 2500 зображень із відокремленими відходами різних типів, 2000 зображень у навчальному наборі та 500 у тестовому.

Використаний у роботі класифікатор отримує на вхід зображення розміром  $768 \times 1024$  із двома або більше об'єктами на звичайному фоні. Водночас на кожному зображенні стандартний фон видаляється, щоби виділити обмежувальне поле навколо кожного об'єкта. Далі на зображення накладається білий фон розміром  $768 \times 1024$ , що об'єднується з 2–6 об'єктами відходів у випадкових місцях. Ці об'єкти можуть перекриватися або перетинатися та мати різний розмір. Загалом модель класифікації налаштовано у такий спосіб, що вона виявляє регіональні пропозиції та класифікує об'єкти за трьома класами: звалище, вторинне перероблення та папір.

За базову архітектуру в роботі використано попередньо навчену модель Fast R-CNN зі зміненими останніми повнозв'язними шарами. Тобто були використані попередньо навчені функції нижчого рівня. Модель навчено на наборі даних із 2000 зображень; кожне зображення із рівномірним поданням кожного класу на різних прикладах. У роботі використано середню точність (*avg\_prec*) для оцінювання продуктивності роботи моделі. Формалізований запис *avg\_prec* має вигляд:

$$avg\_prec = \int_0^1 p(x) dx \quad (1)$$

Водночас крива точності та повноти визначається через обчислення IoU (перетину об'єднання) між передбачуваними та істинними значеннями класифікації. Також у роботі використано функцію втрат, відповідно до [3]:

$$L(p_i, t_i) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \frac{\lambda}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*) \quad (2)$$

Функція втрат (2) є сумою двох нормованих втрат над регіонами зображення з індексами  $i$ , де  $p_i$  – ймовірність представлення регіоном об'єкта  $i$ ,  $p_i^* \in \{0,1\}$  – істинна інформація про те, чи є регіон насправді об'єктом чи ні,  $t_i$  – координати запропонованої моделі обмежувальної рамки,  $t_i^*$  – істинні координати об'єкта.

У результаті проведених обчислювальних експериментів на наборі даних із використанням запропонованого розбиття даних, було досягнуто середню точність *avg\_prec* = 0.691. Розподіл точності класифікації з огляду на різні значення розмірів піднаборів даних зображено на рисунку 2.

Приклад класифікації із виділеними об'єктами відходів проілюстровано на рисунку 3.

Рисунок Б.5 – Третя сторінка тез конференції АПКН-2020  
автора дипломної роботи

Accuracy vs Batch Size

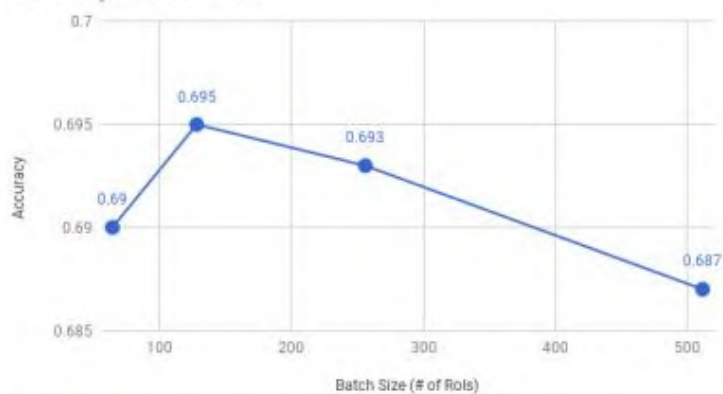


Рисунок 2 – Точність класифікації серед трьох класів для різних значень розмірів піднаборів даних

landfill detections with  $p(\text{landfill} | \text{box}) \geq 0.6$ 

Рисунок 3 – Приклад виведення моделі після попереднього оброблення набору даних

Рисунок Б.6 – Четверта сторінка тез конференції АПКН-2020  
автора дипломної роботи

Фінальні результати класифікації для трьох категорій відходів представлено в таблиці 1.

Таблиця 1 – Остаточна точність класифікації для кожної категорії

Клас	Точність класифікації
Звалище	0.713
Вторинна переробка	0.706
Папір	0.625
<i>avg_prec</i>	0.691

У результаті експериментів помітно, що модель досягла найгірших показників класифікації для категорії «папір». Такий результат міг бути отриманий у зв'язку з білим фоном оброблених зображень, що міг вплинути на виявлення білого паперу, і в такий спосіб погіршити результати класифікації. Подальші дослідження спрямовані на тестування запропонованої моделі на реальних зображеннях груп відходів.

#### Перелік посилань

1. Сафранов Т. А. Сучасний стан та можливості поводження з великогабаритними муніципальними відходами в регіонах України / Т. А. Сафранов, Т. П. Шаніна, В. Ю. Приходько // Український гідрометеорологічний журнал. – 2020. – №25. – С. 115–126. <https://doi.org/10.31481/uhmj.25.2020.11>
2. Мельник О. С. Переробка та утилізація побутових відходів : європейські практики для України [Електронний ресурс] / О. С. Мельник // Український щорічник європейських інтеграційних студій. – К. : APREI, 2020. – Вип. IV. – С. 97–111.
3. Yang, Z. WasNet: A neural network-based garbage collection management system / Z. Yang, D. Li // IEEE Access. – 2020. – Vol. 8. – P. 103984–103993.
4. Ren S. Faster R-CNN: Towards real-time object detection with region proposal networks / S. Ren, K. He, R. B. Girshick, J. Sun // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2015. – Vol. 39. – P. 1137–1149.
5. Thung G. Classification of trash for recyclability status / G. Thung, M. Yang // CS229 Project Report. – Stanford EDU, 2016. – 46 p.

Рисунок Б.7 – П'ята сторінка тез конференції АПКН-2020  
автора дипломної роботи

**ДОДАТОК В**  
**(ОБОВ'ЯЗКОВИЙ)**  
**ЛІСТИНГ ПРОГРАМНОГО КОДУ**

```

import base64
import numpy as np
import io
import PIL
from PIL import Image
import tensorflow.keras
from tensorflow.keras import backend as kerBack
from tensorflow.keras.models import Sequential
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.preprocessing.image import img_to_array
from flask import Flask
from flask import request
from flask import jsonify

app = Flask(__name__)
# app.config['ENV'] = 'development'
# app.config['DEBUG'] = True
# app.config['TESTING'] = True
print(PIL.PILLOW_VERSION)

def prepare_image(input_image, target_size):
    if input_image.mode != 'RGB':
        input_image = input_image.convert('RGB')
    input_image = input_image.resize(target_size)
    input_image = img_to_array(input_image)
    input_image = np.expand_dims(input_image, axis=0)

    return tensorflow.keras.applications.mobilenet.preprocess_input(input_image)
def get_cnn_model():
    global cnn_model
    print('CNN try to load.')
    cnn_model = load_model('CNN_model Sakach.h5')
    print('CNN model was successfully loaded.')

print('Initializing. Loading model...')
get_cnn_model()

@app.route('/classify', methods=['POST'])
def classify():
    req_data = request.get_json(force=True)

    img_coded = req_data['image']
    img_decoded = base64.b64decode(img_coded)
    img = Image.open(io.BytesIO(img_decoded))
    unclassified_img = prepare_image(img, target_size=(224, 224))

```

```

result = cnn_model.predict(unclassified_img).tolist()

response = {
    'classes': {
        'cardboard': result[0][0],
        'food': result[0][1],
        'glass': result[0][2],
        'metal': result[0][3],
        'paper': result[0][4],
        'plastic': result[0][5]
    }
}
return jsonify(response)

```

#### Jupyter Notebook file for CNN manipulation (CNN\_Prepere.inpyb)

```

{
  "cells": [
    {
      "cell_type": "code",
      "execution_count": null,
      "metadata": {},
      "outputs": [],
      "source": [
        "import numpy as np\n",
        "import tensorflow.keras\n",
        "from tensorflow.keras import models\n",
        "from tensorflow.keras import backend as K\n",
        "from tensorflow.keras.layers import Dense\n",
        "from tensorflow.keras.optimizers import Adam\n",
        "from tensorflow.keras.metrics import categorical_crossentropy\n",
        "from tensorflow.keras.preprocessing.image import ImageDataGenerator\n",
        "from tensorflow.keras.preprocessing import image\n",
        "from tensorflow.keras.models import Model\n",
        "from keras.applications import imagenet_utils\n",
        "from sklearn.metrics import confusion_matrix\n",
        "from IPython.display import Image\n",
        "import itertools\n",
        "import matplotlib.pyplot as plt\n",
        "%matplotlib inline"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": null,
      "metadata": {},
      "outputs": [],
      "source": [
        "from keras.preprocessing.image import ImageDataGenerator, array_to_img,\nimg_to_array, load_img\n",
        "from PIL import Image\n",
        "import glob\n",
        "\n",
        "\n",
        "datagen = ImageDataGenerator(\n",
        "    rotation_range=45,\n",
        "    width_shift_range=0.15,\n",
        "    height_shift_range=0.15,\n",
        "    shear_range=0.25,\n",
        "    zoom_range=0.25,\n",
        "    horizontal_flip=True,\n"
      ]
    }
  ]
}

```

```

"        fill_mode='reflect')\n",
"\n",
"image_list = []\n",
"for filename in glob.glob('augment/*.jpg'):\n",
"    im=Image.open(filename)\n",
"    image_list.append(im)\n",
"\n",
"    img = load_img(filename)\n",
"    x = img_to_array(img)\n",
"    x = x.reshape((1,) + x.shape)\n",
"\n",
"# the .flow() command below generates batches of randomly transformed images\n",
"# and saves the results to the `preview/` directory\n",
"    i = 0\n",
"    for batch in datagen.flow(x, batch_size=1,\n",
"                             save_to_dir='preview', save_prefix='food',\n",
save_format='jpeg'):\n",
"        i += 1\n",
"        if i > 7: # count of iterations before data augmentation stops\n",
"            break
]
},
{
"cell_type": "code",
"execution_count": null,
"metadata": {},
"outputs": [],
"source": [
"mobileV2 = tensorflow.keras.applications.mobilenet_v2.MobileNetV2()"
]
},
{
"cell_type": "code",
"execution_count": null,
"metadata": {},
"outputs": [],
"source": [
"def prepare_image(file):\n",
"    img_path = 'waste_images/'\n",
"    img = image.load_img(img_path + file, target_size=(224,224))\n",
"    img_array = image.img_to_array(img)\n",
"    img_array_expanded_dims = np.expand_dims(img_array, axis=0)\n",
"    return keras.applications.mobilenet.preprocess_input(img_array_expanded_dims)"
]
},
{
"cell_type": "code",
"execution_count": null,
"metadata": {},
"outputs": [],
"source": [
"mobileV2.summary()"
]
},
{
"cell_type": "code",
"execution_count": null,
"metadata": {},
"outputs": [],
"source": [
"mobileV2 = tensorflow.keras.applications.mobilenet_v2.MobileNetV2()\n",
"x = mobileV2.layers[-2].output"
]
}

```

```

]
},
{
  "cell_type": "code",
  "execution_count": null,
  "metadata": {},
  "outputs": [],
  "source": [
    "Prediction = Dense(6, activation='softmax')(x)\n",
    "mobV2_model = Model(inputs = mobileV2.input, outputs = Prediction)"
  ]
},
{
  "cell_type": "code",
  "execution_count": null,
  "metadata": {},
  "outputs": [],
  "source": [
    "mobV2_model.summary()"
  ]
},
{
  "cell_type": "code",
  "execution_count": null,
  "metadata": {},
  "outputs": [],
  "source": [
    "for layer in mobV2_model.layers[:-1]:\n",
    "    layer.trainable = False"
  ]
},
{
  "cell_type": "code",
  "execution_count": null,
  "metadata": {},
  "outputs": [],
  "source": [
    "mobV2_model.summary()"
  ]
},
{
  "cell_type": "code",
  "execution_count": null,
  "metadata": {},
  "outputs": [],
  "source": [
    "train_path = 'waste_images/train'\n",
    "validate_path = 'waste_images/validate'\n",
    "test_path = 'waste_images/test'"
  ]
},
{
  "cell_type": "code",
  "execution_count": null,
  "metadata": {},
  "outputs": [],
  "source": [
    "train_batches
    ImageDataGenerator(preprocessing_function=tensorflow.keras.applications.mobilenet.prep
    rocess_input).flow_from_directory(train_path, target_size=(224, 224),
    batch_size=12)\n",
    =

```

```

    "validate_batches" =
ImageDataGenerator(preprocessing_function=tensorflow.keras.applications.mobilenet.prep
rocess_input).flow_from_directory(validate_path, target_size=(224, 224),
batch_size=12)\n",
    "test_batches" =
ImageDataGenerator(preprocessing_function=tensorflow.keras.applications.mobilenet.prep
rocess_input).flow_from_directory(test_path, target_size=(224, 224), batch_size=12)"
    ]
    },
    {
    "cell_type": "code",
    "execution_count": null,
    "metadata": {},
    "outputs": [],
    "source": [
        "mobV2_model.compile(Adam(.00025), loss='categorical_crossentropy',
metrics=['accuracy'])"
    ]
    },
    {
    "cell_type": "code",
    "execution_count": null,
    "metadata": {},
    "outputs": [],
    "source": [
        "mobV2_model.fit_generator(train_batches, steps_per_epoch=27,
validation_data=validate_batches, validation_steps=3, epochs=42) #"
    ]
    },
    {
    "cell_type": "code",
    "execution_count": null,
    "metadata": {},
    "outputs": [],
    "source": [
        "\n",
        "for layer in mobV2_model.layers:\n",
        "    layer.trainable = True\n",
        "\n",
        "for layer in mobV2_model.layers[:-36]:\n",
        "    layer.trainable = False"
    ]
    },
    {
    "cell_type": "code",
    "execution_count": null,
    "metadata": {},
    "outputs": [],
    "source": [
        "mobV2_model.compile(Adam(.00002), loss='categorical_crossentropy',
metrics=['accuracy'])"
    ]
    },
    {
    "cell_type": "code",
    "execution_count": null,
    "metadata": {},
    "outputs": [],
    "source": [
        "mobV2_model.summary()"
    ]
    },
    },

```



```

    "cell_type": "code",
    "execution_count": null,
    "metadata": {},
    "outputs": [],
    "source": []
  },
  {
    "cell_type": "code",
    "execution_count": null,
    "metadata": {},
    "outputs": [],
    "source": []
  },
  {
    "cell_type": "code",
    "execution_count": null,
    "metadata": {},
    "outputs": [],
    "source": []
  }
],
"metadata": {
  "kernelspec": {
    "display_name": "Python 3",
    "language": "python",
    "name": "python3"
  },
  "language_info": {
    "codemirror_mode": {
      "name": "ipython",
      "version": 3
    },
    "file_extension": ".py",
    "mimetype": "text/x-python",
    "name": "python",
    "nbconvert_exporter": "python",
    "pygments_lexer": "ipython3",
    "version": "3.7.3"
  }
},
"nbformat": 4,
"nbformat_minor": 2}

```

**ДОДАТОК Г**  
**(ОБОВ'ЯЗКОВИЙ)**  
**СЛАЙДИ ПРЕЗЕНТАЦІЇ**



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
Кафедра телекомунікацій, медійних та інтелектуальних технологій

**Коломієць О.В.**

**Телеграм-бот для класифікації  
зображень твердих побутових відходів**

Науковий керівник:  
к.ф.-м.н., доц. Грипинська Н.В.

Хмельницький - 2020

Рисунок Г.1 – Перший слайд презентації



## Об'єкт та предмет дослідження

**Об'єктом** дослідження є процес сортування твердих побутових відходів.

**Предметом** дослідження є методи класифікації зображень твердих побутових відходів.

2

Рисунок Г.2 – Другий слайд презентації



## Мета та завдання дослідження

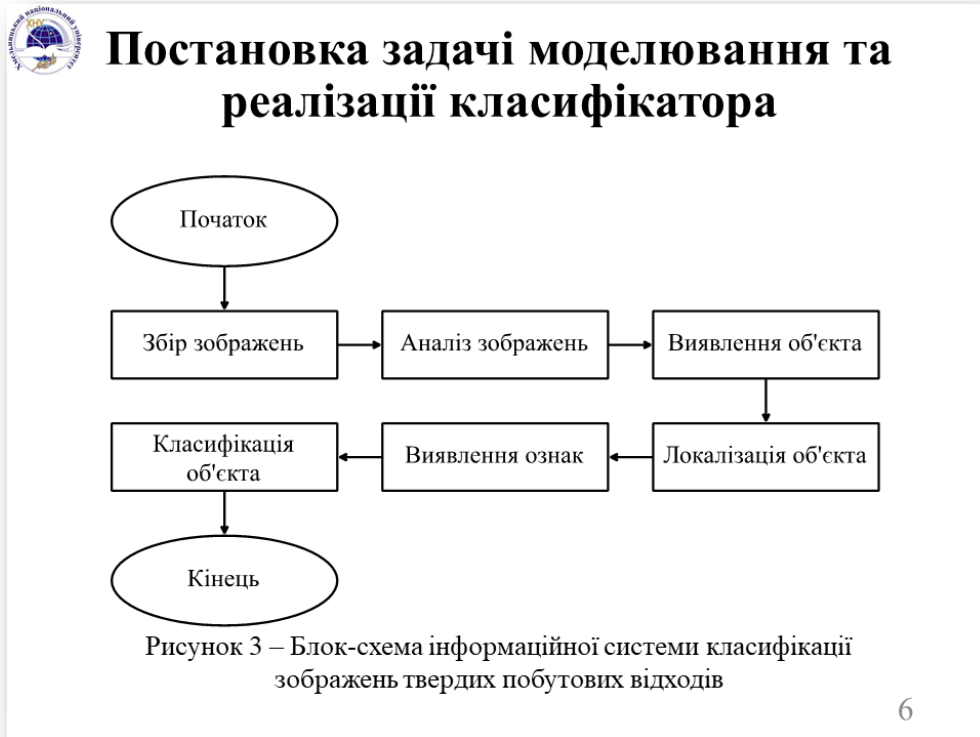
**Метою** дипломної магістерської роботи є клієнт-серверної інформаційної системи у вигляді Телеграм-боту для класифікації твердих побутових відходів.

Для досягнення поставленої мети необхідно виконати такі **завдання**:

1. Провести аналіз стану та тенденцій розвитку методів та систем управління твердими побутовими відходами.
2. Розробити класифікатор зображень твердих побутових відходів.
3. Побудувати модель клієнт-серверної системи управління відходами.
4. Розробити інформаційну систему у вигляді Телеграм-боту для класифікації зображень.

3

Рисунок Г.3 – Третій слайд презентації



6

Рисунок Г.4 – Шостий слайд презентації

## Класифікатор зображень

У роботі використано L2-середньоквадратичну помилку, як функцію втрат виду

$$\mathcal{L}(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^N (\hat{f}(x_i) - y_i)^2 + \frac{\lambda}{2N} \sum_{i=1}^D \|w_i\|_2^2, \quad (1)$$

де  $w_i \in \mathbf{w} \in \mathbb{R}^D$  – множина ваг моделі;  $D$  – розмір зображень у пікселях;  $x_i \in \mathbf{X} \in \mathbb{R}^{N \times D}$  – множина зображень навчального набору даних;  $y_i \in \mathbf{y} \in \mathbb{R}^N$  – множина класів;  $N$  – кількість класів;  $\hat{f}(x_i)$  – цільова функція апроксимації.  $\lambda$  – параметр балансування.

Для виявлення ознак об'єктів на зображенні у даній роботі використано згорткову нейронну мережу з операцією згортки:

$$(K \times I)(u, v) = \sum_{k=-M}^M \sum_{q=-M}^M K(k, q) I(u+k, v+q), \quad (2)$$

де  $K(k, q)$  – ядро згорткового шару;  $M$  – кількість пікселів згорткового ядра;  $I(u+k, v+q)$  – мапа ознак на цільовому зображенні;  $(u, v)$  – позиція пікселя на мапі зображення;  $(k, q)$  – зміщення згорткового ядра по координатній площині зображення.

Рисунок Г.5 – Сьомий слайд презентації



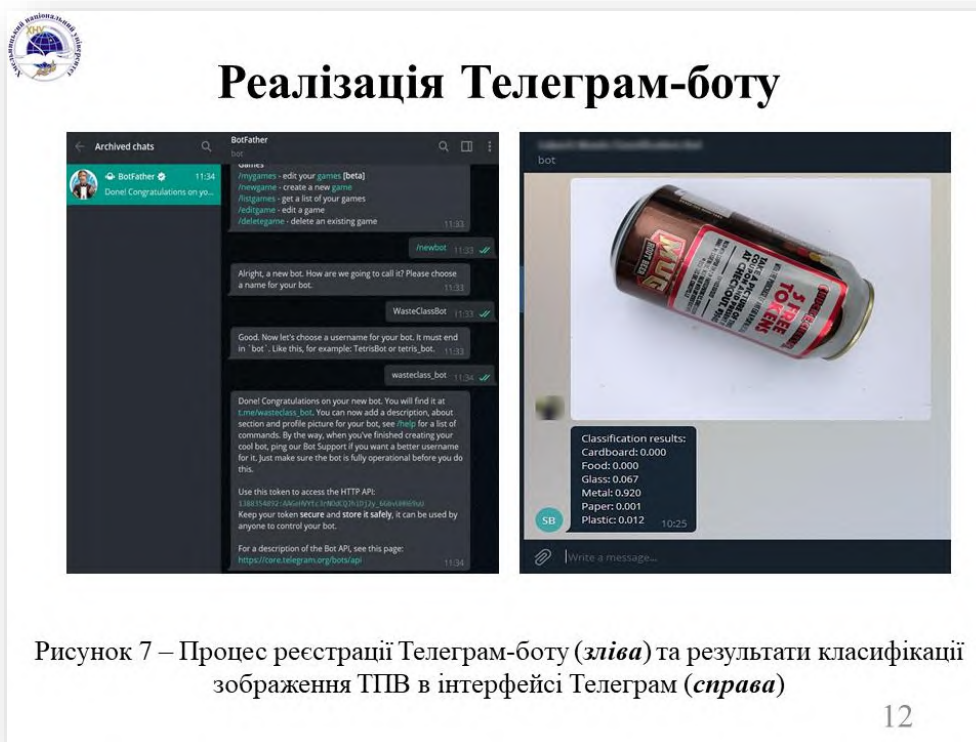
Рисунок Г.6 – Восьмий слайд презентації

**Результати навчання  
нейронної мережі**

Розмір зразків	К-сть ітерацій	Розмір партії зображень	Час тренування, год	Коефіцієнт навчання	Точність класифікації
10 000 (1:4)	1000	10	5,15	$10^{-5} - 10^{-3}$	0,686
		50	3,37		0,701
		100	7,25		0,796
10 000 (1:4)	2000	10	9,43	$10^{-6} - 10^{-4}$	0,736
		50	6,49		0,793
		100	11,04		0,741
10 000 (1:4)	3000	10	12,14	$2 \cdot 10^{-6} - 2 \cdot 10^{-4}$	0,812
		50	13,67		0,823
		100	17,4		0,845

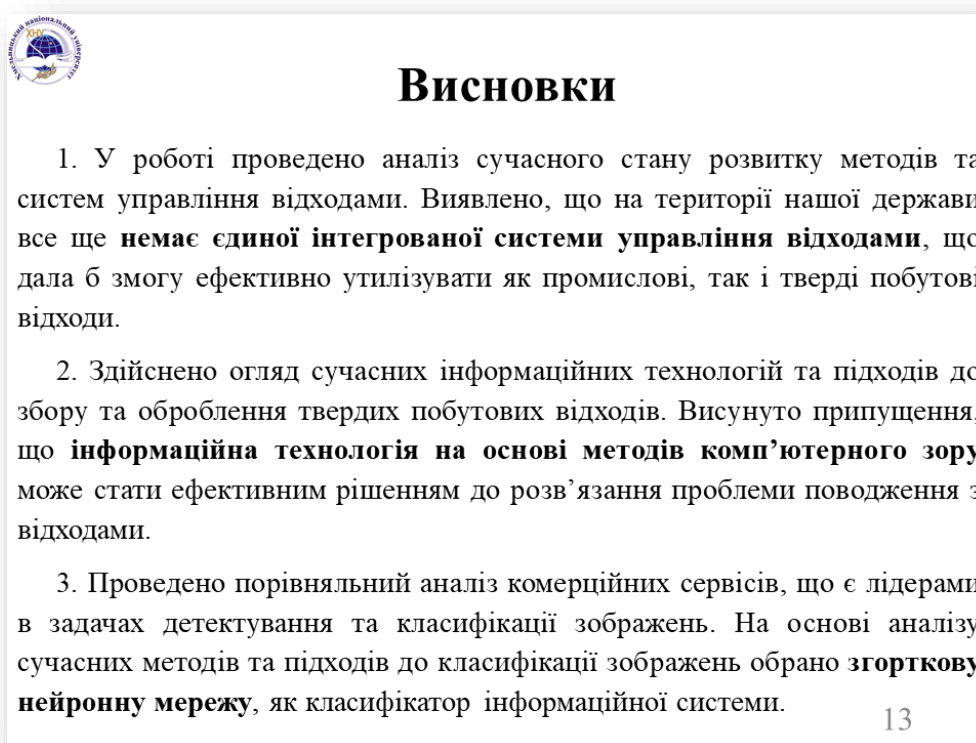
9

Рисунок Г.7 – Дев'ятий слайд презентації



12

Рисунок Г.8 – Дванадцятий слайд презентації



13

Рисунок Г.9 – Тринадцятий слайд презентації



## Висновки

4. Реалізовано модель згорткової нейронної мережі. Нейронну мережу навчено та протестовано на готовому **наборі даних TrashNet**, що містить 2527 зображень і поділений на шість класів. Навчена нейронна мережа **Faster R-CNN** досягла максимальної середньої точності класифікації в **84,5 %**.

5. Реалізовано користувацький інтерфейс у вигляді **Телеграм-боту** для розв'язання поставленої задачі. Проведено валідування програмного забезпечення на реальних прикладах та наведено інструкцію користувача для подальшого розгортання.

6. Розроблений Телеграм-бот може бути використаний у побуті для **особистісного сортування**, як часткова заміна спеціалізованим системам збору та сортування сміття.

Загальний обсяг роботи: 107 сторінок, 41 рисунок, 6 таблиць, 4 додатки, 39 посилань

14

Рисунок Г.10 – Чотирнадцятий слайд презентації



## Апробація роботи

### 1. Тези конференції КМОСС-2020

Нрупунська Н. В., Коломієць О. В. Convolutional neural network for image-based waste classification. VI International scientific and technical conference on the occasion of the 90th anniversary of USCTU “Computer Modeling and Optimization of Complex Systems”, (CMOCS-2020) : proceedings of IX inter. scien.-tech. conf (Dnipro, 4-6 Nov, 2020). Dnipro : Ukrainian State University of Chemical Technology, 2020. P. 112–113. doi:10.32434/CMOCS-2020.

### 2. Тези конференції АПКН-2020

Грипінська Н. В., Коломієць О. В. Автоматизована система виявлення та класифікації твердих побутових відходів на зображеннях. XII Всеукраїнська науково-практична конференція “Актуальні проблеми комп'ютерних наук”, (АПКН-2020) : матеріали XII всеукр. наук.-практ. конф. (м. Хмельницький, 9-10 лис. 2020 р.). Хмельницький : Хмельницький національний університет, 2020. С. 86–90.

15

Рисунок Г.11 – П'ятнадцятий слайд презентації

**ДОДАТОК Д  
(ОБОВ'ЯЗКОВИЙ)  
АНТИПЛАГІАТ**

Mon Nov 23 12:55:46 EET 2020, Стецюк Віктор Іванович, Хмельницький національний університет, ХНУ

**Anti-Plagiarism v-15.257**

**Максимальное совпадение с одним документом 1.0%**

**Словари проверки: en\_US, ru\_RU, ua\_UA. Ошибок в документах: 11%**

ID: 80926 Название: Телеграм-бот для класифікації зображень твердих побутових відходів Добавлено в БД: 2020-11-23 Авторы: Коломієць Ольга Вікторівна Руководители: Гришинська Надія Василівна Консультанты: Оponentы:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	108118	922	2577 (2%)	38 (4%)

Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы

Рисунок Д.1 – Результат перевірки на антиплагіат в системі Anti-Plagiarism

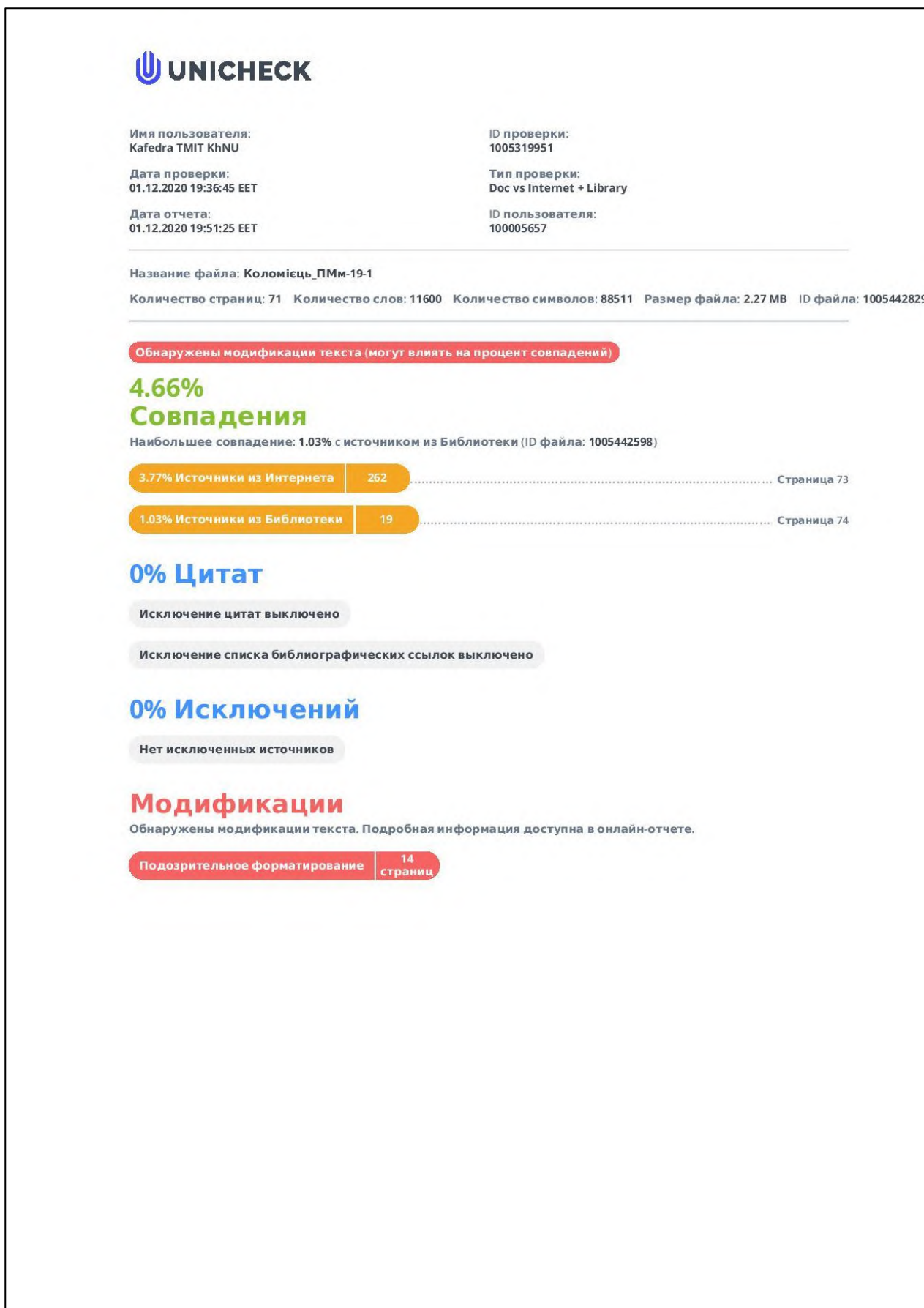


Рисунок Д.2 – Результат перевірки на антиплагіат в системі Unicheck

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА ДИПЛОМНУ РОБОТУ

Дипломник Коломієць Ольга Вікторівна

Тема Telegram-бот для класифікації зображень твердих побутових відходів

Освітній рівень: магістр

Галузь знань: 11 Математика та статистика

Спеціальність: 113 Прикладна математика

**Обсяг дипломної роботи**

Кількість сторінок записки без додатків 80

Кількість сторінок додатків 16

**Характеристика дипломної роботи магістра:**

1. Короткий зміст ДР та прийнятих рішень Метою роботи є розроблення інформаційної системи у вигляді Telegram-боту для класифікації зображень твердих побутових відходів. У роботі спроектовано та розроблено класифікатор зображень твердих побутових відходів з використанням згорткових нейронних мереж. Вперше використано інтерфейс програмного забезпечення Telegram для реалізації системи класифікації таких зображень. Telegram-бот може бути використаний у місцях загального користування, де спеціалізовані системи збору сміття можуть захоплювати відходи всередину, використовувати камеру для виготовлення фотографії та перемістити його у відповідне класифіковане поле.

2. Висновок про відповідність ДР завданню на ДР \_\_\_\_\_

Дипломна робота магістра повністю відповідає завданню

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки, техніки і передових методів роботи: Робота складається з вступу, 3 розділів, висновків, додатків та переліку посилань. У вступі висвітлена актуальність поставленої задачі та мета роботи. У першому розділі проведено аналіз сучасного стану розвитку методів та систем управління відходами; виявлено, що стан поводження з побутовими відходами в Україні є незадовільним. У другому розділі описано процес проектування інформаційної системи для класифікації твердих побутових відходів, а саме: спроектовано класифікатор зображень на основі згорткової нейронної мережі, розглянуто та описано сценарії використання користувацького інтерфейсу, розроблено

*основні модулі клієнт-серверної системи. У третьому розділі наведено реалізацію моделі згорткової нейронної мережі типу Faster R-CNN та описано процес розроблення Телеграм-боту. Перелік посилань складає 33 джерела інформації. Робота виконана з використанням сучасних підходів до моделювання та комп'ютерних технологій.*

4. Позитивні сторони роботи *Реалізовано веб-застосунок із зрозумілим користувацьким інтерфейсом Телеграм-боту для актуальної задачі класифікації об'єктів твердих побутових відходів на зображеннях.*

5. Негативні сторони роботи *У роботі використано лише згорткову нейронну мережу у якості класифікатора зображень. Автором не проведено порівняння результатів нейронної мережі з іншими методами класифікації. Використано малий набір зображень для навчання мережі. У зв'язку з цим можливе перенавчання мережі, яке не розглянуто у роботі.*

6. Оцінка графічного оформлення та пояснювальної записки роботи *Робота виконана відповідно до методичних рекомендацій кваліфікаційної роботи магістра і відповідає нормоконтролю, що прийнятий університетом.*

7. Відгук про роботу в цілому *Робота магістра присвячена розробленню інформаційної системи класифікації твердих побутових відходів з користувацьким інтерфейсом програмного забезпечення Телеграм. З огляду на опис у дипломній роботі, розроблений Телеграм-бот може бути використаний для персонального користування, як зручна тимчасова заміна спеціалізованим системам збору та сортування сміття. Загалом робота відповідає усім критеріям спеціальності «Прикладна математика».*

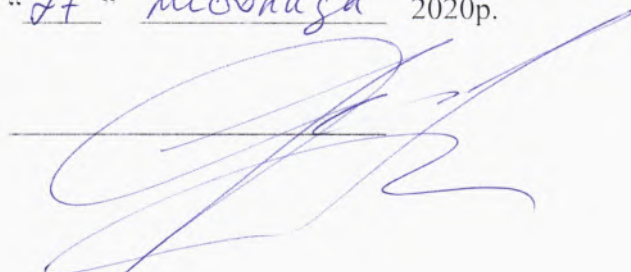
8. Інші зауваження *Помилки, що впливають на кінцевий результат роботи, не знайдено.*

9. Оцінка дипломної роботи *Дипломна робота магістра Коломісць О.В. заслуговує на оцінку «добре».*

РЕЦЕНЗЕНТ *Міхалевський Віталій Цезарійович, кандидат фізико-математичних наук, доцент кафедри КНІТ Хмельницького національного університету*

Прізвище, ім'я, по батькові, посада, місце роботи

*27 листопада 2020р.*



Підпис

Завідувачу кафедри ТМІТ  
д-р.техн.наук Підченку С.К.

*Коломієць О.В*

ІІБ здобувача вищої освіти

ФПКТС, 2 курсу, групи ПМм-19-1

### ЗАЯВА

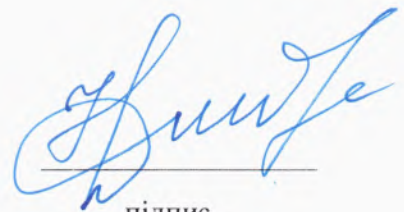
З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіатоповіщений (а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів(Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

2 грудня 2020

дата



підпис

## РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ

### КАФЕДРИ ТЕЛЕКОМУНІКАЦІЙ, МЕДІЙНИХ ТА ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Телеграм-бот для класифікації зображень твердих побутових відходів

Автор: Коломієць Ольга Вікторівна

Спеціальність: 113 – прикладна математика

Освітня програма: освітньо-професійна

Науковий керівник: Грипинська Надія Василівна, к.ф.-м.н., доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	+
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи.	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

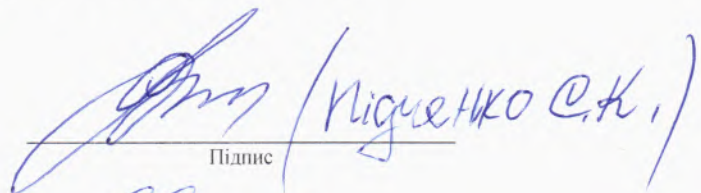
Підтвердження:

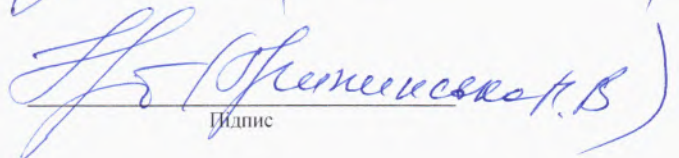
Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) усі запозичення розміщені в розділах, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) частина запозичень відображає аналіз предметної галузі та мають належним чином оформленні посилання;
- 3) інша частина запозичень є сталими виразами кваліфікаційної роботи;
- 4) відсоток співпадінь з кожним із документів не перевищує 1,03%.

3.12.20

Дата

  
Підпис

  
Підпис