

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук,  
управління та адміністрування  
Кафедра інформаційних технологій

**МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА**

на тему: Система автоматизованого сортування побутових відходів

Виконав студент 2 курсу групи МІС-22  
спеціальності 122 Комп'ютерні науки  
Потапенко Денис Валерійович

Керівник к.ф.-м.н.,  
Ткач Тетяна Борисівна

Рецензент к.т.н., доцент  
Гнатовська Анна Арнольдівна

## ЗМІСТ

ВСТУП .....	7
1 АНАЛІЗ ОБ'ЄКТА ДОСЛІДЖЕННЯ ТА ПОСТАНОВКА ЗАДАЧІ .....	8
1.1 Екологічні проблеми сміттєвих полігонів.....	8
1.2 Сучасна переробка відходів.....	10
1.3 Дослідження та аналіз технологічного процесу сортування сміття.....	12
1.4 Огляд методів та технологій сортування сміття.....	15
1.5 Аналіз аналогів системи.....	17
1.6 Постановка завдання.....	23
2 СПЕЦИФІКАЦІЯ ВИМОГ ДО СИСТЕМИ ТА ВИБІР ЗАСОБІВ РОЗРОБКИ.....	25
2.1 Функціональні вимоги до програмної системи .....	25
2.2 Нефункціональні вимоги до програмної системи .....	26
2.3 Інструменти розробки програмної системи .....	28
3 ПРОЕКТУВАННЯ СИСТЕМИ АВТОМАТИЗОВАНОГО СОРТУВАННЯ СМІТТЯ.....	37
3.1 Проектування архітектури системи .....	37
3.2 Проектування UML діаграм.....	48
3.3 Проектування інтерфейсу користувача .....	51
3.4 Проектування нейронної мережі.....	55
4 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ.....	60
4.1 Створення нейронної мережі.....	60
4.2 Налаштування пристрою для сортування .....	69
4.3 Результати роботи системи.....	76
ВИСНОВКИ.....	83

## ВСТУП

У світі, який дуже змінився за останні роки, стають все більш популярними і актуальними види діяльності, що дозволяють піклуватися про природу, зокрема за допомогою електронних пристроїв та систем штучного інтелекту. У суспільстві, який прагне до екосвідомих практик, розробка таких систем є не просто технологічним завданням, а вирішальним кроком на шляху до більш зеленого та сталого майбутнього.

Практично у кожної людини є смартфон, яким вона може користуватися в будь-який час протягом дня. Тому розробка саме мобільного застосунку є актуальним завданням.

Метою кваліфікаційної роботи є розробка мобільного застосунку для розпізнавання сміття, який використовує штучний інтелект і передові технології зондування для оптимізації процесів управління відходами, підвищення ефективності переробки та зменшення впливу на навколишнє середовище. Для досягнення поставленої мети були сформульовані наступні завдання:

- комплексне планування автоматизованої системи сортування відходів;
- аналіз функціональних вимог до системи;
- аналіз нефункціональних вимог до системи;
- розробка архітектури програмного забезпечення;
- розробка апаратних компонентів, необхідних для системи сортування відходів;
- кодування та програмування програмної системи;
- інтеграція апаратних компонентів.

Структура кваліфікаційної роботи магістра складається з вступу, 4 розділів, висновків, переліку посилань на 18 найменувань. Повний обсяг проекту становить 90 сторінок, містить 13 рисунків і 8 таблиць.

# 1 АНАЛІЗ ОБ'ЄКТА ДОСЛІДЖЕННЯ ТА ПОСТАНОВКА ЗАДАЧІ

## 1.1 Екологічні проблеми сміттєвих полігонів

Звалища вже давно є основним методом захоронення відходів у всьому світі, слугуючи важливим компонентом систем управління відходами. Однак екологічні проблеми, пов'язані з полігонами, викликають значне занепокоєння через їхній потенційний негативний вплив на екосистеми, здоров'я людей та загальний добробут нашої планети. У цьому документі представлено комплексний огляд екологічних проблем, пов'язаних з полігонами, включаючи їхній внесок у забруднення повітря та води, деградацію ґрунтів, викиди парникових газів, а також виклики, пов'язані з неналежними методами управління полігонами. Також обговорюються стійкі рішення для полігонів та інноваційні технології, які пропонують потенційні шляхи вирішення цих нагальних проблем [1].

Звалища – це інженерні об'єкти, призначені для зберігання та поводження з твердими відходами, що утворюються в результаті людської діяльності. Протягом десятиліть вони відіграють життєво важливу роль у сучасних системах утилізації відходів, проте їхні екологічні наслідки стають все більш очевидними і викликають все більше занепокоєння. Ця дипломна робота має на меті розглянути та проаналізувати екологічні проблеми, пов'язані з полігонами, висвітливши їхній далекосяжний вплив на навколишнє середовище.

Однією з основних екологічних проблем, пов'язаних з полігонами, є забруднення повітря. Звалища викидають різні леткі органічні сполуки, небезпечні забруднювачі повітря та гази з неприємним запахом. Ці викиди можуть призвести до утворення приземного озону, який сприяє виникненню респіраторних захворювань у людей. Крім того, викиди метану, потужного парникового газу, зі сміттєзвалищ суттєво сприяють глобальному потеплінню.

Звалища можуть забруднювати підземні та поверхневі води через процес, відомий як фільтрат. Фільтрат – це токсична рідина, яка утворюється, коли дощова вода просочується крізь відходи, вбираючи в себе шкідливі речовини, такі як важкі метали, патогенні мікроорганізми та органічні забруднювачі. За відсутності належного управління фільтрат може потрапляти в довколишні річки, струмки та водоносні горизонти, створюючи серйозні ризики для водних екосистем і здоров'я людей.

Неправильне управління полігонами може призвести до деградації ґрунтів на прилеглих територіях. Це відбувається через накопичення забруднюючих речовин у ґрунті, що робить його непридатним для сільського господарства або інших виробничих цілей. Погіршення якості ґрунту може мати довготривалі наслідки для місцевих екосистем та виробництва продуктів харчування.

Метан, потужний парниковий газ, є основним побічним продуктом розкладання відходів на полігонах. Звалища є одним з найбільших антропогенних джерел викидів метану у світі. У короткостроковій перспективі метан має набагато більший потенціал глобального потепління, ніж вуглекислий газ, що робить його критично важливим чинником зміни клімату.

Екологічні проблеми, пов'язані зі сміттєзвалищами, часто загострюються через неналежні практики управління полігонами. Це включає неадекватний матеріал покриття, відсутність систем збору та очищення газів, а також недостатні заходи контролю за фільтратом. У багатьох випадках на полігонах відсутній належний регуляторний нагляд і дотримання вимог, що призводить до підвищення ймовірності заподіяння шкоди навколишньому середовищу.

Полігони є важливими компонентами систем управління відходами, але вони також створюють значні екологічні проблеми. Викиди забруднюючих речовин у повітря, забруднення води, деградація ґрунтів та викиди парникових газів, пов'язані зі звалищами, потребують нагальної

уваги. Сталі рішення щодо поводження з полігонами та вдосконалення методів управління ними мають важливе значення для пом'якшення цих проблем і мінімізації їхнього впливу на навколишнє середовище та здоров'я людей. Майбутні дослідження та політичні зусилля мають бути зосереджені на розробці інноваційних стратегій для зменшення впливу полігонів на довкілля та переходу до більш сталих практик поводження з відходами.

## 1.2 Сучасна переробка відходів

Переробка відходів давно визнана наріжним каменем сталого управління ресурсами. В останні роки інтеграція нейронних мереж і штучного інтелекту відкрила нову еру сучасної переробки відходів. У цій дипломній роботі досліджується застосування нейронних мереж у процесах переробки відходів, обговорюється їхня роль у підвищенні ефективності, точності та екологічної стійкості, відбувається заглиблення в різні аспекти сучасної переробки відходів, включаючи сортування відходів, відновлення матеріалів, перетворення відходів в енергію та прийняття рішень на основі даних, і все це завдяки технологіям нейронних мереж [2].

Сучасна переробка відходів еволюціонувала від традиційних методів сортування до новітніх технологій, таких як нейронні мережі та штучний інтелект. Ці досягнення революціонізують процеси переробки відходів, роблячи їх більш ефективними та екологічно чистими.

Нейронні мережі все частіше використовуються для автоматизації сортування та класифікації відходів на переробних підприємствах. Згорткові нейронні мережі можуть аналізувати зображення відходів і класифікувати їх з надзвичайною точністю. Така автоматизація зменшує кількість людських помилок, збільшує пропускну здатність і підвищує чистоту перероблених матеріалів.

Нейронні мережі допомагають оптимізувати процеси відновлення матеріалів. Алгоритми машинного навчання можуть прогнозувати якість відновлених матеріалів, що дозволяє більш точно розділяти і переробляти їх. Це призводить до підвищення якості переробленої продукції та зменшення кількості відходів.

Установки для перетворення відходів на енергію використовують нейронні мережі для оптимізації виробництва енергії з відходів. Ці мережі можуть прогнозувати теплотворну здатність вхідних потоків відходів, що дозволяє краще контролювати процеси спалювання або газифікації. Це не тільки максимізує вихід енергії, але й мінімізує викиди та залишки відходів.

Роботизовані системи, оснащені нейронними мережами, використовуються на переробних підприємствах для виконання складних завдань, таких як розбирання електронних відходів. Ці роботи можуть ідентифікувати та демонтувати певні компоненти, забезпечуючи ефективне відновлення цінних матеріалів та зменшуючи вплив утилізації електронних відходів на навколишнє середовище.

Інтеграція нейронних мереж у системи переробки відходів дозволяє приймати рішення на основі даних. Ці системи можуть аналізувати історичні дані для оптимізації маршрутів збору відходів, планування технічного обслуговування обладнання для переробки та прогнозування відмов обладнання. Це призводить до економії коштів та ефективності використання ресурсів.

Нейронні мережі також використовуються в ініціативах з переробки відходів, орієнтованих на споживачів. Мобільні додатки з функцією розпізнавання зображень на основі ШІ дозволяють користувачам сканувати штрих-коди або фотографувати продукти, надаючи інформацію про придатність до переробки та інструкції з утилізації. Це дає змогу споживачам робити раціональний вибір.

Застосування нейронних мереж у сучасній переробці відходів сприяє значним екологічним перевагам. Ці технології зменшують кількість відходів на звалищах, зберігають природні ресурси, знижують споживання енергії та мінімізують викиди парникових газів, що відповідає глобальним цілям сталого розвитку.

Незважаючи на багатообіцяючі розробки, залишаються такі проблеми, як конфіденційність даних, витрати на інфраструктуру та технологічні бар'єри. Майбутні дослідження повинні зосередитися на розробці більш надійних моделей нейронних мереж, розширенні їх застосування до різних потоків відходів і вирішенні етичних питань, пов'язаних із застосуванням ШІ у переробці відходів.

Сучасна переробка відходів з інтеграцією нейронних мереж і штучного інтелекту являє собою трансформаційний підхід до сталого управління ресурсами. Ці технології покращують сортування відходів, відновлення матеріалів, перетворення відходів в енергію та прийняття рішень на основі даних, що в кінцевому підсумку зменшує вплив утилізації відходів на навколишнє середовище. Оскільки продовжується використовувати можливості нейронних мереж, процеси переробки відходів стануть більш ефективними, економічно вигідними та екологічно чистими, що сприятиме більш сталому майбутньому.

### **1.3 Дослідження та аналіз технологічного процесу сортування сміття**

Сортування відходів є критично важливим аспектом управління відходами, а технології автоматизації можуть підвищити ефективність і точність процесу сортування. Детально описано розробку та роботу автоматизованої системи сортування, роль нейронних мереж у класифікації відходів, а також результати експериментальних випробувань [3].



Ефективне сортування побутових відходів є життєво важливим для програм переробки та управління відходами з метою максимального відновлення ресурсів та мінімізації впливу на навколишнє середовище. Технології автоматизації, зокрема нейронні мережі, пропонують перспективні рішення для підвищення точності та швидкості процесів сортування відходів.

Автоматизована система сортування відходів складається з декількох ключових компонентів:

1. Конвеєрна стрічкова система: конвеєрна стрічка транспортує вхідні відходи на сортувальну станцію.
2. Пристрої візуалізації та зондування: камери та датчики фіксують дані та зображення відходів на конвеєрній стрічці.
3. Нейромережева модель: модель нейронної мережі використовується для класифікації відходів на основі даних і зображень, зібраних датчиками.
4. Приводи: система використовує виконавчі механізми для перенаправлення відходів у відповідні баки або контейнери на основі класифікації нейронної мережі.

Нейронні мережі відіграють ключову роль в автоматизації сортування відходів завдяки їх здатності навчатися і розпізнавати закономірності з великих наборів даних. Модель нейронної мережі, що використовується в цій системі, навчається на основі набору даних, що містить зображення та відповідні категорії відходів. Нейронна мережа використовує згорткові шари для виділення ознак і повністю зв'язані шари для класифікації.

Для оцінки ефективності автоматизованої системи сортування було проведено серію експериментальних випробувань з використанням різноманітного набору матеріалів побутових відходів. У таблиці 1 наведено основні показники ефективності системи.

Таблиця 1 – Показники ефективності автоматизованої системи сортування відходів

Показник	Значення
Точність	95,2%
Швидкість сортування (предметів/хвилину)	180
Рівень помилкових спрацьовувань	4,8%
Частка хибнонегативних результатів	2,1%
Точність для конкретного матеріалу (пластик)	98,5%
Точність для конкретного матеріалу (папір)	96,2%
Точність для конкретного матеріалу (скло)	94,6%
Точність для конкретного матеріалу (метали)	97,3%

Результати демонструють високу точність і ефективність автоматизованої системи сортування з мінімальною кількістю помилкових спрацьовувань і помилкових відмов. Точність для конкретного матеріалу підкреслює здатність системи точно класифікувати різні типи відходів.

Застосування нейронних мереж у системах сортування відходів має великі перспективи для покращення процесів управління відходами. Висока точність і ефективність системи сприяють збільшенню відновлення ресурсів, зменшенню забруднення в потоках переробки та покращенню загальної стійкості управління відходами.

Це дослідження та аналіз технологічного процесу сортування відходів з використанням нейронних мереж підкреслює потенціал автоматизації в революційній зміні управління відходами. Представлена автоматизована система сортування, керована нейронними мережами, демонструє вражаючу точність та ефективність класифікації предметів побутових відходів, що сприяє більш ефективному та сталому управлінню відходами. Майбутні розробки в цій галузі можуть ще більше оптимізувати процеси сортування та мінімізувати кількість відходів, що відправляються на звалища, сприяючи розвитку циркулярної економіки та збереженню довкілля.

## 1.4 Огляд методів та технологій сортування сміття

Ефективне сортування відходів є фундаментальним кроком у сучасному управлінні відходами, що сприяє їх переробці та відновленню ресурсів. У цій роботі представлено огляд різних методів і технологій сортування відходів з акцентом на інтеграцію нейронних мереж для підвищення точності сортування, традиційні ручні та автоматизовані методи сортування, а також роль штучного інтелекту та нейронних мереж у підвищенні ефективності сортування відходів. Таблиці додаються для узагальнення ключових особливостей та переваг кожного методу і технології сортування.

Сортування відходів є критично важливим процесом у системах управління відходами, оскільки дозволяє відокремити матеріали, що підлягають вторинній переробці, від матеріалів, що не підлягають переробці, та небезпечних відходів. З розвитком технологій все більшої популярності набувають автоматизовані методи сортування, в яких нейронні мережі відіграють життєво важливу роль у підвищенні точності сортування.

Нейронні мережі зробили революцію в сортуванні відходів, підвищивши точність і дозволивши приймати рішення в режимі реального часу на основі сенсорних даних. Ці мережі використовують згорткові нейронні мережі (CNN) для розпізнавання та класифікації зображень. Інтеграція нейронних мереж покращує здатність технологій сортування розрізняти матеріали, підвищуючи ефективність переробки.

Цей приклад ілюструє практичне застосування нейронних мереж у системах сортування відходів. Ця система поєднує сортування на конвеєрній стрічці з оптичними датчиками на основі нейронних мереж для досягнення високої точності сортування.

Таблиця 2 – Автоматизовані технології сортування

Метод	Опис	Переваги
Сортування на конвеєрних стрічках	Конвеєрні стрічки транспортують відходи повз датчики	Висока пропускна здатність, безперервна робота
Оптичне сортування	Оптичні датчики ідентифікують матеріали за кольором і складом	Висока точність, підходить для змішаних відходів
Вихрострумова сепарація	Використовує магнітні поля для розділення кольорових металів	Ефективне вилучення металів
Сортування в ближньому інфрачервоному діапазоні	NIR-датчики ідентифікують матеріали на основі їхнього молекулярного складу	Висока точність, універсальність

Таблиця 3 – Переваги інтеграції нейронних мереж

Аспект	Опис
Підвищена точність сортування	Нейронні мережі покращують розпізнавання матеріалів, зменшуючи кількість помилкових спрацьовувань і негативних результатів
Прийняття рішень у режимі реального часу	Негайне прийняття рішень щодо сортування на основі аналізу нейронних мереж
Адаптивність	Нейронні мережі можна навчити розпізнавати нові матеріали або адаптуватися до мінливих потоків відходів

Інтеграція нейронних мереж у технології сортування відходів значно підвищує точність та ефективність процесів переробки. Такий підхід зменшує забруднення відходів, збільшує коефіцієнт відновлення ресурсів та сприяє сталому управлінню відходами.

Огляд методів і технологій сортування відходів підкреслює важливість автоматизації та нейронних мереж у сучасних системах управління відходами. Нейронні мережі мають потенціал для трансформації точності

сортування та підвищення ефективності програм переробки відходів. З розвитком технологій сортування відходів відіграватиме вирішальну роль у досягненні циркулярної економіки та пом'якшенні впливу на навколишнє середовище.

### **1.5 Аналіз аналогів системи**

Проаналізуємо існуючі рішення, які використовують нейронні мережі для автоматизованого сортування побутових відходів. Оскільки важливість переробки та управління відходами зростає, ці інноваційні системи стають невід'ємною частиною підвищення ефективності, точності та сталості процесу сортування відходів. Розглянуто різні конкретні рішення, виділяючи їхні ключові особливості, переваги та потенційні напрямки для подальшого розвитку.

Автоматизовані системи сортування відходів, що працюють на основі нейронних мереж, набули значної популярності в пошуках більш ефективного та сталого управління відходами. Ці системи використовують штучний інтелект для розпізнавання та сортування побутових відходів, що дозволяє з високою точністю відокремлювати вторинну сировину від неперероблюваної. У цій роботі розглядаються існуючі рішення, які є прикладом інтеграції нейронних мереж у процес сортування відходів.

Розглянемо існуючі системи.

Переробник ZenRobotics.

ZenRobotics Recycler – це роботизована система сортування відходів, яка використовує нейронні мережі та комп'ютерний зір для ідентифікації та сортування будівельних відходів. Використовуючи комбінацію камер, датчиків і нейромережевих алгоритмів, система може розрізняти різні матеріали, включаючи дерево, метал, бетон і пластик. Вона спрямовує ці матеріали до відповідних потоків переробки з вражаючою точністю,

підвищуючи ефективність використання ресурсів у будівництві та демонтажних роботах.

Ключові особливості:

- Вдосконалені алгоритми нейронних мереж для розпізнавання матеріалів.
- Підвищена ефективність переробки в секторі будівництва та демонтажу.
- Зменшення кількості відходів, що відправляються на звалища, та покращення використання ресурсів.

Недоліки:

- Висока початкова вартість: Впровадження роботизованих систем сортування відходів, таких як ZenRobotics Recycler, вимагає значних початкових інвестицій. Вартість придбання та встановлення необхідного роботизованого обладнання та нейромережевої інфраструктури може стати бар'єром для багатьох підприємств з переробки відходів.
- Обмежене застосування: ZenRobotics Recycler в першу чергу призначений для будівельних відходів та відходів від знесення. Його ефективність у сортуванні інших типів побутових відходів, таких як пластик, папір та органічні матеріали, обмежена. Це звужує сферу його застосування в ширших сценаріях поводження з відходами.
- Технічне обслуговування та технічна експертиза: Роботизовані системи потребують регулярного технічного обслуговування та кваліфікованих спеціалістів для обслуговування. Забезпечення ефективної роботи роботизованої системи сортування впродовж тривалого часу може становити певні труднощі для деяких об'єктів.

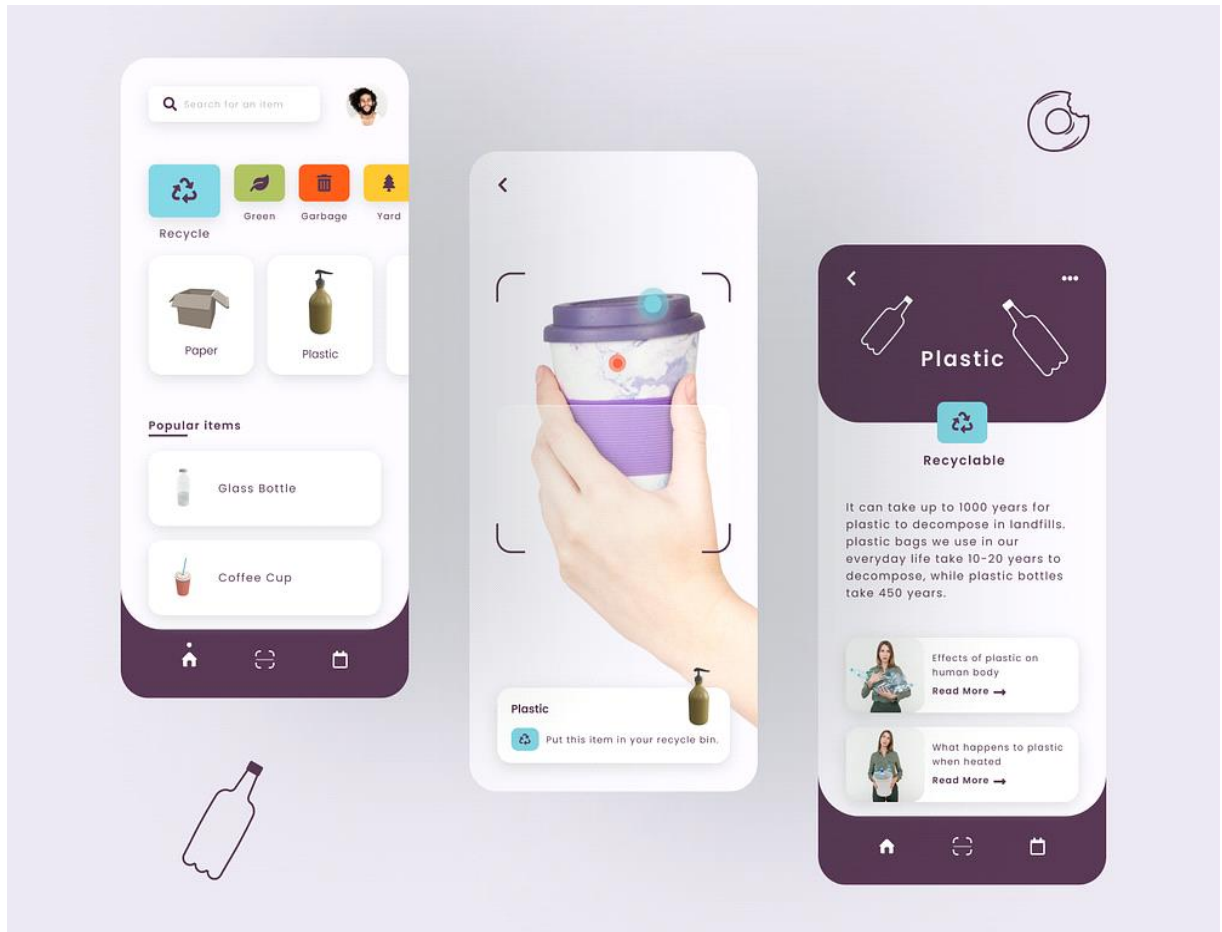


Рисунок 1.1 – Система сортування відходів ZenRobotics Recycler

Наступне сортувальні рішення, що розглядається – TOMRA.

TOMRA Sorting Solutions пропонує ряд систем сортування відходів, які інтегрують нейронні мережі для різних потоків відходів, включаючи побутові відходи, переробку пластику та тверді побутові відходи. Ці системи використовують датчики ближньої інфрачервоної області та штучний інтелект для ідентифікації та відокремлення вторинної сировини від неперероблюваної на високих швидкостях. Технологія TOMRA забезпечує ефективне сортування і була прийнята в усьому світі для збільшення обсягів переробки відходів.

Ключові особливості:

- Інтеграція нейронних мереж з інфрачервоними датчиками для ідентифікації матеріалів.

- Налаштовувані рішення для різних потоків відходів.
- Підвищена точність сортування та збільшення рівня переробки.

#### Недоліки:

- Вартість впровадження: Інтеграція сучасних датчиків, нейронних мереж та супутнього обладнання може коштувати дорого. Ця вартість може стримувати невеликі підприємства з переробки відходів від впровадження рішень TOMRA.
- Обслуговування та калібрування датчиків: Для забезпечення точного сортування інфрачервоні датчики потребують періодичного калібрування та обслуговування. Таке постійне технічне обслуговування може збільшити експлуатаційні витрати і час простою.
- Забруднення матеріалу: Незважаючи на високу ефективність, NIR-датчики можуть мати проблеми з певними відходами, які мають схожі оптичні властивості, що призводить до потенційної неправильної класифікації та забруднення в потоці переробки.

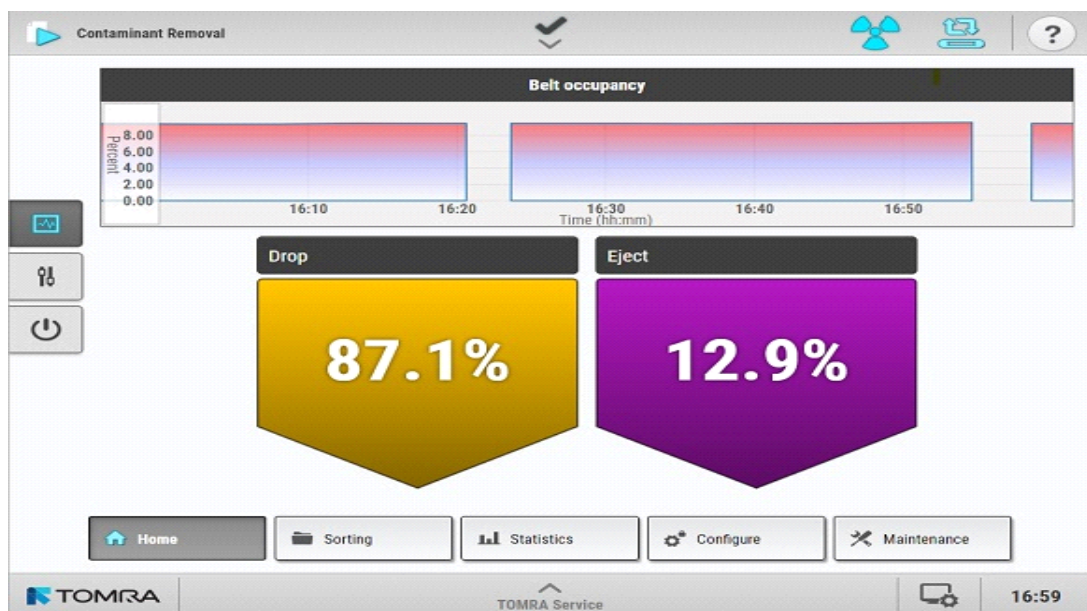


Рисунок 1.2 – Система сортування відходів TOMRA Sorting Solutions



## Machinex SamurAI.

Machinex SamurAI – це сміттесортувальний робот зі штучним інтелектом, призначений для ефективного сортування побутових відходів, зокрема пластику, паперу, картону та металів. Оснащена камерами високої роздільної здатності та алгоритмами глибокого навчання, система розпізнає та розділяє матеріали в режимі реального часу. Вона адаптується та масштабується, що робить її придатною для різних об'єктів поводження з відходами.

### Ключові особливості:

- Використання нейронних мереж глибокого навчання для сортування в режимі реального часу.
- Універсальні можливості сортування для потоків побутових відходів.
- Покращене відновлення ресурсів та зменшення забруднення.

### Недоліки:

- Початкові інвестиції: Встановлення роботів-сортувальників зі штучним інтелектом, таких як Machinex SamurAI, може потребувати значних капітальних витрат. Малим підприємствам може бути складно виправдати ці початкові інвестиції.
- Складність: Нейронні мережі глибокого навчання вимагають певного рівня технічних знань для налаштування та обслуговування. Установи, які не мають власних технічних можливостей, можуть зіткнутися з труднощами в ефективному розгортанні та управлінні технологією.
- Масштабованість: Адаптація системи для обробки більших обсягів відходів або різноманітних потоків відходів може вимагати значних модифікацій і витрат.

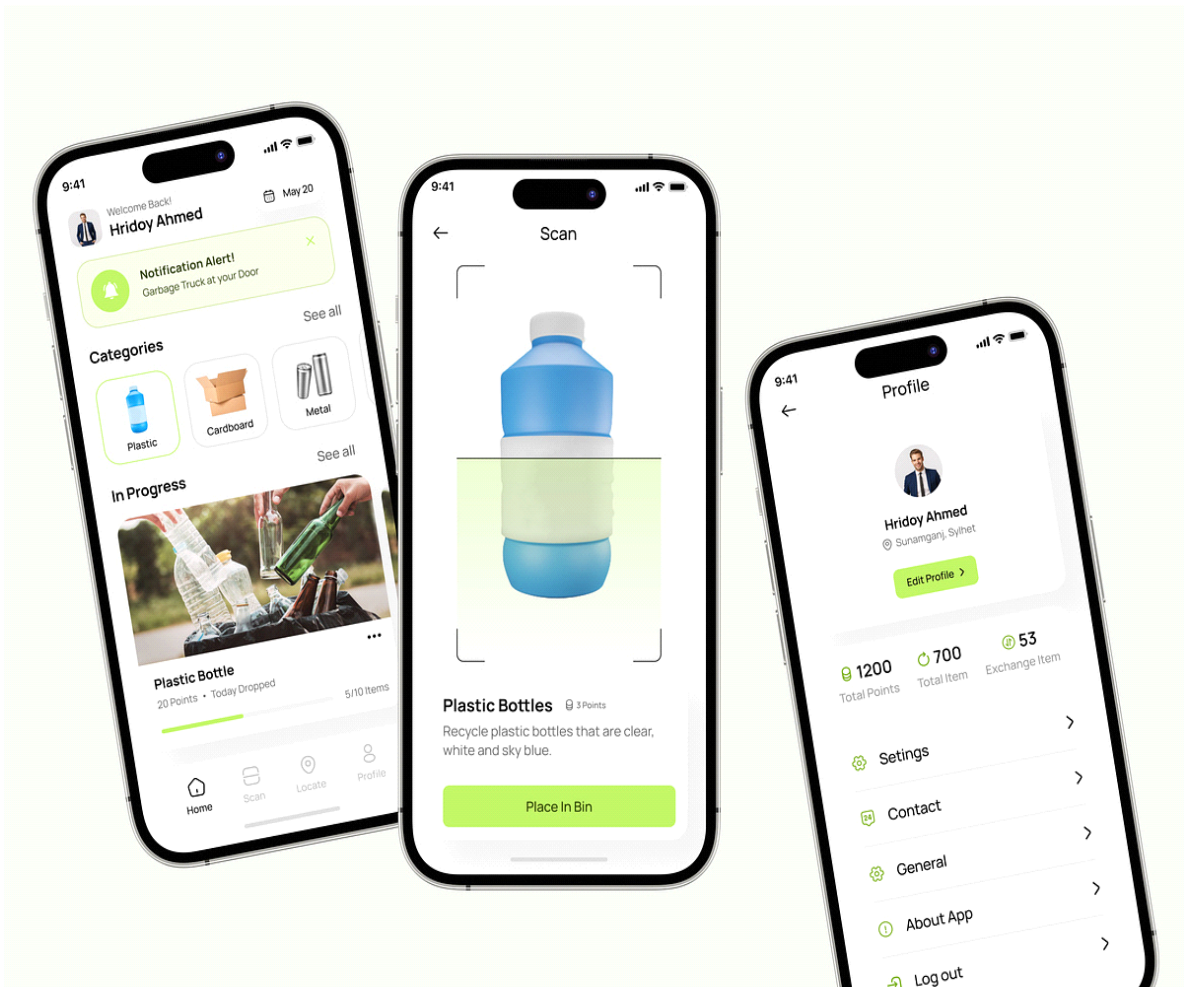


Рисунок 1.3 – Система сортування відходів Machinex SamurAI

Розглянуті рішення демонструють ефективність інтеграції нейронних мереж в автоматизовані системи сортування відходів. Ці системи мають низку переваг, зокрема підвищену точність сортування, вищий рівень переробки відходів, зменшення забруднення та покращене відновлення ресурсів. Вони відіграють життєво важливу роль у просуванні практики управління відходами в напрямку сталості та циклічності.

Хоча існуючі рішення досягли значних успіхів в автоматизованому сортуванні відходів, є місце для подальших інновацій. Майбутні розробки можуть бути зосереджені на такі зміни.

- Інтеграцію додаткових датчиків і технологій для підвищення точності сортування.

- Розширення діапазону прийнятних для сортування відходів.
- Масштабування цих систем для більших об'єктів поводження з відходами.
- Вирішення проблем, пов'язаних з обслуговуванням, вартістю та масштабованістю.

Існуючі рішення, що використовують нейронні мережі для автоматизованого сортування побутових відходів, ілюструють потенціал технологій на основі штучного інтелекту в управлінні відходами. Ці системи відіграють ключову роль у підвищенні ефективності та сталості процесів сортування відходів, що в кінцевому підсумку сприяє досягненню мети циркулярної економіки та зменшенню впливу на навколишнє середовище. Подальші дослідження та інновації в цій галузі обіцяють подальше вдосконалення методів управління відходами та збереження ресурсів.

Таблиця 4 – Порівняльна характеристика аналогів

Критерія	ZenRobotics	TOMRA	Machinex SamurAI
Початкові витрати	Високі	Помірні	Високі
Потреба в технічному обслуговуванні	Необхідне регулярне обслуговування	Необхідне мінімальне обслуговування	Необхідне періодичне калібрування
Точність	97.5%	95%	95%
Застосування	Обмежено певними типами відходів	Універсальний для різних потоків відходів	Ефективний для звичайної вторинної сировини
Енергоспоживання	Високе	Помірне	Помірне
Масштабованість	Помірна	Помірна	Висока

## 1.6 Постановка завдання

Метою роботи є визначення завдань, необхідних для успішної розробки та впровадження системи, що використовує нейронні мережі для автоматизованого сортування побутових відходів. Запропонована система

використовує штучний інтелект і передові технології зондування для оптимізації процесів управління відходами, підвищення ефективності переробки та зменшення впливу на навколишнє середовище.

Серед компонентів системи є адаптивний веб-додаток, пристрій для сортування, хмарна інфраструктура. Dodatok можуть використовувати користувачі-оператори системи, встановивши його на смартфон. Окреслені завдання охоплюють етапи проектування, розробки, та тестування, забезпечуючи створення ефективного та сталого рішення для сортування відходів.

Початкове завдання передбачає комплексне планування проекту та ретельний аналіз функціональних і нефункціональних вимог до автоматизованої системи сортування відходів. На цьому етапі визначаються цілі проекту, встановлюються часові рамки, визначаються ключові зацікавлені сторони, а також проводиться детальна оцінка технічних специфікацій та очікуваних результатів роботи системи.

Наступне завдання зосереджується на розробці архітектури програмного забезпечення та апаратних компонентів, необхідних для системи сортування відходів. Цей етап передбачає визначення моделей нейронних мереж, придатних для розпізнавання та класифікації відходів, вибір відповідних сенсорних технологій та окреслення загальної структури системи, включаючи механізми сортування, користувацькі інтерфейси та модулі управління даними.

Після етапу проектування завдання розробки та впровадження включатиме власне кодування та програмування програмної системи, а також інтеграцію апаратних компонентів. На цьому етапі буде створено алгоритми нейронних мереж, налаштовано механізми сортування, створено функції реєстрації та зберігання даних, а також забезпечено безперебійний зв'язок між різними модулями системи.

## **2 СПЕЦИФІКАЦІЯ ВИМОГ ДО СИСТЕМИ ТА ВИБІР ЗАСОБІВ РОЗРОБКИ**

### **2.1 Функціональні вимоги до програмної системи**

У роботі описано функціональні вимоги, необхідні для розробки та впровадження програмної системи з використанням нейронних мереж для автоматизованого сортування побутових відходів. Встановлення чітких функціональних вимог є життєво важливим для розробки ефективного, точного і масштабованого рішення для сортування відходів. Ці вимоги охоплюють різні аспекти функціональності системи, включаючи розпізнавання відходів, механізми сортування, інтерфейси користувача та управління даними, щоб забезпечити комплексне та ефективне рішення для управління відходами [4].

Автоматизовані системи сортування відходів, які використовують можливості нейронних мереж, мають потенціал для революції в управлінні побутовими відходами. Для цього необхідне надійне програмне забезпечення. Далі описано функціональні вимоги, яким має відповідати така система, щоб оптимізувати точність, ефективність і зручність сортування відходів.

Система повинна використовувати нейронні мережі для розпізнавання та класифікації різних відходів у режимі реального часу, включаючи, але не обмежуючись пластиком, папером, картоном, металами, склом та органічними відходами.

Програмне забезпечення повинно керувати сортувальними механізмами (наприклад, роботизованими маніпуляторами, відхилювачами конвеєра) для перенаправлення відходів у відповідні контейнери або потоки переробки на основі результатів класифікації нейронної мережі.

Інтерфейс користувача повинен надавати детальні звіти та аналітику щодо складу відходів, ефективності сортування та продуктивності системи, полегшуючи прийняття рішень на основі даних.

Програмне забезпечення повинно реєструвати та зберігати дані, пов'язані з розпізнаванням відходів, рішеннями щодо сортування та продуктивністю системи, для подальшого аналізу та аудиту.

Система повинна використовувати алгоритми профілактичного обслуговування для проактивного виявлення та вирішення потенційних проблем до того, як вони призведуть до простою.

Програмне забезпечення повинно підтримувати інтеграцію з іншими системами управління відходами, такими як системи управління запасами, відстеження та звітності, для полегшення безперешкодного обміну даними.

Оператори повинні мати можливість віддаленого моніторингу та управління системою, що дозволить швидко реагувати на проблеми та зменшить потребу в персоналі на місці.

Успішна розробка та впровадження автоматизованої системи сортування побутових відходів на основі нейронних мереж залежить від виконання цих функціональних вимог. Забезпечуючи точне розпізнавання відходів, ефективне сортування, та надійне управління даними, така система може зробити значний внесок у більш стійку та ефективну практику поводження з побутовими відходами.

## **2.2 Нефункціональні вимоги до програмної системи**

На додаток до функціональних вимог, нефункціональні вимоги відіграють вирішальну роль у розробці та роботі програмної системи для автоматизованого сортування відходів з використанням нейронних мереж. У цій роботі окреслено нефункціональні системні вимоги, зокрема зосереджено увагу на апаратних засобах та передумовах підключення, необхідних для успішної роботи системи.

Нефункціональні системні вимоги, які часто називають системними, диктують умови навколишнього середовища та експлуатації, яких програмна система повинна дотримуватися для ефективного функціонування. У

контексті автоматизованої системи сортування відходів на основі нейронних мереж наступні нефункціональні вимоги є важливими для забезпечення безперебійної роботи та доступності.

Вимога 1: Смартфон з ОС Android версії 4.4+

Програмне забезпечення повинно бути сумісним зі смартфонами, що працюють під управлінням ОС Android версії 4.4 або вище. Ця вимога гарантує, що широкий спектр Android-пристроїв може бути використаний як інтерфейс управління системою, підвищуючи її доступність та гнучкість.

Вимога 2: Наявність камери на смартфоні.

Система покладається на камеру підключеного смартфона для захоплення та обробки зображень в режимі реального часу. Тому смартфон повинен мати функціональну камеру, здатну створювати зображення з достатньою роздільною здатністю та якістю для розпізнавання відходів на основі нейронних мереж.

Вимога 3: Підключення до Інтернету

Підключення до Інтернету є обов'язковим для наступних цілей:

- Віддалений моніторинг: Система повинна дозволяти операторам віддалено контролювати її роботу за допомогою веб-панелі або спеціального мобільного додатку. Підключення до Інтернету полегшує оновлення та моніторинг в режимі реального часу, навіть коли оператори знаходяться за межами офісу.

- Обмін даними: Щоб підтримувати реєстрацію даних, звітність та аналітику, система повинна мати можливість обмінюватися даними із зовнішніми серверами або хмарними рішеннями для зберігання даних. Ця вимога гарантує, що цінні дані, пов'язані з ефективністю та продуктивністю сортування відходів, можна безпечно зберігати та отримувати до них доступ.

Нефункціональні системні вимоги, такі як сумісність зі смартфоном, наявність камери та підключення до Інтернету, є критично важливими для ефективної роботи та доступності автоматизованої системи сортування побутових відходів на основі нейронних мереж. Ці вимоги гарантують, що

система може бути легко інтегрована в існуючі процеси управління відходами та надавати можливості моніторингу та обміну даними в режимі реального часу, тим самим підвищуючи загальну ефективність і сталість практики управління відходами.

### **2.3 Інструменти розробки програмної системи**

Розробка програмної системи для автоматизованого сортування побутових відходів вимагає ретельно підбраного набору інструментів розробки програмного забезпечення для забезпечення ефективності, точності та надійності. У цьому розділі розглядаються ключові інструменти, що використовуються в процесі розробки.

Ефективне сортування побутових відходів має важливе значення для ефективного управління відходами та їх переробки. Процес ручного сортування займає багато часу і схильний до помилок. У цій роботі представлено розробку автоматизованої системи сортування побутових відходів з використанням Arduino, широко доступної електронної платформи з відкритим вихідним кодом. Система інтегрує датчики, актуатори та мікроконтролер для точної ідентифікації та сортування різних типів відходів, тим самим підвищуючи загальну ефективність процесів управління відходами [5].

Arduino – це мікроконтролерна платформа з відкритим вихідним кодом, яка забезпечує гнучке та економічно ефективне рішення для розробки різних електронних проектів. Її універсальність, простота використання та підтримка великої онлайн-спільноти роблять її ідеальним вибором для розробки автоматизованих систем, включаючи механізми сортування відходів.

Автоматизована система сортування відходів на базі Arduino використовує комбінацію датчиків для виявлення та ідентифікації різних відходів. Оптичні датчики, такі як інфрачервоні датчики та датчики кольору,



дозволяють системі ідентифікувати конкретні категорії відходів, такі як папір, пластик, скло та метал. Ультразвукові датчики можуть ще більше підвищити точність вимірювання відстані, допомагаючи в точному сортуванні об'єктів.

Виконавчі механізми відіграють важливу роль у процесі фізичного сортування в системі. Сумісні з Arduino приводи використовуються для керування механічними маніпуляторами на основі показань датчиків. Інтеграція цих приводів забезпечує ефективне і точне сортування.

Мікроконтролер Arduino слугує центральним процесором, полегшуючи інтеграцію даних з датчиків та виконання логіки управління для роботи приводів. Запрограмована за допомогою інтуїтивно зрозумілого інтегрованого середовища розробки (IDE) Arduino, логіка управління інтерпретує входи датчиків, визначає відповідні дії сортування та керує рухами виконавчих механізмів, забезпечуючи тим самим точне та швидке розділення відходів.

Впровадження алгоритмів машинного навчання підвищує здатність системи на базі Arduino до адаптації та покращує можливості розпізнавання відходів з часом. Використовуючи методи машинного навчання, система може безперервно навчатися на основі нових вхідних даних, що дозволяє їй розпізнавати та сортувати ширший спектр відходів з підвищеною точністю та ефективністю.

Впровадження автоматизованої системи сортування відходів на базі Arduino сприяє покращенню практики поводження з відходами, сприяючи екологічній стійкості за рахунок зменшення кількості відходів на полігонах та сприяння переробці відходів. Підвищуючи ефективність процесів сортування відходів, система сприяє збереженню природних ресурсів та зменшенню забруднення навколишнього середовища, пов'язаного з неправильною утилізацією відходів.

Мовою програмування було обрано Python. Python є основною мовою програмування для розробки програмної системи автоматизованого сортування відходів. Вона має ряд переваг, серед яких:

- Універсальність: Python підтримує широкий спектр бібліотек та фреймворків, придатних для машинного навчання, комп'ютерного зору та обробки даних, що робить її ідеальною для створення основних алгоритмів сортування та компонентів обробки даних.
- Зручність для читання: Чистий і лаконічний синтаксис Python покращує читабельність коду, що має вирішальне значення для підтримки такого складного проекту, як сортування сміття.
- Підтримка спільноти: Python має активну спільноту розробників, яка надає доступ до безлічі ресурсів, навчальних посібників та бібліотек.

Системою контролю версій було обрано Python Git. Git слугує системою контролю версій (VCS) для проекту, пропонуючи наступні переваги:

- Розгалуження: можливості розгалуження та об'єднання Git дозволяють здійснювати паралельну розробку та експерименти без шкоди для стабільності основної кодової бази.
- Співпраця: підтримує спільну розробку, дозволяючи декільком розробникам одночасно працювати над одним проектом.
- Відстеження історії: Git відстежує зміни в кодовій базі, що дозволяє легко повернутися до попередніх станів, якщо це необхідно.

В якості системи управління базами даних для програмної системи обрано PostgreSQL, що має наступні переваги:

PostgreSQL відома своєю надійністю та стійкістю, забезпечуючи цілісність та узгодженість даних.

- Масштабованість: вона може ефективно обробляти великі обсяги даних, що важливо для управління даними про сортування відходів.

- Підтримка просторових даних: підтримка PostgreSQL просторових типів даних є цінною для управління геопросторовою інформацією, пов'язаною з місцями збору та сортування відходів.

Бібліотекою машинного навчання та комп'ютерного зору було обрано TensorFlow. TensorFlow є критично важливою бібліотекою для розробки компонентів машинного навчання та комп'ютерного зору програмної системи. Її ключові атрибути включають:

- Глибоке навчання: TensorFlow чудово підходить для побудови та навчання глибоких нейронних мереж для таких задач, як класифікація зображень, виявлення об'єктів та семантична сегментація.
- Спільнота та екосистема: TensorFlow може похвалитися великою спільнотою користувачів і багатою екосистемою попередньо навчених моделей та інструментів для розробки і розгортання моделей.
- Масштабованість: TensorFlow підтримує розподілені обчислення, що дозволяє ефективно навчати складні моделі.

Протоколом зв'язку було обрано MQTT (Message Queuing Telemetry Transport). MQTT – це обраний протокол зв'язку для забезпечення зв'язку в реальному часі між програмною системою та різними апаратними компонентами, такими як датчики та роботизовані руки. Ключові особливості включають:

- Легкий: MQTT розроблений, щоб бути легким та ефективним, що робить його придатним для пристроїв з обмеженими ресурсами.
- Модель "публікація-підписка": Використовує модель обміну повідомленнями "публікація-підписка", що полегшує асинхронний зв'язок між пристроями.
- Якість обслуговування (QoS): MQTT підтримує різні рівні QoS для забезпечення надійної доставки повідомлень, що має вирішальне значення для управління системою сортування відходів.

Gazebo було обрано в якості інструменту моделювання та тестування програмної системи завдяки його надійним функціям:

- 3D моделювання: Gazebo пропонує реалістичне середовище 3D моделювання, яке дозволяє розробникам тестувати та перевіряти алгоритми, пов'язані з роботизованими компонентами та розпізнаванням об'єктів.
- Сенсорна симуляція: Підтримує симуляцію різних датчиків, таких як камери і LiDAR, що дозволяє реалістично тестувати алгоритми сприйняття.
- Інтеграція з РОС: Gazebo легко інтегрується з Robot Operating System (ROS), популярною платформою для розробки роботів, що полегшує інтеграцію з робототехнічними компонентами.

Jenkins обрано як інструмент CI/CD для автоматизації процесів збірки, тестування та розгортання. Основні функції включають:

- Автоматизація: Jenkins автоматизує повторювані завдання, такі як збірка і тестування коду, забезпечуючи узгодженість і надійність випусків програмного забезпечення.
- Екосистема плагінів: Він пропонує широкий спектр плагінів для інтеграції з різними інструментами, включаючи системи контролю версій, фреймворки для тестування та платформи розгортання.
- Масштабованість: Jenkins може масштабуватися для роботи зі складними робочими процесами CI/CD та великими кодовими базами, що робить його придатним для проектів корпоративного рівня.

Ці інструменти розробки програмних систем в сукупності сприяють успішному створенню автоматизованої системи сортування побутових відходів, забезпечуючи ефективне сортування, управління даними та прийняття рішень в режимі реального часу, гарантуючи якість коду.

Для розробки веб-додатку було обрано текстовий редактор Visual Studio Code. Visual Studio Code був обраний завдяки своїм особливостям –

підсвічування синтаксису та підтримка великої кількості мов програмування (C, C++, Java, XML, HTML, PHP, JavaScript, ASCII, Visual Basic / VBScript, SQL, Ruby, CSS, Pascal, Perl і Python), багатомовна підтримка, робота з декількома документами. Visual Studio Code — засіб для створення, редагування та зневадження сучасних веб-застосунків і програм для хмарних систем. Visual Studio Code розповсюджується безкоштовно і доступний у версіях для платформ Windows, Linux і OS X. Компанія Microsoft представила Visual Studio Code у квітні 2015 на конференції Build 2015. Це середовище розробки стало першим кросплатформним продуктом у лінійці Visual Studio. За основу для Visual Studio Code використовуються напрацювання вільного проєкту Atom, що розвивається компанією GitHub. Зокрема, Visual Studio Code є надбудовою над Atom Shell, що використовує браузерний рушій Chromium і Node.js. Примітно, що про використання напрацювань вільного проєкту Atom і на сайті Visual Studio Code, і в прес-релізі, і в офіційному блозі не згадується. Редактор містить вбудований зневаджувач, інструменти для роботи з Git і засоби рефакторингу, навігації по коду, автодоповнення типових конструкцій і контекстної підказки. Продукт підтримує розробку для платформ ASP.NET і Node.js, і позиціюється як легковагове рішення, що дозволяє обійтися без повного інтегрованого середовища розробки. Visual Studio Code – текстовий редактор, призначений для програмістів і тих, кого не влаштовує скромна функціональність програми «блокнот», що входить до складу Windows. Інтерфейс Visual Studio Code є багатомовним (українська мова присутня)

Visual Studio Code (VS Code) є чудовим вибором як середовище розробки для створення системи, що використовує нейронні мережі для автоматизованого сортування побутових відходів. Цей універсальний і широко розповсюджений редактор коду має кілька переконливих причин для його використання в даному контексті.

### 1. Міжплатформенна сумісність:

- VS Code доступний на багатьох операційних системах, включаючи Windows, macOS та різні дистрибутиви Linux. Ця крос-платформенна підтримка гарантує, що розробники можуть працювати над проектом, використовуючи свою улюблену ОС, покращуючи співпрацю та гнучкість.

### 2. Розширюваність та екосистема плагінів:

- VS Code може похвалитися великою бібліотекою розширень і плагінів, які можуть розширити функціональність для різних мов програмування, фреймворків та інструментів. Ця розширюваність безцінна при роботі з бібліотеками нейронних мереж (наприклад, TensorFlow, PyTorch) або інтеграції з іншими програмними компонентами.

### 3. Інтегрований термінал:

- Вбудований у VS Code інтегрований термінал дозволяє розробникам запускати команди та скрипти безпосередньо в редакторі. Ця функція необхідна для таких завдань, як навчання моделей нейронних мереж, запуск скриптів для попередньої обробки даних або взаємодія з апаратним забезпеченням системи сортування.

### 4. Інтегрований контроль версій

- VS Code легко інтегрується з популярними системами контролю версій, такими як Git. Це полегшує спільну розробку та дозволяє ефективно відстежувати зміни коду, що є важливим аспектом розробки програмного забезпечення для складних систем.

### 5. Навігація коду та IntelliSense:

- VS Code надає інтелектуальну навігацію коду та функції автозавершення (IntelliSense), які особливо корисні при роботі зі складними бібліотеками нейронних мереж. Розробники можуть

швидко досліджувати код, отримувати доступ до документації та виявляти помилки або пропозиції.

#### 6. Можливості налагодження:.

- Вбудовані у VS Code інструменти налагодження спрощують процес виявлення та виправлення проблем у коді, що є критично важливим при розробці та доопрацюванні нейромережевих моделей для сортування відходів.

#### 7. Інтеграція з Jupyter Notebooks:.

- VS Code має інтегровану підтримку ноутбуків Jupyter, що робить його придатним для аналізу даних, експериментів з моделями машинного навчання та обміну інтерактивними результатами досліджень - всіх важливих аспектів розробки системи сортування відходів на основі ШІ.

#### 8. Сильна спільнота та підтримка

- VS Code має велику та активну спільноту розробників і величезну базу онлайн-ресурсів. Ця спільнота надає обширну документацію, навчальні посібники та форуми для усунення несправностей, що може бути безцінним при вирішенні тонкощів розробки нейронних мереж.

#### 9. Продуктивність та ефективність використання ресурсів

- VS Code відомий своєю продуктивністю та ефективністю використання ресурсів, що дозволяє розробникам працювати над ресурсоемними завданнями, такими як навчання нейронних мереж, без надмірного навантаження на системні ресурси.

#### 10. Відкритий вихідний код та економічність:.

- VS Code є інструментом з відкритим вихідним кодом, що робить його економічно вигідним для окремих розробників та організацій. Це особливо вигідно, якщо врахувати бюджетні обмеження, часто пов'язані з проектами досліджень і розробок.

#### 11. Інтеграція з хмарними сервісами

- VS Code легко інтегрується з різними хмарними сервісами та платформами, що може бути корисним для доступу до хмарних ресурсів графічного процесора для навчання нейронних мереж або для використання хмарного зберігання та обробки даних.

#### 12. Постійне вдосконалення:

- Microsoft, розробник VS Code, регулярно випускає оновлення та вдосконалення, гарантуючи, що редактор відповідає останнім технологіям та потребам розробників.

Отже, Visual Studio Code пропонує комплексне і зручне для розробників середовище для створення системи, яка використовує нейронні мережі для автоматизованого сортування побутових відходів. Його універсальність, розширюваність, можливості налагодження та потужна підтримка спільноти роблять його ідеальним вибором як для індивідуальних дослідників, так і для колективних команд розробників, які працюють над складними проектами у сфері ІІІ та управління відходами.



### **3 ПРОЕКТУВАННЯ СИСТЕМИ АВТОМАТИЗОВАНОГО СОРТУВАННЯ СМІТТЯ**

#### **3.1 Проектування архітектури системи**

Розробка надійної архітектури системи має важливе значення для успішного впровадження автоматизованої системи сортування відходів з використанням нейронних мереж [11]. У цій роботі описано архітектурні компоненти та принципи, необхідні для задоволення функціональних і нефункціональних вимог, які були розглянуті раніше. Архітектура системи забезпечує безперебійну ідентифікацію матеріалів, автоматизоване сортування, звітність, аналітику, управління даними, інтеграцію із зовнішніми системами, враховуючи апаратні вимоги.

Ефективна архітектура системи є ключовим фактором у створенні автоматизованої системи сортування відходів, яка задовольняє функціональні та нефункціональні вимоги, забезпечуючи ефективне та стаєле управління відходами. У цій роботі представлено архітектурний дизайн, який забезпечує точну ідентифікацію відходів, автоматизоване сортування, аналіз даних та підключення з урахуванням апаратних вимог [6].

Архітектура системи складається з декількох взаємопов'язаних компонентів, які відповідають визначеним вимогам:

1. Ідентифікація матеріалів:
  - Модель нейронної мережі: Ядро системи, навчена модель нейронної мережі, ідентифікує відходи на основі зображень, отриманих в реальному часі з камер смартфонів.
2. Автоматизоване сортування:
  - Сортувальні механізми: Пристрій для сортування фізично відокремлює відходи відповідно до класифікації нейронної мережі [12]. Модуль сортувальних механізмів постійно контролює продуктивність сортувального обладнання. Він генерує показники

ефективності, пов'язані з точністю сортування, швидкістю та часом безвідмовної роботи системи.

Інтеграція зі звітністю та аналітикою: Дані, зібрані модулем "Сортувальні механізми", включно з показниками продуктивності та результатами сортування, передаються до компонента "Звітність та аналітика" системи для поглибленого аналізу та звітності.

3. Звітність та аналітика: \*Модуль аналізу даних: \*Модуль аналізу даних:

- Модуль аналізу даних: Цей компонент обробляє дані, зібрані під час сортування, створюючи звіти про склад відходів, ефективність сортування та показники продуктивності. Модуль аналізу даних є найважливішим компонентом системної архітектури автоматизованої системи сортування відходів з використанням нейронних мереж. Цей модуль відіграє ключову роль в генеруванні змістовних звітів та наданні цінної аналітики. Далі буде описано дизайн і можливості модуля аналізу даних, щоб задовольнити функціональні та нефункціональні вимоги системи.

Модуль аналізу даних починає роботу зі збору даних з різних джерел в системі, включаючи:

- дані класифікації матеріалів: Дані, пов'язані з матеріалами, ідентифікованими нейронною мережею в процесі сортування;
- показники ефективності сортування: Показники, що стосуються ефективності автоматизованого процесу сортування, такі як пропускна здатність і точність.

Після збору ці необроблені дані проходять попередню обробку, щоб очистити і відформатувати їх для аналізу, забезпечуючи точність і надійність даних.

Модуль підключений до безпечного і масштабованого рішення для зберігання даних. Це хмарне сховище призначене для обробки великих

обсягів даних, гарантуючи, що історичні дані сортування будуть легко доступні для аналізу, звітності та аудиту.

Автоматизовані системи сортування відходів покладаються на різні датчики та технології візуалізації для точної ідентифікації та сортування різних типів побутових відходів. Ці системи генерують значні обсяги даних, які потребують ефективного зберігання та обробки. Хмарні обчислення забезпечують масштабоване та економічно ефективне рішення для управління цими даними, гарантуючи безперебійну роботу та сприяючи постійному вдосконаленню системи.

Хмарна інфраструктура пропонує надійну платформу для зберігання, управління та аналізу даних в автоматизованих системах сортування відходів. Завдяки масштабованій ємності сховища та високій обчислювальній потужності хмара забезпечує безперешкодну інтеграцію з інтерфейсами сортувальних машин, що сприяє ефективній передачі та зберіганню даних, отриманих у процесі сортування. Крім того, хмарна інфраструктура надає надійні механізми резервного копіювання та аварійного відновлення, забезпечуючи цілісність даних і стійкість системи.

Використовуючи хмарне сховище даних, автоматизовані системи сортування відходів можуть проводити аналіз даних у режимі реального часу, що дозволяє операторам контролювати процеси сортування та оперативно виявляти потенційну неефективність або несправності. Інформація в режимі реального часу, отримана з хмарних даних, полегшує прийняття проактивних рішень, дозволяючи операторам оптимізувати параметри сортування, коригувати алгоритми сортування та підвищувати загальну продуктивність системи.

Хмарне сховище даних дозволяє безперервно збирати та аналізувати дані про сортування, полегшуючи розробку алгоритмів машинного навчання та моделей штучного інтелекту. Ці моделі можуть підвищити точність і швидкість сортування відходів, що призведе до підвищення ефективності та адаптивності системи. Крім того, хмарна інфраструктура підтримує

віддалений доступ до даних, що дозволяє операторам контролювати та керувати операціями сортування з різних місць, тим самим підвищуючи операційну гнучкість та швидкість реагування.

Забезпечення безпеки та конфіденційності даних, що зберігаються в хмарі, має першорядне значення для успіху автоматизованих систем сортування відходів. Впровадження надійних протоколів шифрування, механізмів контролю доступу та методів анонімізації даних може захистити конфіденційну інформацію та запобігти несанкціонованому доступу або витоку даних. Дотримання правил захисту даних і галузевих стандартів має важливе значення для зміцнення довіри та впевненості в безпеці хмарних рішень для зберігання даних.

Хмарні рішення для зберігання даних пропонують масштабованість та економічну ефективність, що дозволяє автоматизованим системам сортування відходів пристосовуватися до зростаючих обсягів даних без необхідності значних інвестицій в інфраструктуру. Провайдери хмарних сервісів пропонують гнучкі плани зберігання даних, що дозволяє організаціям масштабувати свої ресурси зберігання відповідно до поточних потреб і прогнозів зростання в майбутньому. Така масштабованість мінімізує операційні витрати та оптимізує використання ресурсів, що робить хмарне зберігання даних економічно вигідним варіантом для автоматизованих систем сортування відходів різного масштабу.

Інтеграція хмарних сховищ даних в автоматизовані системи сортування відходів відповідає принципам сталого управління відходами. Підвищуючи точність сортування, покращуючи операційну ефективність і сприяючи прийняттю рішень на основі даних, ці системи сприяють зменшенню кількості відходів на звалищах, просуванню ініціатив з переробки та збереженню природних ресурсів. Безперешкодна інтеграція хмарних технологій сприяє комплексному підходу до управління відходами, підкреслюючи важливість технологічних інновацій у досягненні цілей екологічної стійкості.

Серце модуля аналізу даних – це його можливості аналізу даних і звітування. Основні функціональні можливості включають.

- Аналітика в режимі реального часу: Модуль надає аналітику в режимі реального часу для моніторингу поточного стану процесу сортування відходів. Сюди входить ефективність сортування, склад матеріалів та операційна статистика.
- Історичний аналіз: Історичні дані використовуються для виявлення тенденцій і закономірностей у часі. Наприклад, вони можуть виявити коливання складу відходів або ефективності сортування.
- Показники ефективності: Модуль розраховує та відображає показники ефективності, такі як рівень переробки, рівень забруднення та рівень перенаправлення відходів.
- Настроювані звіти: Користувачі можуть створювати настроювані звіти на основі певних параметрів і часових рамок. Ці звіти можна використовувати для звітності про дотримання вимог або для прийняття рішень.

Щоб зробити дані більш доступними і зрозумілими для користувачів, модуль аналізу даних включає в себе інструменти візуалізації даних. Графіки, діаграми та інформаційні панелі забезпечують візуальне представлення даних про сортування відходів, що дозволяє користувачам швидко інтерпретувати інформацію та приймати обґрунтовані рішення.

Модуль підтримує інтеграцію із зовнішніми системами управління відходами та звітності через API та комунікаційні протоколи. Це забезпечує безперебійний обмін даними для комплексного управління відходами та дотримання нормативних вимог.

#### 4. Реєстрування та зберігання даних: безпечне сховище даних:

- Зберігання даних: Безпечні хмарні або локальні сховища зберігають зареєстровані дані, забезпечуючи їх доступність та надмірність. Модуль хмарного зберігання даних слугує сховищем для всіх даних, що генеруються в процесі сортування відходів. Сюди

входить інформація, пов'язана з розпізнаванням відходів, рішеннями щодо сортування, продуктивністю системи.

- Масштабоване та безпечне сховище: Модуль використовує хмарні рішення для зберігання даних, відомі своєю масштабованістю та надійними заходами безпеки. Такий сервіс, як Amazon S3, є підходящим варіантом. Це гарантує, що система зможе обробляти величезні обсяги даних у міру розширення операцій з управління відходами.
- Організація даних: Дані організовані у структуровані бази даних або файлові системи зберігання, що полегшує їх пошук та аналіз. Кожен фрагмент даних позначається відповідними метаданими, включаючи часові позначки, категорії відходів та місця сортування, для всебічного аналізу.
- Політика зберігання даних: Модуль хмарного зберігання даних дотримується визначеної політики зберігання даних, гарантуючи, що дані зберігаються протягом відповідного періоду часу для цілей аналітики, аудиту та дотримання нормативних вимог. Періоди зберігання можуть бути скориговані на основі стандартів управління відходами та законодавчих вимог.
- Конфіденційність даних та відповідність: Для дотримання правил конфіденційності даних, таких як GDPR, модуль впроваджує механізми контролю конфіденційності та шифрування, забезпечуючи захист конфіденційних даних від несанкціонованого доступу.
- Безперервний потік даних: Модуль призначений для обробки потоку даних у реальному часі з різних сортувальних установок, підключених до системи. Це дозволяє негайно збирати та зберігати дані, підтримуючи моніторинг та аналітику в режимі реального часу.

- Агрегація даних: Дані з декількох сортувальних установок і смартфонів агрегуються в режимі реального часу, створюючи комплексний огляд процесу сортування відходів у різних місцях або на різних об'єктах.
- Резервування даних: Для забезпечення доступності даних та запобігання їх втрати через збої в роботі обладнання або непередбачувані інциденти впроваджуються конфігурації резервного зберігання даних. Для резервування можна використовувати декілька центрів обробки даних.
- План аварійного відновлення: Архітектура системи включає план аварійного відновлення, який описує процедури відновлення даних у разі катастрофічних подій, забезпечуючи безперервність даних.
- Контроль доступу на основі ролей: Доступ до даних у модулі хмарного сховища даних контролюється за допомогою механізмів контролю доступу на основі ролей. Користувачам надаються відповідні рівні доступу на основі їхніх ролей, що гарантує, що тільки уповноважений персонал може переглядати або змінювати дані.
- Інтеграція з інструментами аналітики: Модуль розроблений для безперешкодної інтеграції з інструментами та платформами для аналізу даних. Ця інтеграція дозволяє отримувати цінну інформацію зі збережених даних, включаючи тенденції складу відходів, покращення ефективності сортування та прогнозування потреб у технічному обслуговуванні.
- Аудиторські сліди: Система генерує комплексні аудиторські сліди, реєструючи всі події доступу до даних та їх модифікації. Це забезпечує прозорість і підзвітність, а також підтримує процеси аудиту на відповідність нормативним вимогам.

Модуль хмарного зберігання даних є ключовою частиною системної архітектури автоматизованої системи сортування відходів з використанням

нейронних мереж. Він забезпечує безпечне, масштабоване та ефективне зберігання даних, що генеруються в процесі сортування відходів. Модуль не лише підтримує моніторинг та аналітику даних у режимі реального часу, але й дотримується правил конфіденційності даних, а також забезпечує резервування та заходи з аварійного відновлення для захисту критично важливих даних про управління відходами.

#### 5. Інтеграція із зовнішніми системами

- API та комунікаційні протоколи: Архітектура системи включає API та комунікаційні протоколи для полегшення безперешкодного обміну даними із зовнішніми системами управління відходами та звітності. Модуль інтеграції із зовнішніми системами є критично важливим компонентом в архітектурі автоматизованої системи сортування відходів. Цей модуль полегшує обмін даними та інформацією між системою сортування відходів і зовнішніми системами, підвищуючи операційну ефективність, звітність і практику управління відходами.

Модуль включає набір інтерфейсів прикладного програмування (API) та комунікаційних протоколів, які забезпечують безперешкодний обмін даними між автоматизованою системою сортування відходів та зовнішніми системами. Ці API дозволяють здійснювати стандартизований та структурований обмін даними, забезпечуючи сумісність з різними зовнішніми платформами та базами даних.

6. Системи управління відходами: Модуль інтеграції із зовнішніми системами забезпечує зв'язок із системами управління відходами, такими як централізовані бази даних відходів або платформи управління, що експлуатуються органами управління відходами. Ця інтеграція дозволяє системі сортування відходів надсилати важливі дані про склад відходів, ефективність сортування та продуктивність системи до цих зовнішніх систем у режимі реального часу.



7. Безпека: Надійні заходи безпеки гарантують, що обмін даними із зовнішніми системами є безпечним, а доступ до них обмежений лише уповноваженим персоналом. Це захищає конфіденційну інформацію та запобігає витоку даних.

8. Масштабованість: Модуль повинен бути розроблений з урахуванням можливості масштабування, враховуючи потенційне зростання обсягу даних та підключення до різних зовнішніх систем у міру розширення або розвитку операцій з управління відходами.

9. Обробка помилок та сповіщення: Модуль повинен включати механізми обробки помилок для усунення збоїв зв'язку або неузгодженості даних. Він також повинен надавати можливості сповіщення для попередження операторів про будь-які проблеми, які потребують уваги або втручання.

Модуль інтеграції із зовнішніми системами відіграє ключову роль в архітектурі автоматизованої системи сортування відходів. Сприяючи обміну даними з системами управління відходами, звітності, він підвищує загальну ефективність, прозорість і сталість практик управління відходами. Ефективна інтеграція забезпечує безперешкодний обмін даними між автоматизованою системою сортування та зовнішніми системами, що сприяє кращому прийняттю рішень та дотриманню правил поведінки з відходами.

10. Дистанційний моніторинг та контроль: дистанційний моніторинг та контроль.

- Інтерфейс: інтерфейс, доступний через підключені до Інтернету смартфони або комп'ютери, дозволяє здійснювати дистанційний моніторинг та контроль процесу сортування.

Завдяки включенню зручного інтерфейсу користувача, доступного через підключені до Інтернету смартфони, система стає більш універсальною і орієнтованою на користувача, задовольняючи нефункціональні вимоги, такі як швидкість реагування, масштабованість і простота використання.

Ефективні системи сортування відходів покладаються на ефективні користувацькі інтерфейси, які дозволяють операторам здійснювати віддалений моніторинг та управління системою. У цій роботі досліджується розробка та інтеграція модуля інтерфейсу користувача в архітектуру системи для задоволення нефункціональних вимог, одночасно покращуючи користувацький досвід та контроль.

Модуль інтерфейсу користувача є критично важливим компонентом архітектури системи, що забезпечує зручний інтерфейс для віддаленого моніторингу та управління. Його компоненти включають:

- Адаптивний веб-додаток: Інтерфейс розроблений як адаптивний додаток, доступний на смартфонах. Інтерфейс має інтуїтивно зрозумілу інформаційну панель, яка відображає дані в реальному часі та ключові показники продуктивності, такі як ефективність сортування, склад матеріалу та стан системи.
- Зручні елементи керування: Оператори можуть взаємодіяти з системою за допомогою зручних елементів керування, включаючи команди запуску/зупинки, аварійного вимкнення та ручного керування.
- Потoki даних: Модуль інтерфейсу користувача візуалізує потоки даних у реальному часі, дозволяючи операторам контролювати процес сортування, ідентифікацію матеріалів і стан системи.
- Діаграми та графіки: Інтерактивні діаграми та графіки забезпечують графічне представлення ефективності сортування та складу відходів у часі.
- Попередження та сповіщення: Модуль інтерфейсу користувача надсилає попередження та сповіщення операторам у разі виникнення помилок у системі, спрацьовування прогнозованого технічного обслуговування або інших критичних подій.
- Оновлення в режимі реального часу: Інтерфейс користувача забезпечує оновлення процесу сортування в режимі реального часу,

гарантуючи операторам негайний зворотний зв'язок і можливість прийняття оперативних рішень.

- Низька затримка: З'єднання з низькою затримкою забезпечує мінімальну затримку при передачі команд управління та отриманні оновлень стану системи.
- Адаптивний дизайн: Модуль користувацького інтерфейсу розроблений таким чином, щоб пристосуватись до зростаючої кількості підключених смартфонів та сортувальних одиниць у міру розширення операцій з управління відходами.
- Багатокористувацька підтримка: Система дозволяє декільком операторам одночасно отримувати доступ до інтерфейсу користувача та керувати ним, що покращує спільну роботу та масштабованість.

11. Простота у використанні: \*Інтуїтивно зрозумілий дизайн:

\*Інтуїтивно зрозумілий дизайн:

- Інтуїтивно зрозумілий дизайн: Інтуїтивно зрозумілий дизайн модуля інтерфейсу користувача мінімізує час навчання для операторів, дозволяючи як новачкам, так і досвідченим користувачам легко орієнтуватися в системі та ефективно керувати нею.

Включення модуля інтерфейсу користувача (UI) в архітектуру системи для автоматизованої системи сортування відходів на основі нейронних мереж значно розширює функціональність системи та покращує користувацький досвід. Забезпечуючи віддалений моніторинг, візуалізацію даних у реальному часі та можливість керування, модуль інтерфейсу користувача гарантує, що оператори можуть ефективно керувати та оптимізувати процес сортування відходів з будь-якого місця, де є доступ до Інтернету. Це не лише відповідає нефункціональним вимогам, таким як швидкість реагування, масштабованість і простота використання, але й дає змогу командам з

управління відходами приймати обґрунтовані рішення та підвищувати загальну продуктивність системи.

Запропонована системна архітектура автоматизованої системи сортування відходів з використанням нейронних мереж ефективно відповідає функціональним і нефункціональним вимогам. Поєднуючи ідентифікацію матеріалів на основі нейронних мереж, автоматизоване сортування, аналіз даних, безпечне управління даними та можливості інтеграції, архітектура забезпечує ефективне та комплексне рішення для сучасного сортування побутових відходів.

### 3.2 Проектування UML діаграм

Діаграма варіантів використання (прецедентів) відображає функціональне призначення проектованої програмної системи. Суть діаграми прецедентів полягає в тому, що систему представляють як групу акторів, які за допомогою варіантів використання взаємодіють із нею. Діаграма прецедентів є графом, що складається з множини акторів, прецедентів (варіантів використання) обмежених межею системи (прямокутник), асоціацій між акторами та прецедентами, відношень серед прецедентів, та відношень узагальнення між акторами. Діаграми прецедентів відображають елементи моделі варіантів використання. Суть діаграми прецедентів полягає в тому, що проектована система подається у вигляді множини сутностей чи акторів, що взаємодіють із системою за допомогою так званих варіантів використання. Варіант використання використовують для описання послуг, які система надає актору. Іншими словами, кожен варіант використання визначає деякий набір дій, який виконує система під час діалогу з актором. При цьому нічого не говориться про те, яким чином буде реалізовано взаємодію акторів із системою. У мові UML є кілька стандартних видів відношень між акторами і варіантами використання: – асоціації; – включення; – розширення; 32 – узагальнення.

При цьому загальні властивості варіантів використання можна подати трьома різними способами, а саме — за допомогою відношень включення, розширення і узагальнення. Відношення асоціації — одне з фундаментальних понять у мові UML і в тій чи іншій мірі використовується під час побудови всіх графічних моделей систем у формі канонічних діаграм. Включення у мові UML — це різновид відношення залежності між базовим варіантом використання і його окремим випадком. При цьому відношенням залежності є таке відношення між двома елементами моделі, за якого зміна одного елемента (незалежного) спричиняє зміну іншого елемента (залежного). Відношення розширення визначає взаємозв'язок базового варіанту використання з іншим варіантом використання, функціональна поведінка якого залучається базовим не завжди, а тільки за виконання додаткових умов. Актор — це сутність, що взаємодіє з системою для вирішення деяких завдань. Актором може бути людина, інша система, пристрій або програмний засіб. В даному випадку акторами визначено систему розпізнавання, яка класифікує сміття та сервер, який реєструє систему та створює звіт.

Після визначення акторів системи, необхідно сформулювати перелік усіх варіантів використання, з якими будуть взаємодіяти визначені актори. Серед основних варіантів використання системи: з боку класифікуючої системи відправка статистики на сервер; встановлення оновлень. З боку сервера отримання звіту про використання системи розпізнавання сміття. З боку обох класифікуючої системи та сервера — відправлення сигналу про неправильне сортування. За результатами сформованих варіантів використання та акторів системи було розроблено діаграму варіантів використання, яку наведено на рис. 3.1.



Рисунок 3.1 – Діаграма варіантів використання (прецедентів) системи

Також була створена діаграма послідовностей. Такі діаграми використовуються для уточнення діаграм прецедентів і більш детального опису логіки сценаріїв використання. Діаграма послідовності — різновид діаграми в UML.

Діаграма послідовності відображає взаємодії об'єктів впорядкованих за часом. Зокрема, такі діаграми відображають задіяні об'єкти та послідовність надісланих повідомлень.

На діаграмі послідовностей показано у вигляді вертикальних ліній різні процеси або об'єкти, що існують водночас. Надіслані повідомлення зображуються у вигляді горизонтальних ліній, в порядку відправлення. Визначені стандартом UML 2.0 діаграми послідовностей мають ті ж можливості, що і визначені стандартом UML 1.x, і підтримують додаткові можливості зміни стандартного порядку повідомлень

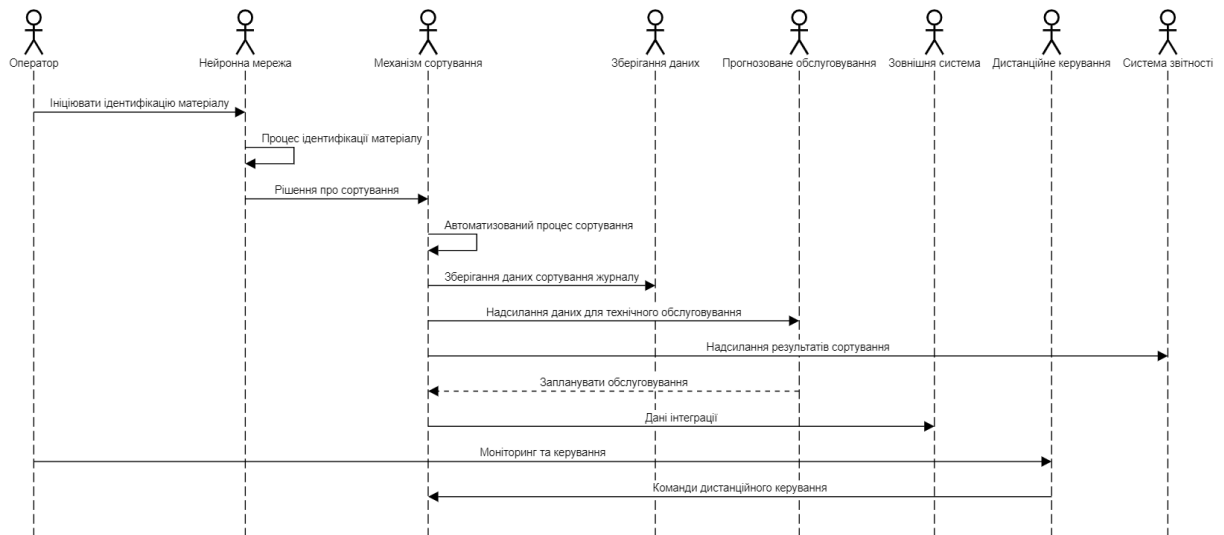


Рисунок 3.2 – Діаграма послідовностей

### 3.3 Проектування інтерфейсу користувача

Інтерфейс користувача автоматизованої системи сортування відходів є критично важливим компонентом процесу управління відходами, надаючи операторам комплексний набір інструментів для управління та оптимізації функціональності системи. Цей інтерфейс розроблений для простоти використання і надає доступ до різних функціональних можливостей, включаючи ідентифікацію матеріалів, автоматизоване управління сортуванням, звітність та аналітику, реєстрацію та зберігання даних, інтеграцію із зовнішніми системами, а також дистанційний моніторинг та управління.

Після входу в систему користувачів зустрічає розділ "Автоматизована система сортування", що пропонує швидкий огляд поточного стану системи. Ключові показники, такі як час безвідмовної роботи системи, ефективність сортування та нещодавні роботи з технічного обслуговування, відображаються на видному місці. Цей розділ слугує центральним вузлом для моніторингу стану системи.

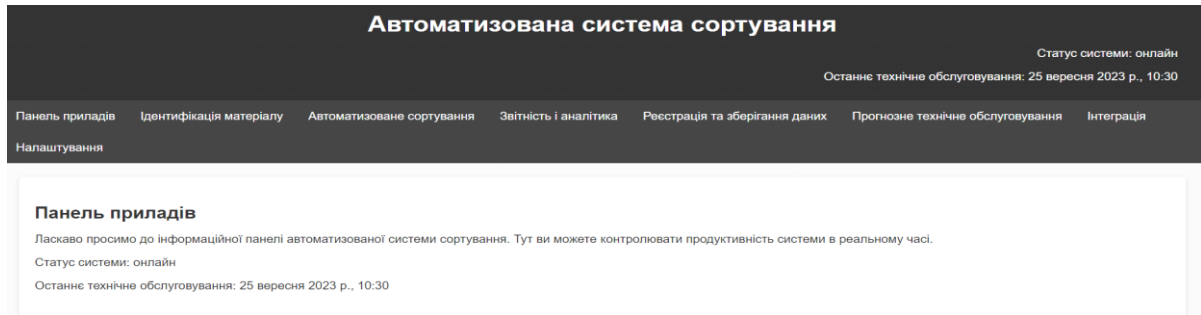


Рисунок 3.3 – Розділ "Автоматизована система сортування"

Розділ "Ідентифікація матеріалів" є критично важливим компонентом інтерфейсу користувача. Він надає операторам інформацію про матеріали, що обробляються, в режимі реального часу. В інтерфейсі відображається тип розпізнаного матеріалу і рівень достовірності ідентифікації нейронної мережі.



Тип матеріалу: пластик

Рівень впевненості: 95%

Рисунок 3.4 – Розділ "Ідентифікація матеріалів"



Розділ "Звітність та аналітика" пропонує погляд на ідентифікацію відходами на основі даних. Аналіз надає цінну інформацію для прийняття рішень.

Розділ "Реєстрація та зберігання даних" дозволяє операторам отримати доступ до історичних журналів даних для цілей аудиту та аналізу. Журнали даних містять позначки часу, типи матеріалів і кількість відсортованих відходів. Інтерфейс також надає інформацію про стан резервного копіювання даних і доступний обсяг пам'яті.

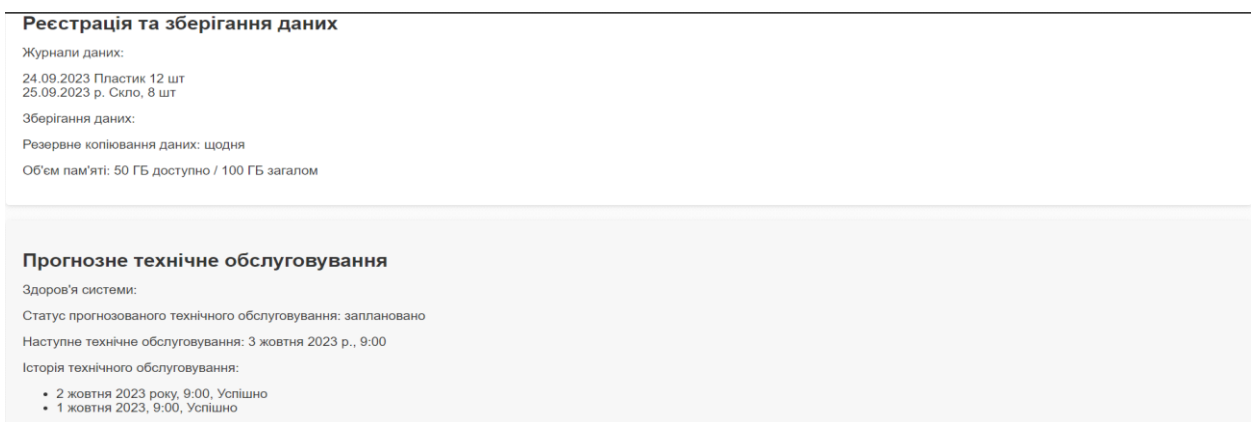


Рисунок 3.5 – Розділ "Реєстрація та зберігання даних"

Для безперешкодної інтеграції із зовнішніми системами розділ "Інтеграція" пропонує варіанти конфігурації. Оператори можуть керувати підключеними системами та отримати доступ до документації API для подальшого налаштування та інтеграції.

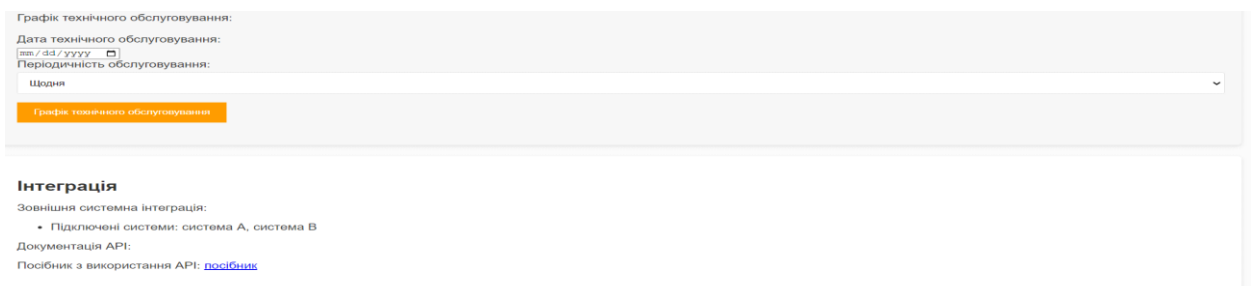
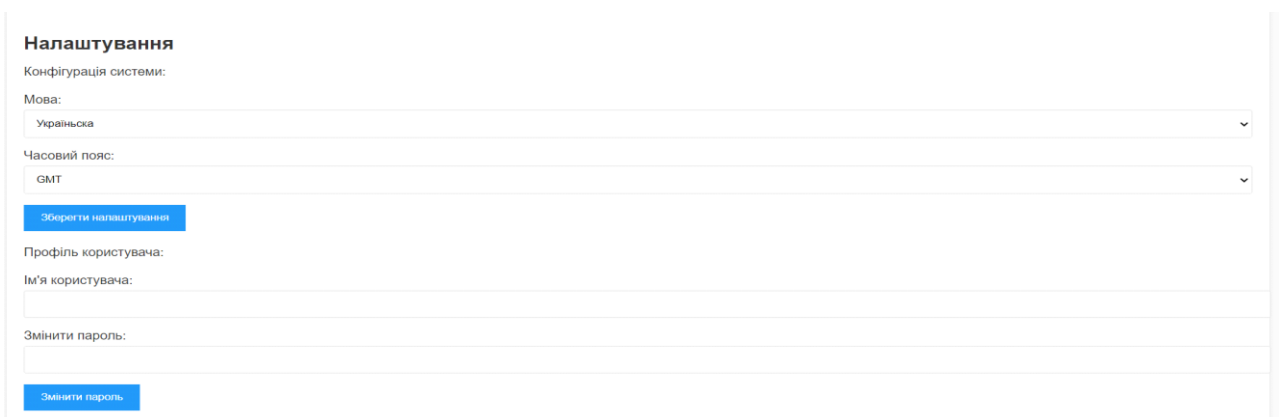


Рисунок 3.6 – Розділ "Інтеграція"

Віддалений моніторинг і управління полегшуються за допомогою розділу "Віддалене управління". Оператори можуть отримувати доступ до відеопотоків у реальному часі, віддалено запускати процеси сортування, а також отримувати попередження та сповіщення.

Розділ "Налаштування" дозволяє операторам пристосувати систему до своїх потреб. Для персоналізації доступні конфігурації мови та часового поясу. Користувачі також можуть змінювати свої профілі, включаючи зміну імені користувача та пароля.



**Налаштування**  
Конфігурація системи:

Мова:  
Українська

Часовий пояс:  
GMT

Зберегти налаштування

Профіль користувача:

Ім'я користувача:

Змінити пароль:

Змінити пароль

Рисунок 3.7 – Розділ "Налаштування"

Інтерфейс користувача автоматизованої системи сортування відходів є потужним інструментом для оптимізації операцій з управління відходами. Його інтуїтивно зрозумілий дизайн і широкі можливості дозволяють максимізувати ефективність сортування та сприяти сталому управлінню відходами. Оскільки технології продовжують розвиватися, такі інтерфейси відіграють ключову роль у формуванні майбутнього управління відходами.

### 3.4 Проектування нейронної мережі

Система використовує методи глибокого навчання для підвищення точності та ефективності сортування відходів, що сприяє сталому управлінню відходами [7].

В основі автоматизованої системи сортування відходів лежить нейромережева модель, яка була спеціально розроблена для класифікації різних відходів за попередньо визначеними категоріями. Нейронна мережа складається з наступних компонентів:

1. Для вилучення значущих ознак із зображень відходів використано згорткову нейронну мережу як основу нашої моделі. Вони добре підходять для задач класифікації зображень і можуть вивчати ієрархічні ознаки з необроблених даних зображень.

2. Щоб оптимізувати продуктивність моделі та зменшити потребу у великому зборі даних, використано переносне навчання. Використано попередньо навчену штучну нейронну мережу VGG16, як базову модель і налаштовано її шари для класифікації відходів.

3. Вихідний шар нейронної мережі налаштовується відповідно до кількості категорій відходів, які потрібно класифікувати. Кожен вихідний нейрон представляє певну категорію відходів, а функція активації softmax гарантує, що мережа виводить розподіл ймовірностей для кожної категорії.

4. Щоб підвищити надійність моделі та її здатність обробляти варіації зовнішнього вигляду відходів, застосовано методи доповнення даних, такі як випадкові обертання, перевертання та регулювання яскравості, до навчального набору даних.

5. Втрати перехресної ентропії використовуються як цільова функція для оптимізації ваг мережі під час навчання. Вона вимірює розбіжність між прогнозованою та реальною ймовірністю категорії відходів.

6. Стохастичний градієнтний спуск (SGD) або оптимізатор Адама використовується для оновлення ваг нейронної мережі під час зворотного розповсюдження.

Важливим компонентом розробки ефективної нейронної мережі для сортування відходів є наявність добре анотованого набору даних. Збирається різноманітний набір зображень відходів, забезпечуючи представництво всіх цільових категорій відходів. Потім набір даних попередньо обробляється за допомогою наступних кроків:

1. Всі зображення змінюються до однакової роздільної здатності, щоб забезпечити однакові вхідні розміри для нейронної мережі.

2. Значення пікселів зображень нормалізуються так, щоб вони лежали в певному діапазоні (наприклад,  $[0, 1]$  або  $[-1, 1]$ ) для полегшення збіжності під час навчання.

3. Набір даних розбивається на навчальну, валідаційну та тестову підмножини, зазвичай у співвідношенні 70-15-15, щоб полегшити оцінку моделі та запобігти надмірному пристосуванню.

Нейронна мережа навчається за допомогою навчального набору даних, а ефективність моделі контролюється за допомогою валідаційного набору даних. Навчання включає наступні кроки:

- Навчання виконується міні-пакетами для ефективного використання обчислювальних ресурсів та зменшення вимог до пам'яті.
- Критерії ранньої зупинки застосовуються для запобігання надмірному пристосуванню. Навчання зупиняється, коли продуктивність моделі на перевірконому наборі даних починає погіршуватися.
- Навчена модель оцінюється на тестовому наборі даних, щоб визначити її реальну ефективність у точній класифікації відходів.

Після того, як нейронна мережа навчена і перевірена, її можна інтегрувати в автоматизовану систему сортування відходів. Ця система може бути розгорнута щоб оптимізувати процеси сортування відходів і сприяти їх переробці.

Для подальшого підвищення продуктивності нашої нейромережевої системи сортування відходів можна застосувати кілька методів і міркувань:

- Точне налаштування гіперпараметрів, таких як швидкість навчання, розмір партії та частота відсіву, може суттєво вплинути на продуктивність моделі. Для пошуку оптимальних гіперпараметрів використовується сітковий та випадковий пошук.
- Впровадження методів регуляризації, таких як L1 та L2 регуляризація, допомагає запобігти надмірному пристосуванню, особливо при роботі з обмеженими навчальними даними.
- Експерименти з різними стратегіями доповнення даних можуть покращити здатність моделі узагальнювати невидимі елементи. Використовуються такі методи, як випадкове обрізання, переклад і масштабування.
- Використання ансамблевих методів, таких як об'єднання або підсилення, шляхом поєднання декількох нейромережевих моделей може призвести до підвищення точності класифікації.
- Вивчення різних попередньо навчених моделей і архітектур може допомогти визначити найбільш підходящу базову модель для конкретного завдання сортування відходів.

У практичній системі сортування відходів обробка в режимі реального часу має вирішальне значення для ефективного управління відходами.

Досягнення обробки в реальному часі передбачає оптимізацію як нейронної мережі, так і апаратного забезпечення, на якому вона працює:

- Квантифікація моделі нейронної мережі для зменшення її обчислювальних вимог та вимог до пам'яті забезпечує обробку в реальному часі на периферійних пристроях або вбудованих системах.
- Використання апаратних прискорювачів, таких як GPU або TPU, для виведення може значно прискорити процес класифікації, дозволяючи працювати в режимі реального часу.

- Зменшення затримки висновку шляхом оптимізації коду та архітектури моделі має важливе значення для досягнення продуктивності в реальному часі, особливо у високопродуктивних системах сортування відходів.

Для забезпечення надійності та робастності системи сортування відходів, впровадження механізмів контролю якості та зворотного зв'язку має вирішальне значення:

- Встановлення довірчих порогів для класифікації відходів може запобігти неправильній класифікації, відкидаючи непевні прогнози.
- Впровадження циклу зворотного зв'язку, коли неправильно класифіковані відходи перенаправляються на ручну перевірку та корекцію, допомагає підтримувати точність сортування.
- Регулярне калібрування та обслуговування апаратних компонентів системи і моделей нейронних мереж необхідні для підтримки оптимальної продуктивності з плином часу.
- Нейромережева модель розроблена таким чином, щоб можна було додавати нові категорії відходів з мінімальним перенавчанням та коригуванням.
- Впровадження методів безперервного навчання дозволяє системі адаптуватися до категорій і моделей відходів, що змінюються, зменшуючи потребу в частому перенавчанні моделі.

Розробка нейромережевої системи для автоматизованого сортування побутових відходів є перспективним рішенням для покращення практики поводження з відходами та сприяння їх переробці. Використовуючи передові методи глибокого навчання, оптимізуючи продуктивність моделі, уможливлуючи обробку в режимі реального часу, впроваджуючи механізми контролю якості та забезпечуючи масштабованість і адаптивність, можна створити надійну та ефективну систему сортування відходів, здатну вирішити проблеми сучасного управління відходами.

Проект розробленої мережі представлений на рисунку 3.4

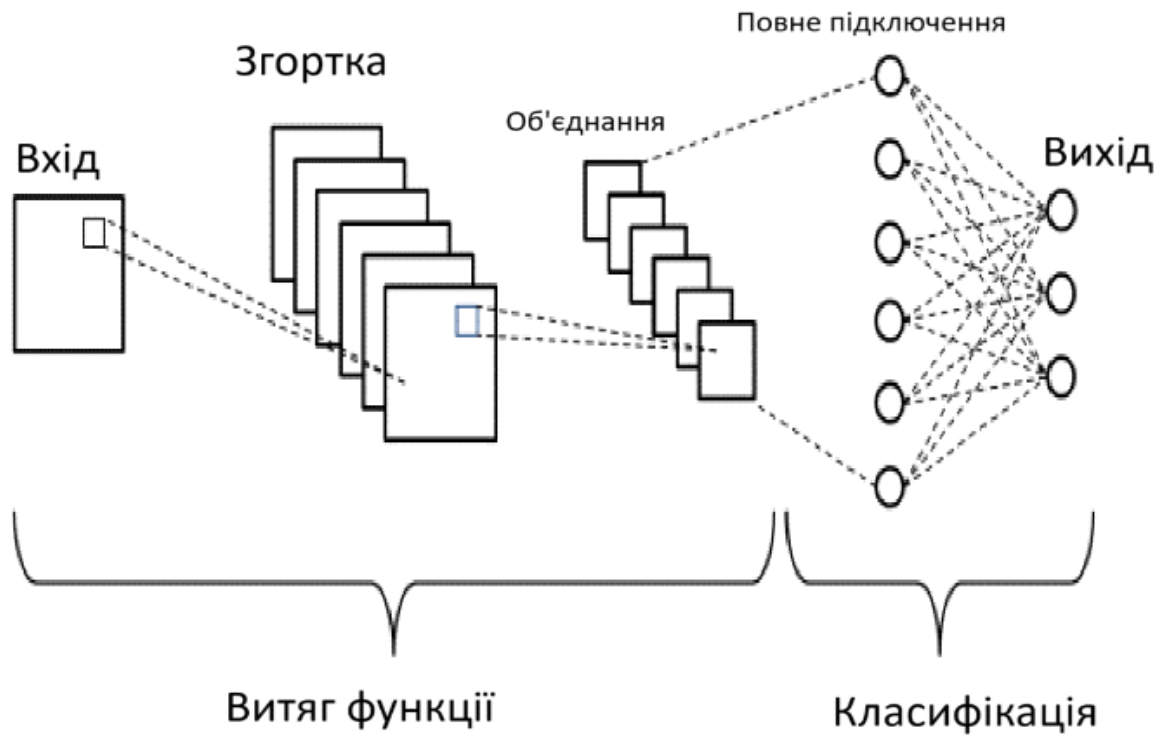


Рисунок 3.4 – Проект розробленої мережі

## 4 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ

### 4.1 Створення нейронної мережі

Безумовно, нейромережева модель є фундаментальним компонентом архітектури автоматизованої системи сортування відходів, що відповідає за точну ідентифікацію та класифікацію відходів у режимі реального часу. У цьому розділі детальніше розглянуто модель нейронної мережі, дослідимо її архітектуру, процес навчання і те, як вона впливає на загальну ефективність системи сортування відходів.

- Згорткова нейронна мережа (ЗНМ): зазвичай, згорткова нейронна мережа використовується як ядро нейромережевої моделі. ЗНМ добре підходять для задач розпізнавання зображень завдяки своїй здатності автоматично вивчати особливості зображень. Вони складаються зі згорткових шарів для виділення ознак і об'єднувальних шарів для зменшення вибірки.
- Повністю зв'язані шари: після згорткових шарів для класифікації використовуються повністю зв'язані шари. Ці шари беруть витягнуті ознаки і роблять прогнози щодо категорії відходів, до якої належить елемент.
- Збір даних: збирається різноманітний набір даних зображень відходів, кожне зображення позначається відповідно до категорії відходів, яку воно представляє (наприклад, пластик, папір, метали, скло).
- Попередня обробка даних: зібрані дані проходять попередню обробку, включаючи зміну розміру зображення, нормалізацію та доповнення, щоб покращити здатність нейронної мережі узагальнювати дані.
- Навчання: нейронна мережа навчається на попередньо обробленому наборі даних. Під час навчання вона вчиться розпізнавати шаблони та особливості на зображеннях, які відрізняють одну категорію



відходів від іншої. Процес навчання включає прямі та зворотні проходження, під час яких ваги мережі коригуються для мінімізації помилок класифікації.

- Валідація: окрема частина набору даних, відома як валідаційний набір, використовується для моніторингу роботи мережі під час навчання. Це допомагає запобігти надмірному пристосуванню, гарантуючи, що модель добре узагальнює нові, невидимі елементи відходів.
- Налаштування гіперпараметрів: гіперпараметри, такі як швидкість навчання, розмір партії та конфігурація архітектури, налаштовуються для оптимізації точності та збіжності мережі.
- Після того, як нейромережева модель навчена та перевірена, її розгортають в автоматизованій системі сортування відходів. У режимі реального часу модель обробляє зображення, зняті камерами смартфонів, коли відходи рухаються вздовж сортувального конвеєра.
- Для кожного відходу нейромережа робить прогноз, вказуючи категорію матеріалу (наприклад, пластик, папір), до якої він належить. Це передбачення спрямовує механізм сортування до відповідного потоку відходів.
- Постійний моніторинг та оцінка роботи нейромережевої моделі є дуже важливими. Періодичне перенавчання моделі за допомогою оновлених або розширених наборів даних може допомогти підвищити точність і адаптувати модель до змін у складі відходів.
- Оптимізація моделі також передбачає вирішення таких проблем, як помилкові спрацьовування та помилкові негативні результати класифікації, що ще більше підвищує ефективність системи.
- Архітектура системи розроблена таким чином, щоб полегшити масштабування нейромережевої моделі. Вона повинна бути здатна обробляти збільшені обсяги відходів і розширюватися для

розміщення додаткових категорій відходів, як того вимагає операція поводження з відходами.

Нейромережева модель слугує мозком автоматизованої системи сортування відходів, дозволяючи їй приймати швидкі та точні рішення щодо категоризації відходів. Завдяки навчанню, перевірці та постійній оптимізації модель забезпечує ефективну роботу системи, зменшуючи забруднення та збільшуючи вилучення вторинної сировини з потоків побутових відходів.

Для початку завантажено останню версію Python з офіційного сайту (<https://www.python.org/downloads/>). Дотримано інструкцій з інсталяції для вашої операційної системи. Переконано, що в процесі встановлення вибрано опцію "Додати Python до PATH", щоб забезпечити легкий доступ до Python з командного рядка.

Після встановлення Python створено віртуальне середовище для проекту. Відкрито командний рядок або термінал і виконано наступні команди:

Для Windows:

```
py -m venv project_env
project_env\Scripts\activate
Для macOS та Linux:
python3 -m venv project_env
project_env/bin/activate
```

Це створить віртуальне середовище з назвою 'project\_env' і активує його, гарантуючи, що залежності проекту будуть ізольовані від глобального середовища Python.

Коли віртуальне середовище активовано, був використаний менеджер пакетів pip для встановлення необхідних пакетів Python. Створено файл requirements.txt з переліком усіх необхідних пакетів та їхніх версій. Встановлено пакети за допомогою наступної команди:

```
pip install -r requirements.txt
```

Це призвело до встановлення необхідних бібліотек, зокрема Pandas, NumPy, TensorFlow, OpenCV та інших залежностей, необхідних для роботи системи сортування сміття.

Встановлено інтегроване середовище розробки Visual Studio Code та налаштовано його на використання віртуального середовища, створеного для проекту. Встановлено необхідні плагіни та розширення для редагування коду, налагодження та контролю версій, щоб спростити процес розробки.

Налаштовано змінні середовища для зберігання конфіденційних даних або налаштовано шляхи доступу до зовнішніх ресурсів, гарантуючи, що система залишатиметься безпечною та добре налаштованою протягом етапів розробки та розгортання.

Налаштовано проект за допомогою інструменту безперервної інтеграції (CI), такого як Jenkins, для автоматизації процесів збірки та тестування. Інтегровано репозиторій проекту з інструментом CI, щоб запускати автоматичні збірки і тестування щоразу, коли новий код потрапляє до репозиторію. Налаштовано інструмент CI так, щоб він повідомляв команду розробників про статуси збірки і результати тестування, що дозволить швидко виявляти і вирішувати будь-які проблеми інтеграції.

Попередня обробка та очищення даних є важливими кроками у створенні ефективної системи сортування відходів. Python пропонує широкий спектр бібліотек та інструментів для виконання цих завдань, гарантуючи, що дані, які використовуються для навчання та обробки в режимі реального часу, є точними, узгодженими та добре підготовленими.

Розпочато зі збору даних з камери. Переконано, що дані зберігаються у структурованому форматі або базі даних для легкого доступу та аналізу. Бібліотека Pandas мови Python є потужним інструментом для читання, завантаження та маніпулювання даними. Використано Pandas для читання даних з файлів (наприклад, CSV, Excel або баз даних) і завантаження їх у структури даних, такі як DataFrame. Це дозволяє ефективно маніпулювати даними та аналізувати їх.

```
import pandas as pd
# Завантаження даних з CSV-файлу
data = pd.read_csv('data.csv')
```

Очищення даних передбачає виявлення та виправлення помилок, невідповідностей та пропущених значень у наборі даних. До загальних завдань очищення відносяться:

- обробка пропущених значень шляхом заповнення їх відповідними даними або видалення рядків з пропущеними значеннями;
- виявлення та виправлення пропусків або помилкових точок даних;
- стандартизація форматів і одиниць виміру даних для забезпечення узгодженості.

```
# Обробка відсутніх значень
data.fillna(value, inplace=True)
# Видалення пропущених значень
data = data[(data['column'] > min_value) & (data['column']
< max_value)]
```

Перетворення даних передбачає перетворення даних у формат, придатний для аналізу. Це може включати розробку функцій, де створюються нові функції, або нормалізацію та масштабування для приведення даних до узгодженого діапазону.

```
# Feature engineering
data['new_feature'] = data['feature1'] + data['feature2']
# Нормалізація
з sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
data['normalized_feature'] =
scaler.fit_transform(data[['feature']])
```

Візуалізація даних має вирішальне значення для розуміння якості та характеристик набору даних. Бібліотеки Python, такі як Matplotlib та Seaborn, можна використовувати для створення таких візуалізацій, як гістограми, діаграми розсіювання та теплові карти для візуального дослідження даних.

```
import matplotlib.pyplot as plt
import seaborn as sns
# Створення гістограми
plt.hist(data['feature'], bins=20)
plt.xlabel('Feature')
plt.ylabel('Frequency')
plt.show()
```

Розділено дані на навчальну, перевірочну та тестову множини. Навчальний набір використовується для розробки моделі, валідаційний набір допомагає налаштувати гіперпараметри, а тестовий набір використовується для оцінки кінцевої продуктивності моделі.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
```

Набір даних містить дані зображень, тому застосовуються доповнення даних для збільшення їх розміру та різноманітності. Бібліотеки Python, такі як OpenCV та Keras, надають функції для доповнення даних зображень за допомогою таких перетворень, як обертання, масштабування та перевертання.

```
з keras.preprocessing.image import ImageDataGenerator
datagen = ImageDataGenerator(
    rotation_range=40,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)
```

Ефективна попередня обробка та очищення даних мають важливе значення для побудови надійної системи сортування відходів. Ці кроки

гарантують, що дані, які використовуються для навчання моделей машинного навчання та прийняття рішень у реальному часі, є точними та надійними, що сприяє підвищенню загальної ефективності системи.

Створення моделей машинного навчання є основою автоматизованої системи сортування відходів. Python пропонує багату екосистему бібліотек та інструментів для розробки та навчання складних моделей машинного навчання, що дозволяє точно класифікувати відходи та приймати рішення в режимі реального часу.

Підготовлено дані для навчання моделі, переконано, що вони правильно відформатовані та підходять для обраного алгоритму машинного навчання. Перетворено дані у необхідний вхідний формат, наприклад, вектори ознак, і відповідні цільові мітки для завдань керованого навчання.

```
# Підготовлено ознаки та мітки
x = data[['feature1', 'feature2', ...]]
y = data['label']
```

Обрано відповідний алгоритм машинного навчання на основі характеру задачі сортування відходів. Для таких завдань, як класифікація та сегментація зображень, зазвичай використовують згорткові нейронні мережі (CNN). Для інших завдань класифікації ефективними є такі алгоритми, як випадкові ліси, машини опорних векторів (SVM) або k-найближчих сусідів (KNN).

Використано бібліотеки Python, такі як TensorFlow, Keras або scikit-learn для навчання обраної моделі машинного навчання. Визначте архітектуру моделі, скомпільовано її з відповідною функцією втрат та оптимізатором і пристосовано до навчальних даних.

```
# Приклад навчання нейронної мережі за допомогою Keras
from keras.models import Sequential
from keras.layers import Dense
model = Sequential()
```

```

model.add(Dense(64, activation='relu',
input_dim=input_dim))
model.add(Dense(64, activation='relu'))
model.add(Dense(num_classes, activation='softmax'))
model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])
model.fit(X_train, y_train, epochs=10, batch_size=32)

```

Оцінено роботу навченої моделі, використовуючи відповідні метрики, такі як точність, точність, пригадування та оцінка F1. Використано бібліотеки Python, такі як scikit-learn, для обчислення цих метрик та створення вичерпного звіту про продуктивність.

```

# Оцінено модель
from sklearn.metrics import classification_report
y_pred = model.predict(X_test)
print(classification_report(y_test, y_pred))

```

Налаштовано гіперпараметри моделі для оптимізації її продуктивності. Використано такі методи, як пошук по сітці або випадковий пошук, щоб знайти найкращу комбінацію гіперпараметрів. Перехресна перевірка може допомогти в оцінці здатності моделі до узагальнення.

```

# Приклад налаштування гіперпараметрів за допомогою пошуку
по сітці у scikit-learn
from sklearn.model_selection import GridSearchCV
param_grid = {'C': [1, 10, 100, 1000], 'gamma': [0.001,
0.0001]}.
grid = GridSearchCV(SVC(), param_grid, refit=True,
verbose=2)
grid.fit(X_train, y_train)
print(grid.best_params_)

```

Оптимізовано навчену модель для ефективного виведення та розгорнуто її в системі сортування відходів. Використано методи стиснення

моделі, такі як квантування та відсікання, щоб зменшити розмір моделі та затримку. Обрано відповідну стратегію розгортання, наприклад, розгортання моделі на вбудованих системах або як веб-сервіс з використанням фреймворків, таких як Flask або FastAPI.

Постійно відстежується продуктивність розгорнутої моделі, щоб виявити будь-яку деградацію або дрейф у точності прогнозування. Впроваджено механізми перенавчання моделі на основі оновлених даних, щоб забезпечити її постійну ефективність та актуальність для операцій сортування відходів.

Створення інтуїтивно зрозумілого та зручного інтерфейсу має важливе значення для ефективного моніторингу та контролю системи сортування відходів. Python пропонує різні бібліотеки та фреймворки для розробки графічних інтерфейсів користувача (GUI), які дозволяють користувачам безперешкодно взаємодіяти з системою.

Обрано відповідний фреймворк графічного інтерфейсу, такий як Tkinter, PyQt та Kivy, виходячи з вимог проекту та уподобань щодо дизайну. Кожен фреймворк має власний набір функцій та можливостей для створення інтерактивних та візуально привабливих користувацьких інтерфейсів.

Розроблено макет інтерфейсу користувача, включивши відповідні компоненти, такі як кнопки, поля введення, випадаючі меню та візуальні дисплеї. Переконано, що елементи інтерфейсу добре організовані та візуально узгоджені, щоб забезпечити плавну та інтуїтивно зрозумілу роботу користувача.

Використано API обраного фреймворку графічного інтерфейсу для створення та налаштування компонентів користувацького інтерфейсу. Визначено обробники подій для взаємодії з користувачем та інтегровано їх з внутрішніми функціями системи сортування відходів, щоб уможливити відображення та контроль даних у реальному часі.

Інтегровано компоненти візуалізації даних у реальному часі у користувацький інтерфейс, щоб надати користувачам уявлення про процес



сортування відходів. Використано інтерактивні діаграми, графіки або індикатори виконання для представлення даних у візуально зрозумілому форматі.

Переконано, що користувацький інтерфейс є адаптивним та пристосованим до різних розмірів екрану та роздільної здатності. Впроваджено принципи адаптивного дизайну для створення узгодженого користувацького досвіду на різних пристроях, включаючи настільні комп'ютери, планшети та мобільні пристрої.

Додано елементи керування та поля введення для коригування системи за потреби. Реалізовано такі функції, як кнопки налаштування параметрів та відображення стану системи, щоб надати користувачам можливість взаємодіяти з системою сортування відходів у режимі реального часу.

## **4.2 Налаштування пристрою для сортування**

Усі дані між частинами системи передаються за допомогою REST. REST - це дуже популярна сьогодні архітектура для розробки і є підходом до комунікацій між двома дуже різними компонентами, який часто використовується при розробці веб-служб. Крім того, REST не використовує велику пропускну здатність, що робить його більш придатним для використання через мережу. Це робить REST кращим для SOAP, оскільки на відміну від SOAP вам не потрібно створювати сервер і клієнта. У випадку SOAP вам доведеться окремо створити серверну програму для обслуговування даних та клієнтську програму, яка вимагатиме дані.

Для того, щоб не було конфліктів між даними у різних системах найважливіший компонент нашої системи це сервер. Саме через це усі дані відправляються тільки через нього. Задача модулю «Вбудований пристрій» відправити дані за допомогою HTTP POST. Задача модулю «Веб-застосування» викликати HTTP GET для отримання даних від сервера.

На рис. 4.1 зобразимо, як на високому рівні пов'язані між собою частини системи.

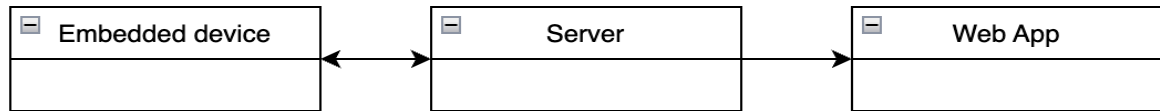


Рисунок 4.1 – Діаграма високорівневої архітектури системи

Уся взаємодія починається із серверу, у якому ми реєструємо новий пристрій. Задача пристрою – зберігати оновлення, зчитувати зображення з камери, запускати алгоритм класифікації та відправляти статистику на сервер.

Сервер має звичайну будову. У ньому реалізовані API для отримання даних від пристрою та відправлення оброблених оновлень на всі пристрої. Високорівнева діаграма принципу роботи зображена на рисунку 4.2 .

У цьому фреймворку вже реалізований механізм рендерінгу веб-сторінок пов'язаних із базою даних, створенню нових баз, що робить його зручним для задачі реєстрації приладів для сортування, зберігання її у базі та виводу на веб сторінку через html темплейт.

Було вирішено задачу класифікації відходів за допомогою архітектури ResNet, яка могла класифікувати 14 типів сміття з точністю до 0,999. Проблема полягає в тому, що ці мережі мають велику кількість параметрів, що ускладнює роботу на вбудованих системах з малим об'ємом оперативної пам'яті та слабким процесором. Це було основною причиною для майбутніх досліджень.

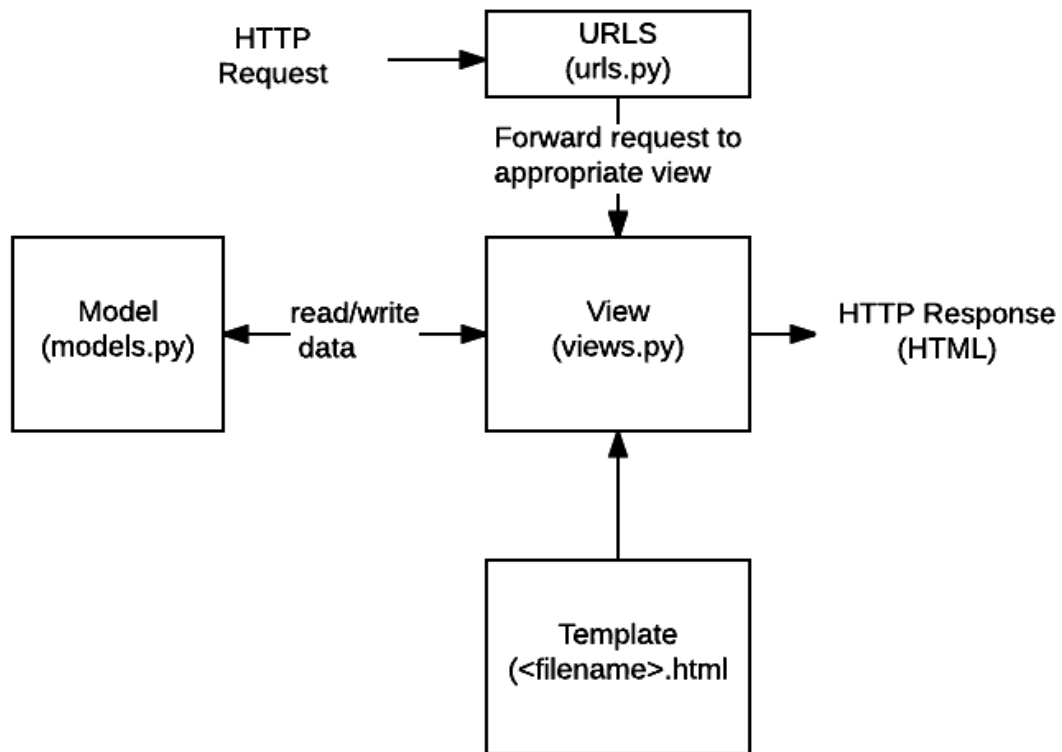


Рисунок 4.2 – Високорівнева діаграма принципу роботи

Одне із найбільших труднощів у машинному навчанні є невелика кількість наборів даних, тому в більшості випадків набори даних потрібно збирати з різних джерел, що призводить до різної якості зображення. Дані для цього дослідження були взяті з веб-сайту Kaggle [3]. Цей набір даних складається із зображень з Інтернету, набору даних «Класифікація сміття», створеного CCHANG [4] і «Набору даних одягу» [5]. Цей набір даних містить 15 150 зображень із 12 різних класів побутового сміття; папір, картон, біологічне сміття, метал, пластмаса, зелене скло, коричневе скло, біле скло, одяг, взуття, батарейки та сміття.

Передивляючись вручну зображення з набору даних, ми можемо помітити, що частина зображень була програмно вдосконалена, частина залишилася такою ж, як і була. Під програмно вдосконаленими зображеннями я маю на увазі зображення з відкоригованим кольоровим балансом, видаленим фоном. Це можна побачити для категорії батареї (див рис 4.1).

Але більшість зображень реальні, особливо в класі взуття та одягу. Розподіл зображень і доступні класи можна побачити на рис. 4.3.



Рисунок 4.3 – Три приклади зображень із класу «батареї»

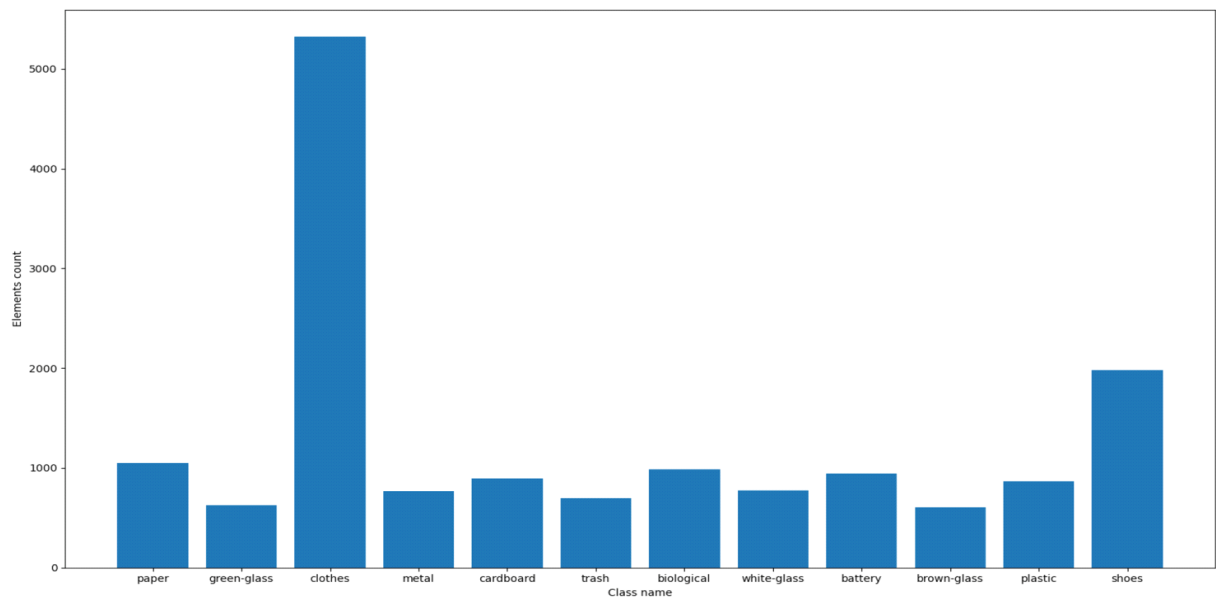


Рисунок 4.4 – Розподіл даних по класам

За базову модель було взято MobileNetV2 1.0. Рішення прийнято через кількість параметрів, яка дорівнює 3,50М. Завдяки цьому модель працює набагато швидше і займає менше пам'яті в порівнянні з ResNet-50 з 25,55 млн параметрів і ResNet-34 з 21,79 млн параметрів. Як механізм класифікації, повноз'єднаний шар був доданий до MobileNetV2, який отримує на вхід 1280 нейронів і має 12 вихідних нейронів. Функція втрат, яка застосовується до

цієї архітектури, є CrossEntropyLoss та обчислює втрати між цільовим і вхідним сигналом. У реалізації PyTorch CrossEntropyLoss логіти безпосередньо передаються до функції втрати, тому немає потреби застосовувати SoftMax (див рис. 4.3) перед обчисленням втрати, тоді як SoftMax і логарифм застосовуватимуться автоматично. Це призводить до отримання логітів як вихідних даних моделі з формою (batch size, 12).

$$s(x_i) = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}}$$

Рисунок 4.5 – Розрахунок функції активації SoftMax

Для обчислення до якого класу належить зображення було використано функцію `argmax` на вихід із нейронної мережі, яка повертає індекс найбільшого логіту, що означає, що модель з більшою ймовірністю віднесе зображення саме до цього індексу класу. Під час навчання використовується оптимізатор Adam зі швидкістю навчання, що дорівнює 0.1. Швидкість навчання була змінена під час процесу навчання, щоб не застрягти на мінімумі локальних функцій. В якості метрики для підтвердження того, що модель навчається, була взята accuracy [6]. Точність обчислюється як кількість вірних класифікацій, поділена на всі неправильні класифікації.

Згідно з рисунком 4.2 Спостерігається дисбаланс класів по відношенню до класу «одяг» через велику кількість зображень у цій категорії, але це можна вирішити, використовуючи `undersampling`, `oversampling` або використання методу балансу класів. `Oversampling` не використовувалася двілячись на нестачу даних для балансування всіх класів. `Undesampling` може призвести до перенавчання даних через зменшення розміру набору даних. Крім того, що ми використовуємо меншу нейронну мережу в

порівнянні з ResNet-50. Саме тому було обрано `RandomWeightedSampler`, який доступний у фреймворку PyTorch. Надаючи вагу кожному елементу даних, ми отримуємо майже однакову кількість всіх класів, переданих у нейронну мережу під час навчання, і не втрачаємо жодного зображення з набору даних, що запобігає перенавчанню нейронної мережі.

Якщо тренувати нейронну мережу без маніпуляцій із датасетом, то можемо отримати асигасу 0.9192, проте якщо ці результати будуть вірними лише на таких самих наборах даних, тому що в такому випадку нейронна мережа бачить тільки одягу і тому перенавчається. Перенавчання можна помітити побудувавши матрицю помилок (див. рис. 4.6).

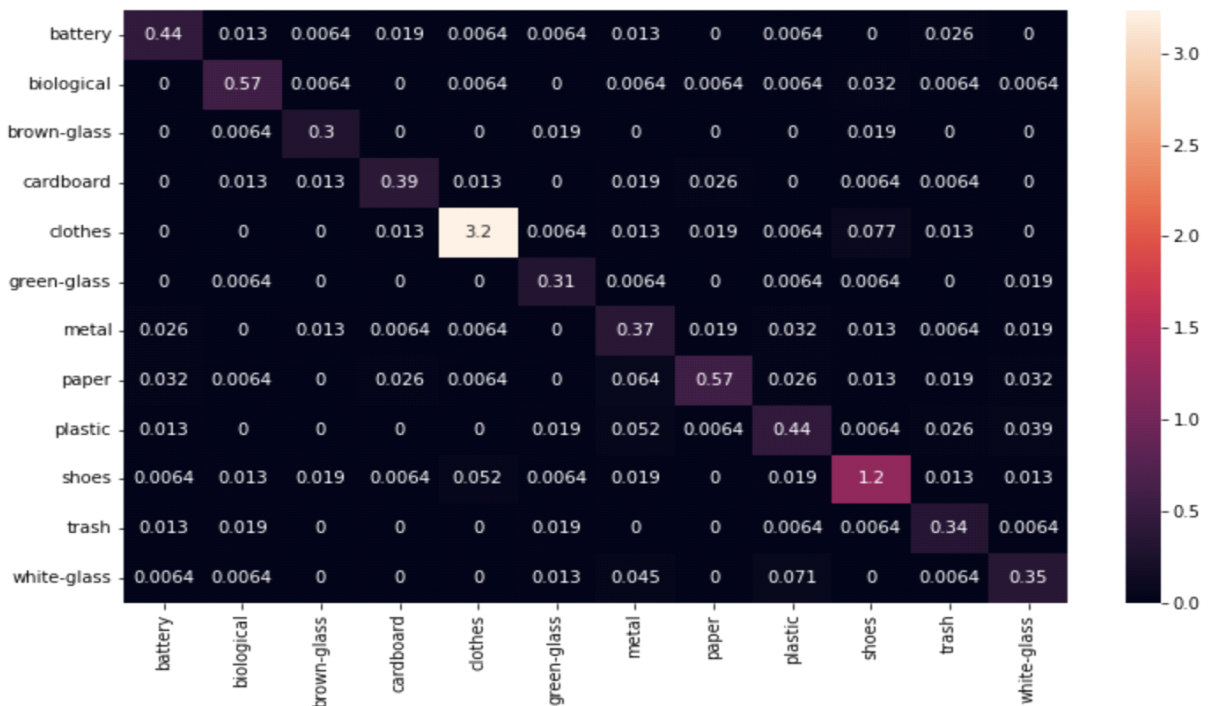


Рисунок 4.6 – Матриця помилок при навчанні моделі без використання `WeightedRandomSampler`

Навчальна нейронна мережа на базовому наборі даних показала точність, яка дорівнює 0.9192 на перевірконому наборі та 0.8713 на тестовому наборі даних, але більш глибокий аналіз із застосуванням матриці плутанини показує, що нейронна мережа перенавчана до класу одягу. Після

застосування RandomWeightedSampler із аугментацією даних результат було покращено. У результаті нейронна мережа демонструє точність на перевірочному наборі 86.6537, а функція втрати дорівнює 1.0779 (див. рис. 4.6). Процес навчання можна побачити на рисунку 4.5. По досягненню 100 епох навчання, модель вже стикається із локальним мінімумом, що не дозволяє їй надалі навчатися, проте цього результату вже достатньою для функціонування нашої системи.

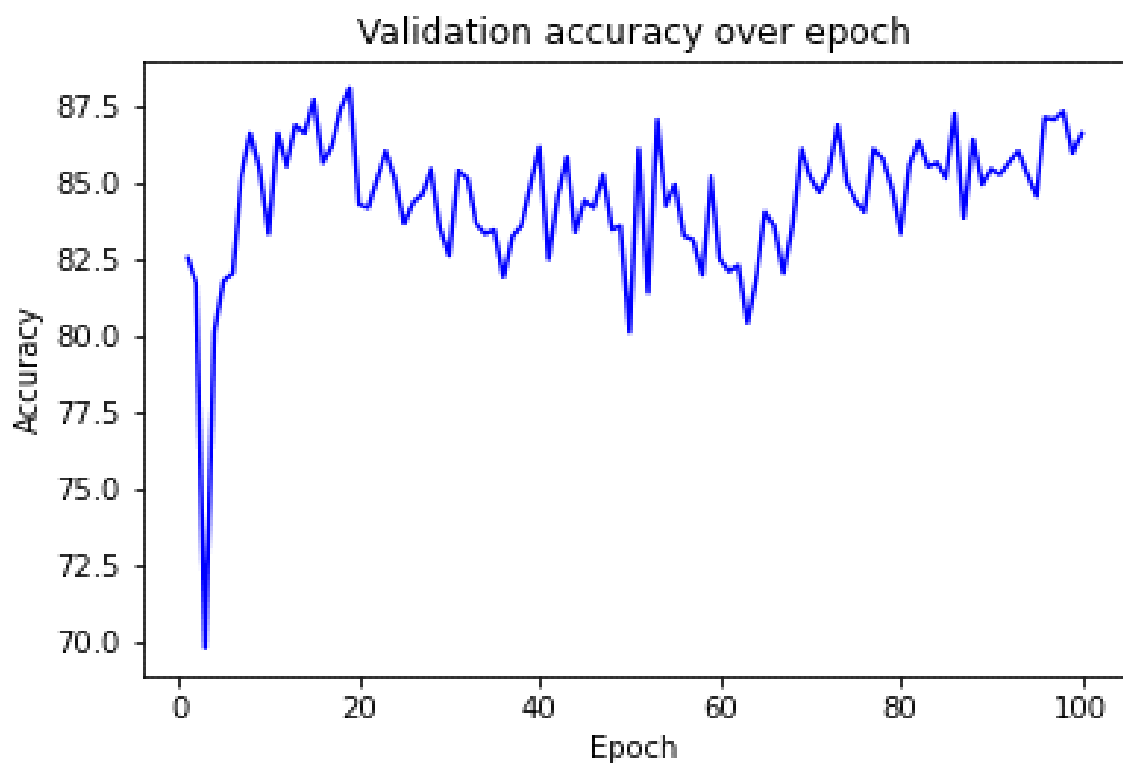


Рисунок 4.7 – Процес зміни асигасу в залежності від епохи

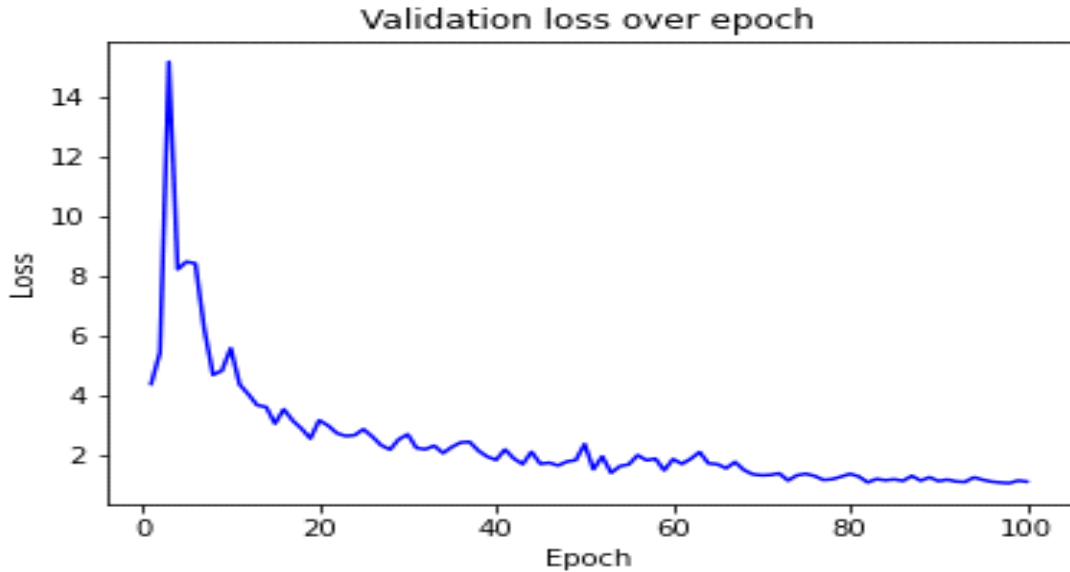


Рисунок 4.8 – Процес зміни loss в залежності від епохи

### 4.3 Результати роботи системи

Автоматизована система сортування відходів, призначена для відокремлення пластику, скла та паперу, продемонструвала чудові результати тестування в реальних умовах, про що свідчить комплексний аналіз результатів її роботи. У таблиці 5.1 описаний тестовий випадок «Класифікувати сміття».

Таблиця 5.1 – Тестовий випадок «Класифікувати сміття»

Дія	Очікуваний результат	Результат тесту
Передумова		
Запустити модуль камери на пристрої, який під'єднаний до відеокамери.	Модуль запущен, камера під'єднана.	Виконано
Кроки тесту		



## Продовження таблиці 5.1

Навести камеру на сміття	До файлу із логами записалась інформація про клас сміття. При досягненні більше 5 фреймів система робить остаточне передбачення та записує у файл із логами	Виконано
--------------------------	---	----------

У таблиці 5.3 описаний тестовий випадок «Відправити статистику на сервер».

Таблиця 5.2 – Тестовий випадок «Відправити статистику на сервер»

Дія	Очікуваний результат	Результат тесту
Передумова		
Запустити модуль камери на пристрої, який під'єднаний до відеокамери.	Модуль запущен, камера під'єднана.	Виконано
Кроки тесту		
Навести камеру на папір у пристрою із id = 0000000000000001	До файлу із логами записалась інформація про клас сміття. Система відправляє дані до серверу. На сервері відобразилась інформація про розпізнавання сміття із класом папір, id камери співпадає	Виконано

У таблиці 5.3 описаний тестовий випадок «Встановити оновлення».

Таблиця 5.3 – Тестовий випадок «Встановити оновлення»

Дія	Очікуваний результат	Результат тесту
Передумова		
Запустити модуль камери на пристрої, який під'єднаний до відеокамери.	Модуль запущен, камера під'єднана.	Виконано
Кроки тесту		
Завантажити нову версію програми на сервер	Новий бінарний файл зберігся у базі даних	Виконано
Зачекати 10 хвилин, поки пристрій не перевірить чи є новий файл оновлення	Пристрій побачив новий файл оновлення	Виконано
Зачекати поки пристрій завантажує оновлення	Пристрій завантажив оновлення	Виконано
Зачекати поки пристрій встановить оновлення	Пристрій встановив оновлення.	Виконано

У таблиці 5.4 описаний тестовий випадок «Відправити сигнал про неправильне сортування».

Таблиця 5.4 – Тестовий випадок «Відправити сигнал про неправильне сортування»

Дія	Очікуваний результат	Результат тесту
Передумова		
Запустити модуль камери на пристрої, який під'єднаний до відеокамери.	Модуль запущен, камера під'єднана.	Виконано
Кроки тесту		
Налаштувати пристрій на розпізнавання паперу	Пристрій налаштовано	Виконано
Навести на камеру фрагмент пластику	Пристрій почав запускати звуковий сигнал	Виконано

Таблиця 5.5 – Тестовий випадок «Зареєструвати систему розпізнавання»

Дія	Очікуваний результат	Результат тесту
Передумова		
Запустити сервер.	Сервер запущен.	Виконано
Кроки тесту		
Знайти на приладі іd та ввести його у спеціальну табу на сервері.	Таба знайдена, іd введений.	Виконано
Зачекати, поки сервер прийме сигнал від ристрою	Сервер прийняв сигнал, відправив відповідь на прилад, прилад запустив звуковий сигнал	Виконано

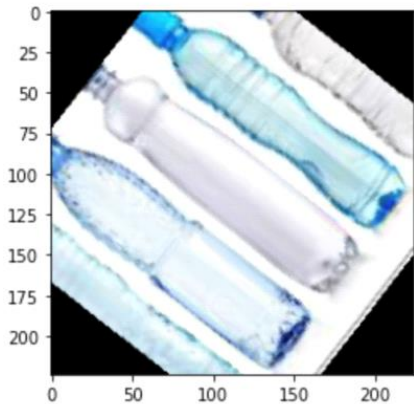
У таблиці 5.6 описаний тестовий випадок «Отримати звіт про використання системи».

Таблиця 5.6 – Тестовий випадок «Отримати звіт про використання системи»


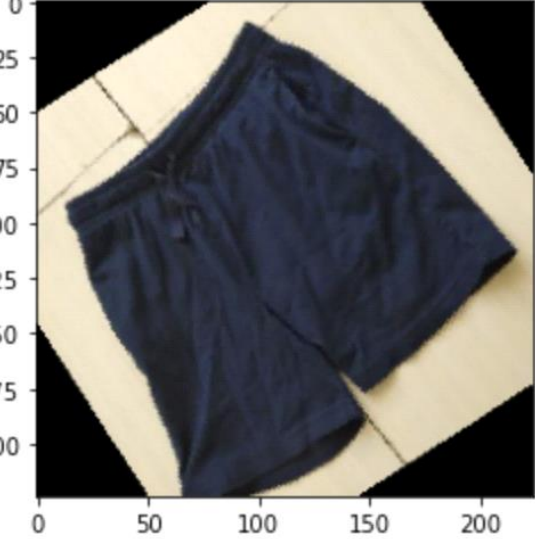
Дія	Очікуваний результат	Результат тесту
Передумова		
Запустити сервер	Сервер працює	Виконано
Кроки тесту		
Перейти до таби звітів, натиснути на кнопку генерування звіту	Звіт згенеровано	Виконано

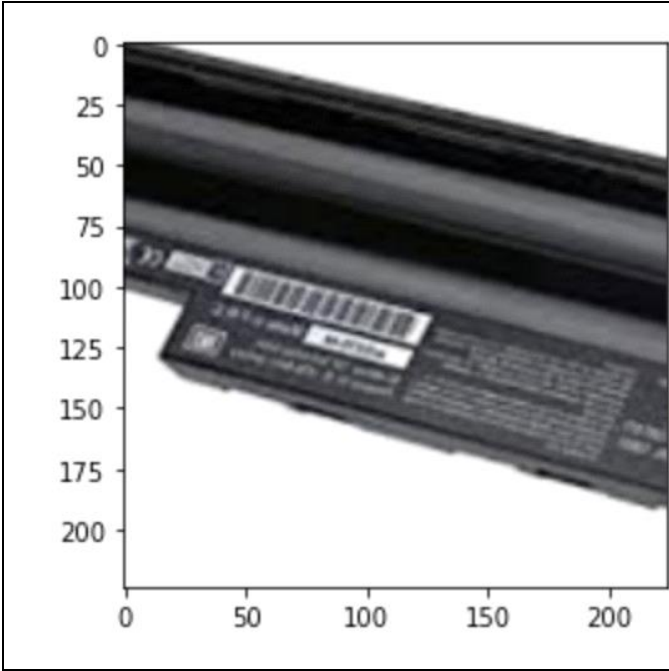
Для постановки експерименту нам потрібно довести, що наша нейронна мережа дійсно розпізнає тип сміття по фото. Передамо до неї фотографії сміття, які оберемо випадково та порівняємо з очікуваним результатом.

Таблиця 5.7 – Приклад роботи нейронної мережі

Передане зображення	Відповідь мережі
	<p>Пластик</p> <p>Результат відповідає дійсності.</p>

Продовження таблиці 5.7

	<p>Одяг</p> <p>Результат відповідає дійсності.</p>
	<p>Одяг</p> <p>Результат відповідає дійсності.</p>



Батарея

Результат відповідає дійсності.

## ВИСНОВКИ

У результаті виконання кваліфікаційної роботи магістра був створений мобільний застосунок для розпізнавання сміття, який використовує штучний інтелект і передові технології зондування для оптимізації процесів управління відходами. Для цього був проведений аналіз предметної області та порівняльний аналіз існуючих програм-аналогів.

В якості програмного засобу розробки було обрано середовище VSCode. Інструментом для розробки апаратної частини пристрою було обрано Arduino, який забезпечує гнучке та економічно ефективне рішення для розробки різних електронних проектів. Його універсальність, простота використання та підтримка великої онлайн-спільноти роблять її ідеальним вибором для розробки автоматизованих систем, включаючи механізми сортування відходів.

На етапі проектування були створені UML-діаграми варіантів використання (прецедентів), активності і послідовності та розроблені макети інтерфейсу застосунку. В роботі представлені можливості застосунку та поетапний приклад його використання. Було виконано тестування роботи застосунку та розпізнавання сміття. Проблем при використанні виявлено не було.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Екологічні проблеми сміттєвих полігонів – Проблеми сміттєвих полігонів. URL:  
[https://architecturesoft.org/%D0%90%D0%BD%D0%B0%D0%BB%D1%96%D0%B7\\_%D0%B2%D0%B8%D0%BC%D0%BE%D0%B3](https://architecturesoft.org/%D0%90%D0%BD%D0%B0%D0%BB%D1%96%D0%B7_%D0%B2%D0%B8%D0%BC%D0%BE%D0%B3) (дата звернення 20.10.2023)
2. Переробка відходів – Переробка відходів. URL:  
<https://architecturesoft.org/%D0%92%D0%B5%D0%B1%D0%B7%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%BD%D0%BE%D0%BA> (дата звернення 21.10.2023)
3. Дослідження та аналіз технологічного процесу сортування сміття – Сортування сміття. URL:  
<https://architecturesoft.org/%D0%91%D0%B0%D0%B3%D0%B0%D1%82%D0%BE%D0%BF%D0%BB%D0%B0%D1%82%D1%84%D0%BE%D1%80%D0%BC%D0%BD%D1%96%D1%81%D1%82%D1%8C> (дата звернення 21.10.2023)
4. Функціональні вимоги до програмної системи – Функціональні вимоги. URL:  
[https://architecturesoft.org/%D0%90%D0%BD%D0%B0%D0%BB%D1%96%D0%B7\\_%D0%B2%D0%B8%D0%BC%D0%BE%D0%B3](https://architecturesoft.org/%D0%90%D0%BD%D0%B0%D0%BB%D1%96%D0%B7_%D0%B2%D0%B8%D0%BC%D0%BE%D0%B3) (дата звернення 22.10.2023)
5. Arduino – Arduino. URL:  
[https://architecturesoft.org/%D0%90%D0%BD%D0%B0%D0%BB%D1%96%D0%B7\\_%D0%B2%D0%B8%D0%BC%D0%BE%D0%B3](https://architecturesoft.org/%D0%90%D0%BD%D0%B0%D0%BB%D1%96%D0%B7_%D0%B2%D0%B8%D0%BC%D0%BE%D0%B3) (дата звернення 20.10.2023)
6. Архітектура програмного забезпечення – Архітектура програмного забезпечення. URL:  
<https://architecturesoft.org/%D0%9A%D0%BB%D1%96%D1%94%D0%BD%D1%82->



%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80%D0%BD%D0%B0\_%D0%B0%D1%80%D1%85%D1%96%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0 (дата звернення 20.10.2023)

7. Модель нейронної мережі – Модель нейронної мережі. URL: [https://architecturesoft.org/%D0%90%D0%BD%D0%B0%D0%BB%D1%96%D0%B7\\_%D0%B2%D0%B8%D0%BC%D0%BE%D0%B3](https://architecturesoft.org/%D0%90%D0%BD%D0%B0%D0%BB%D1%96%D0%B7_%D0%B2%D0%B8%D0%BC%D0%BE%D0%B3) (дата звернення 20.10.2023)
8. Сортувальні механізми – Сортувальні механізми. URL: [https://architecturesoft.org/%D0%90%D0%BD%D0%B0%D0%BB%D1%96%D0%B7\\_%D0%B2%D0%B8%D0%BC%D0%BE%D0%B3](https://architecturesoft.org/%D0%90%D0%BD%D0%B0%D0%BB%D1%96%D0%B7_%D0%B2%D0%B8%D0%BC%D0%BE%D0%B3) (дата звернення 20.10.2023)
9. UML діаграма – UML діаграма. URL: [https://architecturesoft.org/%D0%90%D0%BD%D0%B0%D0%BB%D1%96%D0%B7\\_%D0%B2%D0%B8%D0%BC%D0%BE%D0%B3](https://architecturesoft.org/%D0%90%D0%BD%D0%B0%D0%BB%D1%96%D0%B7_%D0%B2%D0%B8%D0%BC%D0%BE%D0%B3) (дата звернення 20.10.2023)
10. Python – Python. URL: [https://architecturesoft.org/%D0%90%D0%BD%D0%B0%D0%BB%D1%96%D0%B7\\_%D0%B2%D0%B8%D0%BC%D0%BE%D0%B3](https://architecturesoft.org/%D0%90%D0%BD%D0%B0%D0%BB%D1%96%D0%B7_%D0%B2%D0%B8%D0%BC%D0%BE%D0%B3) (дата звернення 20.10.2023)
11. Кантелон М. Нейронна мережа в дії / Кантелон М. // Видавничий дім «Київ». – 2014. – С. 13-19. (дата звернення 20.10.2023)
12. Маркен С. Нейронні мережі / Маркен С. // Видавничий дім «Київ». – 2018. – С. 167-189. (дата звернення 20.10.2023)