https://orcid.org/0009-0007-1089-9223
https://orcid.org/0000-0002-9639-3544

УДК 004.457

Л. ЛІ[*,], О.Є. КОВАЛЕНКО[*]**

МОДЕЛІ ТА ЗАСОБИ ІНТЕЛЕКТУАЛЬНОЇ ОБРОБКИ ЗОБРАЖЕНЬ У СИСТЕМАХ СОРТУВАННЯ ПОБУТОВИХ ВІДХОДІВ

[*]Національний університет біоресурсів і природокористування України, м. Київ, Україна
[**]Університет Бенбу, провінція Аньхой, КНР

Анотація. *Одним із актуальних завдань екологічного менеджменту є ефективна переробка відходів на основі їх сортування. В інтелектуальній системі сортування відходів алгоритм обробки зображень є ключовим компонентом, що визначає її ефективність та точність. У відкритих наборах даних об'єкти на зображеннях зазвичай великі, відрізняються від тих, що були зроблені в реальних умовах сортувальної лінії. Це призводить до відсутності зразків із дрібними об'єктами. В результаті модель може мати труднощі з виявленням, що проявляється в пропущених розпізнаваннях або неточному позиціонуванні. У статті об'єднані матеріали з відкритих джерел та фотографії, отримані в лабораторних умовах, для створення спеціалізованого набору даних для візуального сортування відходів. Він містить зображення з одним об'єктом, кількома об'єктами, розмитими об'єктами, лабораторним фоном та складним фоном. Усі зображення поділені на чотири основні категорії: харчові відходи, відходи, що підлягають переробці, небезпечні відходи та інші відходи. Ефективна легка модель YOLO була обрана як базова завдяки меншій кількості параметрів та високій швидкості обробки для досягнення цілей низького енергоспоживання та низької вартості в інтелектуальній системі сортування. У статті аналізуються популярні алгоритми обробки та розпізнавання зображень із використанням згорткових нейронних мереж та на їх основі пропонується вдосконалений композитний алгоритм для виявлення та класифікації сміття. Також проведено обчислювальний експеримент для підтвердження ефективності запропонованого алгоритму на реальних навчальних вибірках. Показано, що загальна продуктивність запропонованого алгоритму перевищує сучасні популярні алгоритми розпізнавання об'єктів, забезпечуючи ефективне та точне розпізнавання сміття.*
Ключові слова: *обробка зображень, розпізнавання об'єктів, згорткова нейронна мережа, сортування сміття.*

Abstract. *One of the urgent tasks of environmental management is the efficient recycling of waste based on its sorting. In an intelligent waste sorting system, the image processing algorithm is a key component that determines its efficiency and accuracy. In open datasets, objects in images are usually large, different from those taken in real sorting line conditions. This leads to the lack of samples with small objects. As a result, the model may have difficulty with detection, manifested in missed recognitions or inaccurate positioning. The article combines materials from open sources and photographs obtained in laboratory conditions to create a specialized dataset for visual waste sorting. It included images with a single object, multiple objects, blurred objects, a laboratory background, and a complex background. All images are divided into four main categories: food waste, recyclable waste, hazardous waste, and other waste. The efficient, lightweight YOLO model has been chosen as the base model due to its fewer parameters and high processing speed to achieve low energy consumption and cost efficiency in the intelligent sorting system. The paper analyzes popular image processing and recognition algorithms using convolutional neural networks and, based on them, proposes an improved composite algorithm for garbage detection and classification. A computational experiment has also been conducted to confirm the effectiveness of the proposed algorithm on real training samples. It is shown that the overall performance of the proposed algorithm exceeds current popular object recognition algorithms, ensuring efficient and accurate garbage recognition.*
Keywords: *image processing, object recognition, convolutional neural network, garbage sorting.*

DOI: 10.34121/1028-9763-2025-3-4-33-53

1. Вступ

Управління навколишнім середовищем для забезпечення сталого розвитку є актуальною проблемою сьогодення і майбутнього, оскільки вимоги щодо екологічних показників життєвого середовища зростають, а нормативні обмеження посилюються. Впровадження інтелектуалізованих систем управління життєвим середовищем дозволяє підвищити їх ефективність і забезпечити сталий розвиток екосистеми у цілому [1].

Однією з актуальних задач керування життєвим середовищем є ефективна переробка сміття на основі його сортування. В інтелектуальній системі сортування сміття алгоритм обробки зображень є ключовим компонентом, що визначає її ефективність та точність.

Метою статті є розробка моделей і засобів інтелектуальної обробки зображень у системах сортування побутових відходів на основі згорткових мереж.

2. Модель мережі для виявлення об'єктів у смітті

2.1. Алгоритм виявлення об'єктів YOLOv5

Алгоритми виявлення об'єктів на основі глибинного навчання поділяються переважно на алгоритми одного етапу (One-Stage Detectors) та двоетапні алгоритми (Two-Stage Detectors). Двоетапні алгоритми містять два основні кроки: спочатку генерується набір регіонів-претендентів (Region Proposals), а потім виконуються класифікація й уточнення розташування об'єктів у цих регіонах. Типовими представниками є серія R-CNN, зокрема оригінальна R-CNN, а також удосконалені версії Fast R-CNN та Faster R-CNN. Перевагою цих алгоритмів є висока точність виявлення, але за рахунок зниження швидкості обробки.

Алгоритми одного етапу, навпаки, безпосередньо прогнозують межі об'єктів і їхні категорії з вхідного зображення, без попереднього генерування регіонів-претендентів, що забезпечує вищу швидкість обробки. Типовими представниками є серія YOLO (You Only Look Once), включаючи YOLOv1 [2], YOLOv2 [3], YOLOv3 [4], YOLOv4 [5] та ін., а також SSD (Single Shot MultiBox Detector). Ці алгоритми підтримують баланс між точністю та швидкістю, тому мають велике значення в реальному часі, зокрема в вбудованих системах, автономному водінні тощо. Хоча на ранніх етапах точність алгоритмів одного етапу була нижчою за двоетапні, з розвитком технологій сучасні одноетапні моделі досягли або навіть перевищили точність деяких двоетапних, зберігаючи при цьому перевагу у швидкості.

Щоб досягти цілей низького енергоспоживання та низької вартості в інтелектуальній системі сортування, у даному дослідженні обрано ефективну легку модель YOLO як базову завдяки її меншій кількості параметрів та високій швидкості обробки.

Основна ідея YOLO полягає в перетворенні задачі виявлення об'єктів у задачу регресії. Алгоритм розбиває вхідне зображення на сітку та за допомогою згорткової нейронної мережі одночасно прогнозує кілька межових рамок (Bounding Boxes) у кожній комірці сітки, а також відповідну категорію об'єкта та рівень довіри [2]. На відміну від традиційних методів з ковзним вікном або генерації регіонів, YOLO виконує прогнозування в один прохід, що забезпечує надзвичайно високу швидкість. Основні етапи алгоритму YOLO:

1. Розбиття зображення на сітку. Вхідне зображення ділиться на фіксовану кількість сіткових комірок, кожна з яких відповідає за виявлення об'єктів у своєму регіоні.

2. Єдине прогнозування. YOLO виконує один прямий прохід зображення, здійснюючи прогнозування межових рамок та категорій у кожній комірці.

3. Прогнозування межових рамок. Кожна комірка передбачає певну кількість межових рамок із заданими розмірами. Для кожної рамки прогнозуються положення (координати верхнього лівого кута, ширина та висота) і рівень довіри для певного класу.

4. Класифікація об’єктів. Для кожної рамки визначається ймовірність належності об’єкта до певної категорії, представлена коефіцієнтом довіри.

5. Придушення непотрібних рамок (NMS). Щоб усунути надлишкові рамки з перекриттям, YOLO використовує метод неперевершеного максимуму (Non-Maximum Suppression, NMS). Рамки сортуються за рівнем довіри, після чого вибираються найбільш імовірні, а ті, що мають високий коефіцієнт перекриття (Intersection of Union, IoU) з ними, відкидаються.

На сьогодні алгоритм YOLO має кілька версій. Кожна наступна версія удосконалює попередню. YOLOv5 є однією з найпоширеніших серед дослідників. Порівняно з YOLOv4, у YOLOv5 покращено базову мережу, функцію втрат і механізм NMS. Модель має менший розмір, менше параметрів, що знижує споживання пам’яті й обчислювальну складність, а також підвищує швидкість логічного висновку. Структуру мережі зображено на рис. 1.

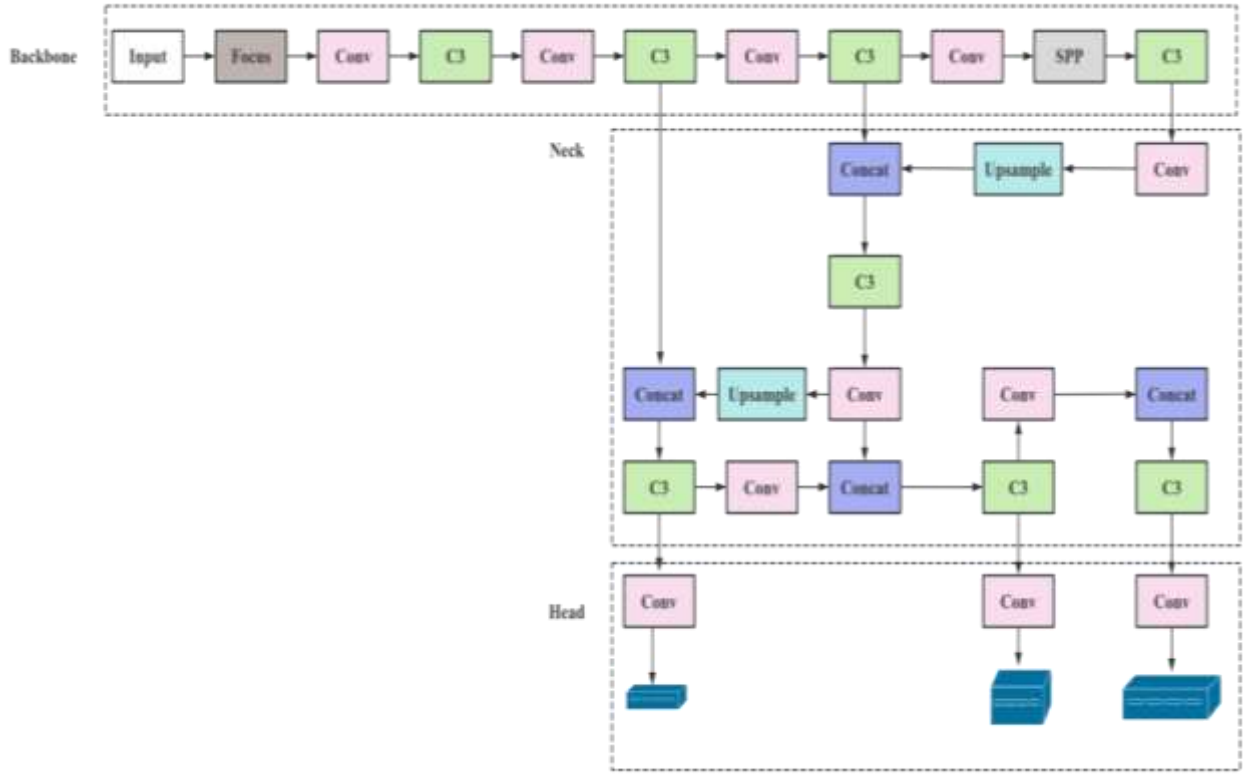


Рисунок 1 — Схематична структура мережі YOLOv5

Ця модель складається з чотирьох основних частин: вхідного блока (Input), основної мережі (Backbone), проміжної частини (Neck) та вихідного блока (Head). На вхідному етапі модель застосовує дві основні техніки — Mosaic-аугментацію та адаптивне обчислення анкерних рамок, які оптимізують процес навчання, підвищуючи узагальнювальну здатність моделі та її точність у виявленні об’єктів. Основна мережа (Backbone) відповідає за витягування ознак: вона виконує багаторівневі згорткові та операції зниження роздільної здатності (downsampling) над вхідним зображенням для формування багатомасштабних карт ознак. Проміжна частина (Neck), яка з’єднує Backbone та Head, виконує операції злиття та підвищення дискретизації (upsampling) ознак із різних рівнів, що підвищує здатність моделі виявляти об’єкти різного розміру. Вихідний блок (Head) містить функцію втрат для рамок (Bounding Box Loss) і механізм придушення непотрібних рамок (NMS 4 — Non-Maximum Suppression), перетворюючи витягнуті ознаки на остаточні результати виявлення — координати об’єктів та їхні категорії. Розглянемо основні структурні моделі.

2.1.1. Мозаїчне розширення

Мозаїчне розширення (mosaic augmentation) [5] — це метод збільшення різноманітності навчальних даних, що полягає у поєднанні чотирьох випадково масштабованих, обрізаних та переставлених зображень в одне нове зображення. Цей підхід підвищує різноманітність навчального набору та покращує здатність моделі до узагальнення, що, зі свого боку, дозволяє краще виявляти об’єкти різного масштабу, положення та орієнтації. Основні кроки:

1. Випадковий вибір 4 зображень.

2. Для кожного зображення виконується випадкове обрізання та масштабування до подібного розміру.

3. Обрізані зображення з’єднуються у випадковому порядку на спільному полотні, утворюючи «мозаїчну» композицію.

4. У результаті зображення містить об’єкти з різних сцен та контекстів, що ускладнює й урізноманітнює навчальні приклади.

Завдяки Mosaic-аугментації модель може в одному проході бачити більше різновидів об'єктів і краще адаптується до складного фону та дрібних цілей. Це прискорює збіжність мережі та покращує підсумкову точність виявлення. Крім того, ця техніка імітує ефект великого датасету, що особливо корисно при обмеженій кількості розмічених даних.

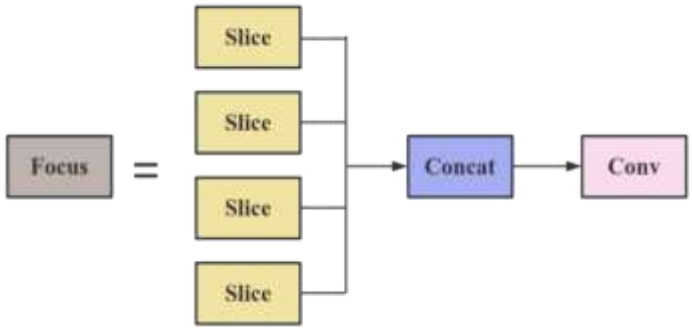


Рисунок 2 — Схематична структура модуля Focus

2.1.2. Модуль Focus

Модуль Focus у YOLOv5 призначений для ефективного зменшення роздільної здатності (downsampling) без втрати важливої інформації на етапі введення зображення. Його головна функція — перетворити просторову інформацію (ширину W і висоту H) у канальну, що полегшує обробку наступними шарами. Структуру модуля Focus наведено на рис. 2.

Принцип роботи:

1. Операція розрізання (slicing). Вхідне зображення обробляється за допомогою стратегії, подібної до downsampling із кроком 2: з кожного 2×2 блока вибирається один піксель, у такий спосіб утворюються чотири взаємодоповнюючі частини оригінального зображення.

2. Операція об’єднання (concatenation). Чотири частини зображення об’єднуються по каналах, тобто трьохканальне зображення (RGB) перетворюється, наприклад, на 12-канальну карту ознак.

3. Згорткове оброблення. Отримана багатоканальна карта ознак подається на згорткові шари, де виконується подальше витягування ознак із супровідним зниженням роздільної здатності.На відміну від традиційного зниження роздільності за допомогою згортки зі кроком 2, Focus-модуль краще зберігає деталі зображення, зменшуючи потенційні помилки в локалізації об’єктів.

У результаті модуль Focus створює карти ознак із нижчою просторовою роздільністю, але більш насичені по інформації, що особливо корисно для точного виявлення об’єктів різного розміру.

2.1.3. Модуль C3

У YOLOv5 модуль C3 (Cross-Compound Convolutional Block) є ключовим компонентом як у структурі Backbone, так і в Neck. Це вдосконалена та оптимізована версія модуля BottleneckCSP, який використовувався в попередніх версіях YOLO. Модуль C3 заснований на концепції часткових міжстадійних з'єднань (CSP, Cross-Stage Partial), однак має змінену архітектуру для покращення продуктивності та ефективності моделі. У традиційних згорткових нейронних мережах кожен шар обробляє всі ознаки з попереднього шару. У C3-модулі вхідна карта ознак розділяється на дві гілки: одна частина ознак передається безпосередньо в наступний згортковий шар, інша частина проходить через серію згорткових операцій, після чого з'єднується (concatenate) з основною гілкою. Схематичну структуру цього модуля наведено на рис. 3. Такий підхід дозволяє мережі краще об'єднувати ознаки з різних рівнів, не збільшуючи істотно обчислювальні витрати. До того ж, він знижує надлишкові обчислення, підвищуючи швидкість роботи моделі та ефективність використання ресурсів.

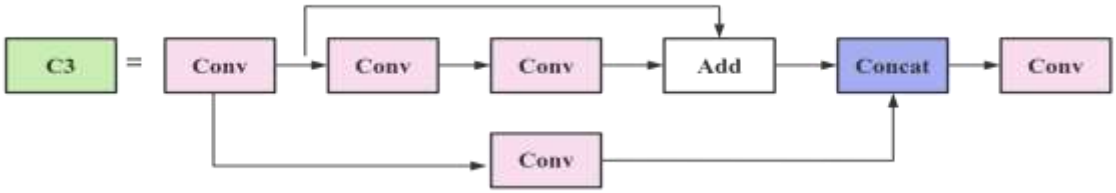


Рисунок 3 — Схематична структура модуля C3

2.1.4. Модуль SPP

У YOLOv5 модуль SPP (Spatial Pyramid Pooling — просторове пірамідальне пулінгування) є структурою для витягування ознак, призначеною для підвищення здатності моделі виявляти об'єкти різного масштабу. SPP-модуль виконує багатомасштабне пулінгування на різних рівнях згорткової нейронної мережі, перетворюючи просторову карту ознак довільного розміру на вихід фіксованої розмірності. Рис. 4 демонструє структуру модуля SPP. Спочатку модуль приймає карти ознак як вхідну інформацію, сформовану попередніми шарами мережі після серії згорткових операцій. Ці карти містять багатий семантичний контекст. Далі застосовуються операції максимального або середнього пулінгу з використанням вікон різного розміру (наприклад, 1×1, 2×2, 4×4 тощо). Таке багатомасштабне пулінгування дозволяє виділити просторові ознаки як на локальному, так і на глобальному рівні. Оскільки розмір пулінг-вікон є фіксованим, результат кожної операції має однакову вихідну форму незалежно від розміру вхідної карти ознак. Отже, незалежно від розмірів вхідного зображення, SPP-модуль може сформувати фіксованої довжини вектор ознак, що представляє інформацію з різних масштабів. У підсумку результати пулінгу з кожного масштабу з'єднуються по каналах, формуючи багатовимірний вектор ознак. Цей вектор поєднує ознаки з різною просторовою роздільністю. У YOLOv5 застосування SPP-модуля істотно розширює поле сприйняття мережі, що дозволяє моделі вловлювати об'єкти на різних рівнях абстракції без збільшення обчислювальної складності. Це покращує точність і узагальнювальну здатність виявлення об'єктів.

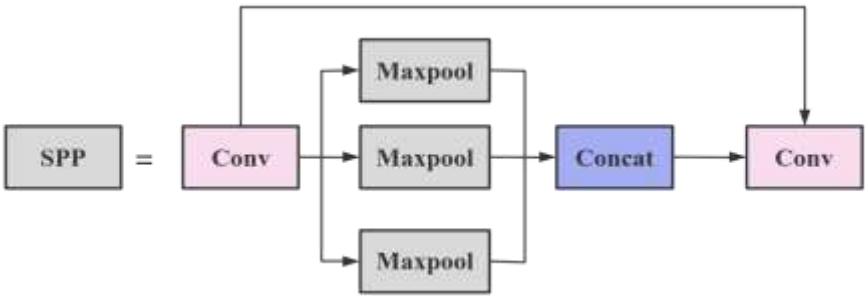


Рисунок 4 — Схематична структура модуля SPP

2.2. Удосконалена основна мережа (Backbone)

Основна мережа (Backbone Network) в алгоритмах розпізнавання об’єктів є базовим компонентом всієї моделі, оскільки вона відповідає за витягування багаторівневих та багатошарових ознак із вхідного зображення. У YOLOv5 як основну мережу використано CSPNet, що є вдосконаленою версією архітектури Darknet-53. Хоча CSPNet вносить певні покращення у напрямі спрощення моделі, вона все ще містить значну кількість надлишкових параметрів, що призводить до високого обчислювального навантаження. З метою зниження обчислювальної складності алгоритму виявлення об’єктів і підвищення ефективності його виконання у цьому дослідженні як основну мережу обрано легку згорткову нейронну мережу MobileNetV3 з подальшими вдосконаленнями для підвищення точності розпізнавання.

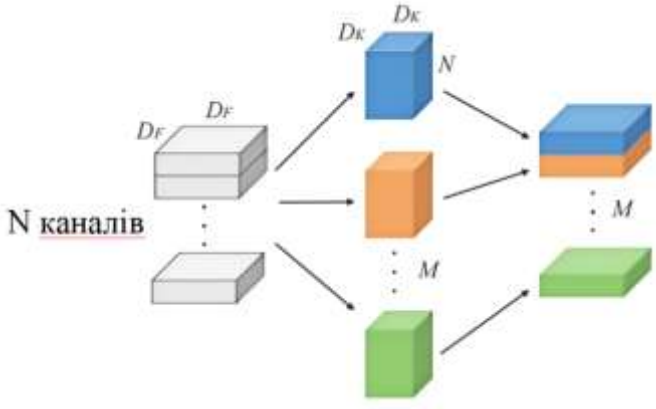


Рисунок 5 — Схема роботи стандартної згортки

MobileNetV3 [6] — це легка згорткова нейронна мережа, запропонована командою Google у 2019 році. Вона має дві версії: MobileNetV3-Large і MobileNetV3-Small. Обидві версії продовжують застосовувати концепцію глибоких роздільних згорток (depthwise separable convolutions) для зменшення обчислювальної складності згорткових шарів. Принципова різниця між стандартною згорткою та глибокою роздільною згорткою зображена на рис. 5 і 6 відповідно.

Глибока роздільна згортка (Depthwise Separable Convolution) складається з двох операцій: глибокої згортки (Depthwise Convolution) та точкової згортки (Pointwise Convolution). На відміну від стандартної згортки, глибинна згортка виконує незалежні операції згортки для кожного вхідного каналу окремо. Кількість згорткових ядер при цьому дорівнює кількості вхідних каналів. Точкова згортка — це операція згортки з ядром розміром 1×1, яка застосовується до отриманих від глибокої згортки карт ознак. Вона служить для лінійного комбінування і інтеграції ознак між різними каналами. Порівняно зі стандартною згорткою, глибока роздільна згортка суттєво зменшує кількість параметрів моделі. Порівняння обчислювальної складності цих двох методів наведено нижче:

$$\frac{P_1}{P_2} = \frac{D_K \times D_K \times D_F \times D_F \times M + M \times N \times D_F \times D_F}{D_K \times D_K \times D_F \times D_F \times M \times N} = \frac{1}{N} + \frac{1}{D_K^{\,2}}. \qquad (1)$$

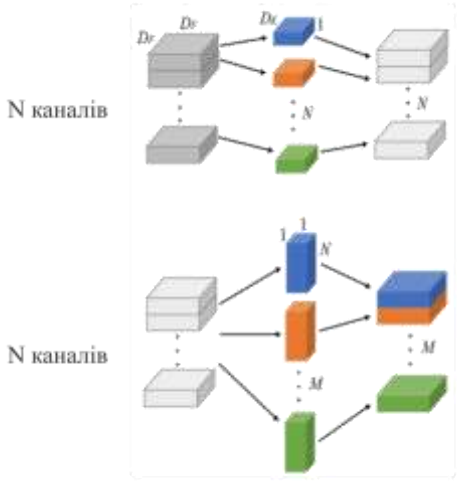


Рисунок 6 — Схема роботи глибокої роздільної згортки

У формулах P_1 та P_2 позначаються обчислювальні витрати глибокої роздільної згортки та стандартної згортки відповідно; D_K — розмір ядра згортки, D_F — розмір карти ознак, N — кількість каналів у карті ознак і ядрі згортки, M — кількість ядер згортки.

У моделі MobileNetV3 використовуються ядра згортки розміром 3×3 та 5×5. З наведених формул видно, що кількість параметрів глибокої роздільної згортки приблизно у 1/9 та 1/25 менша за відповідні параметри стандартної згортки, при цьому точність втрачається дуже незначно. Крім того, у MobileNetV3 на основі MobileNetV2 введено модуль SE (Squeeze and Excitation), який адаптивно регулює ваги для кожного каналу, посилюючи виразність моделі. Також запропоновано функцію активації H-Swish замість ReLU, що дозволяє зменшити обчислювальні витрати при збереженні точності. Загалом модель MobileNetV3 значно зменшує кількість параметрів і час навчання при збереженні високої точності класифікації зображень.

У MobileNetV3 використовується механізм стискання та збудження уваги SE-Net [7] (Squeeze-and-Excitation Network), який адаптивно регулює ваги між каналами, допомагаючи мережі фокусуватися на важливих ознаках. Ключова ідея SE-Net — це навчання набору ваг для адаптивного перенормування ознак кожного каналу. Спочатку застосовується глобальне усереднення (Global Average Pooling, GAP) для стиснення карти ознак кожного каналу у глобальний дескриптор, що дозволяє зменшити обчислювальні витрати та оцінити значущість каналів. Далі через два повнозв’язні (або згорткові) шари цей дескриптор відображається у менший розмір, застосовуються нелінійність ReLU та функція sigmoid, що обмежує ваги у діапазоні від 0 до 1, отримуються ваги уваги для кожного каналу. Нарешті, ці ваги застосовуються до початкової карти ознак множенням для масштабування важливих ознак. Механізм SE-Net дозволяє моделі автоматично навчатися важливості каналів і зосереджуватися на ключових ознаках, покращуючи виразність. Однак SE-Net враховує лише увагу між каналами, не звертаючи безпосередньо увагу на просторову інформацію карти ознак, що може обмежувати покращення продуктивності.

Механізм уваги CBAM [8] (Convolutional Block Attention Module) посилює представлення моделі, адаптивно виділяючи важливі канали та просторові позиції. CBAM складається з двох кроків: спочатку застосовується канальна увага для виділення важливості каналів і корекції ваг між ними, а потім — просторова увага для виділення ключових просторових областей карти ознак. Принципова схема CBAM наведена на рис. 7.

На відміну від SE-Net, CBAM враховує і міжканальні взаємозв’язки, і просторові взаємодії, що дозволяє краще захоплювати кореляції і контекст у карті ознак. Як легкий модуль CBAM істотно покращує продуктивність моделі при незначному збільшенні кількості параметрів і легко інтегрується у існуючі мережеві архітектури. Тому в цій роботі модуль CBAM обрано як заміну модулю SE-Net у мережі MobileNetV3.

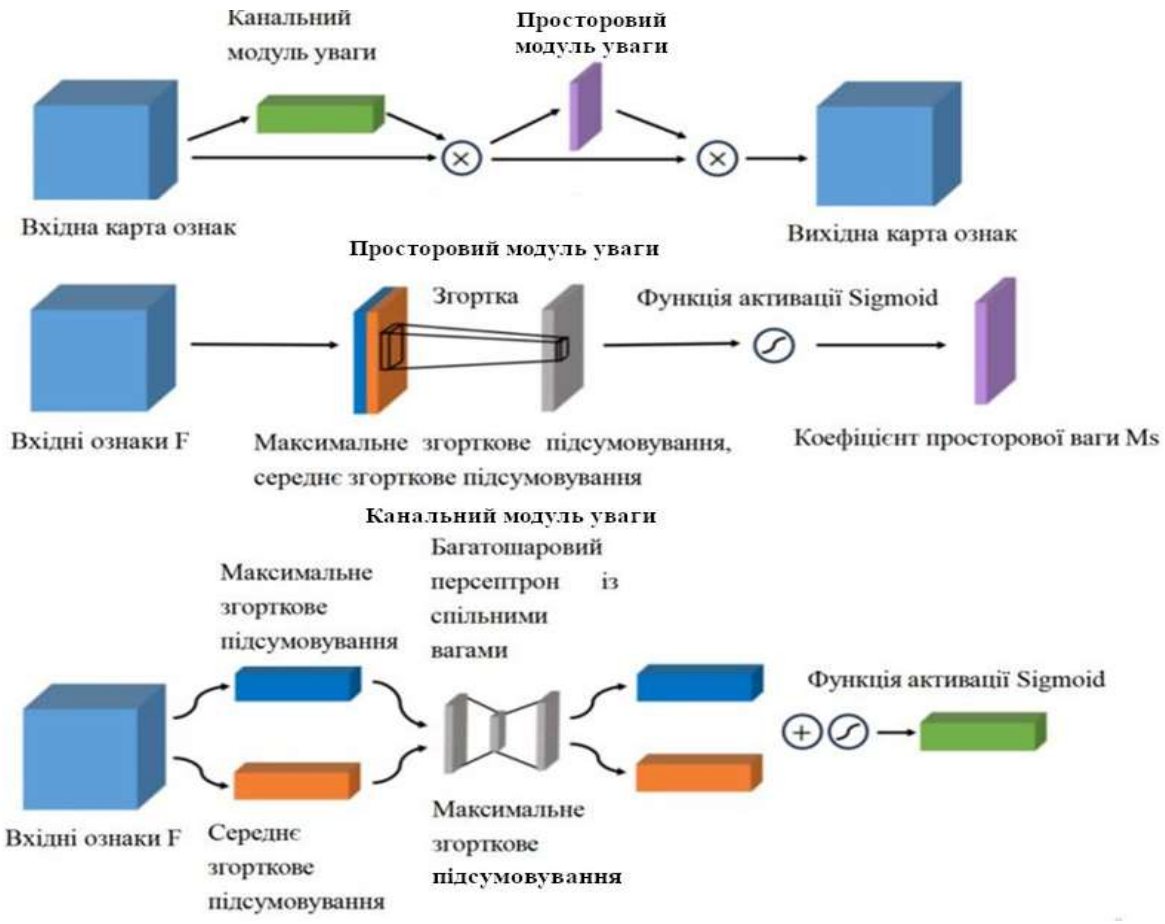


Рисунок 7 — Концептуальна схема механізму уваги CBAM

Функція активації — це нелінійна функція в нейронних мережах, яка застосовується до виходу нейрона. Сума лінійних функцій залишається лінійною, тому якщо в нейронній мережі використовувати лише лінійні функції активації, мережа зможе відображати тільки лінійні залежності. Введення ж нелінійної функції активації дозволяє мережі за допомогою поєднання та послідовних нелінійних перетворень між шарами створювати складні моделі та підвищувати виразність мережі.

Сигмоїдна функція — це нелінійна функція активації, яка відображає вхідні значення в діапазоні від 0 до 1 (рис. 8). Сигмоїдна функція також відома як логістична функція і має такий вид:

$$Sig(x) = (1+e^{-x})^{-1}. \qquad (2)$$

Сигмоїдна функція має плавну S-подібну криву: її вихід швидко змінюється поблизу середини, а на обох кінцях функція насичується, вихідні значення наближаються до 0 або 1. Така плавність допомагає зменшити коливання моделі та проблему перенавчання. Однак, коли вхідні значення значно віддаляються від нуля, градієнт Sigmoid прямує до нуля, що призводить до поступового зменшення градієнтів у глибоких мережах — явище, відоме як зникнення градієнта, що ускладнює навчання. Крім того, діапазон виходу Sigmoid — від 0 до 1 — означає, що при дуже великих або дуже малих вхідних значеннях вихід функції прагне до крайніх значень 0 або 1, що може викликати зміщення у крайні значення. Тому функція активації Sigmoid часто застосовується у задачах бінарної класифікації та в деяких неглибоких мережах, де важлива плавність і диференційованість, але у глибоких мережах її використання обмежене через проблему зникнення градієнта та відповідне зниження продуктивності.

Функція ReLU [9] відображає від'ємні значення в 0, а додатні залишає без змін (рис. 9). Функція ReLU характеризується нелінійністю та простотою обчислення, що забезпечує високу ефективність обчислень на практиці, і описується формулою

$$f(x) = max(0, x). \tag{3}$$

Градієнт ReLU на додатних значеннях є константою 1, що дозволяє уникнути проблеми згасання градієнта та забезпечує кращу передачу градієнта під час зворотного поширення помилки. Одночасно, оскільки вихід ReLU для від'ємних значень дорівнює 0, частина нейронів не активується, що робить мережу розрідженою. Така розрідженість знижує надмірність між ознаками, формує більш компактне й ефективне представлення, що сприяє покращенню узагальнюючої здатності мережі. Однак у функції ReLU є й недоліки: коли вхідне значення від'ємне, градієнт нейрона дорівнює 0, через що параметри цього нейрона не оновлюються, і нейрон перестає реагувати на будь-які вхідні дані. Виникає так зване «вимирання нейронів».

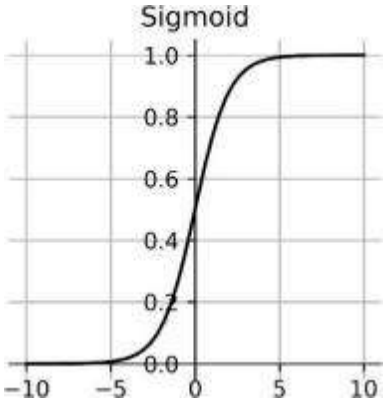


Рисунок 8 — Сигмоїдна функція активації

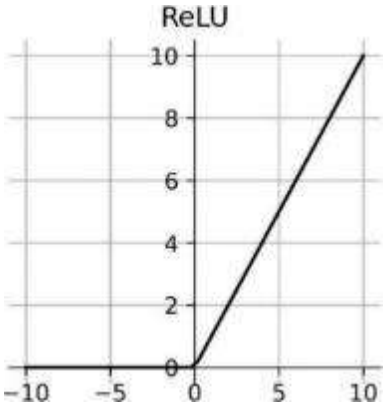


Рисунок 9 — Функція активації ReLU

Функція Mish [10] — це саморегулююча немонотонна функція активації виду

$$Mish(x) = x \times \tanh(\ln(1 + e^{x})). \tag{4}$$

Функція Mish є диференційованою на всій множині дійсних чисел і має неперервну першу похідну (рис.10), що дозволяє уникнути проблеми «смерті» нейронів, характерної для ReLU. Це сприяє кращій обробці градієнтів. Крім того, функція активації Mish здатна швидше сходитися при виконанні складних завдань, а також при збільшенні глибини мережі точність ReLU різко знижується, тоді як Mish краще зберігає точність. Тому в цій роботі функція Mish використовується замість активаційної функції в глибокій мережі MobileNetV3 для підвищення узагальнюючої здатності моделі.

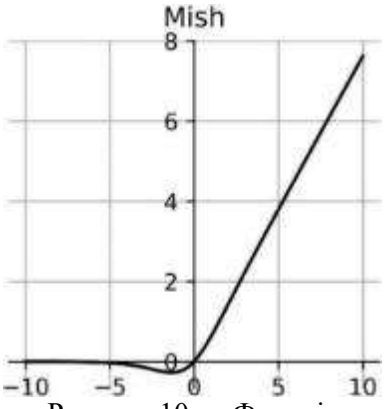


Рисунок 10 — Функція активації Mish

Удосконалена структура основної мережі, представлена в цій роботі, показана на рис. 11. Bneck_s1 позначає згортковий шар із кроком 1, Bneck_s2 — згортковий шар із кроком 2, h означає, що в шарі використовується функція активації H-Swish, m — використовується функція активації Mish. DW та PW відповідно позначають глибокі згортки (depthwise convolution) та поканальні згортки (pointwise convolution). У шарах із кроком 2 інтегровано механізм уваги CBAM, а також застосовано залишкові зв'язки

(residual connections).

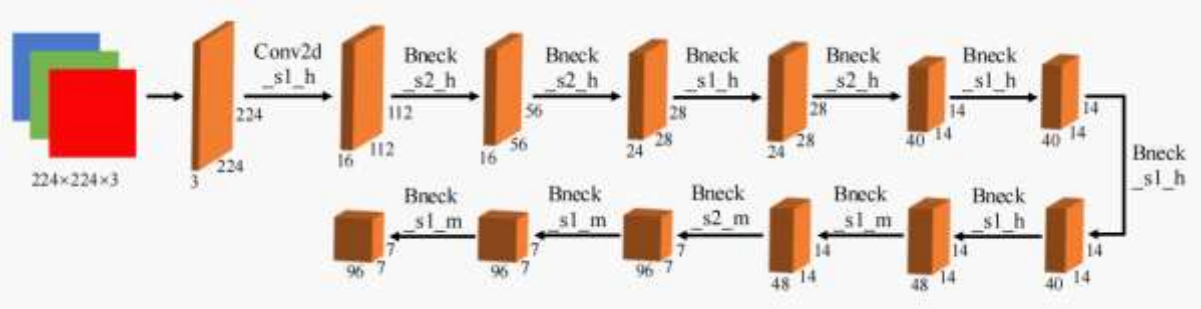


Рисунок 11 — Схема структури удосконаленої основної мережі, запропонованої в цій роботі

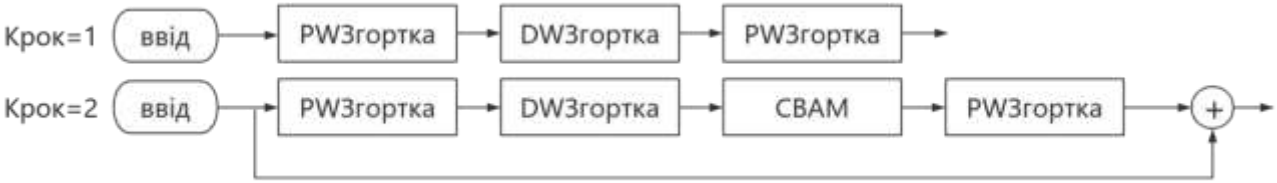


Рисунок 12 — Структура двох типів блоків з різними кроками (stride)

2.3. Поліпшення функції втрат

Функція втрат, яка використовується в алгоритмі YOLO, складається з трьох основних частин: втрати позиції, втрати довіри (confidence loss) та втрати класифікації. Втрата позиції використовується для оцінки зміщення між передбачуваною рамкою (bounding box) та реальною рамкою. Втрата довіри вимірює точність прогнозу довіри рамки. Втрата класифікації оцінює точність передбачуваного класу. Ці три компоненти втрат зважено сумуються, утворюючи кінцеву глобальну втрату. Мінімізуючи цю глобальну втрату, модель YOLO краще навчається визначати положення рамок, довіру та класи об’єктів, що покращує точність і послідовність прогнозів.

Функція втрат IoU [2] (Intersection over Union Loss) є поширеним методом у задачах виявлення об’єктів для оцінки позиційного зміщення між передбачуваною рамкою та реальною рамкою. Вона оцінює точність локалізації, обчислюючи коефіцієнт перетину над об’єднанням (IoU) між передбачуваною та реальною рамками. Функція втрат IoU може бути виражена такою формулою:

$$IoU\ (x, y) = \frac{Intersection(x, y)}{Union(x, y)}. \tag{5}$$

Тут x позначає передбачувану рамку, y — реальну рамку, *Intersection*(*x*,*y*) — площу перетину передбачуваної та реальної рамок, а *Union*(*x*,*y*) — площу їх об’єднання.

Функція втрат IoU зазвичай використовується для навчання моделі шляхом мінімізації різниці між передбачуваною рамкою і реальною, щоб передбачувана рамка максимально відповідала реальній. Коли IoU близький до 1, це означає високу ступінь накладання передбачуваної та реальної рамок, тобто позиційне передбачення є точним; коли IoU близький до 0, це свідчить про відсутність накладання і, відповідно, про неточне позиційне передбачення. Проте, якщо рамки взагалі не перекриваються, за визначенням $IoU = 0$, тоді $loss = 0$, і градієнти не можуть бути передані назад, що унеможливлює навчання.

Метод GIoU [11] додає до IoU мінімальну охоплюючу рамку C, яка містить і передбачувану, і реальну рамки. Формула для обчислення GIoU така:

$$GIoU = IoU\ （x, y）- \frac{C(x, y) - Union(x, y)}{C(x, y)}\ .\qquad (6)$$

Тут x позначає прогнозовану рамку, y — істинну рамку, $IoU(x,y)$ — результат обчислення IoU, а $C(x,y)$ — площа мінімального замкнутого контура, який охоплює як прогнозовану, так і істинну рамку (тобто мінімального прямокутника, що охоплює рамки).

Порівняно з функцією втрат *IoU, GIoU* є більш стабільним у обчисленні градієнтів і враховує ті неперекривні області, які *IoU* ігнорує, що дозволяє краще відображати ступінь перекриття прогнозованої та істинної рамок. Однак для *GIoU* потрібно обчислювати площу замкнутого контура, що збільшує обчислювальні витрати та складність моделі.

DIoU [11] також враховує замкнутий контур C, але обчислює евклідову відстань між прогнозованою та істинною рамками, формула обчислення наведена нижче:

$$DIoU = IoU\ （x, y）- \frac{d^2(x_c, y_c)}{diagonal^2}\ .\qquad (7)$$

Тут *x* позначає передбачену рамку, *y* — реальну рамку, $IoU(x,y)$ — результат обчислення IoU, $d(x_c, y_c)$ — евклідова відстань між центрами передбаченої та реальної рамок, а *diagonal* — довжина діагоналі мінімального замкнутого прямокутника, що охоплює передбачену і реальну рамки.

CIoU [12] на основі *DIoU* враховує співвідношення сторін рамки, його вираз має такий вигляд:

$$CIoU = IoU(x,y) - \frac{d^2(x_c, y_c)}{diagonal^2} - \alpha v.\qquad (8)$$

Тут *x* позначає передбачену рамку, *y* — реальну рамку, $IoU(x,y)$ — результат обчислення IoU, $d(x_c, y_c)$ — евклідова відстань між центрами передбаченої та реальної рамок, diagonal — довжину діагоналі мінімального замкнутого прямокутника, що охоплює передбачену і реальну рамки, α — налаштовуваний параметр, v — коефіцієнт, що оцінює співвідношення сторін рамки і допомагає збалансувати штрафи за положення та розмір.

Функція втрат CIoU вводить урахування співвідношення сторін передбаченої та реальної рамок, що робить регресію рамок більш стабільною, але вона містить більш складні математичні обчислення, що збільшує складність моделі та обсяг обчислень, ускладнюючи навчання і збіжність моделі.

MPDIoU [13] — це покращений метод, запропонований на основі традиційної функції втрат IoU, який враховує концепцію максимального часткового діаметра. Максимальний частковий діаметр — це відстань між двома найвіддаленішими точками двох межових рамок у певному напрямку. *MPDIoU* обчислює відношення перетину максимальних часткових діаметрів двох рамок до їх об’єднання як міру відповідності. Його вираз має вигляд

$$d_1^{\ 2} = (x_1^B - x_1^A)^2 - (y_1^B - y_1^A)^2,\qquad (9)$$

$$d_2^{\ 2} = (x_2^B - x_2^A)^2 - (y_2^B - y_2^A)^2,\qquad (10)$$

$$MPDIoU = IoU(x,y) - \frac{d_1^{\ 2}}{w^2 + h^2} - \frac{d_2^{\ 2}}{w^2 + h^2}\ .\qquad (11)$$

Тут (x_1^A, y_1^A), (x_2^A, y_2^A) позначають координати лівого верхнього та правого нижнього кутів передбаченої рамки, а (x_1^B, y_1^B) (x_2^B, y_2^B) — координати лівого верхнього та правого нижнього кутів реальної рамки, w та h — ширину і висоту вхідного зображення.

Використання *MPDIoU* як функції втрат дозволяє краще враховувати форму обмежувальної рамки та підвищує точність узгодження рамок. Тому у цій роботі *MPDIoU* використовується замість *CIoU*, що застосовується у YOLOv5.

2.4. Покращення алгоритму придушення немаксимумів

Придушення немаксимумів (Non-Maximum Suppression, NMS) [14] використовується у процесі детекції об'єктів для вибору найбільш репрезентативних рамок серед тих, що перекриваються. Основна ідея NMS полягає у тому, що з набору детекційних рамок спочатку обирається рамка з найвищою впевненістю (confidence) як базова, що додається до кінцевого результату. Потім для решти рамок обчислюється ступінь перекриття з базовою рамкою. Якщо перекриття перевищує заданий поріг, такі рамки видаляються зі списку кандидатів. Так фільтруються надмірні рамки з високим перекриттям і залишаються лише найбільш релевантні. Формула цього процесу має вигляд

$$S_i = \begin{cases} S_i, iou(M, b_i) < N_t \\ 0, iou(M, b_i) \geq N_t \end{cases}. \tag{12}$$

Тут S_i — це оцінка впевненості (confidence score) для відповідної обмежувальної рамки, M — рамка з найвищою впевненістю, тобто базова рамка, b_i — кожна з детекційних рамок, N_t — поріг перекриття між детекційною і базовою рамками.

У цій роботі традиційний алгоритм придушення немаксимумів (NMS) замінено на Soft-NMS [15] (Soft Non-Maximum Suppression). Традиційний NMS при видаленні надмірних передбачень використовує фіксований поріг: якщо ступінь перекриття двох рамок перевищує цей поріг, зберігається лише рамка з більшою впевненістю, а рамка з меншою — відкидається. Проте такий фіксований поріг може призводити до значного зниження оцінки рамок із близьким перекриттям, що ускладнює виявлення моделей, особливо для близько розташованих об'єктів.

На відміну від традиційного NMS, Soft-NMS вводить функцію затухання, яка зменшує оцінку перекритих об'єктів замість того, щоб одразу їх відкидати. Вирази для Soft-NMS наведені у формулах (13) і (14). Функція затухання враховує ступінь перекриття і пропорційно знижує впевненість об'єкта, завдяки чому під час немаксимального придушення зберігається більше релевантних передбачень.

$$S_i = \begin{cases} S_i, iou(M, b_i) < N_t \\ S_i(1 - iou(M, b_i)), iou(M, b_i) \geq N_t \end{cases}, \tag{13}$$

$$S_i = S_i e^{-\frac{iou(M, b_i)^2}{\sigma}} \forall b_i \notin D. \tag{14}$$

Тут S_i — це оцінка впевненості для відповідної обмежувальної рамки, M — рамка з найвищою впевненістю, тобто базова рамка, b_i — кожна з детекційних рамок, N_t — поріг перекриття між детекційною і базовою рамками, D — результуюча множина рамок.

3 Алгоритми відстеження об'єктів

Відстеження об'єктів — це завдання безперервного спостереження за цікавими об'єктами у відеопослідовності. У сфері машинного зору існує багато алгоритмів відстеження об'єктів, метою яких є точне відстеження шляхом аналізу положення об'єкта, його візуальних ознак та інформації про рух у відеопослідовності. Поширені алгоритми відстеження містять: відстеження на основі ознак (наприклад, на основі кольору, текстури, форми тощо), відстеження

на основі моделей руху (наприклад, фільтр Калмана, фільтр частинок), а також відстеження на основі глибокого навчання (наприклад, сіамські нейронні мережі, згорткові нейронні мережі). Метою системи сортування відходів є автоматична класифікація або виявлення відходів, що підлягають сортуванню, та виконання відповідних операцій. Використовуючи алгоритми відстеження, система може у режимі реального часу визначати траєкторію руху та координати відходів, надаючи інформацію про їхню поведінку та точне розташування. Крім того, завдяки використанню алгоритмів відстеження під час виявлення об'єктів система не буде повторно розпізнавати ті самі відходи, що дозволяє уникнути передачі великої кількості надлишкової інформації до модуля виконання сортування.

Алгоритм SORT [16] — це метод відстеження об'єктів, заснований на фільтрі Калмана та алгоритмі угорського призначення. Спочатку він використовує детектор об'єктів (наприклад, YOLO, Faster R-CNN тощо) для виявлення об'єктів на кожному кадрі відео, а потім застосовує фільтр Калмана для прогнозування стану відстежуваних об'єктів. Далі за допомогою алгоритму угорського призначення виконується співставлення об'єктів поточного кадру з уже відстежуваними об'єктами попереднього кадру, формуючи нові результати відстеження. Алгоритм SORT простий і ефективний, підходить для відстеження в режимі реального часу, але коли об'єкт частково перекритий або зникає з інших причин, прогнозована фільтром Калмана інформація про стан може не збігатися з результатами виявлення, що призводить до передчасного завершення відстеження.

Алгоритм DeepSORT [17] є вдосконаленою версією SORT, яка додає модуль глибокого навчання для вилучення ознак об'єктів та моделювання їхнього зовнішнього вигляду. DeepSORT використовує глибоку нейронну мережу для вилучення ознак виявлених об'єктів і застосовує їх для оцінювання схожості між об'єктами. Це дозволяє алгоритму зберігати ефективність відстеження навіть тоді, коли об'єкти змінюють свій зовнішній вигляд або потрапляють під часткове перекриття. Крім того, DeepSORT на основі SORT вводить матрицю асоціацій, яка використовується для обчислення оцінок відповідності між об'єктами, що ще більше підвищує точність співставлення. Отже, у порівнянні з SORT, DeepSORT забезпечує вищу точність та стійкість при відстеженні об'єктів.

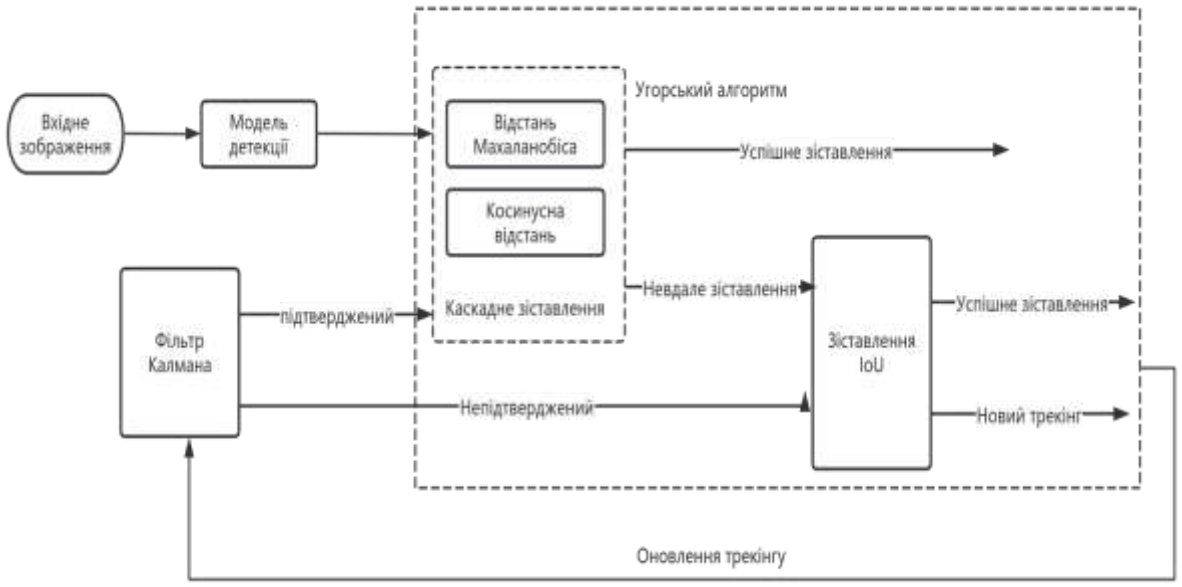


Рисунок 13 — Блок-схема відстеження об'єктів DeepSORT

Ключова ідея алгоритму DeepSort полягає у встановленні відповідності між кожним виявленим об'єктом та вже відстежуваними об'єктами, використовуючи модель глибокого

навчання для їх розпізнавання та вилучення ознак. Робочий процес алгоритму DeepSort, показаний на рис. 13, містить такі основні етапи:

1. Виявлення об’єктів. За допомогою моделі виявлення об’єктів (наприклад, YOLO, Faster R-CNN тощо) виконується пошук об’єктів на кожному кадрі відео. Модель виявлення повинна надавати інформацію про положення кожного об’єкта та координати його обмежувального прямокутника.

2. Вилучення ознак об’єктів. Для кожного виявленого об’єкта використовується глибока нейронна мережа, яка формує його вектор ознак. Ці ознаки застосовуються для вимірювання схожості між об’єктами.

3. Асоціація об’єктів. Об’єкти поточного кадру співставляються з відстежуваними об’єктами попереднього кадру. Для розв’язання цієї задачі зазвичай використовується угорський алгоритм (Hungarian algorithm), який на основі схожості ознак та відстані між об’єктами виконує їхнє зіставлення та формує новий список відстежуваних об’єктів.

4. Прогнозування стану. Для успішно зіставлених об’єктів застосовується фільтр Калмана, який прогнозує їхнє положення та швидкість. Цей процес дозволяє виправляти помилки відстеження та робити обґрунтовані припущення у разі перекриття чи зникнення об’єкта.

5. Оновлення відстеження. На основі нових результатів виявлення поточного кадру виконується оновлення положення та ознак відстежуваних об’єктів, що допомагає забезпечити точність і стійкість процесу відстеження.

4. Навчання моделі виявлення об’єктів відходів та показники оцінювання

4.1. Створення набору даних відходів

Побудова якісного набору даних є ключовим кроком для тренування високопродуктивної моделі. Повний, збалансований і цілеспрямований набір даних гарантує, що модель зможе демонструвати відмінні можливості розпізнавання в реальних умовах застосування.

Для задач класифікації відходів на сьогодні існує лише обмежена кількість відкритих наборів даних, наприклад, набір TrashNet, створений Yang та ін., який містить лише шість класів відходів і не відповідає реальній класифікації побутових відходів [18]. Крім того, у цих відкритих наборах даних об’єкти на зображеннях зазвичай мають великий розмір, що відрізняється від зображень, знятих у реальних умовах сортувальних ліній. Це призводить до браку зразків із малими об’єктами, і, як наслідок, модель може відчувати труднощі з їх виявленням, що проявляється у пропущених розпізнаваннях або неточному позиціонуванні.

З огляду на це, у даній роботі було поєднано матеріали з мережевих джерел та фотографії, отримані в лабораторних умовах, щоб створити спеціалізований набір даних для візуального сортування відходів. До нього увійшли зображення з одним об’єктом, кількома об’єктами, розмитими об’єктами, лабораторним фоном та складним фоном. Усі зображення поділено на чотири основні категорії: харчові відходи, придатні для переробки відходи, небезпечні відходи та інші відходи. У чотирьох основних категоріях загалом представлено 245 різних видів об’єктів. Кількість зразків для кожної категорії та перелік конкретних об’єктів наведено в табл. 1.

Таблиця 1 — Розподіл самостворeного набору даних про сміття

Категорія відходів	Приклади предметів	Кількість
Харчові відходи	Шкірки фруктів, залишки їжі, кондитерські вироби тощо	20 483
Придатні для переробки	Скляні чашки, нержавіюча сталь, друковані плати тощо	49 576

Продовж. табл. 1

Інші відходи	Клейка стрічка, маски, зубні щітки тощо	5 151
Небезпечні відходи	Пляшки з-під ліків, дезінфікуючі засоби, пігулки тощо	4 802

Початкове зображення Випадкове обертання Випадкове обрізання Зміна контрастності Гаусове розмиття

Рисунок 14 — Приклади доповнення даних про зображення сміття

Для розширення кількості зразків та підвищення різноманітності даних було проведено аугментацію експериментальних даних. Зразки зображень було піддано випадковому обертанню (Random Rotation), випадковому обрізанню з масштабуванням (Random Resized Crop), зміні контрастності (Contrast Transformation) та гаусівському розмиттю (Gaussian Blur), як показано на рис. 14. Це дозволяє отримати чотири різні варіанти зображень пляшок із мінеральною водою, що робить навчальні дані максимально близькими до реального розподілу, тим самим покращуючи узагальнюючу здатність та стійкість моделі.

4.2. Навчання мережевої моделі

Навчання мережевої моделі класифікації відходів виконувалося на основі раніше створеного набору зображень відходів. Загальний процес навчання наведено на рис. 15.

Навчання та тестування моделі проводилися на персональному комп’ютері з операційною системою Windows. Апаратні характеристики наведені у табл. 2.

Таблиця 2 — Аппаратне забезпечення експерименту

Пристрій	Модель
Процесор	Intel(R) Xeon(R) Silver 4314
Відеокарта	NVIDIA A30
Оперативна пам’ять	128 ГБ

Використане програмне забезпечення наведено у табл. 3.

Таблиця 3 — Програмне середовище експерименту

Програмне середовище	Версія
Мова програмування	Python 3.10
Фреймворк для глибокого навчання	PyTorch 1.13.0
Версія драйвера GPU	516.94
Версія CUDA	11.2
Операційна система	Windows Server 2019 Standard

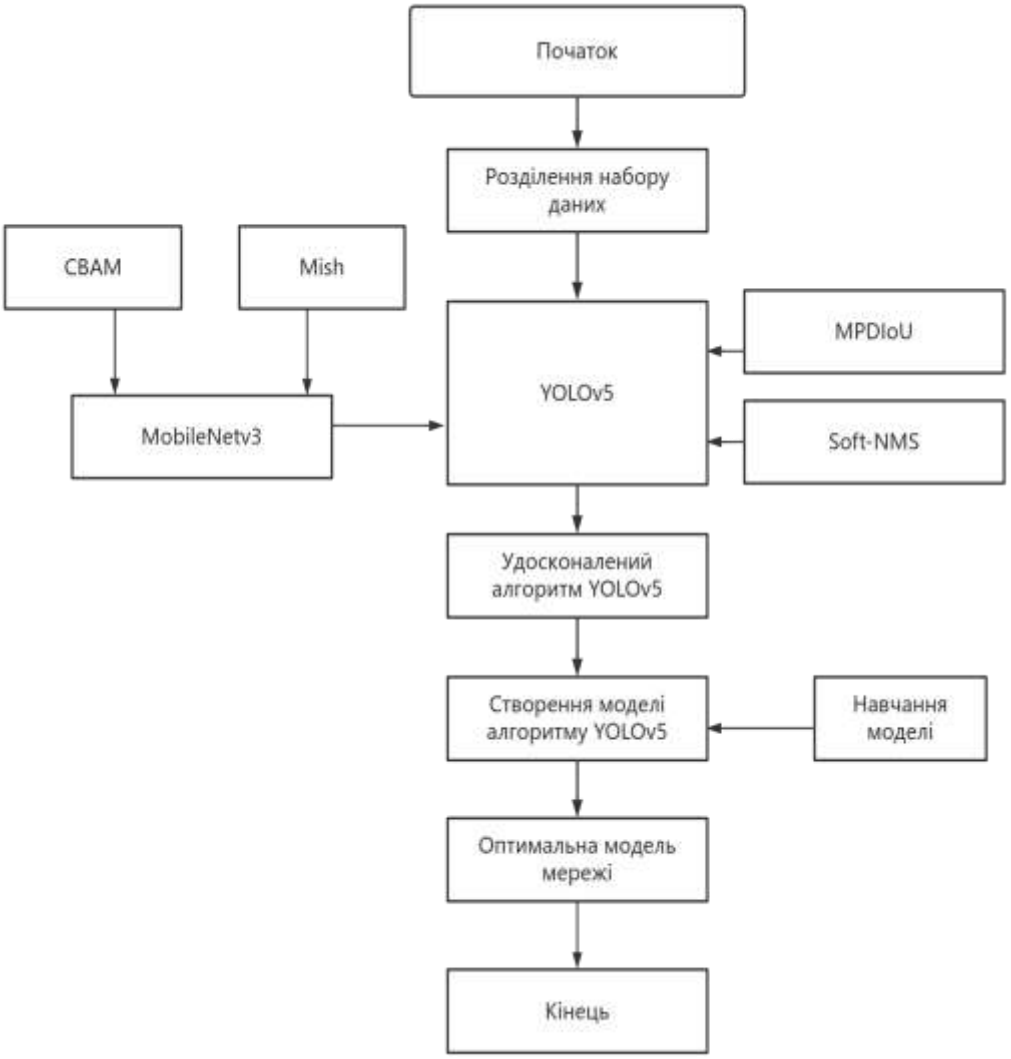


Рисунок 15 — Діаграма процесу тренування моделі класифікації сміття

У цьому експерименті кількість зразків зображень у навчальному, валідаційному та тестовому наборах становить відповідно 2906, 830 та 416. Оскільки в роботі використовується перенесене навчання (transfer learning), для тонкого налаштування моделі достатньо невеликої кількості зразків зображень сміття. Отже, створений набір даних відповідає вимогам експерименту. Основні параметри навчання моделі наведені у табл. 4.

Таблиця 4 — Параметри навчання

Параметр	Значення
Batch-size	32
Кількість епох	200
Оптимізатор	Adam
Розмір вхідного зображення	640×640
Стратегія корекції швидкості навчання	LinearLR

4.3. Метрики оцінювання

Під час розробки та оптимізації алгоритму виявлення об'єктів метрики оцінювання є важливою основою для налаштування параметрів моделі, удосконалення алгоритмічних стратегій і оцінки продуктивності моделі. Відповідно до цілей дослідження, у роботі для оцінки продуктивності моделі використовуються: середнє значення середньої точності (mean Average Precision, mAP), кількість параметрів моделі (Parameters), кількість оброблених зображень за секунду (Frames Per Second, FPS) та розмір пам'яті, яку займає модель (Size).

Середнє значення точності (mAP) комплексно враховує точність (Precision) та повноту (Recall) моделі, при цьому обчислюється для кожного класу значення AP (Average Precision), а потім усі AP усереднюються. Вона не лише оцінює, чи правильно модель розпізнає цільовий клас, а й враховує ступінь перекриття передбачуваних та фактичних межових рамок. Тому mAP дуже добре відображає здатність моделі одночасно локалізувати та класифікувати кілька об'єктів за різних порогів, що робить цю метрику надзвичайно корисною для оцінки продуктивності алгоритмів виявлення об'єктів. Її вираз має такий вигляд:

$$mAP = \frac{1}{n}\sum_{i}^{n} AP_i \,. \tag{15}$$

У формулі n — це загальна кількість класів, а AP — площа під кривою Precision-Recall для i-го класу, де Precision і Recall визначаються за формулами

$$precision = \frac{TP}{TP+FP}\,, \tag{16}$$

$$recall = \frac{TP}{TP+FN}\,, \tag{17}$$

де True Positive (TP) — кількість випадків, коли модель правильно класифікувала позитивний зразок як позитивний;

False Positive (FP) — кількість випадків, коли модель помилково класифікувала негативний зразок як позитивний;

False Negative (FN) — кількість випадків, коли модель помилково класифікувала позитивний зразок як негативний.

FPS у задачах детекції об'єктів використовується для вимірювання швидкості обробки моделі в режимі реального часу при відеопотоках або послідовностях зображень. За обмежених обчислювальних ресурсів FPS допомагає розробникам знайти баланс між продуктивністю моделі та використанням обчислювальних потужностей. Якщо FPS-моделі достатньо високий, це дозволяє знизити вимоги до апаратного забезпечення без втрати якості роботи моделі, що сприяє зменшенню витрат.

Формула для обчислення має вигляд

$$FPS = \frac{FrameNumber}{ElapsedTime}, \tag{18}$$

де *FrameNumber* — загальна кількість зображень для детекції, *ElapsedTime* — загальний час детекції.

Розмір пам'яті, який займає модель, визначає, чи може вона працювати на ресурсообмежених вбудованих пристроях. За умови збереження певного рівня точності детекції, чим менша модель і менше вона споживає пам'яті, тим легше її розгортати та впроваджувати на різних платформах і пристроях.

4.4. Експерименти та аналіз покращених алгоритмів детекції об’єктів

Каркасна мережа алгоритму детекції є його ключовою частиною, що відповідає за вилучення важливих ознак із вхідного зображення. Ці ознаки мають критичне значення для подальшої локалізації та класифікації об’єктів. Тому валідація каркасної мережі є надзвичайно важливою.

На власноруч зібраному наборі даних проводилося навчання оригінальної моделі MobileNetV3 та її покращеної версії, а також виконувалися абляційні експерименти для оцінки внеску кожного покращення у підвищення продуктивності моделі.

Як показано на рис. 16, крива MobileNetV3 ілюструє результати розпізнавання моделі MobileNetV3 з переносним навчанням, CBAM — зміни в розпізнаванні після додавання механізму уваги CBAM до базової MobileNetV3, а Mish — подальша заміна функції активації в моделі. Видно, що при однаковій кількості епох навчання покращена модель демонструє підвищену точність розпізнавання.

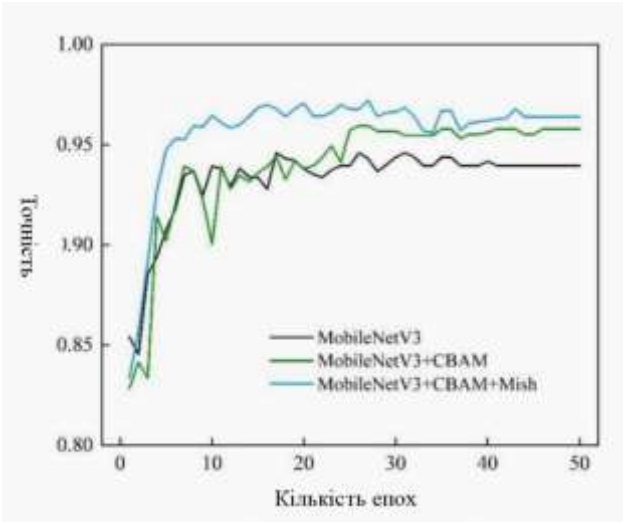


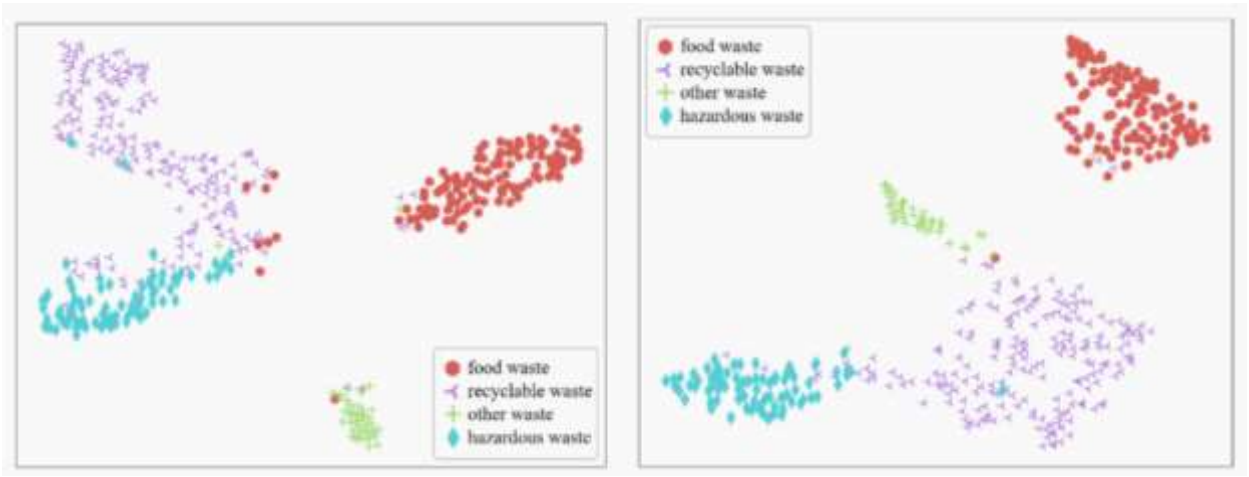
Рисунок 16 — Порівняльні криві абляційного експерименту

Для наочного відображення змін продуктивності моделі в процесі її вдосконалення використовувалися такі показники оцінки: F1-міра (F1-Score), кількість параметрів моделі (Parameters), обсяг пам’яті, яку займає модель (Size), та час розпізнавання одного зображення (Time). З табл. 5 видно, що заміна механізму уваги SE-Net на CBAM дозволила знизити кількість параметрів моделі на 23,03 %, одночасно підвищивши F1-міру на 1,81 %, досягнувши 95,76 %. Використання функції активації Mish у глибоких шарах мережі покращує узагальнювальні властивості згорткових шарів, що підвищило F1-міру моделі на 0,63 %, але при цьому трохи збільшило обчислювальні витрати. В результаті, порівняно з оригінальною моделлю, покращена модель MobileNetV3 показала збільшення F1-міри на 2,44 %, зменшення розміру моделі на 22,05 %, а час розпізнавання одного зображення зріс на 3,5 мс. Отже, при незначному збільшенні часу розпізнавання було досягнуто підвищення точності класифікації та оптимізацію розміру моделі.

Таблиця 5 — Порівняння параметрів покращеної моделі MobileNetV3

Модель	F1-Score/%	Parameters/M	Size/MB	Time/мс
MobileNetv3	93,95	1,52	5,94	22,9
MobileNetV3+CBAM	95,76	1,17	4,63	24,6
MobileNetV3+CBAM+Mish	96,39	1,17	4,63	26,4

Для візуалізації моделі було застосовано метод зниження розмірності t-SNE, який обчислює міру подібності між парами екземплярів у високовимірному та низьковимірному просторах. Як показано на рис. 17, оригінальна модель вважала, що частина характеристик переробного та шкідливого сміття є досить схожими, що пояснює помилки класифікації між цими двома класами у вихідній моделі. У покращеній моделі ступінь подібності між чотирма класами сміття нижчий, кожен клас має унікальні ознаки, що зменшує плутанину під час класифікації та забезпечує більш стабільну роботу моделі.



Початкова модель Покращена модель

Рисунок 17 — Візуалізація покращення моделі t-SNE

Рис. 18 показує результати візуалізації Grad-CAM для прогнозів сміття до та після покращення моделі. Дев'ять карт ознак сміття відповідають тепловим картам дев'яти згорткових шарів у структурі моделі, що відображають області уваги моделі під час розпізнавання та ступінь фокусування на об'єкті. З рисунка видно, що покращена модель краще витягує текстурні ознаки та більш точно визначає контури об'єктів. Крім того, при розпізнаванні розмитих та складних зображень покращена модель демонструє вищий рівень уваги до об'єктів порівняно з оригінальною моделлю. Тому точність класифікації Top-1 покращеної моделі при розпізнаванні сміття є вищою за точність оригінальної моделі.

Для перевірки продуктивності покращеного алгоритму детекції об'єктів у цій роботі обрано алгоритми YOLOv5s, YOLOv5l, YOLOv4 та YOLOv4-Tiny для порівняння з запропонованим алгоритмом. Всі моделі тренувалися та тестувалися на власноруч зібраному наборі даних. Результати порівняння продуктивності наведені в табл. 6.

З таблиці видно, що запропонований алгоритм детекції має вищий показник mAP порівняно з іншими алгоритмами: у порівнянні з YOLOv5l і YOLOv4 mAP вищий на 0,4 % і 1,2% відповідно, а FPS вищий на 36,9 % і 85,3 %. Порівняно з легковаговими версіями YOLOv5s і YOLOv4-Tiny, хоча швидкість детекції трохи зменшилася, mAP підвищився на 3,3 % і 4,9 % відповідно, при цьому використання пам'яті менше, ніж у інших моделей. Отже, загальна продуктивність запропонованої моделі перевищує сучасні популярні алгоритми детекції об'єктів, забезпечуючи ефективне та точне розпізнавання сміття.

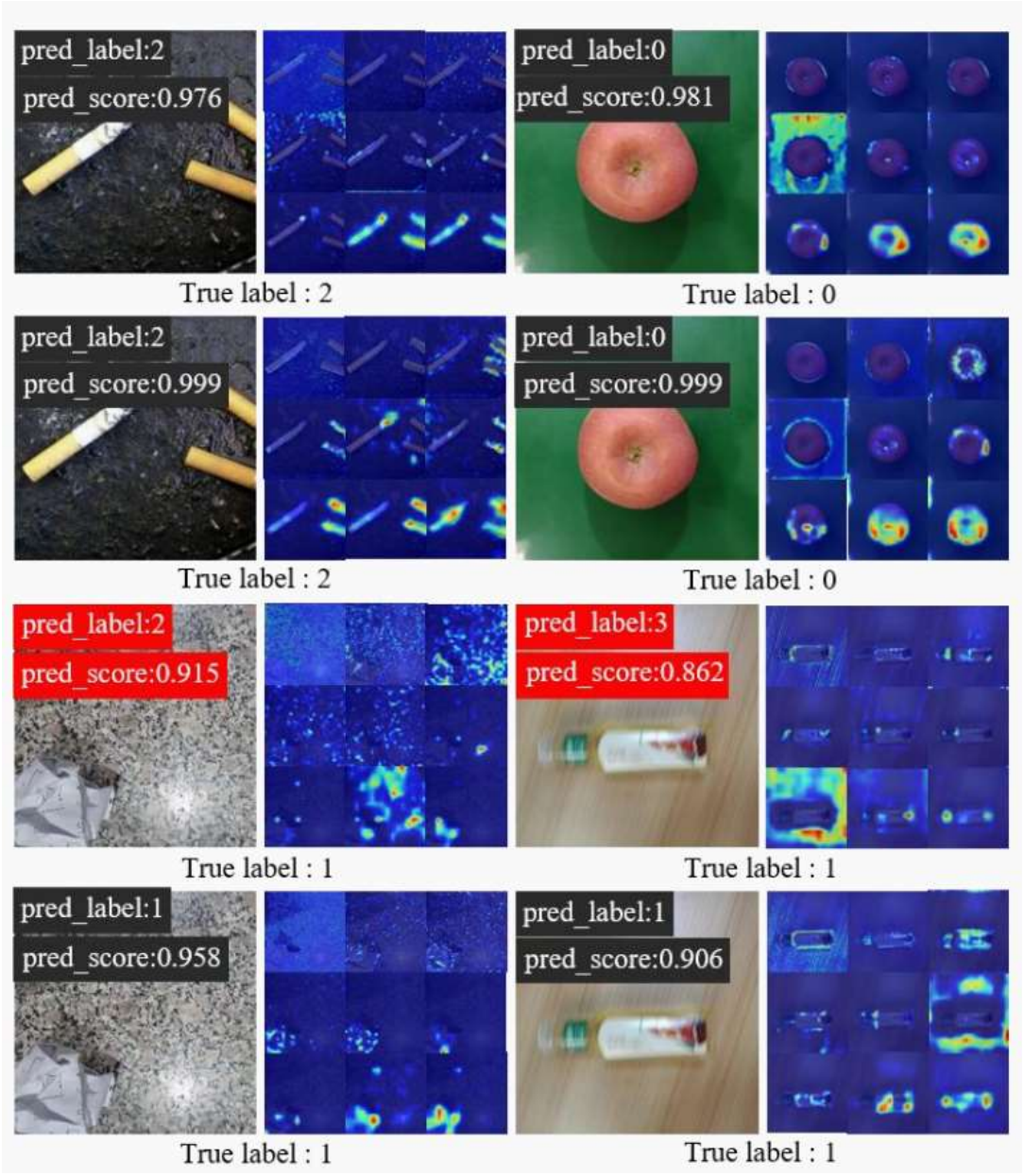


Рисунок 18 — Візуалізація покращення моделі Grad-CAM

Таблиця 6 — Порівняння продуктивності покращеного алгоритму з основними алгоритмами

Модель	mAP/%	Parameters/M	Size/MB	FPS
YOLOv5l	94,2	46,6	93,7	46
YOLOv5s	91,3	7,07	14,4	82
YOLOv4	93,4	64,4	256,2	34
YOLOv4-Tiny	89,7	11,8	23,5	97
Запропонована	94,6	5,03	9,87	6

5. Висновки

Застосування згорткових нейронних мереж для розпізнавання окремих класів об’єктів серед сумішей різних об’єктів дозволяє проводити сортування та виділення цільових

об’єктів у режимі реального часу. Важливою сферою впровадження методів сортування на основі згорткових нейронних мереж є сортування сміття.

На основі аналізу поширених методів розпізнавання зображень із використанням нейромережевого підходу окреслено множину моделей розпізнавання окремих типів сміття та визначено способи їх удосконалення. Запропоновано композиційний алгоритм визначення категорій сміття з використанням удосконалених моделей та проведено його моделювання. Показано покращення параметрів обробки зображень як стосовно точності, так і операційної ефективності.

Подальші дослідження направлені на впровадження розробленого алгоритму і засобів обробки зображень у системі сортування сміття.

СПИСОК ДЖЕРЕЛ

1. Коваленко О.Є., Лі Л. Застосування інтелектуальної обробки зображень у системах управління життєвим середовищем. *Математичні машини і системи*. 2024. № 1. С. 55–69. DOI: https://doi.org/10.34121/1028-9763-2024-1-55-69.
2. Redmon J., Divvala S., Girshick R. You only look once: Unified, real-time object detection. *Proc. of the IEEE conference on computer vision and pattern recognition*. Piscataway, NJ: IEEE, 2016. P. 779–788.
3. Redmon J., Farhadi A. YOLO9000: better, faster, stronger. Proceedings of the IEEE conference on computer vision and pattern recognition, Piscataway, NJ: IEEE, 2017. P. 7263–7271.
4. Redmon J., Farhadi A. YOLOv3: An incremental improvement. arXiv:1804.02767, 2018.
5. Bochkovskiy A., Wang C., Liao H. YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv:2004.10934, 2020.
6. Howard A., Sandler M., Chu G. et al. Searching for MobileNetV3. *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Piscataway, NJ: IEEE, 2019. P. 1314–1324.
7. Hu J., Shen L., Sun G. Squeeze-and-excitation networks. *Proc. of the IEEE conference on computer vision and pattern recognition*. Piscataway. NJ: IEEE, 2018. P. 7132–7141.
8. Woo S., Park J., Lee J. CBAM: convolutional block attention module. *Proc. of the European Conference on Computer Vision (ECCV)*. Berlin: Springer, 2018. P. 3–19.
9. Xu B., Wang N., Chen T. et al. Empirical evaluation of rectified activations in convolutional network. arXiv:1505.00853, 2015.
10. Misra D. Mish: A Self Regularized Non-Monotonic Activation Function. arXiv:1908.08681, 2019.
11. Rezatofighi H., Tsoi N., Gwak J. et al. Generalized intersection over union: A metric and a loss for bounding box regression. *Proc. of the IEEE/CVF conference on computer vision and pattern recognition*. Piscataway, NJ: IEEE, 2019. P. 658–666.
12. Zheng Z., Wang P., Liu W. et al. Distance-IoU loss: Faster and better learning for bounding box regression. *Proc. of the AAAI conference on artificial intelligence*. Palo Alto, CA: AAAI, 2020. N 34 (7). P. 12993–13000.
13. Ma S., Xu Y. A loss for efficient and accurate bounding box regression. arXiv: 2307. 07662, 2023.
14. Neubeck A., Van Gool L. Efficient non-maximum suppression. *18th international conf. on pattern recognition*. Piscataway, NJ: IEEE, 2006. N (3). P. 850–855.
15. Bodla N., Singh B., Chellappa R. et al. Soft-NMS ― improving object detection with one line of code. *Proc. of the IEEE international conference on computer vision*. Piscataway, NJ: IEEE, 2017. P. 5561–5569.
16. Bewley A., Ge Z., Ott L. et al. Simple online and realtime tracking. *2016 IEEE international conference on image processing (ICIP)*. Piscataway, NJ: IEEE, 2016. P. 3464–3468.
17. Wojke N., Bewley A., Paulus D. Simple online and realtime tracking with a deep association metric. *2017 IEEE international conference on image processing (ICIP)*. Piscataway, NJ: IEEE, 2017. P. 3645–3649.
18. Thung G., Yang M. TrashNet: A Dataset for Garbage Detection and Classification. GitHub repository, 2017. URL: https://github.com/garythung/trashnet.

Стаття надійшла до редакції 01.10.2025