

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**Приладобудівний факультет**

**Кафедра автоматизації та систем неруйнівного контролю**

До захисту допущено:

Завідувач кафедри

\_\_\_\_\_ **Юрій КИРИЧУК**

«\_\_» \_\_\_\_\_ 2023

**Дипломний проєкт**

**на здобуття ступеня бакалавра**

**за освітньо-професійною програмою «Комп'ютерно - інтегровані технології  
проєктування приладів»**

**спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології**

**на тему: «Система автоматичного сортування відходів в концепції**

**Industry 5.0»**

Виконав (-ла):

студент (-ка) IV курсу, групи ПМ-91

Денисов Владислав Семенович \_\_\_\_\_

Керівник:

Доцент, к.т.н., Нечай Сергій Олексійович \_\_\_\_\_

Рецензент:

Доцент к.т.н., Козир Олег Васильович \_\_\_\_\_

Засвідчую, що у цьому дипломному проєкті  
немає запозичень з праць інших авторів без  
відповідних посилань.

Студент (-ка) \_\_\_\_\_

Київ – 2023

**Пояснювальна записка**  
**до дипломного проекту**  
**на тему: «Система автоматичного сортування відходів в**  
**концепції Industry 5.0»**

Виконав (-ла):  
студент (-ка) IV курсу, групи ПМ-91  
Денисов Владислав Семенович

\_\_\_\_\_

Керівник:

Доцент, к.т.н., Нечай Сергій Олексійович

\_\_\_\_\_

**Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»**

**Приладобудівний факультет**

**Кафедра автоматизації та систем неруйнівного контролю**

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 151 Автоматизація та комп'ютерно-інтегровані технології

Освітньо-професійна програма «Комп'ютерно-інтегровані технології  
проекування приладів»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ **Юрій КИРИЧУК**

«\_\_» \_\_\_\_\_ 2023

**ЗАВДАННЯ**

**на дипломний проєкт студенту**

**Денисову Владиславу Семеновичу**

1. Тема проєкту «Система автоматичного сортування відходів в концепції Industry 5.0», керівник проєкту Нечай Сергій Олексійович, к.т.н., доцент, затверджені наказом по університету від «30 » травня 2023 р. № 2057-с
2. Термін подання студентом проєкту 17 червня 2023
3. Вихідні дані до проєкту швидкість руху стрічки 1 м/с, продуктивність 150 т/год, довжина проєкції конвеєра 45 м, ширина стрічки 700 мм, діапазон робочих температур від -5 до +50°C. Розробити програмне забезпечення, яке здатне

автоматично класифікувати різні види вторинної сировини на основі зображень, забезпечуючи ефективний процес сортування.

#### 4. Зміст пояснювальної записки

Титульна сторінка; Завдання до дипломного проекту; Анотація; Зміст; Вступ; Розділ 1. Актуальність теми та мета дослідження; 1.5 Огляд існуючих методів та технологій сортування відходів; 1.6 Використання нейронних мереж у сортуванні відходів; 1.7 Огляд нейронних мереж; 1.7.7 Концепція Industry 5.0 та її порівняння з іншими; Розділ 2. Реалізація програмного забезпечення; 2.3 Моделювання програмного забезпечення; Розділ 3. Тестування розробленого програмного забезпечення; 3.1 Опис контрольного прикладу; Висновки; Лістинг коду

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо): складальний кресленик (1 арк. ф. А1); специфікація конвеєрної стрічки (1 арк. ф. А4); схема алгоритму (1 арк. ф. А1); схема ілюстративна (1 арк. ф. А1); презентаційний аркуш (1 арк. ф. А1).

#### 6. Дата видачі завдання 10.04.2023

#### Календарний план

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1	Отримання теми	10.04.2023	
2	Пошук літератури	19.04.2023-26.04.2023	
3	Написання коду	28.04.2023-30.04.2023	
4	Розробка креслень	01.05.2023-17.05.2023	
5	Підготовка ПЗ	18.05.2023-17.06.2023	
6	Подання ДП на перевірку керівнику	17.06.2023	

Студент

Владислав ДЕНИСОВ

Керівник

Сергій НЕЧАЙ

## Анотація

Розробка програмного забезпечення для автоматизованого сортування вторинної сировини є актуальною та важливою проблемою у сучасному світі. У цій роботі було досліджено використання нейронних мереж у сортуванні відходів та розроблено програму, що здатна автоматично класифікувати вторинна сировина. Оглянуто різні методи та технології сортування, зокрема нейронні мережі, згорткові, рекурентні, глибокі, трансформери та віджаті мережі.

Архітектура програмного забезпечення була ретельно розроблена, вимоги до програми були визначені, а необхідні інструменти та бібліотеки були використані для реалізації. Проведено моделювання програми з використанням діаграм використання, послідовності та активності, що показують взаємодію компонентів та послідовність їх виконання.

Також було проведено тестування розробленого програмного забезпечення та наведено опис контрольного прикладу. Результати свідчать про ефективність використання нейронної мережі у процесі сортування вторинної сировини.

Загальним висновком роботи є те, що використання нейронних мереж у сортуванні відходів має великий потенціал для поліпшення точності та швидкості процесу сортування. Проте, для досягнення оптимальних результатів, потрібно продовжувати дослідження та вдосконалювати алгоритми та моделі, щоб забезпечити ефективну та стабільну роботу систем сортування вторинної сировини.

Ключові слова: розробка програмного забезпечення, автоматизоване сортування вторинних відходів, нейронні мережі, класифікація вторинних відходів, ефективність сортування вторинних відходів.

## **Annotation**

Development of software for automated sorting of secondary waste is an urgent and important problem in the modern world. In this work, the use of neural networks in waste sorting was investigated, and a program capable of automatically classifying secondary waste was developed. Various sorting methods and technologies are reviewed, including neural networks, convolutional, recurrent, deep, transformer, and squeezed networks.

The software architecture was carefully designed, the application requirements were defined, and the necessary tools and libraries were used for implementation. The program was modeled using usage, sequence, and activity diagrams, showing the interaction of components and the sequence of their execution.

The developed software was also tested, and a description of the control example was given. The results indicate the effectiveness of using a neural network in the process of sorting secondary waste.

The general conclusion of the work is that the use of neural networks in waste sorting has great potential for improving the accuracy and speed of the sorting process. However, in order to achieve optimal results, it is necessary to continue research and improve algorithms and models to ensure the efficient and stable operation of secondary waste sorting systems.

Keywords: software development, automated sorting of secondary waste, neural networks, classification of secondary waste, efficiency of secondary waste sorting.

# Зміст

ВСТУП.....	9
РОЗДІЛ 1. АКТУАЛЬНІСТЬ ТЕМИ ТА МЕТА ДОСЛІДЖЕННЯ.....	10
1.1 Актуальність теми.....	10
1.2 Мета дослідження .....	11
1.3 Технічне завдання .....	12
1.4 Поняття сортування відходів і його проблеми .....	13
1.4.1 Методи сортування .....	14
1.5 Огляд існуючих методів та технологій сортування відходів .....	18
1.6 Використання нейронних мереж у сортуванні відходів .....	21
1.7 Огляд нейронних мереж .....	24
1.7.1 Згорткові нейронні мережі.....	24
1.7.2 Рекурентні нейронні мережі .....	26
1.7.3 Сумісні нейронні мережі.....	27
1.7.4 Глибокі нейронні мережі.....	29
1.7.5 Віджати нейронні мережі.....	31
1.7.6 Vision Transformer.....	32
1.7.7 Концепція Industry 5.0 та її порівняння з іншими .....	37
Висновки до розділу 1 .....	40
РОЗДІЛ 2. РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	41
2.1 Аналіз вимог до програми та проектування її архітектури .....	41
2.1.1 Функціональні вимоги.....	41
2.1.2 Нефункціональні вимоги.....	42
2.2 Опис використаних засобів.....	44
2.2.1 Обґрунтування вибору мови програмування Python для розробки програми .....	44
2.2.2 Огляд моделі.....	45
2.2.4 Опис необхідних інструментів для реалізації програми .....	46
2.2.4 Огляд використаних бібліотек та фреймворків для нейронних мереж у Python .....	47
2.3 Моделювання програмного забезпечення .....	48
2.3.1 Опис методів.....	48
2.3.2 Діаграма використання.....	51
2.3.3 Діаграма послідовності .....	52

					<i>ДІПБ ПМ-91.11.1760.000 ПЗ</i>		
Зм.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Денисов В. С.			Літера	Аркуш.	Аркушів
Перевір.						7	53
Реценз.		Козир			КПІ ім. Ігоря Сікорського		
Н. Контр.							
Затверд.		Нечай С О					
					Система автоматичного сортування відходів в концепції Industry 5.0		

2.3.4 Діаграма активності .....	53
Висновки до розділу 2 .....	54
<b>РОЗДІЛ 3. ТЕСТУВАННЯ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ</b> .....	<b>55</b>
3.1 Опис контрольного прикладу .....	55
Висновки до розділу 3 .....	55
<b>ВИСНОВКИ</b> .....	<b>56</b>
<b>ДЖЕРЕЛА</b> .....	<b>57</b>
<b>ДОДАТКИ</b> .....	<b>58</b>
Лістинг коду.....	58
Додатки	

					<b>ДПБ ПМ-91.11.1760.000 ПЗ</b>		
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>			
<i>Розроб.</i>		<i>Денисов В. С.</i>			<i>Літера</i>	<i>Аркуш.</i>	<i>Аркушів</i>
<i>Перевір.</i>						8	53
<i>Реценз.</i>		<i>Козир</i>			<i>КПІ ім. Ігоря Сікорського</i>		
<i>Н. Контр.</i>							
<i>Затверд.</i>		<i>Нечай С. О.</i>					
					<i>Система автоматичного сортування відходів в концепції Industry 5.0</i>		



## ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

CNN - ConvolutionalNeuralNetwork

RNN - RecurentNeuralNetwork

DNN - DeepNeuralNetwork

ViT - VisionTransformer

					<i>ДПБ ПМ-91.11.1760.000 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		9

## ВСТУП

Розробка програмного забезпечення для сортування відходів є актуальною та значущою темою у сучасному світі. Завдяки зростаючій свідомості щодо екологічної проблематики та необхідності збереження ресурсів планети, сортування відходів стає все більш важливою складовою нашого повсякденного життя. Оптиміальне сортування дозволяє зменшити кількість відходів, що потрапляє на полігони, та сприяє переробці його відходів для подальшого використання.

В умовах зростаючої кількості відходів та обмежених ресурсів стає важливим впровадження автоматизованих систем сортування відходів. Технології штучного інтелекту та машинного навчання, зокрема нейронні мережі, надають потужний інструментарій для розробки програмного забезпечення, яке може класифікувати різні типи відходів з високою точністю. Це дозволяє зробити процес сортування ефективнішим, швидшим та економічно вигідним.

Основною метою дослідження є розробка програмного забезпечення, що використовує нейронні мережі для автоматичного сортування відходів. Програма повинна бути здатною аналізувати зображення відходів та визначати його тип, такий як: пластик, метал, папір або скло. Метою є створення ефективного і точного інструменту, який сприятиме поліпшенню процесу сортування вторинних матеріалів та сприятиме сталому розвитку.

					<i>ДПБ ПМ-91.11.1760.000 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		10

## РОЗДІЛ 1. АКТУАЛЬНІСТЬ ТЕМИ ТА МЕТА ДОСЛІДЖЕННЯ

### 1.1 Актуальність теми

У сучасному світі проблема відходів стає все більш актуальною та нагальною. Зростаюча кількість відходів, їх неконтрольована накопиченість та неправильна обробка мають серйозний вплив на навколишнє середовище та людське здоров'я. Одним із способів вирішення цієї проблеми є сортування відходів, що дозволяє відокремити корисні компоненти та забезпечити їх подальшу переробку та використання.

Однак, сортування відходів вимагає значних зусиль та ресурсів. Традиційні методи сортування, які базуються на ручному відборі та розділенні матеріалів, мають обмежену швидкість та низьку ефективність. У зв'язку з цим, розробка програмного забезпечення, яке може автоматизувати та поліпшити процес сортування, є насувною потребою.

Використання нейронних мереж у сортуванні відходів вважається одним із перспективних підходів. Нейронні мережі, зокрема глибокі нейронні мережі, володіють здатністю до автоматичного вивчення складних залежностей та патернів у вхідних даних. Їх застосування може значно покращити точність сортування та зменшити вплив людського фактору.

Розробка програмного забезпечення для сортування відходів з використанням нейронних мереж на мові Python відкриває широкі перспективи для вдосконалення цього процесу. Python, як мова програмування, забезпечує зручність розробки та має багатий набір бібліотек для машинного навчання та нейронних мереж. Використання нейронних мереж дозволить автоматизувати сортування відходів, забезпечивши високу швидкість, точність та ефективність.

Таким чином, розробка програмного забезпечення для сортування відходів з використанням нейронних мереж є насувною потребою сучасного суспільства. Це дозволить покращити екологічну ситуацію та забезпечити

					<i>ДПБ ПМ-91.11.1760.000 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		11

раціональне використання ресурсів, сприяючи сталому розвитку та збереженню довкілля для майбутніх поколінь.

Науковий підхід до розробки та використання нейронних мереж у сортуванні відходів забезпечує підвищення якості та надійності розробленого програмного забезпечення. Це дозволить не лише забезпечити більш ефективне сортування відходів, але й сприятиме розвитку наукових досліджень у сфері машинного навчання та нейронних мереж.

Таким чином, тема розробки програмного забезпечення для сортування відходів з використанням нейронних мереж на мові Python має велике значення в контексті екологічної сталості та ефективного управління відходами.

## 1.2 Мета дослідження

Метою дослідження є розробка програмного забезпечення для сортування відходів з використанням нейронних мереж на мові Python. Основними завданнями дослідження є:

1. Аналіз сучасних методів та технологій сортування відходів та їх переваг та обмежень.
2. Вивчення принципів роботи нейронних мереж та їх застосування в сфері сортування відходів.
3. Розробка алгоритму та моделі нейронної мережі для сортування відходів.
4. Реалізація програмного забезпечення на мові Python, що базується на розробленій моделі нейронної мережі.
5. Проведення тестування розробленої програми та оцінка її ефективності та точності сортування відходів.
6. Зробити висновки про результати дослідження та оцінити переваги та недоліки використання нейронних мереж у сортуванні відходів.

7. Надати рекомендації щодо подальшого вдосконалення програмного забезпечення та його можливого впровадження в практику сортування відходів.

Досягнення мети дослідження сприятиме поліпшенню процесу сортування відходів шляхом автоматизації та використання потужних інструментів машинного навчання. Розроблене програмне забезпечення може мати практичне значення в сфері екологічного управління та сприяти збереженню ресурсів та зменшенню негативного впливу відходів на навколишнє середовище.

### **1.3 Технічне завдання**

У даному підрозділі наведено технічне завдання для розробки програмного забезпечення для автоматизованого сортування вторинної сировини з використанням нейронної мережі.

#### ***Опис системи сортування***

Розробити програмне забезпечення, яке здатне автоматично класифікувати різні види вторинної сировини на основі зображень, забезпечуючи ефективний процес сортування.

#### ***Функціонал:***

1. Розпізнавання видів вторинної сировини: пластикових пляшок, паперу, скла, металу тощо.
2. Класифікація відходів на основі зображень, забезпечуючи високу точність (не менше 95%).
3. Здатність обробляти великий обсяг зображень і проводити класифікацію у реальному часі (не менше 10 кадрів в секунду).
4. Сумісність зі зовнішніми датчиками та механізмами для автоматичного сортування відходів.
5. Інтерфейс для візуалізації та моніторингу процесу сортування.

#### ***Вимоги до системи:***

1. Температурний режим роботи: від 10°C до 40°C.
2. Швидкість обробки зображень: не менше 10 кадрів в секунду.

					<b>ДПБ ПМ-91.11.1760.000 ПЗ</b>	Арк.
						13
Зм.	Арк.	№ докум.	Підпис	Дата		

3. Ефективність системи сортування: до 95% точності класифікації.
4. Масштабованість: здатність працювати з великим обсягом зображень та збільшувати продуктивність при потребі.
5. Надійність: система повинна бути стабільною та малопомітною для замовника, мінімізуючи кількість помилкових класифікацій.

### **Обмеження**

1. Використання нейронної мережі для класифікації зображень.
2. Застосування високоефективних алгоритмів обробки зображень та машинного навчання.
3. Використання сучасного апаратного забезпечення з достатньою продуктивністю для забезпечення вимог щодо швидкості та точності.

## **1.4 Поняття сортування відходів і його проблеми**

Сортування відходів є процесом відокремлення та класифікації відходів з метою подальшої переробки та повторного використання. Основна мета сортування полягає в зборі та розділенні матеріалів, які можуть бути використані для виготовлення нових продуктів. [1] Вторинну сировину включає різноманітні матеріали, такі як папір, картон, скло, метал, пластик та інші, які можуть бути перероблені і повторно використані.

Однак, процес сортування відходів стикається з рядом проблем, які впливають на його ефективність та результативність[2]:

1. Неправильне видалення відходів: Багато людей некоректно видаляють відходи, змішуючи різні види матеріалів в один контейнер. Це ускладнює процес сортування та підвищує його вартість.
2. Великий обсяг відходів: Зростаюча кількість відходів ставить під загрозу існуючі системи сортування. Перевантаженість і недостатня місткість сміттєпереробних заводів можуть призводити до незадовільної якості сортування та непродуктивності процесу.
3. Забруднення та забрудненість матеріалів: Деякі матеріали можуть бути забрудненими, що ускладнює їх подальшу переробку та використання.

Наприклад, продукти харчування, масла або інші забруднення можуть змішуватися з матеріалами та зменшувати їх якість.

4. Складність сортування: Деякі матеріали можуть бути складними для сортування, оскільки вони мають схожі властивості або неоднорідну структуру. Наприклад, пластикові упаковки з різних матеріалів можуть виглядати подібно, але потребують різних методів переробки.
5. Витрати та ефективність: Сортування відходів вимагає значних фінансових та технічних ресурсів. Використання традиційних методів сортування, які базуються на ручному відборі, може бути дорогим та неефективним.

Розв'язання цих проблем потребує використання нових технологій та підходів, зокрема використання нейронних мереж та програмного забезпечення, яке може автоматизувати та поліпшити процес сортування відходів. Це може забезпечити ефективніше використання ресурсів, зменшення впливу на довкілля та сприяти сталому розвитку.

#### 1.4.1 Методи сортування

Існують різні методи сортування відходів, включаючи механічні та програмні підходи. Нижче наведено опис кожного з них[3]:

1. Механічні методи сортування: Механічні методи використовують фізичні процеси, механізми та обладнання для відокремлення різних типів матеріалів. Основні методи механічного сортування включають:
  - Конвеєрне сортування: Використовуючи конвеєрну стрічку, вторинну сировину переносять через різні станції, де виконуються операції, такі як ручне відбирання або використання магнітних сепараторів, відсівних машин або оптичних сенсорів.



Рис.1 Конвеєрне сортування відходів

Вібраційне сортування: Відходи розподіляється на вібраційні платформи або сита, де застосовуються різні режими вібрації та кут нахилу, що допомагає відокремити матеріали за їх розміром, формою або вагою.[3]





Рис.2 Вібраційне сортування

Потік повітряного сортування: За допомогою потоку повітря відокремлюються легкі матеріали, наприклад папір та пластик, від важких, наприклад металу чи скла. Цей метод базується на різних швидкостях осадження різних матеріалів у потоці повітря. [4]



Рис.3 Магнітне сортування

Магнітне сортування: Застосовуються магнітні сепаратори для відокремлення феромагнітних матеріалів, наприклад, металевих упаковок, від інших матеріалів. [4]

## 2. Програмні методи сортування:

Програмні методи сортування включають використання комп'ютерних програм, аналітичних алгоритмів та штучного інтелекту для автоматизованого класифікації та сортування відходів. Основні програмні методи включають:

- Використання нейронних мереж: Нейронні мережі є потужним інструментом машинного навчання, який може бути використаний для автоматичного визначення типів матеріалів на основі обробки зображень або сенсорних даних. Нейронні мережі можуть навчитися розпізнавати і відокремлювати різні матеріали. [5]
- Використання комп'ютерного зору: За допомогою алгоритмів комп'ютерного зору, програми можуть аналізувати зображення

відходів та розпізнавати його компоненти, використовуючи характеристики, такі як колір, текстура або форма. [5]

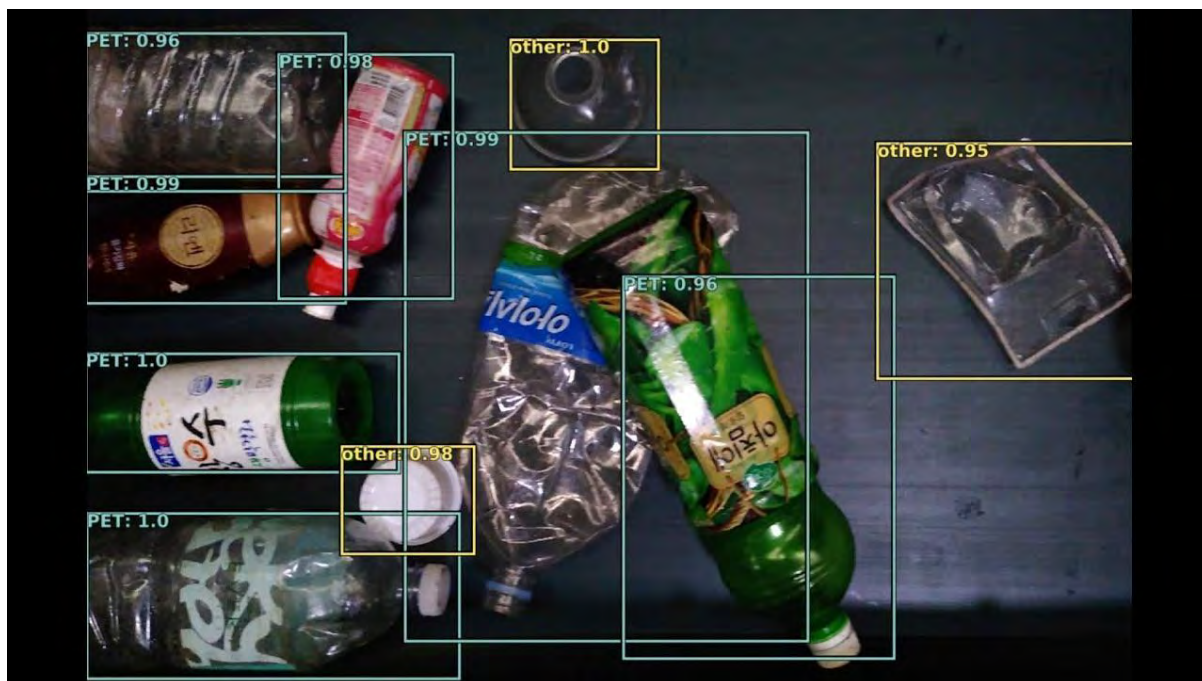


Рис.4 Використання комп'ютерного зору для сортування

- Аналіз сенсорних даних: Сенсорні пристрої, такі як інфрачервоні сенсори, металошукачі або оптичні сенсори, можуть бути використані для збору даних про матеріали та їх характеристики. Ці дані можуть бути оброблені програмами для автоматичного сортування. [5]

Комбінація механічних та програмних методів може забезпечити більш ефективно та точно сортування відходів, що призводить до зменшення втрат та покращення використання ресурсів.

### 1.5 Огляд існуючих методів та технологій сортування відходів

Огляд існуючих методів та технологій сортування відходів включає широкий спектр підходів, які використовуються в сучасних сміттєпереробних заводах та установках. Деякі з них описані нижче:

#### 1. Механічне сортування [5]:

- Балістичне сортування: Цей метод використовує механічні розділювачі, такі як балістичні сита та гідродинамічні класифікатори, для розділення матеріалів за їх розміром та щільністю.
- Магнітна сепарація: Використання магнітних сепараторів допомагає відокремити феромагнітні матеріали, зокрема металеві контейнери, від інших компонентів відходів.
- Потокова повітряна сепарація: Цей метод використовує потік повітря для відокремлення легких матеріалів, наприклад паперу та пластику, від важких, наприклад скла чи металу.

## 2. Оптичне сортування [5]:

- Використання сенсорних камер: Оптичні сенсори та камери використовуються для аналізу зображень відходів та його класифікації на основі кольору, форми, текстури тощо.
- Інфрачервона технологія: Інфрачервоні сенсори можуть використовуватися для виявлення теплового випромінювання різних матеріалів, що дозволяє їх відокремити та класифікувати.

## 3. Автоматизоване сортування з використанням штучного інтелекту (AI) [5]:

- Використання нейронних мереж: Нейронні мережі можуть бути навчені розпізнавати та класифікувати різні типи матеріалів на основі зображень або сенсорних даних.
- Машинне навчання: Методи машинного навчання дозволяють програмам самостійно навчитися розпізнавати та сортувати вторинну сировину на основі великої кількості тренувальних даних.

Ці методи та технології сортування постійно розвиваються і вдосконалюються. Використання автоматизованих систем з програмним забезпеченням, базованих на нейронних мережах та AI, виявляється все більш

					<i>ДПБ ПМ-91.11.1760.000 ПЗ</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		20

ефективним та точним способом сортування відходів, сприяючи покращенню відновлюваності ресурсів та зменшенню впливу на довкілля.

Опишемо переваги та недоліки різних методів сортування відходів в порівняльній таблиці:

Таблиця 1. Порівняння методів

Метод сортування	Переваги	Недоліки
Механічне сортування	<ul style="list-style-type: none"> <li>- Висока швидкість сортування</li> <li>- Можливість обробки великих обсягів відходів</li> <li>- Відокремлення матеріалів за їх фізичними характеристиками</li> </ul>	<ul style="list-style-type: none"> <li>- Обмежена точність сортування</li> <li>- Потребує постійного підтримання та регулювання обладнання</li> <li>- Вимагає великої кількості енергії та ресурсів</li> <li>- Відсутність можливості впливу на якість розсортованого матеріалу</li> </ul>

Продовження таблиці 1

<p>Оптичне сортування</p>	<ul style="list-style-type: none"> <li>- Висока точність класифікації</li> <li>- Можливість сортування за різними параметрами, такими як колір, форма, текстура</li> <li>- Менше впливу людського фактора на процес сортування</li> </ul>	<ul style="list-style-type: none"> <li>- Висока вартість технології та обладнання</li> <li>- Обмежена швидкість обробки</li> <li>- Вразливість до забруднень та змін у яскравості освітлення</li> </ul>
<p>Сортування з використанням нейронних мереж</p>	<ul style="list-style-type: none"> <li>- Здатність до самонавчання та адаптації до нових умов</li> <li>- Висока точність розпізнавання</li> <li>- Можливість розпізнавання різних типів матеріалів</li> </ul>	<ul style="list-style-type: none"> <li>- Потребує великої кількості тренувальних даних</li> <li>- Вимагає великої обчислювальної потужності</li> <li>- Висока складність розробки та підтримки моделей</li> </ul>
<p>Автоматизоване сортування з використанням AI</p>	<ul style="list-style-type: none"> <li>- Висока автоматизація та швидкість обробки</li> <li>- Можливість роботи з різноманітними типами відходів</li> <li>- Здатність до самоорганізації та оптимізації процесу сортування</li> </ul>	<ul style="list-style-type: none"> <li>- Потребує значних обчислювальних ресурсів</li> <li>- Вимагає складних алгоритмів та навчання моделей</li> <li>- Потребує постійного навчання та оновлення моделей</li> </ul>

Ця порівняльна таблиця надає загальний огляд переваг та недоліків різних методів сортування відходів. Важливо враховувати, що ефективність та

придатність різних методів можуть залежати від конкретних умов, типів відходів та технологічних вимог.

## 1.6 Використання нейронних мереж у сортуванні відходів

Використання нейронних мереж у сортуванні відходів є одним з передових методів, що базується на штучному інтелекті (AI). Нейронні мережі - це математичні моделі, які наслідують роботу людського мозку. Вони складаються зі штучних нейронів, які обробляють вхідні дані та генерують відповіді. Застосування нейронних мереж у сортуванні відходів дозволяє автоматично розпізнавати та класифікувати різні типи матеріалів на основі зображень або сенсорних даних.

Детальний процес використання нейронних мереж у сортуванні відходів може включати наступні кроки [5]:

1. Збір та підготовка тренувальних даних: Для тренування нейронної мережі потрібна велика кількість реальних зображень або сенсорних даних відходів, які мають бути позначені згідно з їх типами або категоріями. Ці дані можуть бути отримані шляхом фотографування сміттєвих контейнерів або за допомогою спеціальних сенсорних пристроїв.
2. Побудова архітектури нейронної мережі: Вибирається архітектура нейронної мережі, яка визначає кількість шарів, нейронів та зв'язків між ними. Також встановлюються параметри навчання, такі як швидкість навчання та критерії оцінки функції помилки.
3. Навчання нейронної мережі: За допомогою тренувальних даних нейронна мережа навчається розпізнавати різні типи матеріалів. Під час тренування, вхідні дані подаються на вхід нейронної мережі, а вихідні дані порівнюються з правильними мітками. На основі отриманих результатів мережа коригує свої внутрішні ваги та параметри, щоб зменшити помилки.

					<i>ДПБ ПМ-91.11.1760.000 ПЗ</i>	Арк.
						23
Зм.	Арк.	№ докум.	Підпис	Дата		

4. Тестування та оцінка продуктивності: Після навчання мережа тестується на нових невідомих даних для оцінки її точності та продуктивності. Виконується порівняння між прогнозованими мітками нейронної мережі та дійсними мітками. Зазвичай, навчання та тестування проводяться на окремих наборах даних, щоб оцінити загальну здатність мережі до розпізнавання різних типів відходів.

Переваги використання нейронних мереж у сортуванні відходів включають [5]:

- Висока точність: Нейронні мережі можуть досягати високої точності у розпізнаванні та класифікації різних типів матеріалів, що сприяє ефективному сортуванню.
- Автоматизація: Використання нейронних мереж дозволяє автоматизувати процес сортування, зменшуючи залежність від ручної праці та підвищуючи продуктивність.
- Гнучкість: Нейронні мережі можуть бути навчені розпізнавати нові типи матеріалів або адаптуватися до змін у складі відходів.
- Масштабованість: Використання нейронних мереж дозволяє обробляти великі обсяги відходів, що є важливим для промислових застосувань.

Однак, є також деякі недоліки, пов'язані з використанням нейронних мереж, навчання нейронної мережі вимагає наявності великого обсягу тренувальних даних, які мають бути коректно позначені. Збір та підготовка цих даних можуть бути трудомісткими та часозатратними процесами.

- Вимоги до обчислювальних ресурсів: Нейронні мережі вимагають значних обчислювальних ресурсів для тренування та виконання. Розгортання великих мереж на потужних серверах або використання спеціалізованих пристроїв може бути дорогим. [5]
- Складність розробки та підтримки: Розробка та підтримка нейронних мереж вимагає спеціалізованого знання та експертизи. Вибір архітектури мережі, налаштування параметрів та навчання моделей можуть бути складними завданнями. [5]



- Потреба у постійному навчанні: Якщо склад відходів змінюється або з'являються нові типи матеріалів, нейронна мережа може потребувати постійного навчання для підтримки актуальності та точності сортування. [5]

Усі ці фактори повинні бути враховані при впровадженні нейронних мереж у сортуванні відходів. Однак, незважаючи на недоліки, використання нейронних мереж має значний потенціал для покращення ефективності та точності сортування, зменшення впливу людського фактора та прискорення процесу переробки вторинних матеріалів.

## 1.7 Огляд нейронних мереж

Існує кілька типів нейронних мереж, які можуть бути застосовані у сортуванні відходів.

### 1.7.1 Згорткові нейронні мережі

Згорткові нейронні мережі (Convolutional Neural Networks - CNN): Цей тип мережі особливо ефективний у розпізнаванні образів. Вони використовують згорткові шари для виявлення різних ознак у зображеннях відходів, таких як текстури, форми та кольори. CNN зазвичай застосовуються для класифікації різних типів матеріалів на основі зображень. [6]

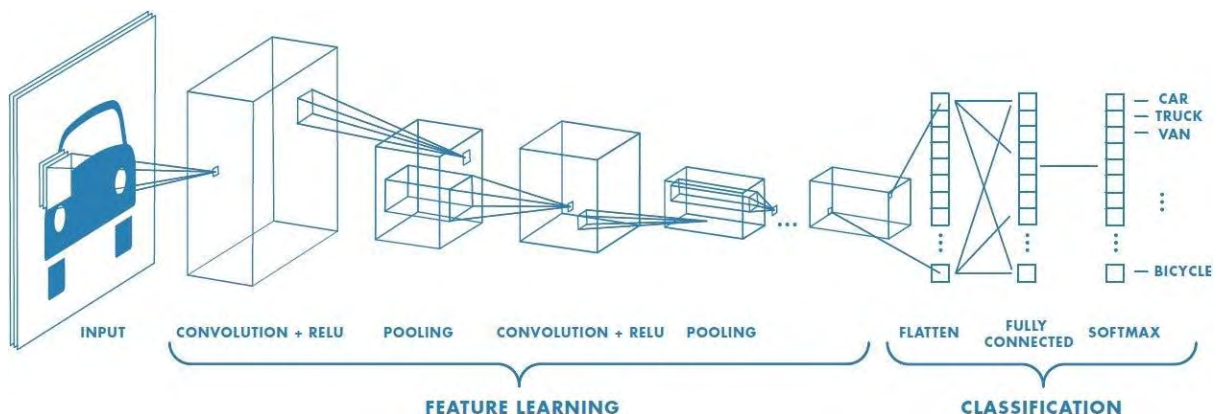


Рис.5 Згорткові нейронні мережі

Згорткові нейронні мережі (Convolutional Neural Networks - CNN) є потужними інструментами для обробки та аналізу зображень. Вони мають

здатність виявляти локальні залежності та шаблони в зображеннях шляхом застосування згорток і пулінгу. [6]

*Основний алгоритм згорткових нейронних мереж можна описати наступним чином [6]:*

1. Введення зображення: Перший шар згорткової нейронної мережі приймає на вхід зображення, яке представлено у вигляді матриці пікселів. Кожен піксель може мати кольорові компоненти (червоний, зелений, синій) або відтінки сірого.
2. Згортка: У згортковому шарі мережі застосовуються фільтри або ядра, які пройшли попереднє навчання. Ці фільтри ковзають по зображенню, виконуючи операцію згортки. Згортка виконує множення значень пікселів зображення та відповідних ваг фільтра, а потім сумує їх. Цей процес виявляє локальні особливості, такі як риси, контури та текстури на зображенні.
3. Активаційна функція: Після згортки до отриманого результату застосовується нелінійна активаційна функція, яка допомагає введеному сигналу проходити через нейрони мережі та визначає їх активацію. Це дозволяє моделі коректніше моделювати складні відношення між функціями в даних.
4. Пулінг: Після активаційної функції застосовується операція пулінгу, яка допомагає зменшити розмір зображення та кількість параметрів. Популярним методом пулінгу є максимальне пулінг, де вибирається максимальне значення у певному регіоні зображення. Це допомагає виділити найважливіші особливості зображення.
5. Повторення шарів: Згортка, активація та пулінг можуть повторюватись кілька разів для отримання більш складних функцій із зображеннями. Кожен наступний шар отримує на вхід результат попереднього шару.
6. Повнозв'язні шари: Після серії згорткових шарів і пулінгу використовуються повнозв'язні шари. Ці шари здійснюють

					<i>ДПБ ПМ-91.11.1760.000 ПЗ</i>	Арк.
						26
Зм.	Арк.	№ докум.	Підпис	Дата		

класифікацію або регресію, перетворюючи отримані ознаки у вихідні прогнози або класи.

7. Вихідний шар: Вихідний шар надає фінальний результат моделі. Зазвичай використовується функція активації, така як softmax, для отримання ймовірностей різних класів або вихідних значень.

Це загальний алгоритм згорткових нейронних мереж. Процес навчання полягає в оптимізації ваг фільтрів і параметрів моделі, щоб забезпечити точність і ефективність сортування відходів.

### 1.7.2 Рекурентні нейронні мережі

Рекурентні нейронні мережі (Recurrent Neural Networks - RNN) є класом нейронних мереж, які призначені для обробки та аналізу послідовних даних, таких як часові ряди, послідовності слів або сигнали. Вони володіють здатністю запам'ятовувати попередні стани та використовувати цю інформацію для аналізу та передбачення наступних елементів послідовності.

[7]

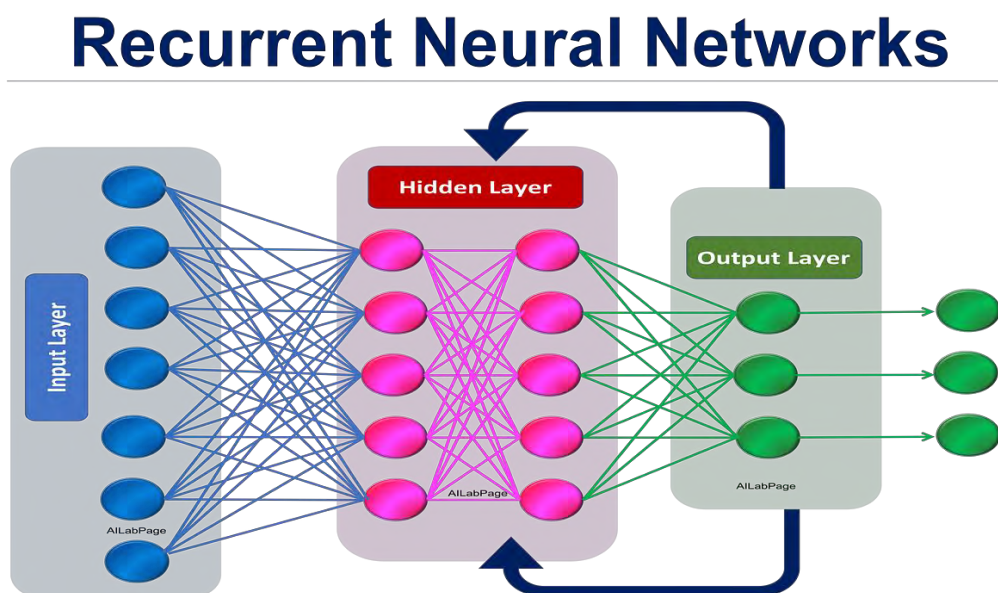


Рис.6 Рекурентні нейронні мережі

*Основний алгоритм рекурентних нейронних мереж можна описати наступним чином[7]:*

1. Введення послідовності: Перший крок полягає у введенні послідовності даних, які можуть бути представлені у вигляді послідовності векторів або послідовності слів. Кожен елемент послідовності розглядається як вхід для мережі.
2. Рекурентний шар: Рекурентний шар мережі використовує один або більше рекурентних нейронів, таких як LongShort-TermMemory (LSTM) або GatedRecurrentUnit (GRU). Ці нейрони мають внутрішні стани, які дозволяють запам'ятовувати інформацію з попередніх елементів послідовності та передавати її наступним елементам. Кожен елемент послідовності оброблюється послідовно шляхом подачі його на рекурентний шар.
3. Передача інформації: Після кожного кроку обробки рекурентний шар генерує вихід та оновлює свої внутрішні стани. Ці внутрішні стани передаються наступному елементу послідовності і використовуються як контекстна інформація для аналізу наступного елемента.
4. Вихідний шар: Після проходження всіх елементів послідовності отримується остаточний вихід від рекурентного шару. Цей вихід може використовуватись для передбачення наступних елементів послідовності, класифікації або регресії.
5. Зворотне поширення помилки: Після отримання вихідного результату використовується зворотне поширення помилки для налаштування ваг рекурентних нейронів. Це дозволяє моделі краще враховувати залежності та покращує її прогностичні можливості.
6. Повторення процесу: У разі потреби рекурентний шар може бути повторено кілька разів для більшої глибини обробки та складніших аналітичних завдань.

Рекурентні нейронні мережі мають здатність моделювати залежності в послідовних даних та запам'ятовувати довготривалі залежності, що робить їх потужним інструментом для аналізу та передбачення у сфері сортування відходів.

### 1.7.3 Сумісні нейронні мережі

Сумісні нейронні мережі (HybridNeuralNetworks) є комбінацією різних типів нейронних мереж з метою отримання кращих результатів аналізу та передбачення даних. Ці мережі поєднують у собі переваги різних архітектур, таких як згорткові нейронні мережі (CNN), рекурентні нейронні мережі (RNN), повнозв'язані шари та інші. [8]

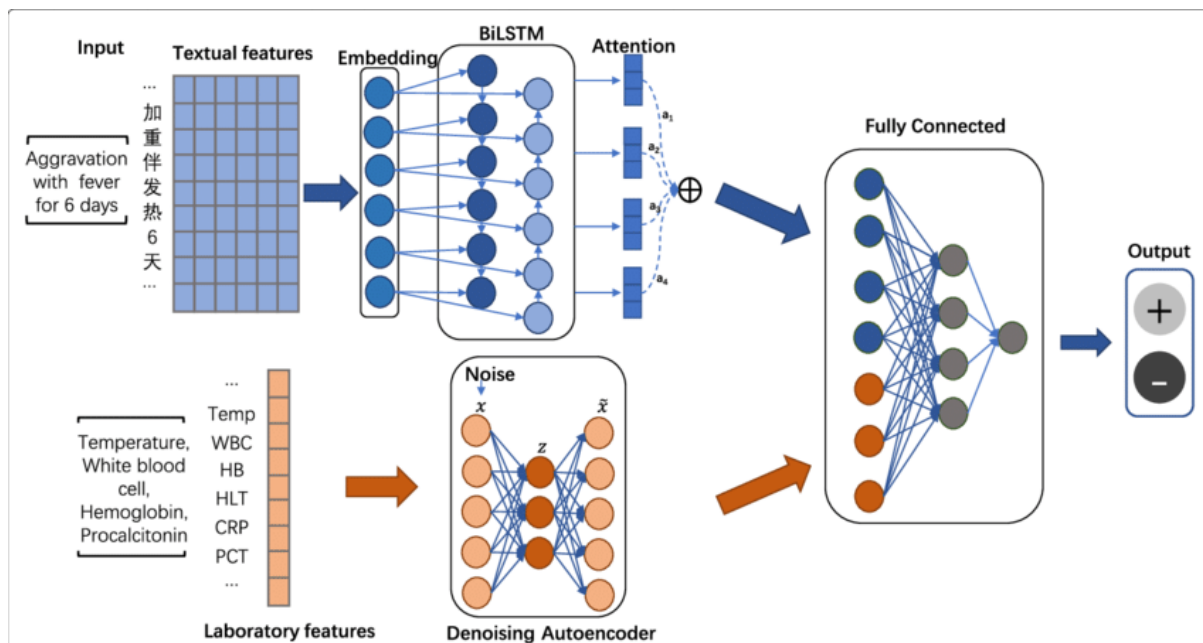


Рис. 7 Сумісні нейронні мережі

Основний алгоритм сумісних нейронних мереж можна описати наступним чином[8]:

1. Визначення архітектури: Початковий крок полягає у визначенні архітектури сумісної нейронної мережі, яка може включати різні типи шарів, такі як згорткові шари, рекурентні шари, повнозв'язані шари та інші. Вибір архітектури залежить від конкретного завдання аналізу сортування відходів.
2. Введення даних: Після визначення архітектури необхідно ввести дані для аналізу. Ці дані можуть бути вхідними зображеннями, текстом або іншими форматами, в залежності від завдання.
3. Проходження крізь мережу: Дані пройшов крізь різні типи шарів у сумісній нейронній мережі. Наприклад, зображення може пройти через

згорткові шари для виявлення особливостей, після чого результати передаються рекурентним шарам для аналізу послідовності чи контексту.

4. Обробка та аналіз даних: Кожен шар мережі обробляє вхідні дані, виконуючи математичні операції та застосовуючи активаційні функції. Результати передаються наступним шарам для подальшої обробки та аналізу.
5. Повнозв'язані шари та вихідний шар: Після проходження через різні типи шарів використовуються повнозв'язані шари для фінальної обробки даних та отримання вихідних результатів. Ці шари можуть здійснювати класифікацію, регресію або інші види аналізу відповідно до поставленої задачі сортування відходів.
6. Зворотнє поширення помилки та навчання: Після отримання вихідного результату застосовується зворотнє поширення помилки для налаштування ваг мережі. Цей процес дозволяє моделі покращувати свої передбачувальні можливості та адаптуватися до нових даних.

Сумісні нейронні мережі є потужним інструментом для сортування відходів, оскільки вони можуть використовувати різні типи шарів для аналізу різних аспектів даних. Це дозволяє отримувати більш точні та комплексні результати в порівнянні з використанням окремих типів нейронних мереж.

#### 1.7.4 Глибокі нейронні мережі

Глибокі нейронні мережі (DeepNeuralNetworks - DNN) є потужним інструментом для аналізу та передбачення даних. Вони складаються з багатьох шарів нейронів, які виконують послідовну обробку вхідних даних для отримання вихідного результату. [9]

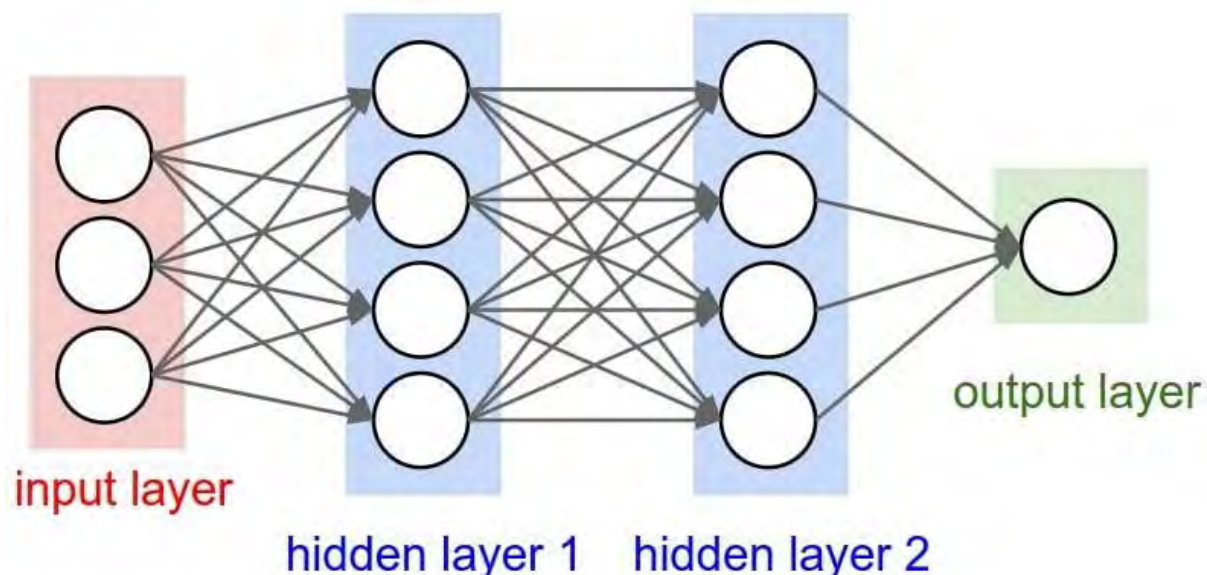


Рис. 8 Глибокі нейронні мережі

*Основний алгоритм глибоких нейронних мереж можна описати наступним чином[9]:*

1. **Визначення архітектури:** Перший крок полягає у визначенні архітектури глибокої нейронної мережі. Це включає вибір кількості шарів та кількості нейронів у кожному шарі. Глибокі нейронні мережі можуть мати від кількох до декількох сотень шарів, залежно від складності завдання та обсягу даних.
2. **Введення даних:** Дані для аналізу вводяться в глибоку нейронну мережу. Це може бути набір зображень, текстові дані або інші формати, залежно від конкретної задачі.
3. **Проходження крізь мережу:** Дані проходять через різні шари нейронів у глибокій мережі. Кожен шар обробляє вхідні дані, виконуючи математичні операції та застосовуючи активаційні функції. Вихід кожного шару стає вхідним для наступного шару, і процес продовжується до досягнення остаточного шару.
4. **Вихідний шар:** Після проходження через всі шари отримується вихідний результат від глибокої нейронної мережі. Вихідний шар може здійснювати класифікацію, регресію або інші види аналізу, відповідно до поставленої задачі.

5. Зворотнє поширення помилки та навчання: Після отримання вихідного результату використовується зворотнє поширення помилки для налаштування ваг мережі. Цей процес включає обчислення помилки між прогнозованими та очікуваними результатами та оновлення ваг для зменшення помилки. Цикл зворотного поширення помилки повторюється протягом тренувального процесу з метою покращення прогнозування.
6. Тестування та оцінка: Після навчання мережі її можна піддати тестуванню на нових невідомих даних для оцінки її точності та ефективності.

Глибокі нейронні мережі дозволяють моделювати складні залежності в даних та отримувати більш точні результати порівняно з неглибокими мережами. Вони широко використовуються у багатьох галузях, включаючи сортування відходів, завдяки своїй здатності виявляти складні шаблони та залежності у великих обсягах даних.

### 1.7.5 Мережі стискання та збудження

Мережі стискання та збудження (SE) — це тип архітектури нейронної мережі, яка спрямована на покращення продуктивності згорткових нейронних мереж (CNN) шляхом явного моделювання взаємозалежностей між каналами. Вони були представлені в статті «Мережі стискання та збудження» Цзе Ху, Лі Шен і ГанСуня в 2018 році. [10]

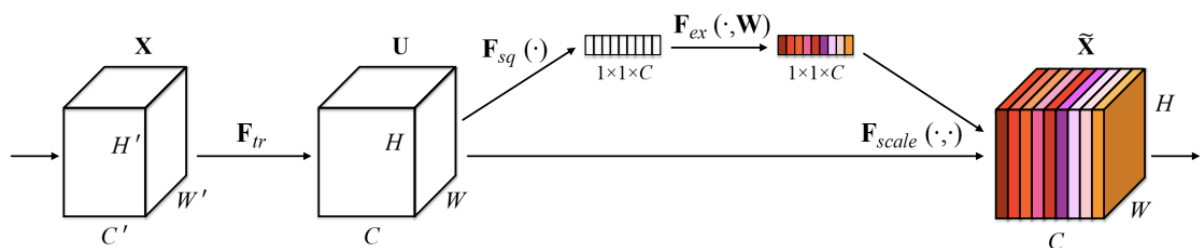


Рис. 9 Віджати нейронні мережі



*Основний алгоритм віджатих нейронних мереж можна описати наступним чином[10]:*

1. Функція зжимання (Firemodule): Основною ідеєю SqueezeNet є використання спеціальних модулів зжимання, відомих як Firemodules. Кожен Fire модуль має два шари: squeeze шар і expand шар.
2. Squeeze шар: У squeeze шарі використовується згортковий шар з невеликими фільтрами, що допомагає зменшити кількість вхідних каналів. Це зводить розмірність даних до більш компактної форми.
3. Expand шар: Після squeeze шару використовується комбінація згорткового шару з меншою кількістю фільтрів і 1x1 згорткового шару для розширення кількості каналів. Це дозволяє моделі отримувати більшу кількість признаков з меншою кількістю параметрів.
4. Послідовність Fire модулів: Віджаті нейронні мережі будуються шляхом послідовного розміщення Fire модулів один за одним. Це дозволяє моделі зберігати компакту архітектуру та знижувати кількість параметрів.
5. Глобальне середнє зведення (GlobalAveragePooling): У кінці мережі використовується глобальне середнє зведення, яке зменшує просторові розміри даних до одного значення на кожен канал. Це дозволяє знизити кількість параметрів та обчислювальні витрати.
6. Вихідний шар: Після глобального середнього зведення використовується повнозв'язаний шар або конволюційний шар для фінальної обробки та отримання вихідного результату.

Віджаті нейронні мережі, такі як SqueezeNet, зазвичай мають набагато меншу кількість параметрів порівняно зі стандартними глибокими мережами, при цьому досягаючи прийнятної точності. Це робить їх ефективними для застосувань з обмеженими обчислювальними ресурсами, включаючи сортування відходів, де швидкість та ефективність є важливими факторами.

## 1.7.6 VisionTransformer

VisionTransformer (ViT) - це передова архітектура нейромережі для обробки зображень, яка використовує трансформерну модель для досягнення високої точності в завданнях бачення комп'ютера. Він відрізняється від традиційних згорткових нейромереж (CNN), які домінують у баченні комп'ютера, тим, що використовує механізм трансформера для аналізу залежностей між пікселями у зображенні. [11]

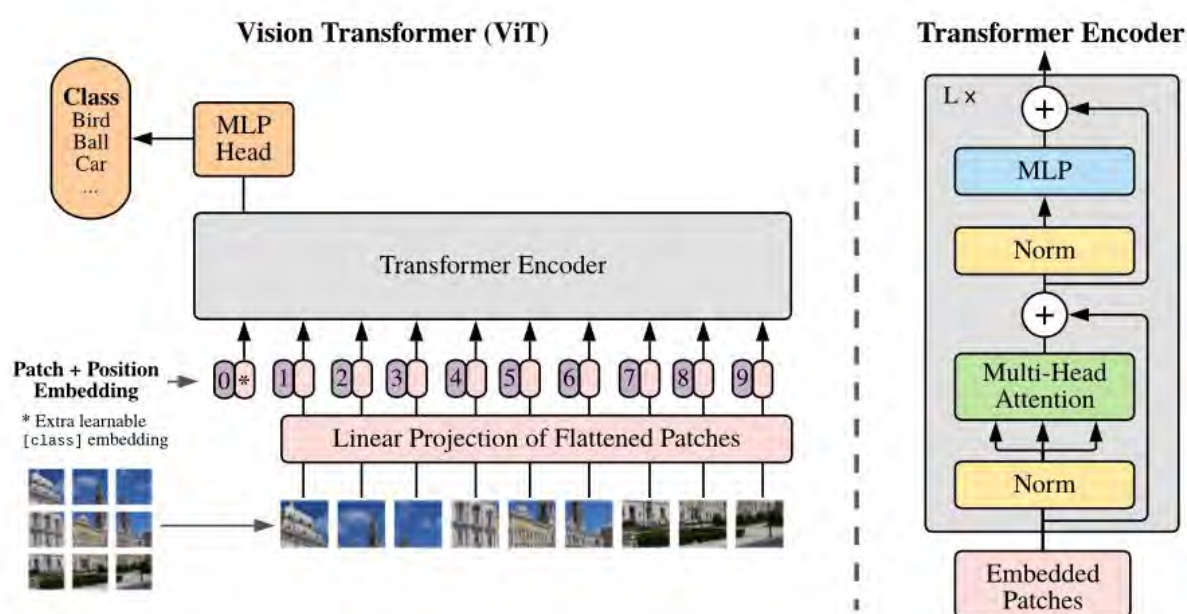


Рис. 10 VisionTransformer

Основна ідея за архітектурою ViT полягає в тому, щоб розглядати зображення як послідовність патчів, де кожен патч є вектором функцій. Потім ці патчі проходять через набір шарів трансформера для взаємодії між ними та отримання контекстної інформації. Нарешті, отримані контекстуалізовані патчі подаються до класифікатора для здійснення класифікації.

Основні переваги VisionTransformer[11]:

1. Ефективна обробка послідовностей: Використання трансформера дозволяє ефективно обробляти послідовність патчів у зображенні, взаємодіючи між ними та збираючи контекстуальну інформацію.

2. Розширюваність: Архітектура ViT може бути застосована до зображень різних розмірів, що робить її більш гнучкою та розширюваною в порівнянні з традиційними згортковими нейронними мережами.
3. Здатність до використання передньо навчених моделей: За допомогою передньо навчених моделей, таких як "Aalaa/Fine\_tuned\_Vit\_trash\_classification", можна використовувати готові ваги та використовувати їх для класифікації зображень в конкретних завданнях.

Однак, ViT також має свої обмеження:

1. Вимоги до обчислювальних ресурсів: Трансформерні моделі, включаючи ViT, вимагають значних обчислювальних ресурсів для тренування та інференсу. Вони можуть бути обмежені при застосуванні до великих наборів даних або на ресурсно обмежених пристроях.
2. Залежність від послідовностей: Використання патчів та послідовностей може бути менш ефективним для вирішення завдань, де важлива локальна інформація, наприклад, обробка текстів на зображенні.

В цілому, VisionTransformer є потужним інструментом для обробки зображень, який відкриває нові перспективи в області бачення комп'ютера та машинного навчання.

Таблиця 2. Порівняння нейронних мереж

Нейро нна мереж а	Алгоритм	Деталі	Переваги	Недоліки
Згортк ові	Використовують згорткові шари для виявлення	<ul style="list-style-type: none"> <li>- Здатність розпізнавати візуальні шаблони та особливості зображень</li> <li>- Ефективна обробка великих обсягів даних</li> <li>- Застосовується у багатьох візуальних завданнях, включаючи сортування відходів</li> </ul>	<ul style="list-style-type: none"> <li>- Висока точність у виявленні зображень</li> <li>- Швидкість обробки великих обсягів даних</li> </ul>	<ul style="list-style-type: none"> <li>- Вимагає значних обчислювальних ресурсів</li> <li>- Потребує великої кількості навчальних даних</li> <li>- Вразлива до перенесення на нові типи відходів</li> </ul>

Рекурентні	Використовують повернені зв'язки для	<ul style="list-style-type: none"> <li>- Здатність моделювати послідовність даних</li> <li>- Використовується для обробки текстових та послідовних даних</li> </ul>	<ul style="list-style-type: none"> <li>- Здатність розпізнавати залежності в послідовних даних</li> <li>- Ефективні у вирішенні задач машинного перекладу та обробки мови</li> </ul>	<ul style="list-style-type: none"> <li>- Вимагає великої кількості обчислювальних ресурсів</li> <li>- Схильність до проблем з довготривалим залежними послідовностями</li> </ul>
Гібридні	Поєднання різних типів мереж для	<ul style="list-style-type: none"> <li>- Комбінування переваг різних типів мереж</li> <li>- Можливість моделювати різноманітні види даних та завдань</li> </ul>	<ul style="list-style-type: none"> <li>- Покращена адаптивність та гнучкість моделі</li> <li>- Краща адаптація до різних типів даних</li> </ul>	<ul style="list-style-type: none"> <li>- Вимагає додаткової настройки та налаштування параметрів</li> </ul>
Глибокі	Використовують багат шарову архітектуру	<ul style="list-style-type: none"> <li>- Здатність моделювати складні залежності в даних</li> <li>- Висока точність у вирішенні складних завдань</li> </ul>	<ul style="list-style-type: none"> <li>- Здатність виявляти складні шаблони та залежності</li> <li>- Висока точність результатів</li> </ul>	<ul style="list-style-type: none"> <li>- Вимагають значних обчислювальних ресурсів</li> <li>- Потребують великих навчальних наборів даних</li> </ul>

Зм.	Арк.	№ докум.	Підпис	Дата
-----	------	----------	--------	------

ДПБ ПМ-91.11.1760.000 ПЗ

Арк.

37

				- Схильність до перенавчання
Віджат і (Squee zeNet)	Використовують спеціальні модулі зжимання	- Компактна архітектура з меншою кількістю параметрів та обчислювальних операцій	- Висока ефективність в умовах обмежених ресурсів	- Може втратити деяку точність порівняно з більш складними мережами

Ця таблиця надає порівняльний огляд розглянутих типів нейронних мереж, їх алгоритмів, переваг та недоліків у контексті застосування для сортування відходів. Залежно від конкретних вимог проекту і наявних ресурсів, можна обрати підходящий тип мережі для розробки програмного забезпечення сортування відходів на мові Python.

### 1.7.7 Концепція Industry 5.0 та її порівняння з іншими

Концепція Industry 5.0 є останньою еволюційною стадією в розвитку промисловості. Вона виникає на основі концепцій Industry 4.0 та Industry 3.0, але зосереджується на співробітництві між людьми та автоматизованими системами. Industry 5.0 покликана забезпечити гармонійний розвиток технологій та людського потенціалу, використовуючи інноваційні технології, такі як штучний інтелект, робототехніка, Інтернет речей та інші. [12]

Однією з основних характеристик Industry 5.0 є акцент на людині. У попередніх концепціях, таких як Industry 4.0, автоматизація та роботизація були центральними. Однак у Industry 5.0 пріоритетом є використання автоматизованих систем для полегшення та підтримки роботи людей, а не для їх заміщення. Головною метою Industry 5.0 є створення гнучкої та зручної робочої середовища, в якому люди та машини можуть взаємодіяти та співпрацювати. [12]

Одна з істотних рис Industry 5.0 - це поєднання людської креативності та здібностей з технологічними можливостями. Люди можуть вносити свої унікальні знання, досвід та інтуїцію в роботу з автоматизованими системами, що дозволяє розв'язувати складні завдання та створювати нові інновації. Наприклад, в медицині Industry 5.0 може сприяти появі інноваційних методів діагностики та лікування за участю лікарів та розумних медичних пристроїв.

Ще однією характеристикою Industry 5.0 є збільшення ролі соціальної відповідальності підприємств. При впровадженні інноваційних технологій та автоматизованих систем, компанії повинні враховувати екологічні та соціальні наслідки своїх дій. Створення стійкого розвитку та забезпечення благополуччя для всіх стає важливим завданням. [12]

Тепер давайте порівняємо Industry 5.0 з концепціями Industry 4.0 та Industry 3.0.

Перша істотна відмінність полягає у фокусі. Industry 3.0 була характеризована використанням автоматизованих систем в масовому виробництві. Industry 4.0 піднімає цю концепцію на новий рівень, впроваджуючи інтернет речей, аналітику даних та штучний інтелект. Однак Industry 4.0 все ще зберігає центральне місце для технологій та автоматизації. У свою чергу, Industry 5.0 ставить людину у центр і сприяє співробітництву між людьми та автоматизованими системами.

Друга відмінність полягає у взаємодії. У Industry 3.0 роботи між різними процесами були великим викликом, оскільки вони працювали окремо один від одного. Industry 4.0 пропонує підключення систем та даних для автоматизації

та оптимізації виробничих процесів. Industry 5.0 вдосконалює цю взаємодію, забезпечуючи спільну роботу між людьми та автоматизованими системами для досягнення кращих результатів.

Нарешті, третя відмінність - соціальна відповідальність. Industry 3.0 та Industry 4.0 більше зосереджені на підвищенні продуктивності та збільшенні прибутків. Усе ж таки, Industry 5.0 вводить новий елемент соціальної відповідальності, де увага приділяється екологічним та соціальним аспектам, а також створенню сприятливого середовища для праці та співіснування людей та технологій.

Отже, Industry 5.0 є новою етапом у розвитку промисловості, де співробітництво між людьми та автоматизованими системами, людська креативність та соціальна відповідальність виявляються на передньому плані. Ця концепція відмінна від попередніх Industry 3.0 та Industry 4.0 і покликана забезпечити гармонійний розвиток суспільства, економіки та технологій.

					<i>ДПБ ПМ-91.11.1760.000 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		40



## Висновки до розділу 1

У першому розділі роботи було розглянуто актуальність теми сортування відходів і визначена мета дослідження. Було з'ясовано, що сортування відходів є важливою проблемою в сучасному суспільстві, оскільки сприяє ефективному утилізації та переробці сміттєвих матеріалів, а також зменшенню негативного впливу на навколишнє середовище.

Метою дослідження є розробка програмного забезпечення для сортування відходів з використанням нейронних мереж на мові Python. Для досягнення цієї мети, в роботі проводиться огляд існуючих методів та технологій сортування відходів, а також вивчаються різні типи нейронних мереж і їх можливе застосування.

Було розглянуто поняття сортування відходів, його проблеми та методи сортування, які використовуються в цій галузі. Також був проведений огляд існуючих методів та технологій сортування відходів, що включає механічні та програмні засоби.

Окрема увага була приділена використанню нейронних мереж у сортуванні відходів. Було розглянуто загальний огляд нейронних мереж, включаючи згорткові, рекурентні, сумісні, глибокі та віджаті мережі. Кожен тип мережі був детально описаний, включаючи їх алгоритми та особливості.

					<i>ДПБ ПМ-91.11.1760.000 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		41

## РОЗДІЛ 2. РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 2.1 Аналіз вимог до програми та проектування її архітектури

#### 2.1.1 Функціональні вимоги

**Функціональні вимоги** - це вимоги, які описують функції, функціональність та поведінку програмного забезпечення. Вони визначають, що програма повинна робити і які функції вона має виконувати. Функціональні вимоги описують дії, введення та виведення, які має підтримувати програма. [13]

*Функціональні вимоги до програми сортування відходів за допомогою камери включають наступне[13]:*

1. **Захоплення зображення:** програма повинна мати можливість захоплювати зображення відходів за допомогою підключеної камери або вбудованої камери на пристрої.
2. **Обробка зображення:** програма повинна здійснювати обробку зображення для виділення сміттєвих матеріалів та їх характеристик. Це може включати розпізнавання об'єктів, сегментацію зображення, виявлення контурів тощо.
3. **Класифікація відходів:** на основі обробленого зображення програма повинна здійснювати класифікацію різних типів відходів. Наприклад, вона може визначати, чи належить предмет до пластику, скла, металу або паперу.
4. **Відображення результатів:** програма повинна відображати результати класифікації на екрані, надавати інформацію про тип відходів та, можливо, вказувати рекомендації щодо його подальшої обробки чи утилізації.
5. **Операційна швидкість:** програма повинна працювати достатньо швидко, щоб забезпечити реальний час або прийнятний час очікування при обробці зображення та виведенні результатів.

6. Навчання та адаптація: програма може мати можливість навчатися на основі зібраних даних для покращення класифікації та адаптації до нових типів відходів.
7. Інтерфейс користувача: програма повинна мати зручний та інтуїтивно зрозумілий інтерфейс користувача, що дозволить легко керувати та отримувати доступ до функцій сортування відходів.
8. Сумісність з різними пристроями: програма повинна бути сумісною з різними типами пристроїв, що мають камеру, наприклад, смартфони, планшети, веб-камери тощо.
9. Системні вимоги: програма повинна працювати на операційних системах, які підтримуються цільовими пристроями, і вимагати прийнятних обчислювальних ресурсів та пам'яті.

Ці функціональні вимоги визначають основні можливості програми сортування відходів за допомогою камери і служать як основа для подальшого проектування та розробки програмного забезпечення.

### 2.1.2 Нефункціональні вимоги

Нефункціональні вимоги - це вимоги, які описують характеристики, якості та властивості програмного забезпечення, які не відносяться безпосередньо до його функціональності. Вони визначають вимоги до продуктивності, надійності, безпеки, ефективності, зручності використання та інших аспектів програми. [13]

Нижче наведено приклади нефункціональних вимог до програми сортування відходів за допомогою камери:

1. Ефективність: програма повинна працювати ефективно, забезпечуючи швидку обробку зображення та класифікацію відходів без помітних затримок.
2. Точність класифікації: програма повинна забезпечувати високу точність класифікації відходів, знижуючи кількість помилок та неправильних результатів.

3. Надійність: програма повинна бути надійною, забезпечуючи стабільну та безперебійну роботу під час захоплення, обробки зображення та виведення результатів.
4. Сумісність з різними пристроями та операційними системами: програма повинна бути сумісною з різними пристроями, такими як смартфони, планшети, комп'ютери, і працювати на різних операційних системах, таких як Android, iOS, Windows, MacOS і Linux.
5. Зручність використання: програма повинна мати простий та інтуїтивно зрозумілий інтерфейс користувача, що дозволить легко керувати та отримувати доступ до функцій сортування відходів.
6. Безпека: програма повинна забезпечувати захист персональних даних користувачів та забезпечувати безпеку передачі і зберігання інформації.
7. Масштабованість: програма повинна мати можливість масштабуватись для обробки більшої кількості зображень та розширення функціональності для включення нових типів відходів.
8. Доступність: програма повинна бути доступною для людей з різними здібностями та потребами, забезпечуючи наявність адаптивного дизайну та можливостей налаштування.
9. Супровід та підтримка: програма повинна мати доступну технічну підтримку, оновлення та виправлення помилок для забезпечення безперебійної роботи та вирішення проблем користувачів.
10. Вартість: програма повинна бути доступною за вартістю та враховувати економічну цінність для користувачів та організацій, що впроваджують її.

Ці нефункціональні вимоги доповнюють функціональні вимоги та визначають властивості програми сортування відходів, які є важливими для її успішного використання та задоволення потреб користувачів.

					<i>ДПБ ПМ-91.11.1760.000 ПЗ</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		44

## 2.2 Опис використаних засобів

### 2.2.1 Обґрунтування вибору мови програмування Python для розробки програми

Вибір мови програмування для розробки програми сортування відходів залежить від різних факторів, таких як доступність засобів, продуктивність, екосистема, спільнота розробників та інші.

*Ось деякі обґрунтування вибору мови програмування Python для нашої програми:*

1. Простота використання: Python є однією з найпопулярніших і легких для вивчення мов програмування. Вона має простий синтаксис, який дозволяє розробникам швидко створювати програми та легко читати існуючий код. Це особливо важливо для команди розробників, які можуть мати різний рівень навичок програмування.
2. Багата екосистема: Python має широкую колекцію бібліотек та інструментів, що дозволяють розробникам швидко реалізовувати різноманітні функціональності. Для задачі сортування відходів, Python надає доступ до багатьох пакетів машинного навчання, обробки зображень та комп'ютерного зору, що можуть бути використані для реалізації алгоритмів класифікації та розпізнавання відходів на основі зображень.
3. Підтримка машинного навчання: Python є популярним вибором для розробки моделей машинного навчання та нейронних мереж. Є потужні бібліотеки, такі як TensorFlow, Keras та PyTorch, які надають зручні інтерфейси для побудови, тренування та використання нейронних мереж. Це дозволяє використовувати потужні алгоритми машинного навчання для класифікації відходів на основі зображень.
4. Переносимість: Python є крос-платформеною мовою програмування, що означає, що програми, написані на Python, можуть працювати на різних операційних системах, таких як Windows, macOS та Linux. Це забезпечує

більшу гнучкість та доступність для розгортання програми на різних платформах.

5. Швидкість розробки: Python має велику кількість готових до використання бібліотек та фреймворків, що дозволяють розробникам прискорити процес розробки. Наявність цих ресурсів знижує необхідність у високорівневому програмуванні і дозволяє фокусуватися на реалізації специфічних функцій сортування відходів.

Враховуючи ці фактори, Python є гарним вибором для розробки програми сортування відходів. Вона надає зручний і простий для використання інтерфейс, має потужну екосистему і підтримку для машинного навчання, що робить її ідеальним інструментом для реалізації даного проекту.

### 2.2.2 Огляд моделі

Використана модель в прикладах класифікації зображень - "Aalaa/Fine\_tuned\_Vit\_trash\_classification". Ця модель базується на архітектурі VisionTransformer (ViT), яка є однією з передових архітектур для обробки зображень. [14]

VisionTransformer використовує механізм трансформера, який є популярним у сфері обробки природної мови. Він використовує множину механізмів атенції для моделювання залежностей між пікселями зображення. Модель має претреновані ваги, що дозволяє їй класифікувати зображення на певні категорії.

"Aalaa/Fine\_tuned\_Vit\_trash\_classification" є моделлю, яка була навчена на великому наборі зображень з відповідною міткою. Це означає, що вона навчена розпізнавати різні категорії відходів, такі як пластик, метал, папір, скло тощо. [14]

Модель "Aalaa/Fine\_tuned\_Vit\_trash\_classification" може бути завантажена з використанням бібліотеки transformers, як показано в попередніх прикладах коду. Вона має попередньо навчені ваги, які дозволяють їй класифікувати зображення на основі навченого контексту. [14]

Ця модель є простою для використання і надає зручний інтерфейс для класифікації зображень. Вона може бути використана для різних завдань, пов'язаних з класифікацією відходів або інших типів зображень.

При використанні моделі важливо забезпечити правильну підготовку зображень та вказати вірні налаштування моделі.

Модель "Aalaa/Fine\_tuned\_Vit\_trash\_classification" має свої плюси та мінуси:

Плюси:

1. Висока точність: Модель базується на передовій архітектурі VisionTransformer (ViT) та була попередньо навчена на великому наборі зображень. Це дозволяє їй досягати високої точності при класифікації зображень відходів.
2. Зручний інтерфейс: Використання моделі здійснюється за допомогою бібліотеки transformers, яка надає зручний інтерфейс для завантаження та використання моделі.
3. Переносимість: Модель може бути використана на різних платформах та середовищах, що робить її переносимою та доступною для широкого кола користувачів.

Мінуси:

1. Обмежений набір категорій: Модель була навчена розпізнавати лише обмежений набір категорій відходів, таких як пластик, метал, папір, скло тощо. Якщо потрібно класифікувати інші типи відходів або використовувати модель для інших завдань, може знадобитися додаткове навчання або зміни у моделі.
2. Залежність від якості даних: Точність моделі може залежати від якості та репрезентативності навчального набору зображень. Якщо навчальний набір не включає різноманітність зображень або не відображає реальні умови використання, точність моделі може бути обмеженою.

#### 2.2.4 Опис необхідних інструментів для реалізації програми

Для реалізації програми для класифікації відходів на зображеннях використовуються наступні інструменти:

					<i>ДПБ ПМ-91.11.1760.000 ПЗ</i>	Арк.
						47
Зм.	Арк.	№ докум.	Підпис	Дата		

1. PIL (PythonImagingLibrary): Ця бібліотека надає функціональність для роботи з зображеннями, включаючи завантаження, копіювання, малювання та маніпулювання зображеннями. [15]
2. Requests: Ця бібліотека дозволяє здійснювати HTTP-запити для завантаження зображень з веб-серверів. [15]
3. Transformers: Ця бібліотека від компанії HuggingFace надає широкий набір інструментів для роботи з натренованими моделями глибокого навчання, зокрема зображеннями. У даному випадку використовуються класи ViTFeatureExtractor та ViTForImageClassification для попередньої обробки зображення та класифікації за допомогою моделі VisionTransformer. [15]
4. ImageDraw: Цей клас з бібліотеки PIL дозволяє малювати різні форми, текст та інші об'єкти на зображенні. [15]
5. ImageFont: Цей клас з бібліотеки PIL дозволяє вибрати та використати шрифт для надпису на зображенні. [15]

#### 2.2.4 Огляд використаних бібліотек та фреймворків для нейронних мереж у Python

При розробці нейронних мереж у Python, існує кілька популярних бібліотек та фреймворків, які надають потужні інструменти для побудови, навчання та використання моделей машинного навчання. Ось огляд деяких з них:

1. TensorFlow: TensorFlow є однією з найпопулярніших бібліотек для глибокого навчання. Вона надає широкий набір функцій для побудови та тренування нейронних мереж. TensorFlow також має високо розвинутий інтерфейс для роботи з великими наборами даних та можливостями розподіленого обчислення. [16]
2. Keras: Keras є високорівневим API для побудови та тренування нейронних мереж. Вона базується на TensorFlow і надає зручний та



простий у використанні інтерфейс. Keras дозволяє швидко створювати моделі та виконувати навчання з мінімальними зусиллями. [16]

3. PyTorch: PyTorch є ще одним популярним фреймворком для глибокого навчання, який набуває все більшої популярності. Вона пропонує динамічний обчислювальний граф, що дозволяє більшу гнучкість при визначенні та зміні моделей. PyTorch також надає багато інструментів для роботи зі зображеннями та іншими типами даних. [17]
4. Scikit-learn: Scikit-learn є бібліотекою машинного навчання, яка надає широкий спектр алгоритмів класифікації, регресії, кластеризації та інших методів. Вона має простий у використанні інтерфейс та добре підходить для початківців. [17]
5. Theano: Theano є однією з перших бібліотек для глибокого навчання у Python. Вона надає низькорівневі операції для обчислень на графах та ефективні обчислювальні оптимізації. Theano використовується в багатьох великих проектах з машинного навчання. [17]

Ці бібліотеки та фреймворки є потужними інструментами для розробки нейронних мереж у Python. Вони надають широкий спектр функцій та можливостей для побудови та навчання моделей, роботи з даними та визначення різноманітних алгоритмів машинного навчання. Кожна з цих бібліотек має свої особливості та переваги, тому вибір залежить від потреб та вимог вашого проекту.

## 2.3 Моделювання програмного забезпечення

### 2.3.1 Опис методів

Таблиця 2.1 Опис методів

Метод/Елемент	опис

ViTFeatureExtractor	Видалення функцій для моделі VisionTransformer (ViT). Попередньо обробляє зображення для введення в модель.
ViTForImageClassification	Попередньо підготовлена модель ViT, налаштована для класифікації зображень. Приймає попередньо оброблене зображення як вхідні дані та створює передбачувані логіти.
Image.open()	Відкриває файл зображення за наданою URL-адресою за допомогою бібліотеки PIL.
feature_extractor	Примірник ViTFeatureExtractor для попередньої обробки зображення та отримання вхідних тензорів.
model	Екземпляр ViTForImageClassification для виконання прямого проходу вхідних тензорів і генерації передбачених логітів.
inputs	Попередньо оброблені тензори зображень, отримані шляхом пропускання зображення через

	feature_extractor. Використовується як вхідні дані для modelпередбачення.
outputs	Вихід modelпісля виконання прямого проходу на вхідних тензорах. Містить передбачені логіти.
logits	Прогнозовані логіти для кожного класу, отримані з outputs.
predicted_class_idx	Індекс класу з найвищими прогнозованими логітами.
predicted_class_label	Мітка, що відповідає прогнозованому індексу класу, отриманому з конфігурації моделі.
ImageDraw.Draw()	Заняття з бібліотеки PIL для малювання на зображенні.
image_with_label	Копія оригінального зображення, до якого буде додано передбачену мітку класу.

draw.text()	Спосіб нанесення тексту на зображення. Приймає текст мітки, положення, колір заливки та шрифт як аргументи.
ImageFont.load_default( )	Функція з бібліотеки PIL для завантаження шрифту за замовчуванням для малювання тексту етикетки.
label_position	Позиція на зображенні, де буде намальовано текст мітки.
image_with_label.show( )	Метод відображення зміненого зображення з передбачуваною міткою класу.

### 2.3.2 Діаграма використання

Діаграма використання — в UML, це діаграма, на якій зображено, які дії може виконати користувач відповідно до функціоналу програми. [18]

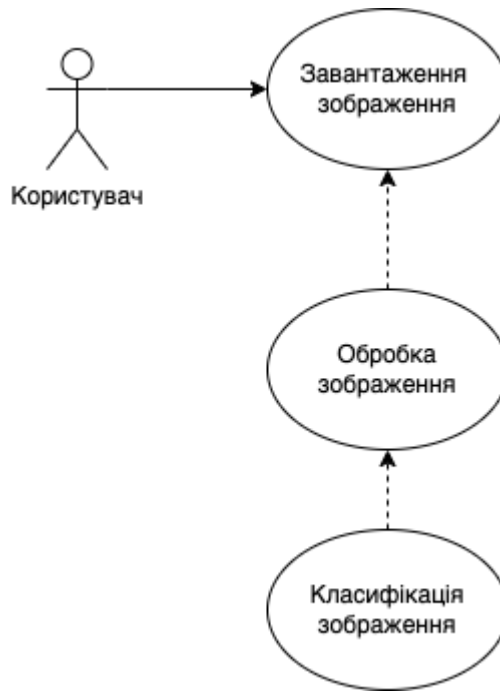


Рис.2.1 Діаграма використання

На цій діаграмі «Користувач» взаємодіє з «Системою» для виконання різних варіантів використання, таких як завантаження зображення, попередня обробка зображення, класифікація зображення та відображення результату.

### 2.3.3 Діаграма послідовності

Діаграма послідовності, також відома як діаграма взаємодії, є графічним зображенням взаємодії між різними об'єктами або компонентами в рамках системи. Вона показує послідовність повідомлень, які обмінюються об'єкти, та порядок, в якому ці повідомлення передаються. [18]



Рис.2.2 Діаграма послідовності

У цій діаграмі послідовності "Користувач" надсилає запит "Завантажити Зображення" до "Система". Система активується та виконує послідовність дій "Передпроцесування Зображення", "Класифікування Зображення" та "Відображення Результату". "Класифікування Зображення" включає виклик зовнішнього компонента "Класифікатор", який повертає результат класифікації до "Система".

### 2.3.4 Діаграма активності

Діаграма діяльності (діаграма активності) дозволяє відобразити послідовності дій, що реалізовані у програмному забезпеченні. [18]

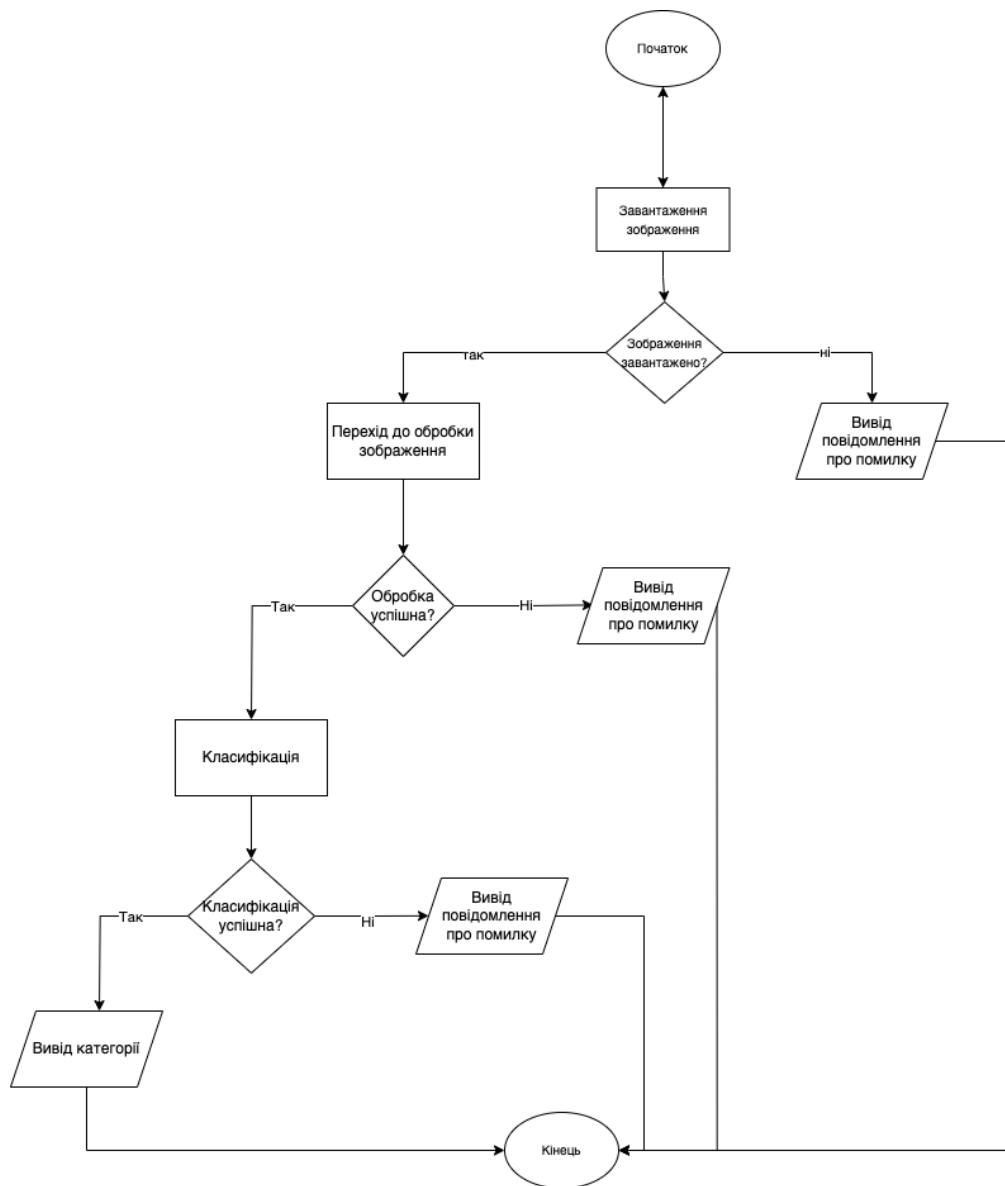


Рис.2.3 Діаграма активності

У цій діаграмі активності спочатку користувач вибирає опцію "Завантажити Зображення". Система отримує запит та перевіряє, чи існує зображення. Якщо зображення існує, система переходить до обробки зображення. Після успішної обробки система класифікує зображення. Якщо класифікація пройшла успішно, система відображає результат класифікації. У протилежному випадку система відображає повідомлення про помилку класифікації. Якщо зображення відсутнє, система відображає повідомлення про відсутність зображення. Діаграма завершується.

## Висновки до розділу 2

Було проведено аналіз вимог до програми та проектування її архітектури. Було розглянуто функціональні та нефункціональні вимоги до програми, які визначають основні функції та характеристики, що має мати програма.

У розділі також був описаний вибір мови програмування Python для реалізації програми. Python обрано через його простоту, широку підтримку та наявність багатофункціональних бібліотек для машинного навчання та обробки зображень.

Далі було проведено огляд використаної моделі - VisionTransformer (ViT) для класифікації зображень. Описано його основні переваги, такі як здатність до розпізнавання складних залежностей у зображеннях та висока точність класифікації.

Також були наведені описи необхідних інструментів для реалізації програми, таких як бібліотеки PIL (PythonImagingLibrary), requests і transformers. Ці інструменти забезпечують функціональність для роботи з зображеннями, включаючи завантаження, обробку, класифікацію та візуалізацію результатів.

Для нейронних мереж у Python був проведений огляд використаних бібліотек та фреймворків, таких як TensorFlow, PyTorch, Keras і scikit-learn. Ці інструменти надають широкі можливості для розробки, навчання та застосування нейронних мереж у різних областях.

В розділі також було проведено моделювання програмного забезпечення, де описано методи, які використовуються у програмі, а також наведено діаграму використання, діаграму послідовності та діаграму активності, що ілюструють взаємодію компонентів програми та послідовність їх виконання.

Загалом, розділ 2 надає повний огляд вимог, архітектури та інструментів, необхідних для реалізації програми для класифікації відходів на зображеннях. Він встановлює основу для подальшого розвитку та реалізації програмного забезпечення.



## РОЗДІЛ 3. ТЕХНОЛОГІЧНИЙ РОЗДІЛ

### 3.1. Вибір технологічної одиниці

У технологічній частині описані етапи проектування та конструювання оптико-механічної сортувальної машини. Етапи виробничого процесу пристрою та очікувана вартість чітко відображаються. Поступово пояснюються технічні вимоги та оцінюється виробничий потенціал компонента. Компонентом у цьому випадку є оптико-механічний сортувальний комплекс, до ключових компонентів якого входять конвеєрна стрічка, датчик оптичного спектру, який працює за тими ж принципами, що й спектрометр, і система продування, яка може включати повітродувку високого тиску.

Натяжний шнековий пристрій, кінцевий барабан, верхня та нижня роликові опори, стрічка, відхиляючий барабан, привід і приводний барабан є складовими стрічкового конвеєра.

Корпус, ротор, вал, ведуча шестерня, підшипник, гумова прокладка, всмоктувальний патрубок і напірний патрубок складають конструкцію типової повітродувки.

Будемо вважати, що конструкція однакова, оскільки оптичний датчик заснований на принципі дії спектрометра: джерело світла, дзеркало, вхідна зіниця, коліматор, дифракційна решітка, вихідна лінза, окуляр, детектор, записуючий пристрій.

Загалом ми маємо 9 компонентів в оптичному датчику, 8 частин у повітродувці та 8 частин у конвеєрі. Всього отримуємо 25 штук.

### 3.2. Визначення основних показників технологічності.

Абсолютний показник трудомісткості виготовлення виробу  $T_v$  можна розрахувати як:

$$(1) \quad T_v = \sum T_i,$$

де  $T_i$  – штучно-калькуляційний час.

На виготовлення 25 деталей, з яких складається оптико-механічна сортувальна машина було витрачено 150 нормо-годин, тобто  $T_B = 150$  нормо-годин.

Знаючи показник трудомісткості  $T_B$ , можна розрахувати критерій технологічності  $K_T$ :

$$K_T = T_B / T_{B0}, \quad (2)$$

Де  $T_{B0}$  – базовий показник трудомісткості ( $T_{B0} = 105$  нормо-годин)

Маємо:

$$K_{BT} = 150 / 105 = 1.42$$

Наступним кроком є визначення критерія уніфікації:

$$K_y = (N_y + n_y) / (N + n), \quad (3)$$

де  $N, n$  – всі складальні одиниці і деталі у виробі,  
 $N_y, n_y$  – кількість уніфікованих складальних одиниць і деталей в конструкції.

У нашому випадку уніфіковані складальні одиниці присутні:  $N_y = 6, n_y = 14, N = 6, n = 35$ .

$$K_y = (6 + 14) / (6 + 35) = 0.48.$$

Так як, отримане значення знаходиться в діапазоні  $K_y = 0.25 \dots 0.5$ , то можемо зробити висновок, що виріб має задовільну технологічність.

Коефіцієнт уніфікації складальних одиниць виробу  $K_{yN}$  розраховується як відношення кількості уніфікованих складальних одиниць до загального числа складальних одиниць у виробі:

$$K_{yN} = N_y / N, \quad (4)$$
$$K_{yN} = 6 / 6 = 1.$$

Отриманий результат  $K_{yN} > 0.4$ , це означає, що технологічність виробу прийнятна.

Коефіцієнт уніфікації деталей виробу  $K_{yn}$  розраховується як відношення кількості уніфікованих деталей до загального числа деталей у виробі:

$$K_{yn} = n_y / n, \tag{5}$$

$$K_{yn} = 14 / 35 = 0.4.$$

Отримавши результат, можна сказати що виріб задовольняє технологічність, оскільки потрапляє в діапазон  $K_{yn} = 0.3 \dots 0.6$ .

### 3.3. Структурна схема складання.

Створення структурної схеми виконується перед складанням виробу. Це необхідно для вказівки фундаментальної конструкції виробу, кількості складальних одиниць, кількості основних частин і назви. На цьому малюнку показано основні етапи процесу складання виробу та кількість операцій складання.

Схема складання та технологічна схема відрізняються головним чином тим, що на схемі складання чітко не зображено порядок складання основних компонентів виробу.

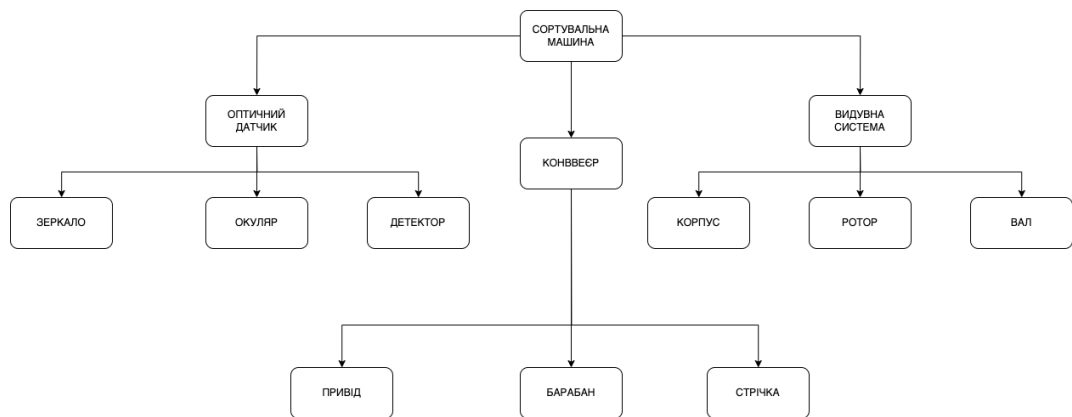


Рис. 3.3.1 Структурна схема

### 3.4. Технологічна схема складання.

Технічна монтажна схема призначена для відображення порядку монтажу, періодичності, змісту процесу налагодження, порядку забезпечення з'єднань. Так само, як і на структурній схемі, показано компоненти та частини. Ви можете змінити конструкцію пристрою, створивши технологічний план складання, який упорядковує технологічний процес.

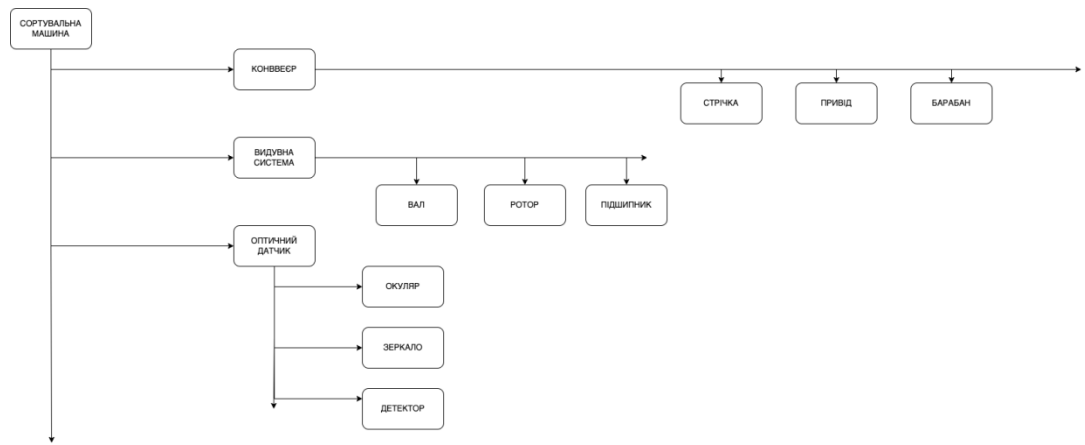


Рис. 3.4.1 Технологічна схема.

### 3.5. Розрахунок геометричної точності складання

Ланка	Номинальний розмір	Допустиме відхилення	Тип ланки
A <sub>1</sub>	50	+0.2	Збільшуюча
A <sub>2</sub>	56	-0.01	Зменшуюча
A <sub>3</sub>	123	-0.1	Збільшуюча
A <sub>4</sub>	150	+0.3	Зменшуюча
A <sub>5</sub>	35	-0.15	Зменшуюча
A <sub>6</sub>	250	-0.10	Збільшуюча

Номинальний розмір вихідної ланки подаємо у вигляді формули:

$$A_{\Delta} = \sum A_{i \text{ збіл}} - \sum A_{i \text{ зменш}}, (6)$$

$$A_{\Delta} = 50 - 56 + 123 - 150 - 35 + 250 = 182 \text{ [мм]},$$

Розрахуємо допуск замикаючої ланки за умови повної взаємозаміни:

$$\delta_{\Delta} = \sum_{i=1} |\delta_i|, (7)$$

$$\delta_{\Delta} = 0.2 + 0.01 + 0.1 + 0.3 + 0.15 + 0.1 = 0.86 \text{ [мм]},$$

Знайдемо середину поля допуску замикаючої ланки:

$$K_3 = \sum K_{i \text{ збіл}} - \sum K_{i \text{ зменш}}, (8)$$

$$K_3 = 0.055 - 0.005 - (-0.05 + 0.02 - 0.10 - 0.05) = 0.23 \text{ [мм]},$$

Знайшовши середину поля допуску замикаючої ланки, знайдемо верхнє та нижнє відхилення розміру допуску замикаючої ланки:

$$\delta_{\Delta B} = K_3 + 0.5 * \delta_{\Delta} \quad (9)$$

$$\delta_{\Delta H} = K_3 - 0.5 * \delta_{\Delta} \quad (10)$$

$$\delta_{\Delta B} = 0.23 + 0.5 * 0.86 = 0.66 \text{ [мм]}$$

$$\delta_{\Delta H} = 0.23 - 0.5 * 0.86 = -0.2 \text{ [мм]}$$

Отже, ми отримали наступні розміри, що дозволять провести повну взаємозаміну  $A_{\Delta} = 182^{+0.715}_{-0.175} \text{ [мм]}$

Тепер розрахуємо допуск замикаючої ланки, але методом неповної взаємозаміни:

$$\delta_{\Delta} = \frac{1}{3} * 3 * \sqrt{\sum_{i=1}^5 (\delta_i)^2},$$

(11)

$$\delta_{\Delta} = \frac{1}{3} * 3 * \sqrt{\sum_{i=1}^5 (0.2 + 0.01 + 0.1 + 0.3 + 0.15 + 0.1)^2} = 5.58 \text{ [мм]},$$

Розрахуємо середину поля розсіювання, беручи до уваги коефіцієнт для відносного розсіювання з ланками  $\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = \alpha_5 = \alpha_6 = 0$ :

$$K_3 = K_1 + 0.5 * \alpha_1 * \delta_1 + K_2 + 0.5 * \alpha_2 * \delta_2 -$$

$$-[(K_3 + 0.5 * \alpha_3 * \delta_3) + (K_4 + 0.5 * \alpha_4 * \delta_4) + (K_5 + 0.5 * \alpha_5 * \delta_5) + (K_6 + 0.5 * \alpha_6 * \delta_6)], (56)$$

$$K_3 = 0.055 + 0.5 * 0 * (-0.05) - 0.2 + 0.5 * 0 * (-0.15) -$$

$$-[-0.05 + 0.5 * 0 * (-0.1)] + (0.2 + 0.5 * 0 * (-0.06)) + (-0.105 * 0 * (-0.15)) + (-0.06 * 0 * (-0.06))]$$

$$= 0.23 \text{ [мм]}.$$

Для верхньої та нижньої границі поля розсіювання замикаючої ланки маємо:

$$\delta_{\Delta B} = K_3 + 0.5 * \delta_{\Delta},$$

(12)

$$\delta_{\Delta H} = K_3 - 0.5 * \delta_{\Delta},$$

(13)

$$\delta_{\Delta B} = 0.23 + 0.5 * 5.58 = 3.02 \text{ [мм]},$$

$$\delta_{\Delta H} = 0.23 - 0.5 * 5.58 = -2.56 \text{ [мм]},$$

Отримуємо:

$$A_{\Delta} = 182_{-2.545}^{+3.085} \text{ [мм]}.$$

### Висновки до технологічного розділу:

В процесі роботи над технологічним розділом визначено необхідні показники, складено технологічну і складальну схеми. В результаті розрахунків отримали коефіцієнт уніфікації деталей виробу 0.32, що задовольняє вимогам на технологічність. Також розраховано розміри, які дозволять провести повну взаємозаміну

					<i>ДПБ ПМ-91.11.1760.000 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		62

$A_{\Delta} = 182_{-0.175}^{+0.715}$  [мм] і  $A_{\Delta} = 182_{-2.545}^{+3.085}$  [мм]. Можна зробити висновок, що апарат є достатньо технологічним для запуску у серійне виробництво.

					<i>ДПБ ПМ-91.11.1760.000 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		63

## РОЗДІЛ 4. ТЕСТУВАННЯ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 3.1 Опис контрольного прикладу

Запустимо модель з використанням картинки зі склом. Після запуску отримуємо картинку з зазначеною категорією.



Рис.3.1 Приклад класифікації

Запустимо модель з використанням картинки з пластиком. Після запуску отримуємо картинку з зазначеною категорією.



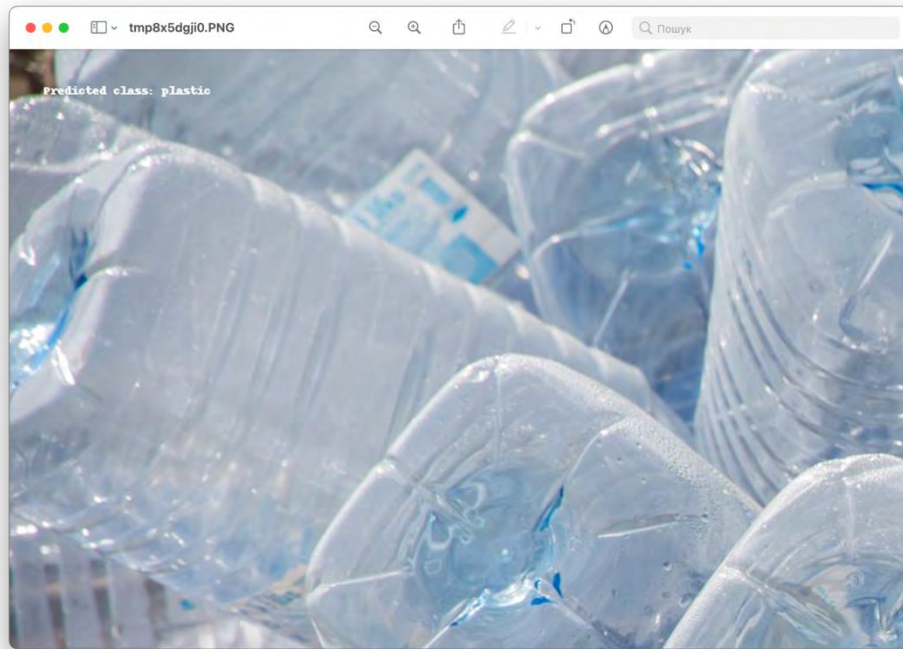


Рис.3.2 Приклад класифікації

Запустимо модель з використанням картинки з металом. Після запуску отримуємо картинку з зазначеною категорією.

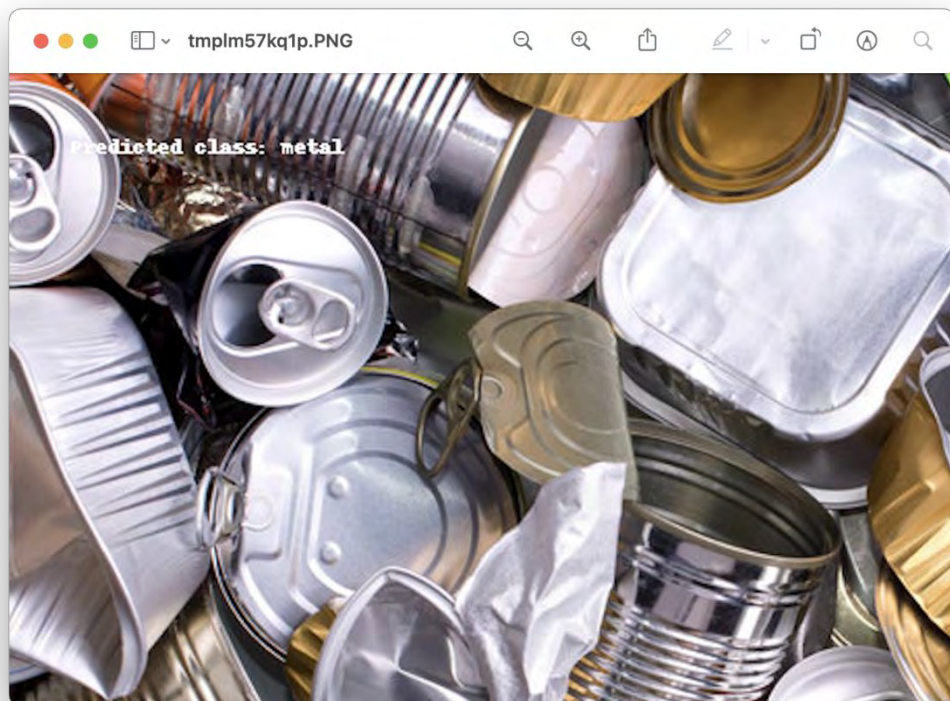


Рис.3.3 Приклад класифікації

### Висновки до розділу 3

Контрольний приклад використовується для оцінки ефективності та точності розробленого програмного забезпечення для автоматизованого сортування відходів. У контрольному прикладі використовується зображення з предметом, який потрібно класифікувати (наприклад, пластиковий виріб). Застосовується модель, побудована на основі нейронної мережі VisionTransformer (ViT), яка навчена на відповідних даних для класифікації відходів. Зображення піддається попередньому опрацюванню, а потім передається через модель для отримання передбачень класифікації. Висновок здійснюється на основі найвищого значення передбаченого класу та відповідного позначення класу. Результати класифікації відображаються на зображенні, що дозволяє візуально перевірити правильність класифікації.

					<i>ДПБ ПМ-91.11.1760.000 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		66

## ВИСНОВКИ

Проблема сортування відходів є актуальною у сучасному світі, де екологічна свідомість та необхідність в управлінні відходами стають все більш важливими. Використання нейронних мереж у сортуванні відходів є потенційно ефективним інструментом для покращення процесу сортування.

Метою дослідження було розробити програмне забезпечення, яке здатне автоматично класифікувати відходи на основі використання нейронної мережі. Головною метою було покращити точність та швидкість процесу сортування, зменшити навантаження на людей та забезпечити ефективне використання ресурсів.

Був проведений огляд існуючих методів та технологій сортування відходів, зокрема згорткових, рекурентних, сумісних, глибоких та віджатих нейронних мереж. Також була детально розглянута модель VisionTransformer, яка показала високу точність у завданнях класифікації зображень.

Було проведено аналіз вимог до програми та проектування її архітектури. Вибір мови програмування Python був обґрунтований його простотою та наявністю потрібних бібліотек. Для реалізації були використані необхідні інструменти та бібліотеки для роботи з зображеннями та нейронними мережами.

Були описані методи, використані для розробки програмного забезпечення. Були побудовані діаграми використання, послідовності та активності, які демонструють взаємодію компонентів програми та послідовність їх виконання. Був описаний метод тестування та наведено опис контрольного прикладу. Це дозволяє перевірити функціональність та коректність роботи програми.

Загальною висновком розробки програмного забезпечення для автоматизованого сортування відходів є те, що використання нейронних мереж у сортуванні відходів може принести значні покращення у впровадженні ефективних систем сортування. Однак, для досягнення

					<i>ДПБ ПМ-91.11.1760.000 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		67

оптимальних результатів, потрібно продовжувати дослідження та вдосконалювати алгоритми та моделі, щоб забезпечити точність, швидкість та стабільність роботи системи.

					<i>ДПБ ПМ-91.11.1760.000 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		68

## ДЖЕРЕЛА

1. Waste-to-Energy (SecondEdition), 2011 [Електронний ресурс] – Режим доступу до ресурсу: <https://www.sciencedirect.com/topics/earth-and-planetary-sciences/waste-sorting>.
2. WasteSorting: A CompleteGuidetoHowandWhyYourBusinessMustDoIt [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://www.anis-trend.com/waste-sorting-a-complete-guide-to-how-and-why-your-business-must-do-it/>.
3. A MethodforWasteSegregationusingConvolutionalNeuralNetworks [Електронний ресурс] – Режим доступу до ресурсу: <https://arxiv.org/pdf/2202.12258.pdf>.
4. AI andWasteRecognition – WhyIt Works SoWell [Електронний ресурс] – Режим доступу до ресурсу: <https://recycleye.com/ai-and-waste-recognition-why-it-works-so-well/>.
5. UsingNeuralNetworkstoImproveWasteManagement, OnePropagationat a Time. [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://towardsdatascience.com/using-neural-networks-to-improve-waste-management-one-propagation-at-a-time-b2b7414d329d>.
6. ConvolutionalNeuralNetworks, Explained [Електронний ресурс] – Режим доступу до ресурсу: <https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939>.
7. Whatarerecurrentneuralnetworks? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.ibm.com/topics/recurrent-neural-networks>.
8. HybridArtificialNeuralNetworks: Models, AlgorithmsandData [Електронний ресурс] – Режим доступу до ресурсу: [https://link.springer.com/chapter/10.1007/978-3-642-21498-1\\_23](https://link.springer.com/chapter/10.1007/978-3-642-21498-1_23).
9. DeepNeuralNetworks [Електронний ресурс] – Режим доступу до ресурсу: [https://www.tutorialspoint.com/python\\_deep\\_learning/python\\_deep\\_learning\\_deep\\_neural\\_networks.htm](https://www.tutorialspoint.com/python_deep_learning/python_deep_learning_deep_neural_networks.htm).
10. Squeeze-and-ExcitationNetworks [Електронний ресурс] – Режим доступу до ресурсу: <https://towardsdatascience.com/squeeze-and-excitation-networks-9ef5e71eacd7>.
11. VisionTransformer [Електронний ресурс] – Режим доступу до ресурсу: <https://paperswithcode.com/method/vision-transformer>.
12. Industry 5.0 [Електронний ресурс] – Режим доступу до ресурсу: [https://research-and-innovation.ec.europa.eu/research-area/industrial-research-and-innovation/industry-50\\_en](https://research-and-innovation.ec.europa.eu/research-area/industrial-research-and-innovation/industry-50_en).
13. WhatareFunctionalandNon-FunctionalRequirementsandHowtoDocumentThese [Електронний ресурс] –

					<i>ДПБ ПМ-91.11.1760.000 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		69

- Режим доступу до ресурсу: <https://enkonix.com/blog/functional-requirements-vs-non-functional/>.
14. VisionTransformer (base-sized model) [Електронний ресурс] – Режим доступу до ресурсу: [https://huggingface.co/Aalaa/Fine\\_tuned\\_Vit\\_trash\\_classification](https://huggingface.co/Aalaa/Fine_tuned_Vit_trash_classification).
  15. Zelle, J. M. (2010). Python Programming: An Introduction to Computer Science. Franklin, Beedle & Associates.
  16. Shaw, Z. A. (2013). Learn Python the Hard Way. Addison-Wesley Professional.
  17. Matthes, E. (2019). Python Crash Course. No Starch Press.
  18. Types of UML Diagrams [Електронний ресурс] – Режим доступу до ресурсу: <https://www.lucidchart.com/blog/types-of-UML-diagrams>.
  19. Огляд автоматизації процесу сортування сміття оптико-механічним методом / Д. В. Клименко, С. О. Нечай // Погляд у майбутнє приладобудування: збірник праць XIV Всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених, 18-19 травня 2021 р. – К.: ПБФ, КПІ ім. Ігоря Сікорського, 2021. – С. 196-198.
  20. Автоматизована система сортування відходів на базі візуальної спектроскопії / О. І. Бельман, Н. В. Стельмах // Погляд у майбутнє приладобудування: збірник праць XIII Всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених, 13-14 травня 2020 р. – К.: ПБФ, КПІ ім. Ігоря Сікорського, 2020. – с. 94-96.
  21. Дипломний проект бакалавра: організація, вимоги до структури, змісту та оформлення [Текст]: навч. посіб. для студ. спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології», освітньо-професійна програма «Комп'ютерно-інтегровані технології проектування приладів» / КПІ ім. Ігоря Сікорського: уклад.: О. К. Нікітін, В. М. Зайцев. – Київ: КПІ ім. Ігоря Сікорського, 2020. – 173 с.

## ДОДАТКИ

### Лістинг коду

Для доступу звертатись до авторів проекту

					<i>ДПБ ПМ-91.11.1760.000 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		71