

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ПРИРОДОКОРИСТУВАННЯ**

**ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ  
ТЕХНОЛОГІЙ  
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

# **КВАЛІФІКАЦІЙНА РОБОТА**

першого (бакалаврського) рівня вищої освіти

на тему: **«Інтелектуальна інформаційна система управління  
відходами з використанням глибокого навчання»**

Виконав: студент 4 курсу групи Кн-41сп

Спеціальності 122 «Комп'ютерні науки»

(шифр і назва)

Базюк Андрій Романович

(Прізвище та ініціали)

Керівник: д.т.н., професор Тригуба А.М.

(Прізвище та ініціали)

Рецензент: к.т.н., доцент Бабич М.І.

(Прізвище та ініціали)

**ДУБЛЯНИ-2024**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ПРИРОДОКОРИСТУВАННЯ  
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ  
ТЕХНОЛОГІЙ  
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Перший (бакалаврський) рівень вищої освіти  
Спеціальність 122 «Комп'ютерні науки»

«ЗАТВЕРДЖУЮ»

Завідувач кафедри \_\_\_\_\_

д.т.н., проф. А. М. Тригуба

«\_\_\_\_» \_\_\_\_\_ 2024 р.

## ЗАВДАННЯ

на кваліфікаційну роботу студенту

Базюку Андрію Романовичу

1. Тема роботи: «Інтелектуальна інформаційна система управління відходами з використанням глибокого навчання»

Керівник роботи Тригуба Анатолій Миколайович, професор  
затверджені наказом по університету від 27.11.2023 року № 641/к-с.

2. Строк подання студентом роботи 10.06.2024 р.

3. Вихідні дані до роботи: вимоги до системи управління відходами з використанням глибокого навчання; статистичні дані отримання відходів.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити) \_\_\_\_\_

Вступ.

1. Аналіз стану питання в практиці та теорії.

2. Вибір інструментарію та підготовка даних для інтелектуальної інформаційної системи управління відходами.

3. Структурна схема та алгоритми модулів інтелектуальної системи управління відходами з використанням глибокого навчання.

4. Створення програмних модулів для інтелектуальної інформаційної системи управління відходами з використанням глибокого навчання.

5. Охорона праці.

Висновки та пропозиції.

Список використаної літератури.

5. Перелік ілюстраційного матеріалу (з точним зазначенням обов'язкових креслень): аналіз стану питання в практиці та теорії; вибір інструментарію та підготовка даних для інтелектуальної інформаційної системи управління відходами; структурна схема та алгоритми модулів інтелектуальної системи управління відходами з використанням глибокого навчання; створення програмних модулів для інтелектуальної інформаційної системи управління відходами з використанням глибокого навчання.

6. Консультанти з розділів:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1, 2, 3, 4	<i>Тригуба А.М., зав. кафедри ІТ</i>		
5	<i>Тимочко В.О., доцент кафедри фізики, інженерної механіки та безпеки виробництва</i>		

7. Дата видачі завдання

27 листопада 2023 р.

#### Календарний план

№ з/п	Назва етапів кваліфікаційної роботи	Терміни виконання етапів роботи	Примітка
1	<i>Написання першого розділу</i>	<i>27.11-31.12.23</i>	
2	<i>Виконання другого розділу та аркушів ілюстраційного матеріалу до нього</i>	<i>01.02-05.03.24</i>	
3.	<i>Виконання третього розділу та аркушів ілюстраційного матеріалу до нього</i>	<i>06.03-14.04.24</i>	
4.	<i>Виконання четвертого розділу та аркушів ілюстраційного матеріалу до нього</i>	<i>15.04-10.05.24</i>	
5.	<i>Написання розділу «Охорона праці»</i>	<i>11.05-23.05.24</i>	
6.	<i>Завершення оформлення розрахунково-пояснювальної записки та аркушів ілюстраційного матеріалу</i>	<i>24-31.05.24</i>	
7.	<i>Завершення роботи в цілому</i>	<i>01-10.06.24</i>	

Студент \_\_\_\_\_ Базюк А.Р.  
(підпис)

Керівник роботи \_\_\_\_\_ Тригуба А.М.  
(підпис)

УДК 004.89:004.032.26:628.4

Інтелектуальна інформаційна система управління відходами з використанням глибокого навчання.

Базюк А.Р. Кафедра ІТ – Дубляни, Львівський НУП, 2024.

Кваліфікаційна робота: 70 с. текст. част., 22 рис., 3 табл., 14 арк. ілюстраційного матеріалу, 49 джерела.

Виконано аналіз стану та особливості управління відходами. Проаналізовано стан використання глибокого навчання під час управління відходами. Виконано аналіз існуючих інтелектуальних інформаційних систем управління відходами. Здійснена ідентифікація ідеї проекту.

Виконано вибір технологій та засобів для розробки інтелектуальної інформаційної системи управління відходами. Підготовлено зображення для навчання штучної нейронної мережі. Наведена структурна схема системи управління відходами з використанням глибокого навчання. Розроблено алгоритм роботи модуля із камерою та сервоприводом. Подано блок-схему та принцип роботи розумного сміттєвого контейнера. Розроблено алгоритм роботи ультразвукового датчика та алгоритм роботи датчика вимірювання навантаження. Запропоновано згорткову нейронну мережу CNN для класифікації відходів.

Розроблено код для програмного модуля управління відходами. Створено веб додаток для моніторингу наповнення контейнерів для відходів. Створено моделі нейронної мережі CNN.

Розроблено заходи безпеки під час під час розробки інтелектуальної інформаційної системи управління відходами з використанням глибокого навчання.

## ЗМІСТ

ВСТУП .....	7
РОЗДІЛ 1. АНАЛІЗ СТАНУ ПИТАННЯ В ПРАКТИЦІ ТА ТЕОРІЇ.....	9
1.1. Стан та особливості управління відходами .....	9
1.2. Аналіз стану використання глибокого навчання під час управління відходами.....	13
1.3. Аналіз існуючих інтелектуальних інформаційних систем управління відходами.....	16
1.4. Ідентифікація ідеї проекту.....	19
РОЗДІЛ 2. ВИБІР ІНСТРУМЕНТАРІЮ ТА ПІДГОТОВКА ДАНИХ ДЛЯ ІНТЕЛЕКТУАЛЬНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ВІДХОДАМИ .....	21
2.1. Мета та завдання роботи .....	21
2.2. Вибір технологій та засобів для розробки інтелектуальної інформаційної системи управління відходами .....	22
2.3. Підготовка зображень для навчання штучної нейронної мережі .....	23
РОЗДІЛ 3. СТРУКТУРНА СХЕМА ТА АЛГОРИТМИ МОДУЛІВ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ УПРАВЛІННЯ ВІДХОДАМИ З ВИКОРИСТАННЯМ ГЛИБОКОГО НАВЧАННЯ .....	28
3.1. Структурна схема системи управління відходами з використанням глибокого навчання.....	28
3.2. Алгоритм роботи модуля із камерою та сервоприводом.....	30
3.3. Блок-схема та принцип роботи розумного сміттевого контейнера .....	31
3.4. Алгоритм роботи ультразвукового датчика .....	33
3.5. Алгоритм роботи датчика вимірювання навантаження.....	35
3.6. Згорточна нейронна мережа CNN для класифікації відходів.....	36
РОЗДІЛ 4. СТВОРЕННЯ ПРОГРАМНИХ МОДУЛІВ ДЛЯ ІНТЕЛЕКТУАЛЬНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ВІДХОДАМИ З ВИКОРИСТАННЯМ ГЛИБОКОГО НАВЧАННЯ .....	40

4.1. Створення програмного модуля для управління відходами.....	40
4.2. Створення веб додатку для моніторингу наповнення контейнерів для відходів .....	44
4.3. Створення моделі нейронної мережі CNN .....	46
РОЗДІЛ 5. ОХОРОНА ПРАЦІ .....	49
5.1. Аналіз стану робочого місця та умов праці.....	49
5.2. небезпечні та шкідливі чинники .....	50
5.3. Інструкція із охорони праці під час розробки інтелектуальної інформаційної системи управління відходами з використанням глибокого навчання .....	52
ВИСНОВКИ І ПРОПОЗИЦІЇ .....	55
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	58
ДОДАТКИ.....	64

## ВСТУП

Сучасні тенденції урбанізації та зростання населення спричиняють значне збільшення обсягів твердих побутових відходів. Це створює серйозні проблеми для екологічної стабільності та ефективного управління відходами. Традиційні методи управління відходами часто є недостатньо ефективними та потребують значних людських і фінансових ресурсів [45]. В умовах сучасного технологічного розвитку потреба у впровадженні інноваційних підходів, що дозволяє підвищити ефективність і автоматизацію процесів управління відходами.

Одним із таких підходів є застосування інтелектуальних інформаційних систем, які використовують методи глибокого навчання для аналізу та оптимізації процесів збору, сортування та переробки відходів. Глибоке навчання, як підгалузь машинного навчання, дозволяє створювати моделі, які можуть автоматично навчатися і вдосконалюватися з часом на історичних даних, забезпечуючи високу точність і адаптивність у складних і динамічних умовах [39].

Інтелектуальна інформаційна система управління відходами з використанням глибокого навчання забезпечує комплексний підхід до вирішення проблем із відходами. Вона включає автоматизований моніторинг та прогнозування обсягів відходів, оптимізацію маршрутів збору, покращене сортування та підвищення ефективності переробки. Така система забезпечує зниження витрат, мінімізацію впливу на навколишнє середовище та підвищення якості життя населення.

У даній кваліфікаційній роботі розглядається розробка та впровадження інтелектуальної інформаційної системи управління відходами, що базується на глибокому навчання. Робота включає аналіз існуючих методів і технологій, розробку концептуальної моделі системи, а також оцінку ефективності впровадження запропонованих рішень. Особливо важливо використовувати

алгоритми глибокого навчання для вирішення завдань класифікації прогнозу та в рамках системи управління відходами.

Основною метою цієї роботи є демонстрація можливостей та переваг використання інтелектуальних систем на основі глибокого навчання для підвищення ефективності управління відходами та створення умов для сталого розвитку міських середовищ.



## РОЗДІЛ 1. АНАЛІЗ СТАНУ ПИТАННЯ В ПРАКТИЦІ ТА ТЕОРІЇ

### 1.1. Стан та особливості управління відходами

Очікується, що до 2025 року річний обсяг твердих відходів досягне 2,2 мільярдів тон, що коштуватиме 375,5 мільярдів доларів США на управління відходами [27]. Неналежне поводження з відходами матиме величезний негативний вплив на економіку, здоров'я населення та навколишнє середовище. Агентство з охорони навколишнього середовища (EPA) визнало переробку твердих побутових відходів (ТПВ) другою «найбільш екологічно безпечною» стратегією поводження з міськими відходами [37]. Ефективна переробка відходів є як економічною, так і екологічною. Це може допомогти у відновленні сировинних ресурсів, збереженні енергії, зменшенні викидів парникових газів, забруднення води, скороченні нових звалищ тощо [18].

У країнах, що розвиваються, переробка твердих побутових відходів залежить від відокремлення домогосподарств за допомогою збирачів сміття та збирачів, які торгують вторинною сировиною для отримання прибутку. У розвинених країнах громади більше залучені до програми переробки. Кілька методів, таких як механічне сортування та хімічне сортування, доступні в розвинених країнах для автоматичного сортування відходів [23]. Проте існує величезний потенціал для покращення переробки відходів навіть у розвинених країнах. Рівень переробки муніципальних відходів у США та Європейському Союзі становить близько 34% і 50% відповідно, що значно нижче цільового рівня переробки в 75% [48].

Парадигма утилізації відходів, яка характеризується лінійним потоком матеріалів від первинного видобутку через обробку, виробництво, використання та закінчується утилізацією (рис. 1.1).

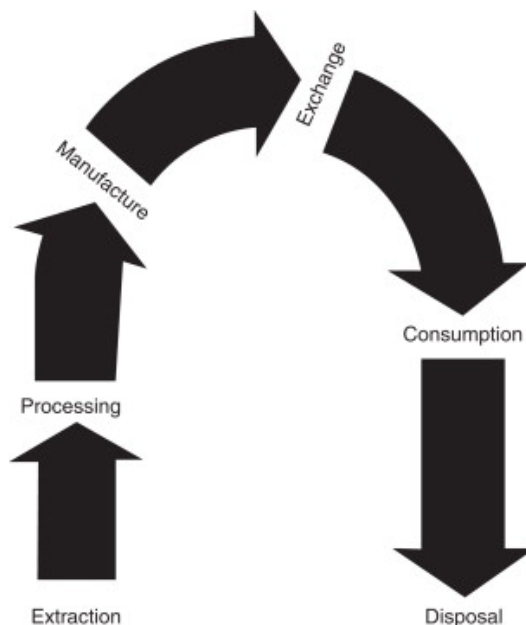


Рисунок 1.1 – Парадигма утилізації відходів

Урбанізація посилює проблеми управління відходами, перш за все через концентрацію людей, які виробляють відходи, і супутню відсутність місця для роботи з ними, що створює значні практичні труднощі в управлінні відходами, які утворюються домогосподарствами та міськими підприємствами.

Урбанізація також тягне за собою просторову концентрацію споживачів, просторово ізольованих від різноманітних ланцюгів виробництва та постачання, через які споживчі товари збираються в місті, що ускладнює повернення відходів для продуктивного використання.

Управління відходами, як це визначено Директивою Європейського Парламенту та Ради 2008/98/ЄС про відходи, включає збирання, перевезення, оброблення (включаючи відновлення та видалення), нагляд за такими операціями і подальший контроль, догляд за об'єктами видалення відходів після їх закриття, а також діяльність брокерів та дилерів.

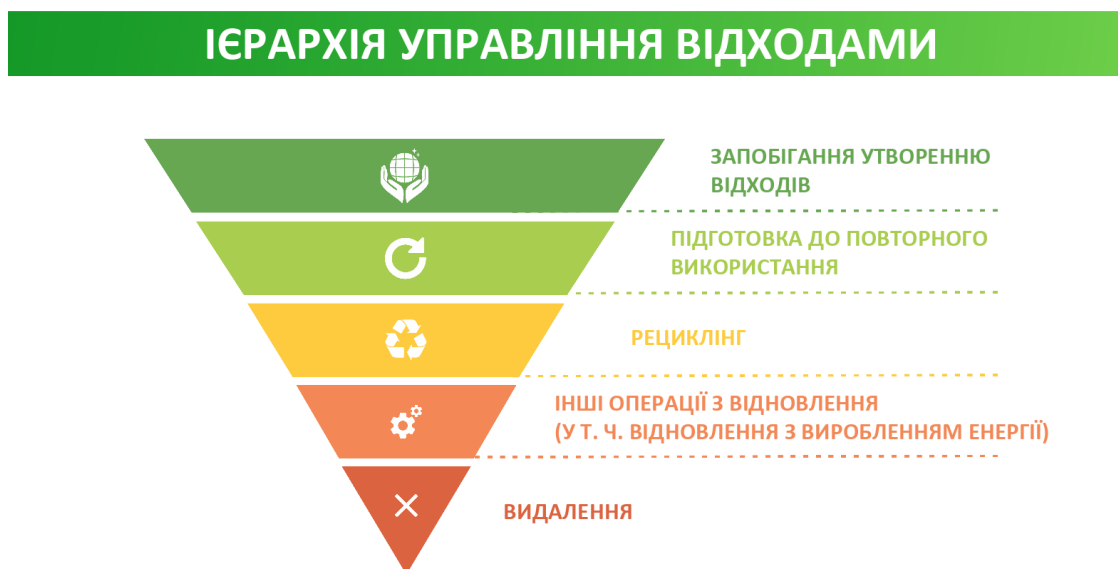


Рисунок 1.2 – Ієрархія управління відходами [21]

До основних принципів реформи управління відходами належать:

1. Відповідність засадам економіки замкненого циклу. Економіка замкненого циклу (циркулярна економіка) є моделлю економічного розвитку, яка базується на відновленні та раціональному споживанні ресурсів. Ця модель є альтернативою традиційній лінійній економіці і спрямована на мінімізацію негативного впливу на довкілля, безвідходне виробництво та досягнення цілей сталого розвитку.

2. Ієрархія управління відходами. Ієрархія управління відходами складається з п'яти рівнів, які визначають пріоритетні заходи в процесі управління відходами.

Запобігання утворенню відходів є найвищий пріоритет, що забезпечує уникнення створення відходів.

Повторне використання практикують, якщо уникнути відходів не вдається, необхідно максимально повторно використовувати матеріали.

Рециклінг передбачає перероблення відходів на нову продукцію, матеріали або речовини. Включає перероблення органічного матеріалу, але не включає відновлення енергії чи використання відходів як паливо.



Рисунок 1.3 – Складові управління відходами

Інші види утилізації передбачають, якщо рециклінг неможливий, використовуються інші методи утилізації, такі як відновлення енергії або перероблення на паливо чи матеріали для зворотного заповнення.

Видалення відходів використовують коли інші методи неможливі, відходи захоронюються на спеціально обладнаних місцях або знищуються на установках, які не відповідають екологічним нормативам.

Ця структура пріоритетів допомагає мінімізувати негативний вплив відходів на довкілля, сприяє сталому розвитку та підтримує економіку замкненого циклу.

Дорожньою картою для вирішення проблем у сфері управління відходами є затверджена Кабінетом Міністрів України в 2017 році Національна стратегія управління відходами в Україні до 2030 року [18]. Ця стратегія базується на дотриманні ієрархії відходів, де видалення відходів є найменш бажаним варіантом, тоді як запобігання, повторне використання, переробка та виробництво альтернативної енергії з відходів є пріоритетними. Стратегія передбачає модернізацію інфраструктури для зменшення кількості відходів та підвищення її ефективності, оцифрування галузі управління відходами та централізацію інформації на єдиному порталі.

Документ закликає до створення інклюзивної тарифної системи, яка покриває всі витрати, включаючи догляд за сміттєзвалищем протягом 35 років. Також пропонується запровадити нові принципи, такі як «забруднювач платить» та «розширена відповідальність виробника», а також покращити механізми моніторингу та контролю, включаючи встановлення пунктів прийому роздільних відходів. Це має підвищити рівень переробки до 23% до 2023 року та збільшити роздільний збір до 23% [18].

Метою стратегії є переробка 65% всіх відходів і 70% муніципальних відходів до 2030 року. Національна стратегія управління відходами в Україні до 2030 року відповідає вимогам рамкової директиви ЄС щодо ієрархії поводження з відходами та визначає ключові орієнтири поступового переходу до економіки замкненого циклу.

## **1.2. Аналіз стану використання глибокого навчання під час управління відходами**

Недавній прогрес у глибокому навчанні сприяв безпрецедентним покращенням комп'ютерного зору. Згорткова нейронна мережа (CNN) є одним із найбільш визнаних алгоритмів глибокого навчання для його широкого застосування в класифікації зображень, сегментації та виявленні [26]. Тому пропонується виконувати класифікацію відходів.

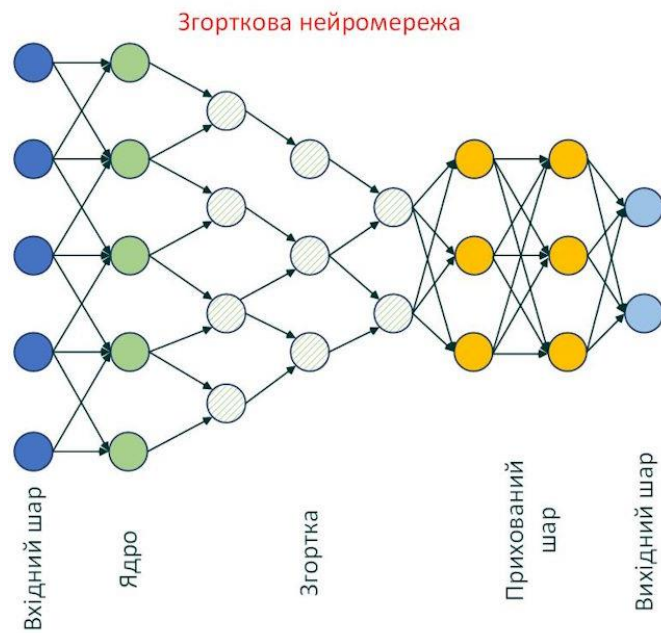


Рисунок 1.4 – Згорткова нейронна мережа (CNN)

Аве та ін. [22] пропонують експериментальний проект із використанням моделі Faster R-CNN для класифікації відходів на три категорії: папір, вторинна переробка та звалище.

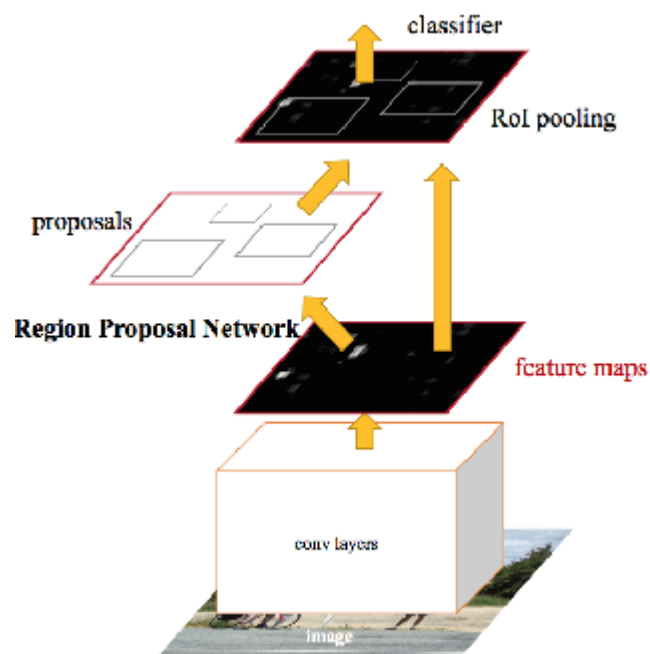


Рисунок 1.7 – Архітектура нейронної мережі на основі RCNN [22]

Цей метод досягає середньої точності 68%. Thung і Yang [38] розгорнули опорну векторну машину (SVM) і згорткову нейронну мережу (CNN) для класифікації відходів на шість категорій.

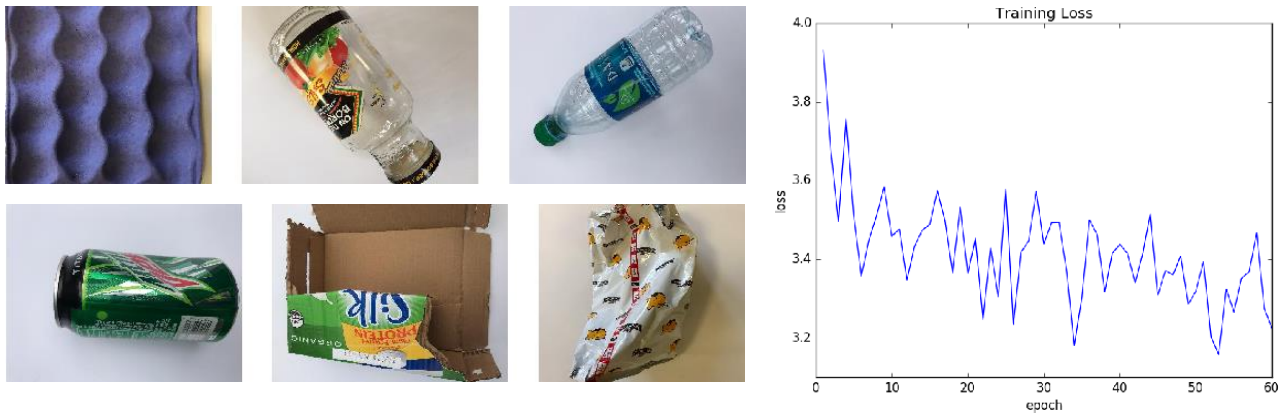


Рисунок 1.8 – Класифікація сміття за статусом повторної переробки та результати навчання згорткової нейронної мережі (CNN): 1 – папір; 2 – скло; 3 – пластик [38]

Він досягає рівня точності 63% для SVM і 23% для CNN. Рад та ін. [35] розробив програму бачення на основі GoogLeNet для локалізації та класифікації міських відходів.

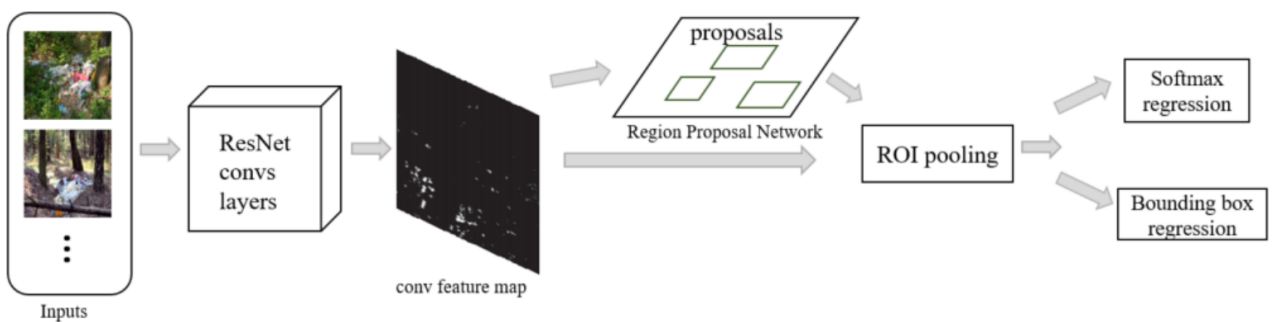


Рисунок 1.9 – Загальна архітектура мережі [35]

Процес впровадження мережі показано на рисунку 1.9. Під час навчання вводять набір даних спочатку через спільний шар згортки ResNet, генеруючи карту ознак. Потім шар RPN отримує вихідні дані і генерує велику кількість пропозицій регіонів. Дослідження стверджує, що рівень точності коливається від 63% до 77% для різних типів відходів.

Донован [25] запропонував використовувати Google TensorFlow і зйомку камери для автоматичного сортування відходів як компосту та вторинної переробки. Проте як концептуального проекту експериментального результату поки що немає.

### 1.3. Аналіз існуючих інтелектуальних інформаційних систем управління відходами

Сьогодні відомо декілька існуючих інтелектуальних інформаційних систем управління відходами. Вони мають різну архітектуру та використовують різних інструментарій.

Міттал та ін. [32] розробив проект, щоб визначити, чи містить зображення сміття чи ні.

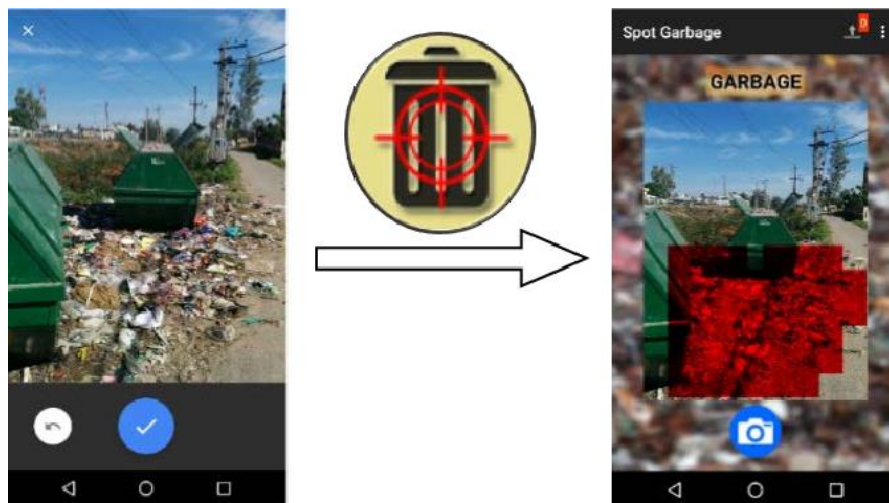


Рисунок 1.10 – Додаток SpotGarbage для смартфонів для виявлення сміття за допомогою глибокого навчання: ліворуч вхідне зображення, що показує сміття, праворуч сегментоване вихідне зображення з програми [32]

Цей проект використовує модель AlexNet перед навчанням і досягає середньої точності 87,69%. Однак цей проект спрямований на сегментування сміття на зображенні без надання функцій класифікації відходів.



Як було розглянуто, методології автоматичної класифікації, доступні в літературі, використовують виключно CNN на основі зображень і призводять до обмеженої точності.

У цій роботі [24] запропоновано багат шаровий гібридний метод (MHS) для класифікації відходів у громадських місцях. Коли відходи надходять у гібридну систему, камера та датчики активуються для спостереження за предметом. Система обробки зображень складається з камери для захоплення зображення, яке аналізується CNN. Сенсорна система, одночасно, функціонує для отримання числової інформації від об'єктів. Остаточні результати (двійковий вихід) отримані за допомогою системи MLP, входами якої є 22 виходи з CNN та числова інформація від датчиків. У цьому відношенні систему MLP можна навчити незалежно від моделі CNN, а параметри ваги та зміщення моделі CNN можуть залишатися незмінними. З іншого боку, оскільки вихідні дані CNN є входами для MLP, ці дві моделі фактично функціонують одночасно для генерації результатів двійкової класифікації.

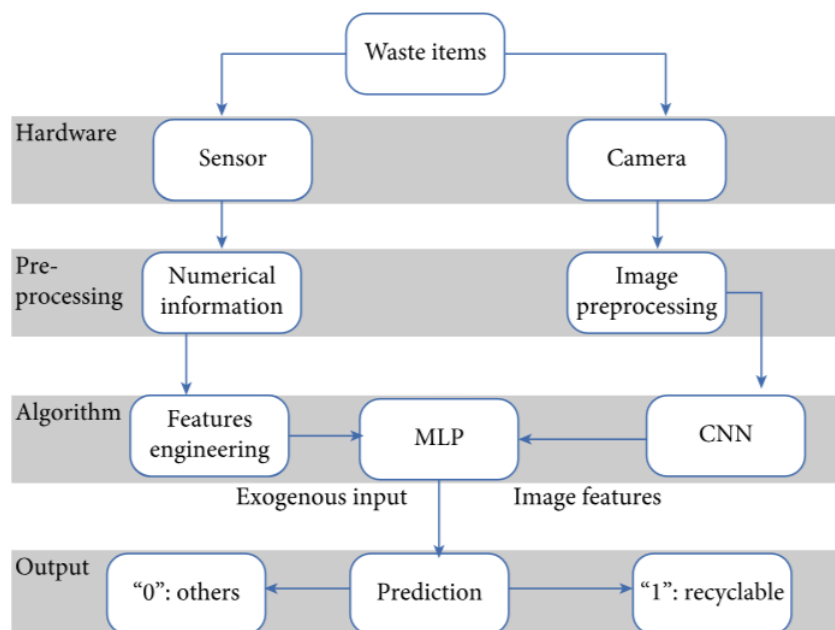


Рисунок 1.11 – Багат шарова гібридна система (MHS) [24]

Відпрацьовані зображення, пов'язані з іншою цифровою інформацією, вимірною датчиками, надходять у систему. Система може автоматично сортувати сміття як придатне для переробки чи інше. Запропонований MHS

досягає середньої точності вище 90%, що значно перевершує метод на основі еталонного зображення.

У роботі [23] систему називають «інтелектуальною», оскільки її було створено з використанням інтелектуального підсилювача GA–fuzzy для виявлення відходів і класифікації зображень сміття.

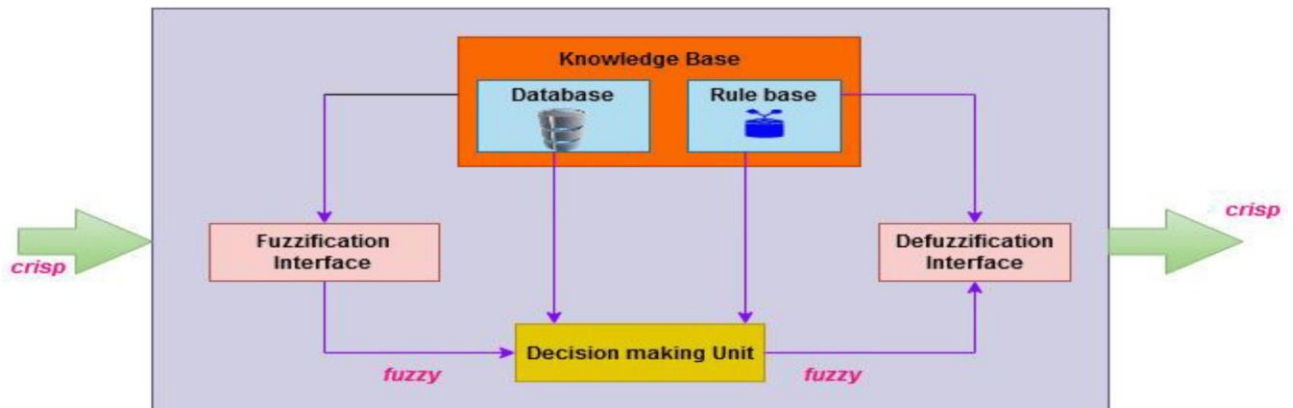


Рисунок 1.12 – Функціональні блоки FIS

Ця робота відіграє важливу роль у розбудові розумних та екологічно чистих міст. Відповідне застосування цієї моделі допомагає ідентифікувати та переробляти відходи, відокремлювати канцерогенні відходи та знищувати їх, що значно допомагає у створенні гігієнічного зеленого середовища та, таким чином, удосконалює циркулярну економіку нації.

В останні кілька років FIS застосовувався в різних предметах, які виражалися в термінах правил «якщо-тоді» на основі досвіду. Розробляється набір нечітких правил з відповідними функціями приналежності для визначення необхідних пар вхід-вихід. FIS також відомий як нечіткі системи на основі правил, нечіткі моделі, нечіткі асоціативні пам'яті (FAM) або нечіткі контролери. FIS складається з п'яти функціональних блоків, як показано на рис. 1.12.

Архітектура запропонованої системи показана на рисунку 1.13. Об'єкт падає на середню частину всередині контейнера, і вага удару зчитується тензодатчиком при контакті з об'єктом.

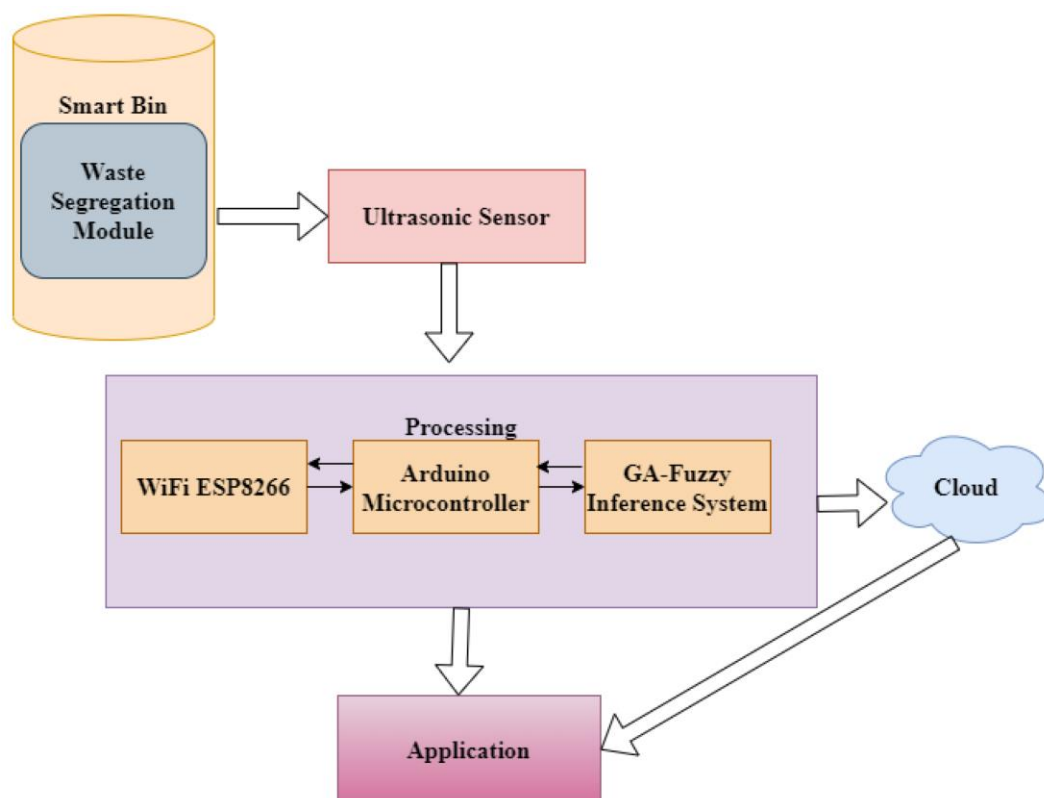


Рисунок 1.13 – Архітектура запропонованої системи управління відходами

Наступні ваги також зчитуються до тих пір, поки вага об'єкта не стане постійною. На попередньому етапі було виміряно 30 безперервних показань ваги. Показання розділені на два набори: перший набір – це показання удару та відскоку об'єкта. Це змінні показники, оскільки об'єкт відскакує після удару. З цього набору отримують найбільше значення.

#### 1.4. Ідентифікація ідеї проекту

Проблема управління відходами є однією з найбільш гострих і складних задач сучасного суспільства. Зі зростанням населення та урбанізацією обсяги твердих побутових відходів стрімко зростають, що призводить до серйозних екологічних, економічних та соціальних наслідків. Традиційні методи управління відходами часто є неефективними через недостатню автоматизацію, відсутність точних прогнозів та складнощі у сортуванні відходів.

Впровадження інтелектуальних інформаційних систем з використанням глибокого навчання може суттєво покращити ці процеси. Глибоке навчання дозволяє створювати високоточні моделі для аналізу великих обсягів даних, що дає змогу автоматизувати і оптимізувати різні аспекти управління відходами. Це, в свою чергу, сприяє зниженню витрат, зменшенню негативного впливу на довкілля та підвищенню ефективності системи управління відходами в цілому.

Ідентифікація ідеї проекту «Інтелектуальна інформаційна система управління відходами з використанням глибокого навчання» базується на актуальності проблеми управління відходами, перевагах глибокого навчання для вирішення складних задач та очікуваних результатах, які мають значний позитивний вплив на екологічну ситуацію та ефективність управління відходами. Реалізація даного проекту сприятиме впровадженню інноваційних рішень у сфері управління відходами та створенню умов для сталого розвитку.

## РОЗДІЛ 2.

### ВИБІР ІНСТРУМЕНТАРІЮ ТА ПІДГОТОВКА ДАНИХ ДЛЯ ІНТЕЛЕКТУАЛЬНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ВІДХОДАМИ

#### 2.1. Мета та завдання роботи

Метою проекту є розробка інтелектуальної інформаційної системи управління відходами з використанням методів глибокого навчання для підвищення ефективності процесів збору, сортування та переробки відходів.

Для досягнення поставленої мети необхідно вирішити такі задачі:

- ✓ провести аналіз існуючих інтелектуальних інформаційних систем управління відходами та визначити їх недоліки;
- ✓ дослідити можливості застосування глибокого навчання для автоматизації та оптимізації процесів управління відходами;
- ✓ розробити концептуальну модель інтелектуальної інформаційної системи управління відходами;
- ✓ створити прототип системи, що включає модулі для моніторингу, прогнозування, оптимізації маршрутів збору та сортування відходів;
- ✓ надати рекомендації щодо впровадження розробленої системи у практику управління відходами.

Очікується, що розроблена інтелектуальна інформаційна система управління відходами забезпечить підвищення точності прогнозів обсягів відходів. Вона передбачатиме автоматизацію процесів сортування відходів за допомогою методів комп'ютерного зору. На її основі можна здійснювати оптимізацію маршрутів збору відходів, що знизить витрати на транспорт та час збору. Вона сприятиме зменшенню негативного впливу на навколишнє середовище завдяки ефективнішій переробці відходів.

## **2.2. Вибір технологій та засобів для розробки інтелектуальної інформаційної системи управління відходами**

Запропонована інтелектуальна інформаційна система управління відходами базується на інтеграції глибокого навчання та Інтернету речей (IoT). Основні компоненти системи включають згорткову нейронну мережу (CNN) для сортування відходів, розумні сміттєві баки, оснащені мікроконтролерами та датчиками, а також засоби зв'язку через IoT та Bluetooth для моніторингу та управління в реальному часі.

Для автоматизованого сортування перетравлюваних і неперетравлюваних відходів використовується згорткова нейронна мережа (CNN). CNN є популярною парадигмою глибокого навчання, яка демонструє високу ефективність у задачах класифікації зображень. Вибір CNN обґрунтований її здатністю розпізнавати та класифікувати об'єкти на основі візуальних даних з високою точністю.

Розумні сміттєві баки є ключовим компонентом системи, що забезпечує автоматизацію збору та сортування відходів. Вони оснащені мікроконтролерами та кількома датчиками, які дозволяють здійснювати моніторинг і управління в реальному часі.

IoT та Bluetooth забезпечують безперервний моніторинг та управління системою в реальному часі. Підключення IoT дозволяє відстежувати стан сміттєвих баків та надсилати дані на централізований сервер з будь-якої точки світу, тоді як Bluetooth забезпечує локальну передачу даних на невеликих відстанях через портативний комп'ютер.

Перевагами використання IoT та Bluetooth є можливість отримувати та обробляти дані в реальному часі. Контроль та управління системою з будь-якого місця, тобто віддалено. Є можливість швидкої передачі даних на невеликих відстанях для оперативного реагування.

Для реалізації проекту було обрано наступні технології та інструменти. Запропоновано використовувати фреймворки TensorFlow та PyTorch для

розробки та тренування моделей глибокого навчання. Використано бібліотеку OpenCV для комп'ютерного зору, яка використовується для обробки зображень відходів. Платформи Arduino та Raspberry Pi використовуються для створення прототипів мікроконтролерів і датчиків для розумних смітєвих баків. Протоколи MQTT та HTTP для передачі даних через IoT. Також пропонується використати інструменти для розробки веб додатку для локального моніторингу через Bluetooth.

Для перевірки ефективності розробленої моделі буде проведено наступні тести. Оцінка точності класифікації відходів за допомогою CNN. Передбачена перевірка точності та надійності даних, зібраних датчиками. Також передбачено використання шкали зручності використання системи (SUS) кінцевими користувачами.

Вибір технологій та засобів для розробки інтелектуальної інформаційної системи управління відходами забезпечує створення ефективної, автоматизованої та зручної у використанні системи. Інтеграція глибокого навчання та IoT дозволяє значно підвищити точність і ефективність управління відходами, сприяючи зниженню витрат та покращенню екологічної ситуації.

### **2.3. Підготовка зображень для навчання штучної нейронної мережі**

Нами використано загально доступний набір даних із зображеннями відходів [14]. Він включає зображення скла, паперу, картону, пластику, металу та сміття. Нами написано код, який виконує кілька кроків для аналізу, підготовки та візуалізації даних для моделі нейронної мережі CNN для класифікації відходів.

Насамперед виконано розпакування архіву з даними (рис. 2.1). Спочатку архів з даними розпаковується за допомогою модуля zipfile.

```

import zipfile
import os

# Шлях до архіву
zip_path = r'G:\Dipl_2024\Bak_2024\Kn\Базюк\dataset-resized.zip'
extract_path = r'G:\Dipl_2024\Bak_2024\Kn\Базюк\dataset-resized'

# Розпакування архіву
with zipfile.ZipFile(zip_path, 'r') as zip_ref:
    zip_ref.extractall(extract_path)

```

Рисунок 2.1 – Фрагмент коду для розпакування архіву з даними

У подальшому використовується ImageDataGenerator, який автоматично завантажує дані з папок, містять зображення різних класів (рис. 2.2).

```

import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import matplotlib.pyplot as plt

# Шлях до папки з даними
data_dir = os.path.join(extract_path, 'dataset-resized')

# Визначення параметрів генератора зображень
datagen = ImageDataGenerator(
    rescale=1./255,
    validation_split=0.2
)

# Завантаження даних
train_generator = datagen.flow_from_directory(
    data_dir,
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical',
    subset='training'
)

validation_generator = datagen.flow_from_directory(
    data_dir,
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical',
    subset='validation'
)

# Виведення інформації про класи
class_labels = list(train_generator.class_indices.keys())
print("Класи:", class_labels)

```

Рисунок 2.2 – Фрагмент коду для виведення інформації про класи та візуалізації зразків зображень

Створюється об'єкт ImageDataGenerator, який визначає параметри для генерування зображень. У цьому випадку встановлюється параметр rescale, який масштабує пікселі до діапазону [0, 1], і параметр validation\_split, який визначає частку даних, що будуть використовуватися для валідації.

Дані завантажуються за допомогою методу flow\_from\_directory. Зображення зчитуються з папки data\_dir та розміщуються в пакетах (батчах)



розміром 32x32. Крім того, використовується параметр `subset`, щоб визначити, чи є цей набір даних тренувальним або валідаційним. Виводиться список класів, які ідентифікуються у тренувальному генераторі, за допомогою методу `class_indices` (рис. 2.3).

```
Found 2024 images belonging to 6 classes.
Found 503 images belonging to 6 classes.
Класи: ['cardboard', 'glass', 'metal', 'paper', 'plastic', 'trash']
```

Рисунок 2.3 – Результати виведення списку класів, які ідентифікуються у тренувальному генераторі

Наступний фрагмент коду (рис. 2.4) відповідає за візуалізацію зразків зображень з тренувального набору даних.

```
# Функція для візуалізації зразків зображень
def plot_sample_images(generator, class_labels):
    plt.figure(figsize=(12, 12))
    for i in range(9):
        # Отримання зображення та мітки
        x, y = next(generator)
        ax = plt.subplot(3, 3, i + 1)
        plt.imshow(x[0])
        plt.title(class_labels[y[0].argmax()])
        plt.axis("off")
    plt.show()

# Візуалізація зразків тренувальних даних
plot_sample_images(train_generator, class_labels)
```

Рисунок 2.4 – Фрагмент коду для візуалізації зразків зображень з тренувального набору даних

Насамперед створюється функція `plot_sample_images`, яка отримує два аргументи: `generator`, який є генератором зображень, і `class_labels`, який є списком міток класів. Функція призначена для виведення зразків зображень разом з відповідними мітками.

Після цього застосовується функція `plot_sample_images` до тренувального генератора `train_generator` та списку міток класів `class_labels`. Дев'ять зразків зображень виводяться у вигляді сітки 3x3 за допомогою функції `imshow`, яка відображає зображення, та функції `title`, яка відображає відповідну мітку класу (рис. 2.5).

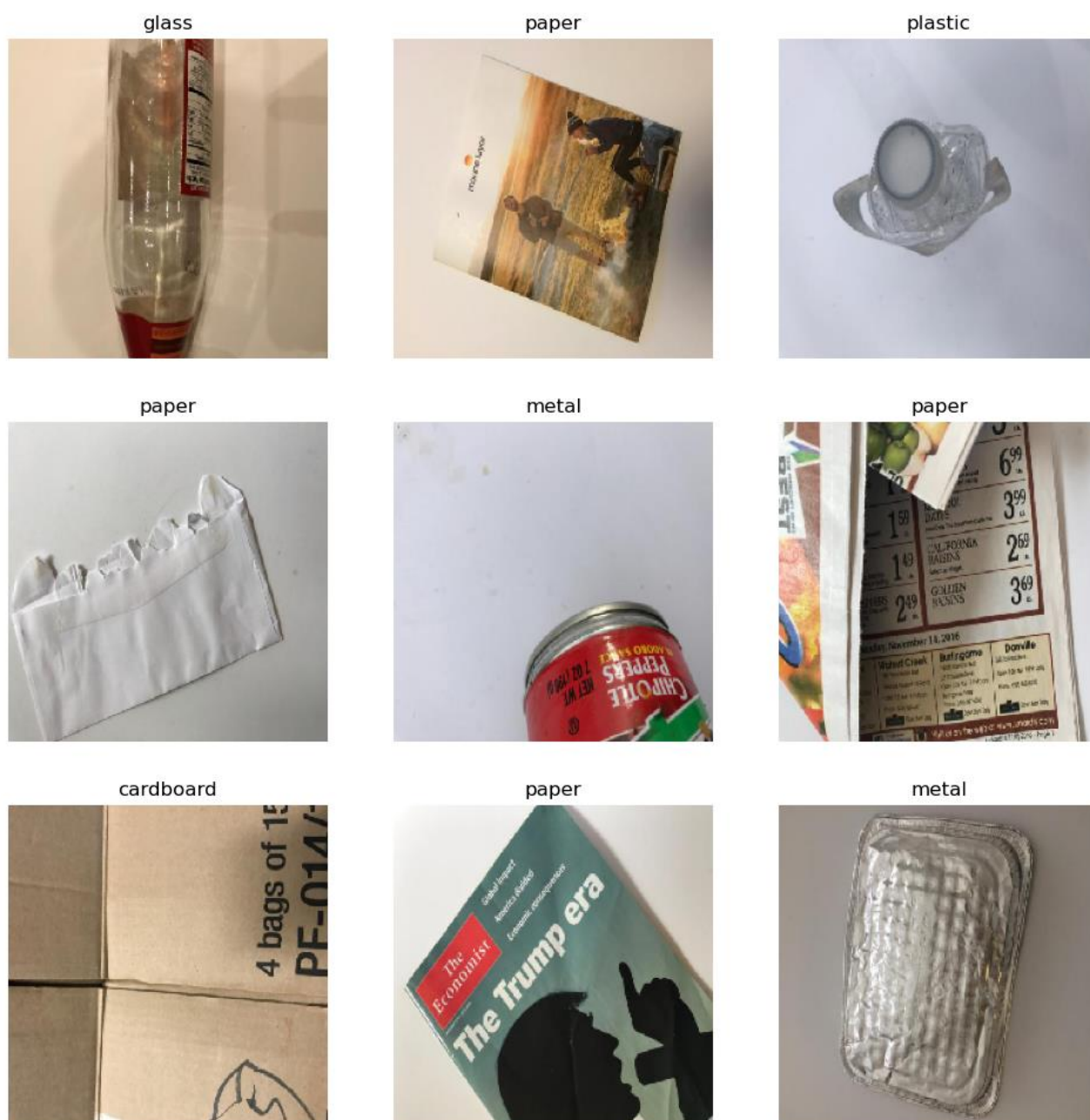


Рисунок 2.5 – Візуалізація окремих зразків зображень

Нами візуалізовано деякі зразки зображень для перегляду та перевірки правильності підготовки даних перед тренуванням моделі. Їх аналіз дав можливість побудувати гістограму відображення кількості зразків у кожному класі даних (рис. 2.6).

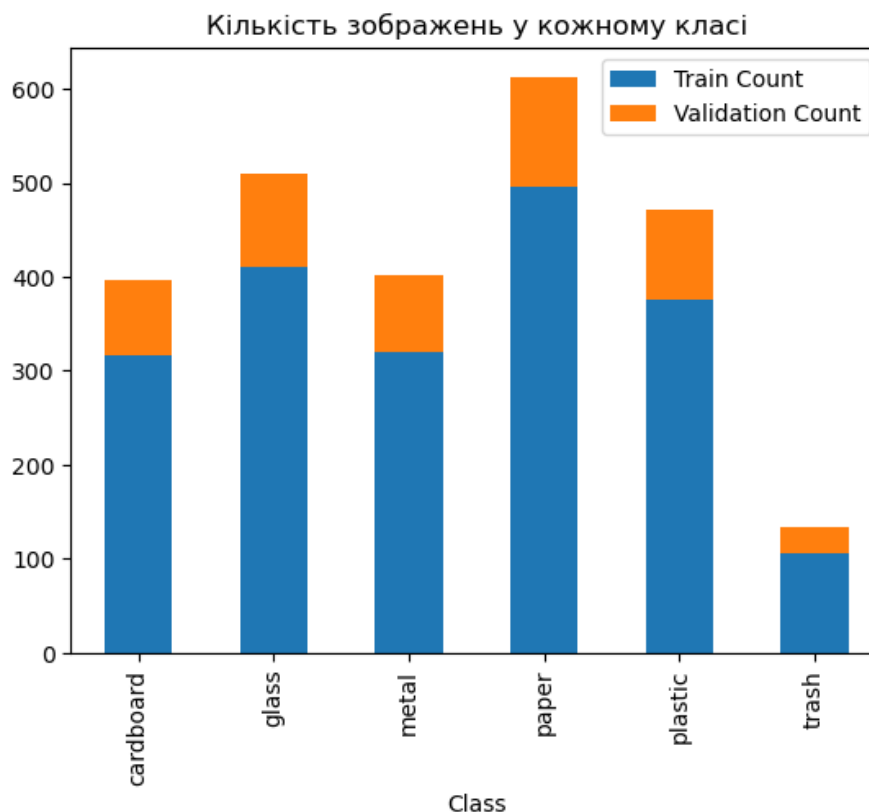


Рисунок 2.6 – Гістограма кількості зразків у кожному класі даних

Отримана гістограма кількості зразків у кожному класі даних свідчить про те, що наявно шість класів – «cardboard» (картон), «glass» (скло), «metal» (метал), «paper» (папір), «plastic» (пластик) і «trash» (сміття). Кожен клас має відповідну кількість зображень у тренувальному наборі даних. Найбільша кількість зображень належить до класу «paper» (495), а найменша – до класу «trash» (106).

Для кожного класу також вказана кількість зображень у валідаційному наборі даних. Ці дані допомагають забезпечити, що модель буде правильно перевірена на незалежних даних після тренування. В цьому випадку також кількість зображень для валідації найбільша для класу «paper» (118), а найменша – для класу «trash» (27).

Цей аналіз дає можливість отримати оцінку балансу даних у різних класах та допомагає у визначенні необхідності застосування додаткових стратегій збалансування класів під час тренування моделі.

### РОЗДІЛ 3.

## СТРУКТУРНА СХЕМА ТА АЛГОРИТМИ МОДУЛІВ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ УПРАВЛІННЯ ВІДХОДАМИ З ВИКОРИСТАННЯМ ГЛИБОКОГО НАВЧАННЯ

### 3.1. Структурна схема системи управління відходами з використанням глибокого навчання

Запропонована структурна схема системи управління відходами з використанням глибокого навчання являє собою поєднання двох частин, а саме класифікації відходів за допомогою згорткової нейронної мережі та розумних сміттєвих контейнерів, що сприяє моніторингу даних у реальному часі за допомогою Інтернету речей.

Дві структурні моделі об'єднані, щоб отримати бажані результати в галузі управління відходами. Класифікація відходів за відповідними категоріями допомагає ідентифікувати відходи, придатні для повторного використання. Визначення відходів, які підлягають переробці, дозволить нам утилізувати їх без погіршення якості.

Пропонується розділити відходи на дві широкі категорії – розкладні (органічні відходи) та нерозкладні. Через відсутність достатньої кількості даних для класифікації відходів використали точно налаштовані моделі для класифікації відходів. Класифікація відходів за допомогою технології глибокого навчання допомагає отримати категорії відходів із зображень.

Архітектура сміттєвих ящиків дозволяє кільком датчикам знімати показання та передавати дані для моніторингу. На рис. 2.1 представлена структурна схема системи.

У запропонованій схемі модуль камери сканує відходи. Після успішного завершення процесу сканування відходів та отримання зображень виконується компонент попередньої обробки зображень, зроблених камерою в режимі реального часу.

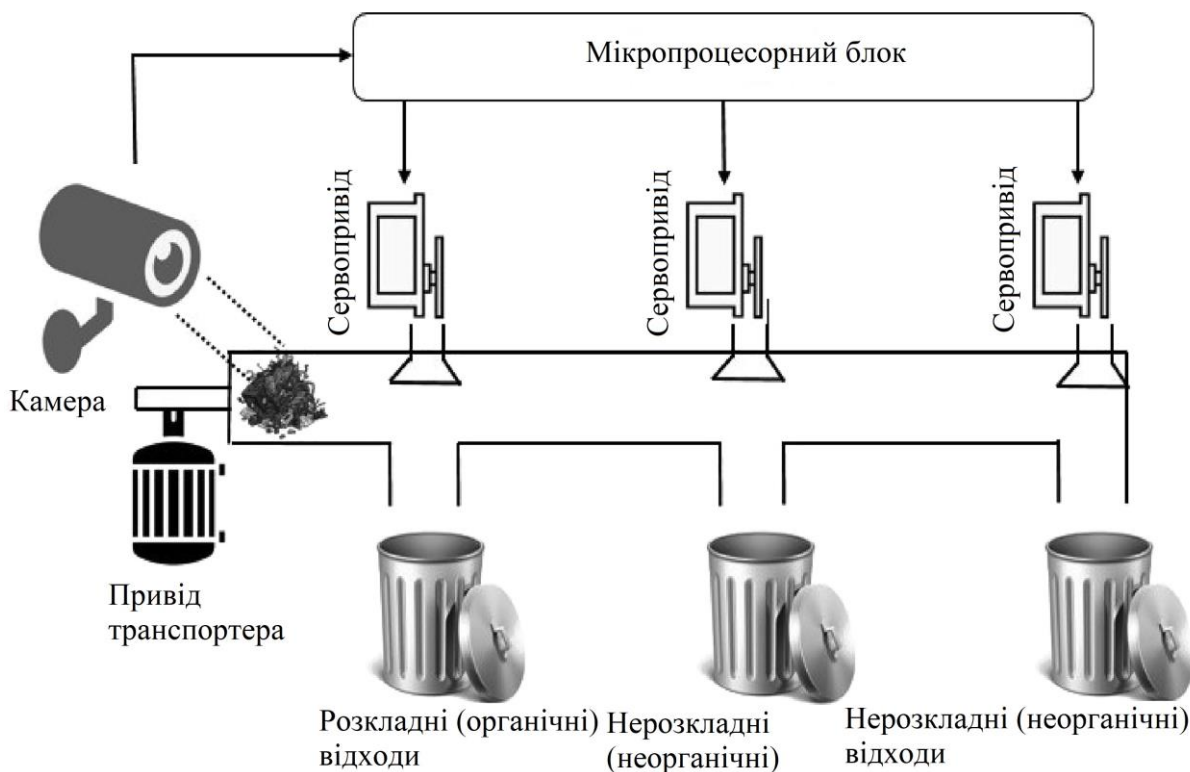


Рисунок 3.1 – Структурна схема запропонованої інтелектуальної системи управління відходами з використанням глибокого навчання

Модель використовує лише зміну розміру зображення для забезпечення меншої складності. Після цього попередньо оброблені зображення обробляються мікропроцесором (Raspberry Pi). Мікропроцесор класифікує зображення за допомогою класифікатора і надсилає команду серводвигуну, щоб той поклав відходи у відповідний сміттєвий бак.

Мікроконтролер смітника надсилатиме дані на портативний комп'ютер для моніторингу в режимі реального часу. Ця система також включає в себе транспортер, який здатний переносити відходи відповідно до інструкцій від переробного блоку. Щоразу, коли блок обробки класифікує відходи, він надсилає сигнал на транспортер, щоб той переносив ці відходи до серводвигуна. Після цього транспортер припиняє рух і чекає наступної команди від переробного блоку.

### 3.2. Алгоритм роботи модуля із камерою та сервоприводом

Одним із вагомих модулів інтелектуальної інформаційної системи управління відходами є модуль камери та сервоприводу. Модуль камери підключений до мікроконтролера запропонованої системи і відповідає за отримання зображень зі станом відходів.

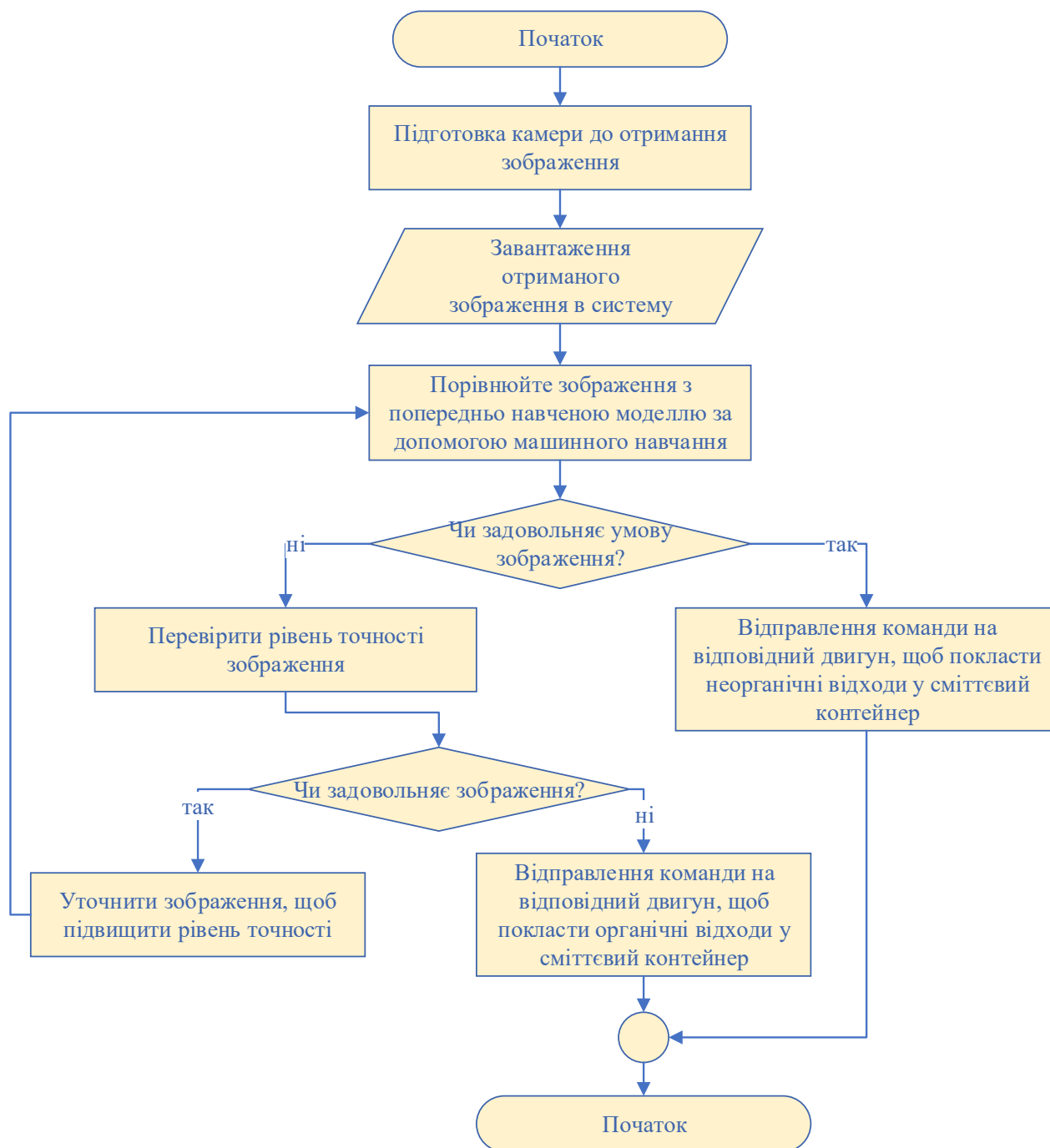


Рисунок 3.2 – Блок-схема алгоритму роботи модуля із камерою та сервоприводом

На рис. 3.2 представлено блок-схему модуля із камерою та сервоприводом.

Перш за все, запропонована система ініціалізується та проходить підготовку до отримання зображень. Модуль камери робить знімок і надсилає його до мікроконтролера. Отримавши зображення, мікроконтролер передає його вже навченій моделі CNN, і модель дає відповідь щодо цього зображення.

Мікроконтролер використовує відповідь CNN, щоб подати сигнал до сервоприводу для того, щоб транспортувати відходи у відповідний сміттєвий бак.

Мікроконтролер приймає рішення на основі ймовірності того, що окремі відходи належать до органічних (тих що розкладаються) або неорганічних (тих що не розкладаються). Після цього серводвигун виконує свою роботу, забираючи відходи і подаючи їх через транспортер у відповідний сміттєвий контейнер.

### **3.3. Блок-схема та принцип роботи розумного сміттєвого контейнера**

Розглянемо принцип проектування та процес розробки запропонованої архітектури розумного сміттєвого ящика. На рис. 3.3 показано структурну схему розумного контейнера для сміття. Для нього використовується мікроконтролер під назвою «ESP8266» node MCU.

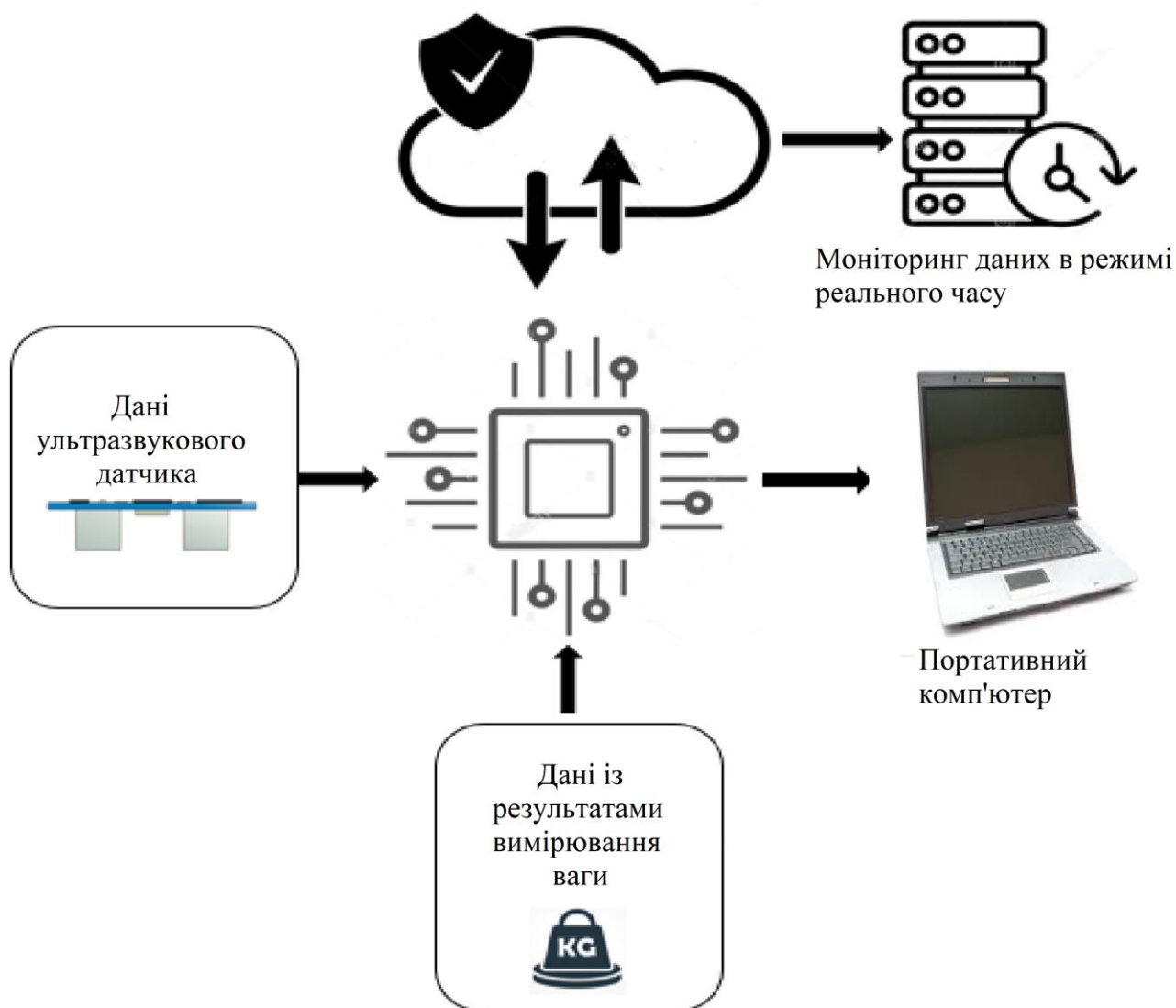


Рисунок 3.3 – Структурна схема розумного контейнера для сміття

Ультразвуковий датчик підключений до мікроконтролера для вимірювання рівня заповнення контейнера для сміття. Ультразвуковий датчик розташований у верхній частині розробленої моделі. Ультразвук передається та приймається для обчислення рівня наповнення сміттьового відра. Розраховане значення надсилається на мікроконтролер. Датчик вимірювання навантаження також розміщений у нижній частині. Цей датчик відповідає за обчислення ваги відходів у кг.

Коли кількість відходів у сміттьовому ящику періодично збільшується, значення навантаження також збільшується. Потім оновлене значення передається на мікроконтролер. Значення ультразвукового перетворювача та



датчика вимірювання навантаження надсилаються на портативний комп'ютер. Користувачі можуть легко спостерігати за вагою сміття та поточним порожнім шаром контейнера за допомогою веб додатку. Якщо підключення до Інтернету доступне, відповідні значення надсилатимуться на хмарний сервер, і користувачі зможуть відстежувати дані в режимі реального часу за допомогою веб додатку.

### **3.4. Алгоритм роботи ультразвукового датчика**

На рис. 3.4 зображено блок-схему роботи ультразвукового датчика. На цьому рисунку система спочатку ініціалізує ультразвуковий датчик. Датчик надсилає та приймає ультразвук для вимірювання рівня порожнього сміттевого бака. Також ультразвуковий датчик надсилає відповідні виміряні дані на мікроконтролер.

Завдяки доступному з'єднанню через Bluetooth забезпечується надсилання виміряних даних до веб додатку портативного комп'ютера. За відсутності з'єднання Bluetooth виконується перевірка інтернет-з'єднання. Якщо інтернет-з'єднання через Wi-Fi доступне, система відправить дані через хмарний сервер на веб додаток.

Отримані дані як з хмарного сервера, так і через Bluetooth з'єднання забезпечують моніторинг даних через веб додаток. Для перевірки достовірності наповненості контейнера встановлено поріг шкали.

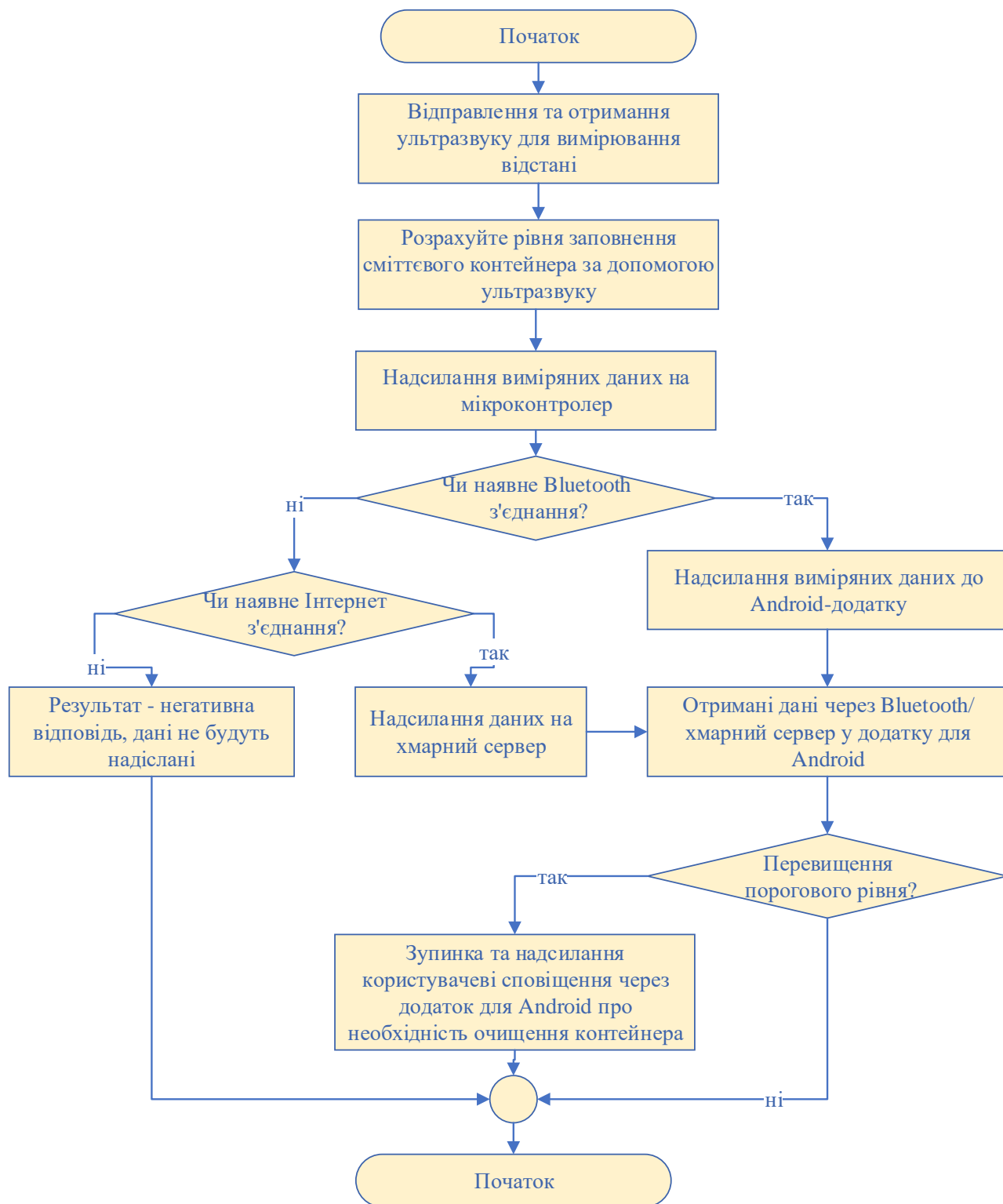


Рисунок 3.4 – Блок-схема алгоритму роботи ультразвукового датчика

Якщо відходи перетинають пороговий рівень, система зупиняє відповідний процес і попереджає користувача, надсилаючи повідомлення про необхідність очистити та замінити сміттєвий контейнер через веб додаток. У разі відсутності інтернет-з'єднання дані не надсилатимуться.

### 3.5. Алгоритм роботи датчика вимірювання навантаження

Для вимірювання ваги сміття передбачено датчик вимірювання навантаження. На рис. 3.5 представлено блок-схему принципу роботи датчика вимірювання навантаження.

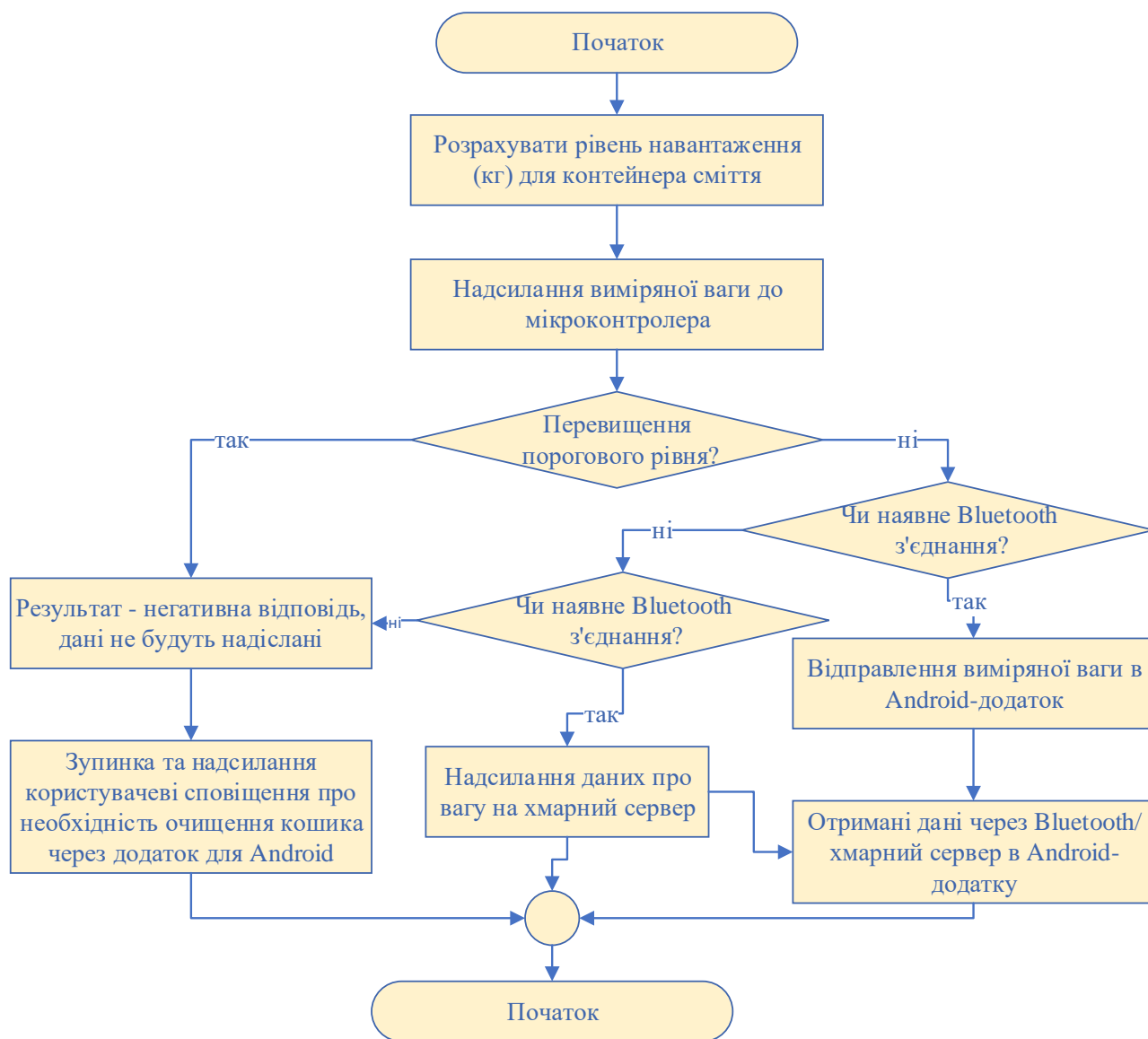


Рисунок 3.5 – Блок-схема алгоритму роботи датчика вимірювання навантаження

Спочатку ініціалізували датчик для зчитування даних. Датчик вимірювання навантаження обчислює навантаження в кг і надсилає дані назад до мікроконтролера. Мікроконтролер перевіряє, чи не перевищує вага максимальний пороговий рівень. Негативна відповідь про перевищення максимального рівня призводить до пошуку з'єднання Bluetooth. Якщо

з'єднання Bluetooth доступне, система надсилає дані до веб додатку портативного комп'ютера.

Якщо з'єднання Bluetooth неможливе, система перейде далі і перевірить підключення до Інтернету. Якщо з'єднання з Інтернетом відсутнє, система припинить вимірювання, і дані не будуть надіслані.

Система дозволяє передавати повідомлення в веб додаток, яке вказує користувачеві на необхідність почистити і замінити контейнер для сміття. Якщо інтернет-з'єднання працює, система надсилає виміряну вагу до хмари. Позитивна спроба отримувати дані як з хмарного сервера, так і через Bluetooth з'єднання дозволила відстежувати дані через веб додаток у режимі реального часу.

### **3.6. Згорточна нейронна мережа CNN для класифікації відходів**

Згорткова нейронна мережа (CNN) – це складна мережа прямого поширення. Вона може вирішувати широкий спектр завдань, які раніше були нерозв'язними. CNN працює шляхом вилучення ознак із зображень. CNN широко використовуються для класифікації зображень завдяки своїй високій точності. Класифікація зображень – це процедура введення зображення і виведення відповідного класу, до якого воно належить. У запропонованій нами системі розділили відходи на дві широкі категорії – органічні (які розкладаються і утворюють біогаз) та неорганічні (які нерозкладаються) відходи. До неорганічних відходів належать картон, скло, метал, папір та інші.

CNN складається з вхідного шару, групи прихованих шарів та вихідного шару. Приховані шари зазвичай включають згорткові шари, шари випрямлених лінійних одиниць (Rectified Linear Unit, ReLU), об'єднані шари та повністю з'єднані шари. Вхідне зображення пропускається через приховані шари і, нарешті, генерує вихідне. За допомогою серії згорткових шарів комп'ютер досягає більш абстрактного уявлення про зображення об'єкта (рис. 3.6).

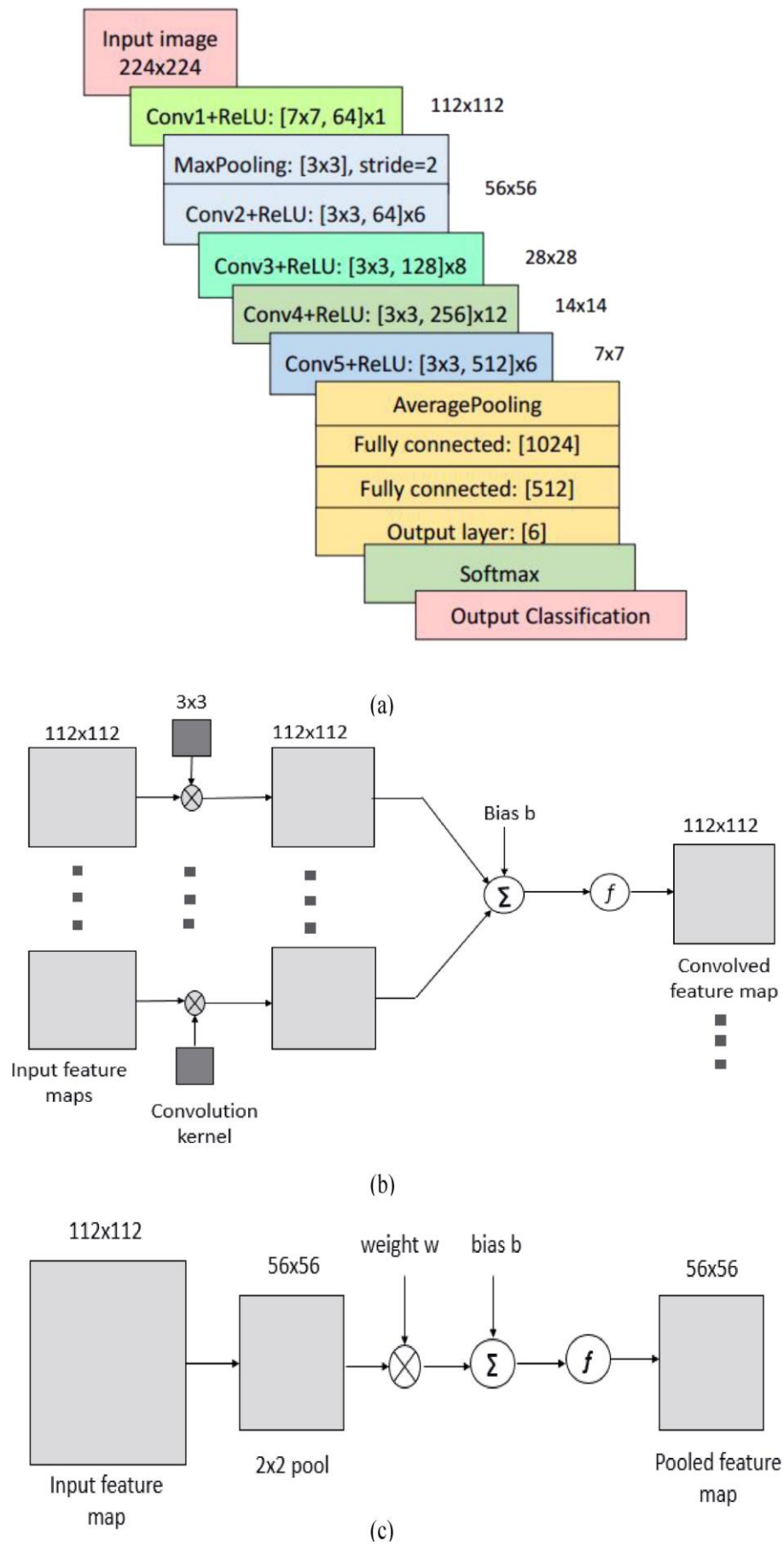


Рисунок 3.6 – Топологія CNN для запропонованої системи

Мережа приймає на вхід двовимірне зображення і згортає його за допомогою згорткового фільтра Габора  $7 \times 7$ . Різні шари CNN генерують кілька рівнів ознак. Існуючі шари виконують різні математичні обчислення і передають згенеровані дані на вхід наступним шарам.

Нами вибрано архітектуру ResNet34, яка працює значно краще за інших. Рис. 3.6, б ілюструє топологію CNN, реалізовану на основі ResNet-34 для запропонованої системи. Мережа приймає на вхід зображення розміром  $224 \times 224$  і генерує ознаки за допомогою фільтра згортки. Кожен шар виконує різні математичні обчислення і передає згенеровані дані на вхід наступним шарам.

Фільтр  $7 \times 7$  з кроком 2 використовується для першого шару згортки над вхідним зображенням  $224 \times 224$ , як показано на рис. 3.6, с. Решта шарів згортки виконують згортку  $3 \times 3$  з розміром карти ознак 64, 128, 256 і 512 відповідно. В операціях також застосовуються зовнішній зсув і функція ReLU. Повна архітектура ResNet-34, що складається з 34 шарів, показана на Рисунку 6(а). Кожен шар згортки містить карту ознак у трьох вимірах: висота, ширина, глибина. Шар згортки виробляє і отримує тензор розміром  $[W_1 \times H_1 \times D_1]$ , де  $W_1, H_1$  та  $D_1$  виражають розмірність тензора. Згорнута карта вихідних ознак є тензором розміром  $[W_2 \times H_2 \times D_2]$ , де  $W_2, H_2$  та  $D_2$  обчислюються за формулами (3.1), (3.2) та (3.3):

$$W_2 = \frac{W_1 - F + 2P}{S} + 1, \quad (3.1)$$

$$H_2 = \frac{H_1 - F + 2P}{S} + 1, \quad (3.2)$$

$$D_2 = K, \quad (3.3)$$

де  $K$  – позначає кількість фільтрів;  $F$  – просторове розширення фільтра;  $P$  – нульове заповнення;  $S$  – крок.

Після шару згортки розміщується шар субдискретизації для зменшення просторової роздільної здатності згорнутої карти ознак. Зображення розбивається на частини, що не перекриваються. На цьому шарі застосовується

операція максимального об'єднання для зменшення дискретизації. Тут максимальне об'єднання виконується за допомогою матриці об'єднання  $3 \times 3$  з кроком 2, як показано на рис. 3.6(с). Шар об'єднання створює тензор розміром  $[W_3 \times H_3 \times D_3]$ , де  $W_3, H_3$  та  $D_3$  визначаються за допомогою рівнянь (3.4), (3.5) та (3.6):

$$W_3 = \frac{W_2 - F}{S} + 1, \quad (3.4)$$

$$H_3 = \frac{H_2 - F}{S} + 1, \quad (3.5)$$

$$D_3 = D_2 \cdot D_2. \quad (3.6)$$

Запропонована мережа CNN використовує два повністю зв'язані шари з 1024 та 512 нейронами. Щоб усунути ефект перенавчання, ми використали методи пакетної нормалізації та відсіювання на відповідних шарах. Потім як функцію активації на вихідному шарі використовується soft-max, яка нормалізує вхідний вектор  $z$  з  $k$  дійсних чисел до розподілу ймовірностей  $i$ , нарешті, класифікує зображення в мітку. Функція задається рівнянням (3.7):

$$f_j(z) = \frac{e^{z_j}}{\sum_k e^{z_k}} \text{ for } j = 1, 2, \dots, k. \quad (3.7)$$

## РОЗДІЛ 4.

# СТВОРЕННЯ ПРОГРАМНИХ МОДУЛІВ ДЛЯ ІНТЕЛЕКТУАЛЬНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ВІДХОДАМИ З ВИКОРИСТАННЯМ ГЛИБОКОГО НАВЧАННЯ

### 4.1. Створення програмного модуля для управління відходами

Для реалізації вашого проекту «Інтелектуальна інформаційна система управління відходами з використанням глибокого навчання» написано програмний модуль для роботи з камерою та сервоприводом. Використовуємо бібліотеки Python, такі як cv2 для роботи з камерою, tensorflow для завантаження моделі CNN, та gpiozero для керування серводвигуном на Raspberry Pi.

Написаний код, який реалізує основні функції вашого проекту, представлено у додатку А.

Насамперед нами виконано завантаження потрібних бібліотек (рис. 4.1).

```
import cv2
import numpy as np
import tensorflow as tf
from gpiozero import Servo
from time import sleep

# Ініціалізація камери
cap = cv2.VideoCapture(0)

# Завантаження попередньо навченої моделі CNN
model = tf.keras.models.load_model('path_to_your_model.h5')

# Ініціалізація серводвигуна
servo = Servo(17) # Приклад для GPIO 17
```

Рисунок 4.1 – Фрагмент коду завантаження потрібних бібліотек

Бібліотека cv2 (OpenCV) використовується для обробки зображень. Вона широко використовується для захоплення зображень з камер, обробки зображень. Пропонується використовувати бібліотеку numpy (np) для роботи з масивами та виконання математичних операцій. Вона часто використовується разом з OpenCV для обробки зображень.



Бібліотека `tensorflow (tf)` для машинного навчання, яка включає інструменти для створення, навчання та використання нейронних мереж. У даному випадку використовується для завантаження попередньо навченої моделі CNN (Convolutional Neural Network).

Бібліотека `gpiozero` використовується для спрощеного керування GPIO (General Purpose Input/Output) на Raspberry Pi. Використовується для керування серводвигуном.

Функція `time.sleep` з бібліотеки `time` призупиняє виконання програми на вказану кількість секунд. Використовується для створення затримок у програмі.

Після цього виконується ініціалізація камери. Де код `cv2.VideoCapture(0)` відкриває першу доступну камеру на системі (0 зазвичай відповідає першій підключеній камері). Цей об'єкт `cap` буде використовуватися для захоплення кадрів з відео потоку.

У подальшому виконується завантаження попередньо навченої моделі CNN. У нашому випадку, файл з моделлю зберігається за адресою `'path_to_your_model.h5'`. Ця модель буде використовуватися для класифікації зображень.

Після цього виконується ініціалізація серводвигуна. Ініціалізує серводвигун, підключений до GPIO піна номер 17 на Raspberry Pi. Об'єкт `servo` буде використовуватися для керування позицією серводвигуна.

Отже, код (рис. 4.2) імпортує необхідні бібліотеки для роботи з відео, обробки зображень, нейронних мереж та керування апаратними компонентами на Raspberry Pi.

Функція `preprocess_image` призначена для попередньої обробки зображень перед подачею їх на вхід моделі CNN для класифікації (рис. 4.2). Попередня обробка зображення включає зміну розміру, нормалізацію та додавання осі `batch`.

```
def preprocess_image(image):
    """
    функція для попередньої обробки зображення
    """
    image = cv2.resize(image, (224, 224)) # Зміна розміру відповідно до вимог моделі
    image = image / 255.0 # Нормалізація зображення
    image = np.expand_dims(image, axis=0) # Додавання осі batch
    return image
```

Рисунок 4.2 – Фрагмент коду із функцією preprocess\_image для попередньої обробки зображень

Функція preprocess\_image виконує три основні кроки для підготовки зображення – змінює розмір зображення до 224x224 пікселів, нормалізує піксельні значення до діапазону від 0 до 1, і додає вісь batch для сумісності з вимогами моделі CNN. Ці кроки забезпечують правильне форматування вхідних даних для ефективною класифікації зображень моделлю нейронної мережі.

Функція classify\_waste призначена для класифікації зображення за допомогою попередньо навченої моделі CNN (рис. 4.3). Ця функція приймає зображення як вхідний параметр, обробляє його та використовує модель для передбачення класу зображення.

```
def classify_waste(image):
    """
    функція для класифікації зображення за допомогою моделі CNN
    """
    processed_image = preprocess_image(image)
    prediction = model.predict(processed_image)
    return np.argmax(prediction) # Повернення класу з найвищою ймовірністю
```

Рисунок 4.3 – Фрагмент коду із функцією classify\_waste для класифікації зображення

Функція classify\_waste виконує такі кроки. Викликає функцію preprocess\_image для попередньої обробки вхідного зображення. Використовує модель CNN для передбачення класу зображення. Визначає клас зображення з найвищою ймовірністю за допомогою np.argmax. Повертає індекс класу з найвищою ймовірністю, який відповідає передбаченому класу зображення.

Ця функція є ключовою для класифікації відходів у системі управління відходами, оскільки вона визначає тип відходів і дозволяє системі відповідно сортувати їх.

Функція `control_servo` призначена для керування серводвигуном на основі результатів класифікації зображення (рис. 4.4). Вона приймає як вхідний параметр мітку класу (`class_label`), яка визначає тип відходів, і відповідно змінює позицію серводвигуна для сортування відходів.

```
def control_servo(class_label):
    """
    Функція для керування серводвигуном відповідно до результатів класифікації
    """
    if class_label == 0: # Приклад для органічних відходів
        servo.min() # Позиція серводвигуна для органічних відходів
    elif class_label == 1: # Приклад для неорганічних відходів
        servo.max() # Позиція серводвигуна для неорганічних відходів
    sleep(1) # Затримка для завершення руху серводвигуна
    servo.mid() # Повернення серводвигуна в початкову позицію
```

Рисунок 4.4 – Фрагмент коду із функцією `control_servo` для керування серводвигуном

Функція `control_servo` виконує наступні кроки. Перевіряє мітку класу відходів. Встановлює серводвигун у мінімальну позицію, якщо це органічні відходи (мітка класу 0). Встановлює серводвигун у максимальну позицію, якщо це неорганічні відходи (мітка класу 1). Виконує затримку на 1 секунду, щоб дати серводвигуну час для завершення руху. Повертає серводвигун у середню позицію, готуючи його до обробки наступного зображення.

Ця функція забезпечує правильне сортування відходів на основі класифікації, керуючи серводвигуном для переміщення відходів у відповідні контейнери.

Наступний блок коду (рис. 4.5) реалізує безперервний цикл для отримання зображень з камери, класифікації відходів за допомогою моделі CNN, керування серводвигуном на основі результатів класифікації та відображення отриманих зображень. У разі завершення або виникнення помилок, ресурси будуть належним чином звільнені.

```

try:
    while True:
        # Захоплення зображення з камери
        ret, frame = cap.read()
        if not ret:
            break

        # Класифікація зображення
        class_label = classify_waste(frame)

        # Керування серводвигуном на основі результатів класифікації
        control_servo(class_label)

        # Відображення захопленого зображення
        cv2.imshow('Waste Classification', frame)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

finally:
    # Звільнення ресурсів
    cap.release()
    cv2.destroyAllWindows()

```

Рисунок 4.5 – Фрагмент коду безперервного циклу для отримання зображень з камери, класифікації відходів за допомогою моделі CNN та керування серводвигуном

Отриманий код створює безперервний цикл, який отримує зображення з камери, класифікує це зображення на основі моделі CNN та керує серводвигуном відповідно до результатів класифікації. При цьому відображається отримане зображення у вікні. Завершує роботу при натисканні клавіші «q». В разі будь-яких проблем або завершення роботи, ресурси камери та вікна будуть належним чином звільнені.

## 4.2. Створення веб додатку для моніторингу наповнення контейнерів для відходів

Створення веб додатку для ПК, що забезпечує моніторинг рівня заповнення та ваги сміттєвих контейнерів потребує кількох кроків, включаючи розробку інтерфейсу користувача (UI), отримання даних з мікроконтролера та обробку цих даних для відображення. Нами використано Coderep.io як середовище розробки.

Для створення веб-додатку, який відобразить вагу сміття та рівень заповнення контейнера в реальному часі, використано HTML, CSS та JavaScript (рис. 4.6).

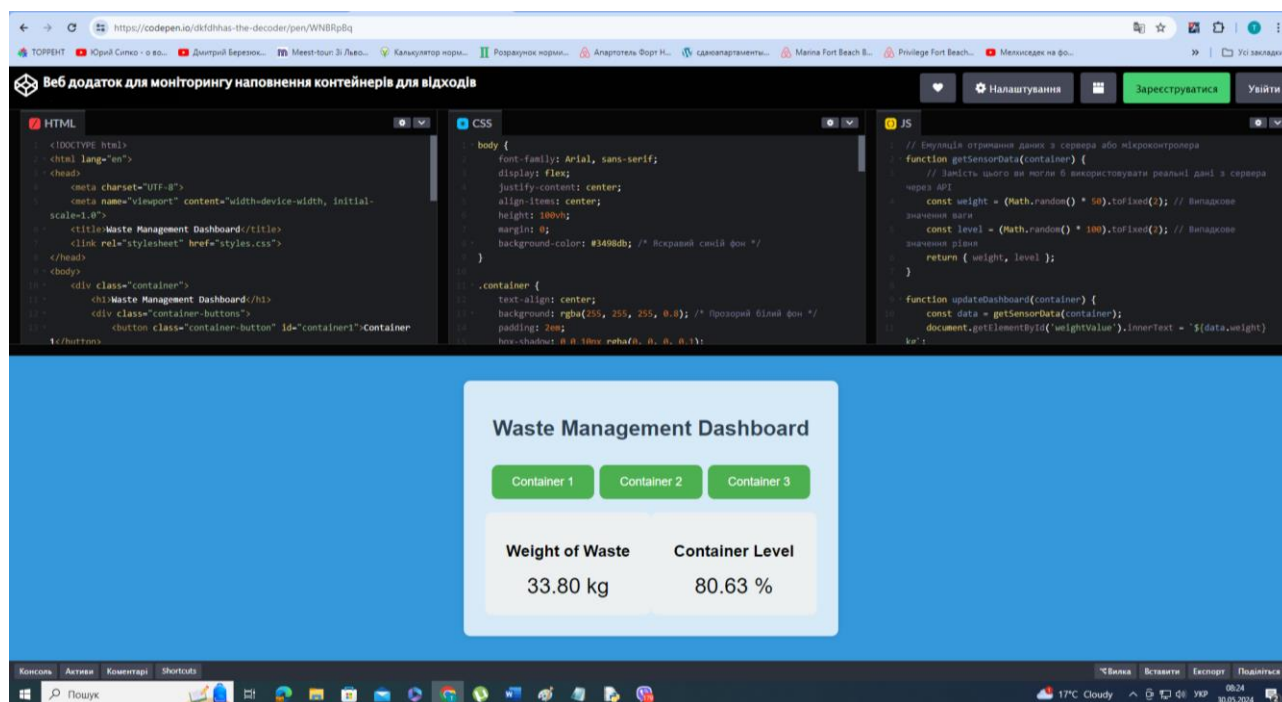


Рисунок 4.6 – Створення веб-додатку для моніторингу наповнення контейнерів для відходів

Написаний код веб-додатку для моніторингу наповнення контейнерів для відходів представлено у додатку А. Насамперед було встановлено базову структуру HTML. Запропоновано два елементи для відображення ваги сміття та рівня заповнення контейнера. Передбачено підключення стилів CSS та скрипт JavaScript.

При цьому styles.css встановлює основні стилі для додатку, забезпечуючи простий та чистий інтерфейс. Для інтеграції з реальними даними слід замінити функцію `getSensorData()` на реальне API для отримання даних з мікроконтролера або сервера. При цьому передбачено використання `fetch` для запитів до API. Є можливості розширення функціональності. Зокрема, можна додати більше інформації, такої як історія даних, графіки тощо.

### 4.3. Створення моделі нейронної мережі CNN

Створимо модель нейронної мережі CNN для класифікації відходів, використовуючи архітектуру ResNet-34. Для цього використовуємо бібліотеки TensorFlow та Keras для реалізації запропонованої архітектури. Код для створення моделі нейронної мережі CNN для класифікації відходів наведено у додатку А.

Насамперед виконується імпорт потрібних бібліотек (рис. 4.7).

```
import tensorflow as tf
from tensorflow.keras.applications import ResNet34
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D
from tensorflow.keras.models import Model
from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Завантаження базової моделі ResNet-34 з попередньо навченими вагами
base_model = ResNet34(weights='imagenet', include_top=False, input_shape=(224, 224, 3))
```

Рисунок 4.7 – Фрагмент коду для імпорту потрібних бібліотек

Нами використано бібліотеку tensorflow як основну для побудови та тренування моделі нейронної мережі. Вибрано архітектуру ResNet-34, попередньо натренованої на наборі даних ImageNet. Після цього виконано додавання шарів до моделі Dense та GlobalAveragePooling2D з tensorflow.keras.layers.

Цей фрагмент коду підготовляє базову модель ResNet-34 для подальшого додавання та тренування нових шарів для задачі класифікації відходів.

Наступним кроком здійснено додавання власних шарів для задачі класифікації відходів (рис. 4.8).

```
# Додавання власних шарів для задачі класифікації відходів
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(512, activation='relu')(x) # Додатковий повнозв'язний шар
predictions = Dense(3, activation='softmax')(x) # Вихідний шар з 3 класами (наприклад: органічні, неорганічні, інші)

# Створення повної моделі
model = Model(inputs=base_model.input, outputs=predictions)

# Замороження wag базової моделі
for layer in base_model.layers:
    layer.trainable = False

# Компіляція моделі
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

Рисунок 4.8 – Фрагмент коду для нових шарів до базової моделі ResNet-34 та її компіляції

При цьому передбачено додавання шару глобального середнього об'єднання (Global Average Pooling), який зменшує розмірність даних та узагальнює ознаки. Виконано додавання повнозв'язного (Dense) шару з 512 нейронами та активацією ReLU. Це дозволяє моделі вивчати складніші представлення даних. Окрім того, виконано додавання вихідного шару з 3 нейронами та softmax активацією для класифікації на три класи (наприклад, органічні, неорганічні, інші). Після цього виконується створення повної моделі, замороження ваг базової моделі та компіляція створеної моделі.

Цей код (рис. 4.8) додає нові шари до базової моделі ResNet-34, заморожує ваги базової моделі, щоб зосередитися на навчанні нових шарів, і компілює модель для подальшого тренування.

Наступний код (рис. 4.9) забезпечує автоматичне зчитування, підготовку, трансформацію та генерацію зображень для навчання та валідації моделі нейронної мережі. Це дозволяє уникнути перенавчання моделі та покращити її узагальнювальну здатність на нових даних.

```
# Підготовка даних
train_datagen = ImageDataGenerator(
    rescale=1./255,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    validation_split=0.2
)

train_generator = train_datagen.flow_from_directory(
    'path_to_your_data/train', # Шлях до тренувальних даних
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical',
    subset='training'
)

validation_generator = train_datagen.flow_from_directory(
    'path_to_your_data/train', # Шлях до тренувальних даних
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical',
    subset='validation'
)
```

Рисунок 4.9 – Фрагмент коду для автоматичного зчитування, підготовки, трансформації та генерації зображень для навчання та валідації моделі нейронної мережі

Наступний код (рис. 4.10) виконує тренування моделі нейронної мережі на даних, які генеруються в реальному часі з допомогою ImageDataGenerator.

```

# Тренування моделі
model.fit(
    train_generator,
    steps_per_epoch=train_generator.samples // train_generator.batch_size,
    validation_data=validation_generator,
    validation_steps=validation_generator.samples // validation_generator.batch_size,
    epochs=10
)

# Збереження моделі
model.save('waste_classification_model.h5')

```

Рисунок 4.10 – Фрагмент коду для тренування та навчання моделі нейронної мережі

Тренування відбувається протягом 20 епох, і на кожній епосі модель перевіряється на валідаційних даних. Після завершення тренування модель зберігається у файл для подальшого використання.

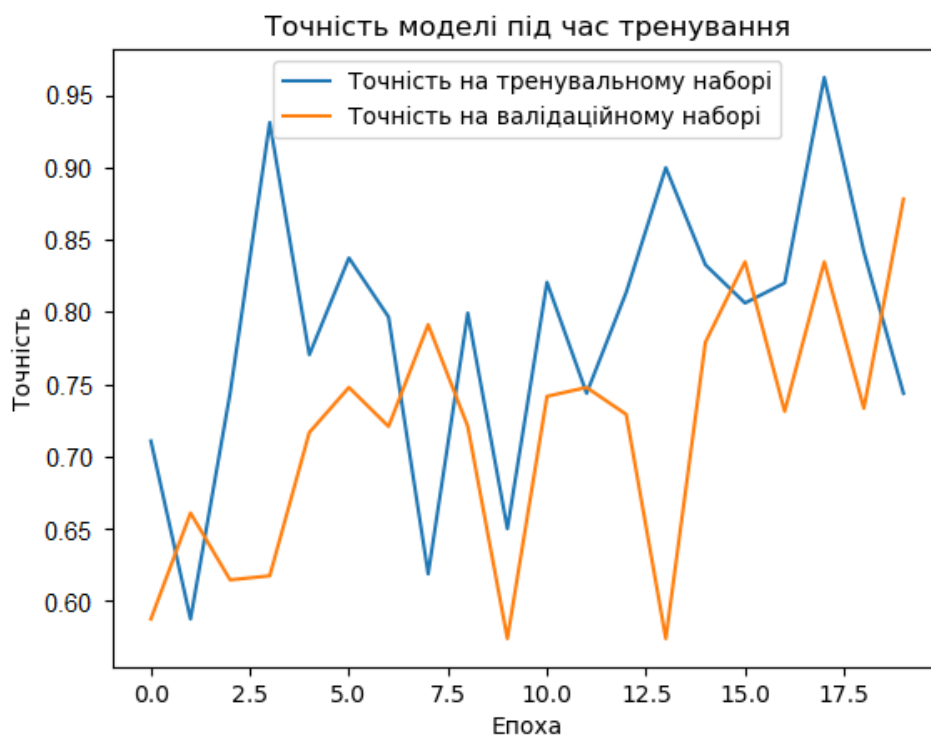


Рисунок 4.11 – Результати визначення точності моделі під час тренування

Встановлено, що модель дає можливість класифікувати зображення відходів із точністю 85%, що достатньо для використання її у запропонованій інтелектуальній системі.



## **РОЗДІЛ 5.**

### **ОХОРОНА ПРАЦІ**

Розробка інтелектуальних інформаційних систем, зокрема систем управління відходами з використанням глибокого навчання, є складним та багатограним процесом. Важливим аспектом цього процесу є забезпечення належного рівня охорони праці для всіх учасників проєкту. Охорона праці під час розробки подібних систем охоплює широкий спектр питань, включаючи фізичну безпеку, психоемоційне здоров'я працівників, забезпечення ергономічних умов праці та дотримання стандартів безпеки роботи з комп'ютерним обладнанням та програмним забезпеченням.

Проєкти з впровадження інтелектуальних систем управління відходами, особливо із залученням технологій глибокого навчання, потребують висококваліфікованих спеціалістів у галузях програмування, аналізу даних, інженерії та екології. Тривала робота за комп'ютером, часті стресові ситуації, пов'язані з дедлайнами, та необхідність постійного навчання можуть впливати на фізичне та психоемоційне здоров'я розробників. Відтак, важливо створити такі умови праці, що сприятимуть не лише продуктивності, але й добробуту всіх учасників проєкту.

#### **5.1. Аналіз стану робочого місця та умов праці**

Аналіз стану робочого місця та умов праці є ключовим етапом для забезпечення ефективної та безпечної роботи команди, залученої до розробки Інтелектуальної інформаційної системи управління відходами з використанням глибокого навчання. Цей аналіз дозволяє виявити потенційні ризики, визначити проблемні зони та розробити заходи для їх усунення.

Робочі місця повинні бути розташовані у просторах та добре вентильованих приміщеннях. Важливо забезпечити належне природне та штучне освітлення для зниження навантаження на зір.

Забезпечити використання ергономічних крісел та столів, що дозволяють налаштовувати висоту та кут нахилу. Монітори повинні розташовуватись на рівні очей, щоб зменшити навантаження на шию та спину. Клавіатура та миша повинні бути розташовані на комфортній висоті для запобігання напруженню рук та зап'ясть.

Робоче місце повинно бути чистим та організованим, з достатньою кількістю простору для зручної роботи. Наявність полиць, шухляд та інших засобів зберігання для розташування документів та особистих речей.

Слід передбачити використання сучасних комп'ютерів та іншого обладнання, що відповідає вимогам продуктивності та безпеки. Виконувати регулярне технічне обслуговування та оновлення обладнання для забезпечення його належного функціонування.

Аналіз стану робочого місця та умов праці дозволяє створити оптимальні умови для роботи команди, що займається розробкою інтелектуальної інформаційної системи управління відходами з використанням глибокого навчання. Дотримання вимог щодо фізичних, технічних, психоемоційних та організаційних умов праці сприяє підвищенню продуктивності, зниженню ризиків для здоров'я та забезпеченню безпеки працівників.

## **5.2. Небезпечні та шкідливі чинники**

Розробка інтелектуальної інформаційної системи управління відходами з використанням глибокого навчання, як і будь-яка інша діяльність у сфері інформаційних технологій, має потенційні небезпечні та шкідливі виробничі чинники, які можуть впливати на здоров'я та безпеку працівників. Аналіз цих

чинників допомагає визначити заходи, необхідні для мінімізації ризиків і забезпечення комфортних та безпечних умов праці.

Одним з основних небезпечних факторів є неправильна організація робочого місця, що може призвести до проблем з опорно-руховим апаратом, таких як біль у спині, шиї та руках. Тривала робота за комп'ютером без перерв може спричинити синдром зап'ястного каналу, м'язову напругу та інші ергономічні проблеми. Важливо забезпечити правильну висоту столу та крісла, використовувати підставки для ніг, клавіатури та моніторів, а також заохочувати регулярні перерви та розминки.

Робота з комп'ютерами та іншими електронними пристроями супроводжується впливом електромагнітного випромінювання. Хоча сучасні пристрої виробляють мінімальні рівні випромінювання, тривала експозиція може мати негативний вплив на здоров'я. Рекомендується дотримуватися безпечних відстаней від моніторів, використовувати захисні екрани та регулярно провітрювати приміщення.

Робота над складними проектами, такими як розробка інтелектуальних систем, часто пов'язана з високими психоемоційними навантаженнями. Постійний стрес, дедлайни, велика кількість інформації для обробки можуть призвести до вигоряння, тривожних станів та інших психологічних проблем. Важливо забезпечити підтримку психоемоційного здоров'я працівників через організацію сприятливої атмосфери на роботі, проведення командних заходів та надання психологічної допомоги при необхідності.

Тривала робота за комп'ютером без належних перерв може призвести до зорового напруження та розвитку комп'ютерного зорового синдрому. Симптоми включають сухість очей, затуманення зору, головний біль та втому очей. Для мінімізації ризиків рекомендується дотримуватися правил «20-20-20» (кожні 20 хвилин робити 20-секундну перерву, дивлячись на об'єкт, що знаходиться на відстані 20 футів), використовувати монітори з належними параметрами яскравості та контрасту, а також регулярні огляди у офтальмолога.

Враховуючи всі вищезазначені чинники, розробка та впровадження заходів для забезпечення безпечних та комфортних умов праці є необхідним елементом успішної роботи над інтелектуальною інформаційною системою управління відходами.

### **5.3. Інструкція із охорони праці під час розробки інтелектуальної інформаційної системи управління відходами з використанням глибокого навчання**

#### *1. Загальні положення*

1.1. Ця інструкція встановлює вимоги охорони праці під час розробки інтелектуальної інформаційної системи управління відходами з використанням глибокого навчання.

1.2. Інструкція обов'язкова для виконання всіма працівниками, залученими до процесу розробки системи.

1.3. Відповідальність за забезпечення охорони праці несе керівник проекту.

#### *2. Організація робочого місця*

2.1. Робоче місце повинно бути обладнане ергономічними меблями (стіл, крісло), що регулюються за висотою та нахилом.

2.2. Монітор комп'ютера має бути розташований на рівні очей на відстані 50-70 см від користувача. Клавіатура і миша повинні бути на зручній висоті для рук.

2.3. Освітлення робочого місця повинно бути достатнім, з перевагою природного світла. Необхідно уникати відблисків на екрані монітора.

2.4. Робоче місце має бути чистим і організованим, з достатньою кількістю місця для розташування обладнання та документів.

#### *3. Використання обладнання*

3.1. Використовувати лише справне та сертифіковане обладнання.

3.2. Регулярно проводити технічне обслуговування комп'ютерної техніки.

3.3. При роботі з електронним обладнанням дотримуватися правил електробезпеки.

3.4. Використовувати спеціальні підставки та органайзери для кабелів, щоб уникнути спотикання та пошкоджень.

#### *4. Робота з програмним забезпеченням*

4.1. Встановлювати та використовувати тільки ліцензійне програмне забезпечення.

4.2. Регулярно оновлювати програмне забезпечення для забезпечення безпеки та функціональності.

4.3. Використовувати антивірусне програмне забезпечення та інші засоби кібербезпеки для захисту даних.

4.4. Дотримуватись політики безпеки інформації, включаючи зберігання конфіденційних даних у захищених місцях.

#### *5. Психоемоційне здоров'я*

5.1. Забезпечувати підтримку сприятливого психологічного клімату в колективі, сприяти відкритій комунікації.

5.2. Регулярно проводити тренінги з управління стресом та психоемоційною стійкістю.

5.3. Організувати робочий графік таким чином, щоб уникати перевантажень та надмірної напруги.

5.4. Надати можливість працівникам отримувати психологічну підтримку при необхідності.

#### *6. Гігієна праці*

6.1. Працівники повинні регулярно мити руки, особливо перед їжею та після використання туалету.

6.2. Робоче місце повинно бути оснащено засобами особистої гігієни, включаючи антисептики та серветки.

6.3. Регулярно провітрювати приміщення для забезпечення свіжого повітря.

6.4. Підтримувати чистоту та порядок на робочому місці, проводити регулярне прибирання.

### *7. Надання першої допомоги*

7.1. У разі нещасного випадку надавати першу допомогу згідно з інструкціями та викликати медичну допомогу.

7.2. Робоче місце повинно бути оснащено аптечкою першої допомоги з необхідними медикаментами та засобами.

7.3. Працівники повинні бути навчені наданню першої допомоги та знати розташування аптечки.

### *8. Контроль та відповідальність*

8.1. Керівник проекту повинен регулярно контролювати дотримання вимог цієї інструкції.

8.2. За порушення вимог охорони праці працівники несуть відповідальність згідно з чинним законодавством.

8.3. Всі працівники повинні бути ознайомлені з цією інструкцією під розпис.

### *9. Заключні положення*

9.1. Інструкція переглядається та оновлюється у разі змін умов праці або законодавства.

9.2. Всі пропозиції та зауваження щодо покращення умов праці розглядаються керівником проекту та, за необхідності, впроваджуються в робочий процес.

## ВИСНОВКИ І ПРОПОЗИЦІЇ

У нашій роботі розглядається розробка та впровадження інтелектуальної інформаційної системи управління відходами, що базується на глибокому навчанні. Робота включає аналіз існуючих методів і технологій, розробку концептуальної моделі системи, а також оцінку ефективності впровадження запропонованих рішень. Особливо важливо використовувати алгоритми глибокого навчання для вирішення завдань класифікації прогнозу та в рамках системи управління відходами.

Нами проаналізовано стан та особливості управління відходами. Зокрема визначено складові управління відходами та особливості державної політики у цьому напрямі. Національна стратегія управління відходами в Україні до 2030 року вказує на повну відповідність вимогам рамкової директиви ЄС щодо ієрархії поводження з відходами та відповідно доцільність розробок у цьому напрямі.

Виконаний аналіз стану використання глибокого навчання під час управління відходами свідчить про те, що заслуговують на увагу згорткові нейронні мережі (CNN). Вона є одним із найбільш визнаних алгоритмів глибокого навчання для його широкого застосування в класифікації зображень та їх сегментації. Тому пропонується виконувати класифікацію відходів із її використанням.

Сьогодні відомо декілька існуючих інтелектуальних інформаційних систем управління відходами. Вони мають різну архітектуру та використовують різних інструментарій.

Розробка та впровадження інтелектуальних інформаційних систем з використанням глибокого навчання може суттєво покращити процеси управління відходами. Глибоке навчання дозволяє створювати високоточні моделі для аналізу великих обсягів даних, що дає змогу автоматизувати і оптимізувати різні аспекти управління відходами. Це, в свою чергу, сприяє

зниженню витрат, зменшенню негативного впливу на довкілля та підвищенню ефективності системи управління відходами.

Нами здійснено вибір технологій та засобів для розробки інтелектуальної інформаційної системи управління відходами. Запропоновано використовувати фреймворки TensorFlow та PyTorch для розробки та тренування моделей глибокого навчання. Використано бібліотеку OpenCV для комп'ютерного зору, яка використовується для обробки зображень відходів.

Нами використано загально доступний набір даних із зображеннями відходів. Він включає зображення скла, паперу, картону, пластику, металу та сміття. Нами написано код, який виконує кілька кроків для аналізу, підготовки та візуалізації даних для моделі нейронної мережі CNN для класифікації відходів.

Запропонована інтелектуальна інформаційна система управління відходами базується на інтеграції глибокого навчання та Інтернету речей (IoT). Розроблена структурна схема системи управління відходами з використанням глибокого навчання являє собою поєднання двох частин, а саме класифікації відходів за допомогою згорткової нейронної мережі та розумних сміттєвих контейнерів, що сприяє моніторингу даних у реальному часі за допомогою Інтернету речей.

Одним із вагомих модулів інтелектуальної інформаційної системи управління відходами є модуль камери та сервоприводу. Модуль камери підключений до мікроконтролера запропонованої системи і відповідає за отримання зображень зі станом відходів. На рис. 2 представлено блок-схему модуля із камерою та сервоприводом.

Нами запропоновану структурну схему розумного контейнера для сміття. Для нього використовується мікроконтролер під назвою «ESP8266» node MCU. Ультразвуковий датчик підключений до мікроконтролера для вимірювання рівня заповнення контейнера для сміття.

На рис. 3.4 зображено блок-схему роботи ультразвукового датчика. На цьому рисунку система спочатку ініціалізує ультразвуковий датчик. Датчик



надсилає та приймає ультразвук для вимірювання рівня порожнього сміттевого бака. Також ультразвуковий датчик надсилає відповідні виміряні дані на мікроконтролер.

Для вимірювання ваги сміття передбачено датчик вимірювання навантаження. На рис. 3.5 представлено блок-схему принципу роботи датчика вимірювання навантаження.

Нами вибрано архітектуру моделі нейронної мережі ResNet34, яка працює значно краще за інших. Рис. 3.6, б ілюструє топологію CNN, реалізовану на основі ResNet-34 для запропонованої системи. Мережа приймає на вхід зображення розміром  $224 \times 224$  і генерує ознаки за допомогою фільтра згортки. Кожен шар виконує різні математичні обчислення і передає згенеровані дані на вхід наступним шарам.

Для реалізації вашого проекту «Інтелектуальна інформаційна система управління відходами з використанням глибокого навчання» написано програмний модуль для роботи з камерою та сервоприводом. Використовуємо бібліотеки Python, такі як cv2 для роботи з камерою, tensorflow для завантаження моделі CNN, та gpiozero для керування серводвигуном на Raspberry Pi.

Створено веб додатку для ПК, що забезпечує моніторинг рівня заповнення та ваги сміттевих контейнерів. Для цього виконано кілька кроків, включаючи розробку інтерфейсу користувача (UI), отримання даних з мікроконтролера та обробку цих даних для відображення. Нами використано Codepen.io як середовище розробки.

Створено модель нейронної мережі CNN для класифікації відходів, використовуючи архітектуру ResNet-34. Для цього використано бібліотеки TensorFlow та Keras для реалізації запропонованої архітектури. Наведено код для створення моделі нейронної мережі CNN для класифікації відходів.

Розроблені заходи із охорони праці забезпечують покращення умов праці під час розробка інтелектуальної інформаційної системи управління відходами, що базується на глибокому навчання.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Береза А. М. Основи створення інформаційних систем. Навч. посібник. 2-е вид., перероб. і доп. К.: КНЕУ, 2001. 214 с.
2. Берко А. Системи баз даних та знань. Книга 1. Організація баз даних та знань: Навч. посібник / А. Берко, О. Верес, В. Пасічник. Львів: Магнолія 2006, 2011. 456 с.
3. Боюн В. П. Інтелектуальні комп'ютерні системи сприйняття і обробки фізичної інформації: вісник Національної академії наук України. Київ, 2015. С. 82–84.
4. Вахнюк С.В. Технологія створення програмних та інтелектуальних систем: навчальний посібник / С. В. Вахнюк. Суми: ДВНЗ “УАБС НБУ”, 2011. 254 с.
5. Вергунов В. В. Інтелектуальні системи збереження даних як інструмент для підтримки оптимальної структури даних та архітектури бази даних: вісн. КНУ ім. Тараса Шевченка. Київ, 2013. С. 122–127.
6. Використання штучних нейронних мереж для прогнозування складових гібридних проєктів / А. Тригуба, В. Пташник, А. Татомир, Н. Коваль, І. Кондисюк. Теорія і практика розвитку агропромислового комплексу та сільських територій: матеріали XXII Міжнар. наук.-практ. форуму, 5-7 жовт. 2021р.: у 2 т. Львів: ННБК «АТБ», 2021. Т. 2. С. 96-100.
7. Гандзюк М. П., Желібо Е. П., Халімовський М. О. Основи охорони праці / За ред. Гандзюка М. П. К.: Каравела 2003. 405 с.
8. Губський Ю. В., Іванов С. М. Системи штучного інтелекту. Навч. посіб. К.: Академвидав, 2010. 236с.
9. Жидецький В.Ц. Основи охорони праці: Підручник. К.: Основа, 2002. 320с.
10. Зайченко Ю. П. Основи проєктування інтелектуальних систем. Київ: Слово, 2006. 352 с.

11. Любарський С.В., Шаціло П.В. Методологія вибору моделі подання знань в інтелектуальних навчальних системах: збірник наукових праць ВІПІ НТУУ «КПІ». Київ, 2010. С. 65–71.
12. Мельник Л.Г., Воляннюк Н.В. Машинне навчання: підручник. К.: Ліра-К, 2018. 364с.
13. Москальова В. М. Основи охорони праці. Підручник. Київ: ВД Професіонал, 2005. 666 с.
14. Набір зображень із відходами. URL: <https://raw.githubusercontent.com/garythung/trashnet/master/data/dataset-resized.zip>
15. Недашківський О.Л. Планування та проектування інформаційних систем: навчальний посібник / О.Л. Недашківський. Київ, 2014. 215с.
16. Нестеренко О.В., Ковтунець О.В., Фаловськнй О.О. Інтелектуальні системи і технології. Ввідний курс: Навч. посібник. К.: Національна академія управління, 2017. 90 с.
17. Програмування числових методів мовою Python : підруч. / А. В. Анісімов, А. Ю. Дорошенко, С. Д. Погорілий, Я. Ю. Дорогий ; за ред. А. В. Анісімова. К. : Видавничо-поліграфічний центр «Київський університет», 2014. 640 с.
18. Стан сфери поводження з побутовими відходами в Україні за 2013-2020 роки. URL: <https://www.minregion.gov.ua/napryamkidiyalnosti/zhkh/terretory/>
19. Tryhuba A., Boyarchuk V., Tryhuba I., Ftoma O., Padyuka R., Rudynets M., Forecasting the Risk of the Resource Demand for Dairy Farms Basing on Machine Learning, Proceedings of the 2nd International Workshop on Modern Machine Learning Technologies and Data Science (MoMLeT+DS 2020). Volume I: Main Conference. Lviv-Shatsk, Ukraine, June 2-3, 2020. pp.327-340.
20. Тригуба А. Системна модель цифрової трансформації сільських територіальних громад на основі обчислювального інтелекту. Вісник Львівського національного екологічного університету. Агроінженерні дослідження, 2022, № 26, р. 177-184.

21. Управління відходами. URL: <https://mepr.gov.ua/upravlinnya-vidhodamy/>
22. Awe O., Mengistu R., and Sreedhar V. Smart trash net: waste localization and classification. arXiv Preprint, 2017.
23. Christensen T. H., E. Gentil, A. Boldrin, A. W. Larsen, B. P. Weidema, and M. Hauschild, “C balance, carbon dioxide emissions and global warming potentials in LCA-modelling of waste management systems,” Waste Management & Research, vol. 27, no. 8, pp. 707–715, 2009.
24. Chu Y., Huang C., Xie X., Tan B., Kamal S., Xiong X. Multilayer Hybrid Deep-Learning Method for Waste Classification and Recycling. Comput Intell Neurosci. 2018 Nov 1:2018:5060857.
25. Donovan J. Auto-Trash Sorts Garbage Automatically at the TechCrunch Disrupt Hackathon, 2018. URL: <https://techcrunch.com/2016/09/13/auto-trash-sorts-garbage-automatically-at-the-techcrunch-disrupt-hackathon/>
26. He K., Zhang X., Ren S., and Sun J. Deep residual learning for image recognition. Proceedings of Conference on Computer Vision and Pattern Recognition, Las Vegas Valley, NV, USA, June 2016.
27. Hoornweg D. and Bhada-Tata P., What a Waste: A Global Review of Solid Waste Management, World Bank, Washington, DC, USA, 2012.
28. Komatineni S. Pro Android 4 (Professional Apress) / Satya Komatineni, Dave MacLean. New York: Apress. 2012. 1020 p.
29. Koval N., Tryhuba A., Kondysiuk I., Tryhuba I., Boiarchuk O. Forecasting the Fund of Time for Performance of Works in Hybrid Projects Using Machine Training Technologies. MoMLLeT+ DS, 196-206.
30. Malanchuk O., Tryhuba A., Tryhuba I., Sholudko R., Pankiv O., A Neural Network Model-based Decision Support System for Time Management in Pediatric Diabetes Care Projects. International Scientific and Technical Conference on Computer Sciences and Information Technologies, 2023.

31. Malanchuk O.M., Triguba A.M., Pankiv O.V., Sholudko R.Y. Architecture of an intelligent information system for forecasting components of medical projects. *Applied aspects of information technology* 6 (4), 376-390-376-390.
32. Mittal G., Yagnik K. B., Garg M., and Krishnan N. C. SpotGarbage: smartphone app to detect garbage using deep learning. *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, New York, NY, USA, 2016.
33. Pandas [Электронный ресурс]. URL: <http://pandas.pydata.org/pandas-docs/stable/overview.html#license>.
34. Python (programming language). URL: <https://bitly su/paHJe>.
35. Rad M. S., Kaenel A. V., Droux A. et al. A computer vision system to localize and classify wastes on the streets. *Computer Vision System*, pp. 195–204, Springer, Cham, Switzerland, 2017.
36. Rahman W., Islam R., Hasan A., Bithi N.I., Hasan Md. M., Motiur Rahman M.M. Intelligent waste management system using deep learning with IoT. *Journal of King Saud University - Computer and Information Sciences*. Volume 34, Issue 5, May 2022, Pages 2072-2087.
37. Sanderson R. E. Environmental Protection Agency Office of Federal Activities' Guidance on Incorporating EPA's Pollution Prevention Strategy into the Environmental Review Process, EPA, Washington, DC, USA, 1993.
38. Thung G. and Yang M. Classification of trash for recyclability status. *arXiv Preprint*, 2016.
39. Tryhuba A., Boyarchuk V., Tryhuba I., Francik S., Rudynets M., Method and software of planning of the substantial risks in the projects of production of raw material for biofuel, *CEUR Workshop Proceedings*, 2020, 2565, pp. 116–129.
40. Tryhuba A., Kondysiuk I., Tryhuba I., Boiarchuk O., Tatomyr A., Intellectual information system for formation of portfolio projects of motor transport enterprises, *CEUR Workshop Proceedings*, 2022, 3109, pp. 44–52.

41. Tryhuba A., Kotenko V. Intelligent information system for resource planning in grain crops delivery projects on the basis of machine learning. IEEE 18th International Conference on Computer Science and Information Technologies (CSIT) 2023. P. 1-4.

42. Tryhuba A., Koval N., Tryhuba I., Boiarchuk O., Application of Sarima Models in Information Systems Forecasting Seasonal Volumes of Food Raw Materials of Procurement on the Territory of Communities. CEUR Workshop Proceedings, 2022, 3295, pp. 64–75.

43. Tryhuba A., Padyuka R., Tymochko V., Lub P., Mathematical model for forecasting product losses in crop production projects. CEUR Workshop Proceedings, 2022, 3109, pp. 25–31.

44. Tryhuba A., Tryhuba I., Ftoma O., Boyarchuk O., Method of quantitative evaluation of the risk of benefits for investors of fodder-producing cooperatives. International Scientific and Technical Conference on Computer Sciences and Information Technologies, 2019, 3, pp. 55–58, 8929788.

45. Tryhuba I., Hutsol T., Tryhuba A., Cieszewska A., Kovalenko N., Mudryk K., Glowacki S., Bryś A., Tulej W., Sojak M. Analysis and Quantification of the Generation of Organic Waste by Households for Energy Production. Sustainability 2023 , 15 (22), 15922; <https://doi.org/10.3390/su152215922>

46. Tryhuba I., Tryhuba A., Hutsol T., Cieszewska A., Andrushkiv O., Glowacki S., Bryś A., Slobodian S., Tulej W., Sojak M. Prediction of Biogas Production Volumes from Household Organic Waste Based on Machine Learning. Energies 2024, 17(7), 1786; <https://doi.org/10.3390/en17071786>

47. Tryhuba I., Tryhuba A., Hutsol T., Lopushniak V., Cieszewska A., Andrushkiv O., Barabasz W., Pikulicka A., Kowalczyk Z., Vasyuk V. European Green Deal: Justification of the Relationships between the Functional Indicators of Bioenergy Production Systems Using Organic Residential Waste Based on the Analysis of the State of Theory and Practice. Energies. Енергія 2024 , 17 (6), 1461; <https://doi.org/10.3390/en17061461>

48. US EPA, Facts and Figures about Materials, Waste and Recycling, EPA, Washington, DC, USA, 2018. URL: <https://www.epa.gov/facts-and-figures-about-materials-waste-and-recycling/advancing-sustainable-materials-management-0>
49. Williams P. T., Waste Treatment and Disposal, Wiley, West Sussex, UK, 2005.

# Додатки



## Додаток А

### Фрагмент програмного коду створення модуля для управління відходами

```

import cv2
import numpy as np
import tensorflow as tf
from gpiozero import Servo
from time import sleep

# Ініціалізація камери
cap = cv2.VideoCapture(0)

# Завантаження попередньо навченої моделі CNN
model = tf.keras.models.load_model('path_to_your_model.h5')

# Ініціалізація серводвигуна
servo = Servo(17) # Приклад для GPIO 17

def preprocess_image(image):
    """
    Функція для попередньої обробки зображення
    """
    image = cv2.resize(image, (224, 224)) # Зміна розміру відповідно до вимог
    моделі
    image = image / 255.0 # Нормалізація зображення
    image = np.expand_dims(image, axis=0) # Додавання осі batch
    return image

def classify_waste(image):
    """
    Функція для класифікації зображення за допомогою моделі CNN
    """
    processed_image = preprocess_image(image)
    prediction = model.predict(processed_image)
    return np.argmax(prediction) # Повернення класу з найвищою ймовірністю

def control_servo(class_label):
    """
    Функція для керування серводвигуном відповідно до результатів класифікації
    """
    if class_label == 0: # Приклад для органічних відходів
        servo.min() # Позиція серводвигуна для органічних відходів
    elif class_label == 1: # Приклад для неорганічних відходів
        servo.max() # Позиція серводвигуна для неорганічних відходів
    sleep(1) # Затримка для завершення руху серводвигуна
    servo.mid() # Повернення серводвигуна в початкову позицію

try:
    while True:
        # Захоплення зображення з камери
        ret, frame = cap.read()
        if not ret:
            break

        # Класифікація зображення
        class_label = classify_waste(frame)

        # Керування серводвигуном на основі результатів класифікації
        control_servo(class_label)

        # Відображення захопленого зображення
        cv2.imshow('Waste Classification', frame)

```

```

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

finally:
    # Звільнення ресурсів
    cap.release()
    cv2.destroyAllWindows()

```

### Фрагмент програмного коду створення HTML документа

```

<!DOCTYPE html>
<html lang=«en»>
<head>
  <meta charset=«UTF-8»>
  <meta name=«viewport» content=«width=device-width, initial-scale=1.0»>
  <title>Waste Management Dashboard</title>
  <link rel=«stylesheet» href=«styles.css»>
</head>
<body>
  <div class=«container»>
    <h1>Waste Management Dashboard</h1>
    <div class=«data»>
      <div class=«data-item»>
        <h2>Weight of Waste</h2>
        <p id=«weightValue»>0 kg</p>
      </div>
      <div class=«data-item»>
        <h2>Container Level</h2>
        <p id=«levelValue»>0 %</p>
      </div>
    </div>
  </div>
  <script src=«script.js»></script>
</body>
</html>

```

### Фрагмент програмного коду CSS для забезпечення адаптивності інтерфейсу

```

body {
  font-family: Arial, sans-serif;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  margin: 0;
  background-color: #3498db; /* Яскравий синій фон */
}

.container {
  text-align: center;
  background: rgba(255, 255, 255, 0.8); /* Прозорий білий фон */
  padding: 2em;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  border-radius: 10px;
}

```

```

h1 {
  margin-bottom: 1em;
  color: #2c3e50; /* Темно-синій колір тексту */
}

.container-buttons {
  margin-bottom: 1em;
}

.container-button {
  background-color: #4CAF50;
  border: none;
  color: white;
  padding: 15px 30px; /* Більший розмір кнопки */
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 18px; /* Збільшений розмір шрифту */
  margin: 4px 2px;
  cursor: pointer;
  border-radius: 10px;
  transition: background-color 0.3s ease; /* Плавна анімація зміни кольору при
наведенні */
}

.container-button:hover {
  background-color: #45a049;
}

.container-button.active {
  background-color: #2980b9; /* Колір підсвічення активного контейнера */
}

.data {
  display: flex;
  justify-content: space-around;
  margin-top: 1em;
}

.data-item {
  background: #ecf0f1; /* Світло-сірий колір поля */
  padding: 1.5em;
  border-radius: 10px;
  width: 200px; /* Більший розмір поля */
}

p {
  font-size: 2em;
  margin: 0;
}

#recommendation {
  margin-top: 1em;
  font-size: 1.5em;
  color: #e74c3c; /* Червоний колір для підказок */
}

```

## Фрагмент програмного коду створення нейронної мережі CNN

```

import tensorflow as tf
from tensorflow.keras.applications import ResNet34

```

```

from tensorflow.keras.layers import Dense, GlobalAveragePooling2D
from tensorflow.keras.models import Model
from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Завантаження базової моделі ResNet-34 з попередньо навченими вагами
base_model = ResNet34(weights='imagenet', include_top=False, input_shape=(224,
224, 3))

# Додавання власних шарів для задачі класифікації відходів
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(512, activation='relu')(x) # Додатковий повнозв'язний шар
predictions = Dense(3, activation='softmax')(x) # Вихідний шар з 3 класами
(наприклад: органічні, неорганічні, інші)

# Створення повної моделі
model = Model(inputs=base_model.input, outputs=predictions)

# Замороження ваг базової моделі
for layer in base_model.layers:
    layer.trainable = False

# Компіляція моделі
model.compile(optimizer='adam', loss='categorical_crossentropy',
metrics=['accuracy'])

# Підготовка даних
train_datagen = ImageDataGenerator(
    rescale=1./255,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    validation_split=0.2
)

train_generator = train_datagen.flow_from_directory(
    'path_to_your_data/train', # Шлях до тренувальних даних
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical',
    subset='training'
)

validation_generator = train_datagen.flow_from_directory(
    'path_to_your_data/train', # Шлях до тренувальних даних
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical',
    subset='validation'
)

# Тренування моделі
model.fit(
    train_generator,
    steps_per_epoch=train_generator.samples // train_generator.batch_size,
    validation_data=validation_generator,
    validation_steps=validation_generator.samples //
validation_generator.batch_size,
    epochs=10
)

# Збереження моделі
model.save('waste_classification_model.h5')

```