

УДК 004.8:[623.438+623.55.021]

М. А. Назаркевич, Н. Т. Олексів

Національний університет "Львівська політехніка", м. Львів, Україна

СИСТЕМА РОЗПІЗНАВАННЯ ОБ'ЄКТІВ НА ОСНОВІ МОДЕЛІ YOLO

Побудовано систему розпізнавання об'єктів, знятих у режимі реального часу на відеокамеру в зашумленому та змінному щодо навколишніх умов середовищі. Досліджено методику наповнення бази даних для мобільних військових об'єктів. Для розпізнавання об'єктів використано нейромережу YOLO v8, яка дає змогу відстежувати рухomi та ідентифікувати об'єкти, які потрапляють на відео із відеокамери. Ця нейромережа дає змогу відстежувати об'єкти зі зміною масштабу, під час руху з перешкодами. З'ясовано, що розпізнавання об'єктів здійснюється на основі контурного аналізу, зіставлення із шаблоном та виявлення і встановлення відповідності ознакам. Використано методи штучного інтелекту на основі YOLO v8 для розпізнавання військової техніки. Здійснено навчання для різних моделей YOLO із використанням оптимізаторів Adam W, Adam, SGD та роздільної здатності 512×512, 640×640, 1024×1024 px зображень. Поліпшення розпізнавання об'єктів досягається завдяки аналізу контурів, порівнянню шаблонів і введених особливих точок. Різні роздільні здатності зображень та оптимізатори по-різному впливали на продуктивність моделі, а стандартні метрики оцінки не надають найточнішого вигляду. Найефективнішим оптимізатором є метод градієнтного спуску (SGD), який показав найкращі показники точності для розпізнавання бойових машин. Градієнт зазвичай розглядають як суму градієнтів, зумовлених кожним елементом навчання, і використовують для коригування параметрів моделі. Внаслідок розроблення системи сформовано показники із точністю розпізнавання (accuracy) 92 %, F1-оцінка (F1 score) – 89 %, середній показник точності (mAP) – 90 %. Запропоновано спосіб наповнення набору даних та створення класифікатора. Побудовано модель розпізнавання бойових машин. Наведено графіки, результати розпізнавання рухомих об'єктів у нейромережі Yolo8 x.

Ключові слова: розпізнавання, YOLO, штучний інтелект, відстеження об'єктів, нейронна мережа.

Вступ / Introduction

У сфері комп'ютерного зору технології штучного інтелекту передбачають автоматичне розпізнавання рухомих об'єктів, виявлення перешкод під час руху об'єктів та здійснення їх ідентифікації [1].

Однією із наймасштабніших розробок для розпізнавання рухомих об'єктів у відеопотоці є нейромережа YOLO. Зокрема, в [2] запропоновано детектор об'єктів на основі глибинних ознак. Він покращує здатність тренувати мережу за допомогою методів навчання із кількома масштабами без збільшення розміру мережі та інтегрує численні функції відстеження моделей спостереження для точного визначення місця розташування об'єктів. Однак цей метод недостатньо ідентифікує об'єкти, до того ж виявлення у складних сценах з великою кількістю об'єктів хибне.

Застосовуючи методи комп'ютерного зору для пошуку об'єктів на зображенні, як правило, використовують три основні способи розпізнавання об'єктів на зображенні:

- контурний аналіз [3];
- відповідність шаблону [4];
- виявлення ознак, опис і надання їм відповідності [5].

Одним зі способів розпізнавання об'єктів із відеопотоку є використання методів пошуку за шаблоном. Цей метод враховує інформацію про те, як виглядає потрібний об'єкт, який у нього може бути фон, як

виглядають певні контури об'єкта і в якому положенні вони можуть перебувати. Також враховується можливе місце виявлення об'єкта. Метод дає змогу досягти високої якості розпізнавання, у нього доволі висока швидкість. Однак, коли відеокамера фіксує кілька схожих один на одного об'єктів, які відповідають різним шаблонам, якість розпізнавання погіршується.

Об'єкт дослідження – процеси збирання, опрацювання та пошуку для виявлення об'єктів в умовах зашумленого фону та навчання на основі розробленого власного датасету.

Предмет дослідження – моделі, методи та засоби дослідження ефективності розпізнавання об'єктів військового призначення, зокрема метод градієнтного спуску.

Мета роботи – формування dataset для розпізнавання рухомих об'єктів та вдосконалення згорткових мереж для зменшення часу розпізнавання.

Для досягнення зазначеної мети визначено такі основні завдання дослідження:

- створення dataset для розпізнавання рухомих об'єктів;
- попереднє оброблення даних для зменшення часу виявлення об'єктів;
- удосконалення згорткових мереж із розпаралеленням процесу виявлення об'єктів.

Аналіз останніх досліджень та публікацій. Дослідження [6] стосується виявлення та відстеження об'єктів. Запропоновано новий підхід за допомогою інтеграції архітектури You Only Look Once (YOLO) для виявлення об'єктів із фільтром із метою відстеження цілей. У поєднанні зі структурою виявлення реалізований інноваційний ковзний фільтр, новий фільтр, для підвищення надійності відстеження в умовах перешкод. Зокрема, ковзний інноваційний фільтр реалізовано для підвищення продуктивності відстеження за допомогою оцінювання оптимального розташування центроїда в кожному кадрі та оновлення траєкторії об'єкта. Відстеження цілей традиційно покладається на методи теорії оцінки, такі як фільтр Калмана, а ковзний інноваційний фільтр запропонований як надійна альтернатива, особливо придатна для сценаріїв із апіорною інформацією про динаміку системи.

У [7] розглянуто детектори об'єктів, які поділено на дві категорії – двоступеневі та одноступеневі. Двоступеневі детектори зосереджені переважно на вибірковій стратегії регіональних пропозицій через складну архітектуру; однак одноступеневі детектори зосереджені на всіх координатах зображення для виявлення об'єктів за допомогою простішої архітектури одним проходом.

У [8] реалізовано навчання на основі YOLO, який є найшвидшим детектором об'єктів загального призначення. І YOLOv2 забезпечує найсучасніше і краще співвідношення між ними – швидкість у реальному часі та відмінну точність для виявлення об'єктів, ніж інші системи виявлення, завдяки різноманітності наборів даних виявлення. Крім того, YOLO краще узагальнює представлення об'єкта, ніж інші моделі, що робить його ідеальним для програм, які покладаються на швидке та надійне виявлення об'єктів. Зважаючи на визначні та цінні переваги, він гідний того, щоб його настійно рекомендували та популяризували. Окрім структури кожного алгоритму, найактуальнішою проблемою для машинного навчання є обсяг набору даних. Наявність відповідних навчальних даних може бути життєво важливою частиною навчання, щоб досягти ідеї результатів.

Мета дослідження [9] – виявлення об'єктів на відео у реальному часі на вбудованих пристроях. Використано платформу для розвитку архітектури мережі YOLOv2, яка має у 2,8 рази менше параметрів. Для подальшого зменшення енергоспоживання на вбудованих пристроях зі збереженням продуктивності введено метод логічного зв'язку із адаптацією до руху для зменшення частоти на основі характеристик тимчасового руху.

У [10] подано зважену двонаправлену пірамідну мережу ознак BiFPN. Крім того, інші методи, зокрема доповнення тесту часу (TTA) й об'єднання зважених коробів (WBF), допомагають досягти кращої точності та міцності. Наші експерименти продемонстрували, що ViT-YOLO істотно перевершує найсучасніші детектори, та дали змогу досягти одного з найкращих результатів.

Щоб вирішити проблеми з виявленням об'єктів на інфрачервоних зображеннях, такі як низька швидкість розпізнавання та висока частота помилкових тривог, спричинена великими відстанями, слабкою енергією та низькою роздільною здатністю, у [11] запропоновано безрегіональний режим детектора об'єктів під назвою YOLO-FIR для інфрачервоних зображень з ядром YOLOv5 зі стисненням каналів, оптимізацією параметрів тощо. Покращено інфрачервону мережу виявлення об'єктів YOLO-FIRI. Зокрема, під час проектування

мережі із виділення ознак (міжступеневі часткові з'єднання (CSP)) модуль у дрібному шарі розширюється та повторюється, щоб максимізувати використання дрібних функцій. Крім того, у залишкових блоках введено вдосконалений модуль уваги для фокусування на об'єктах. Крім того, для підвищення точності виявлення малих об'єктів додано багатомасштабне виявлення. Експериментальні результати на наборах даних KAIST і FLIR показують, що YOLO-FIRI демонструє якісне покращення порівняно із найсучаснішими детекторами. Порівняно з YOLOv4 середня точність (mAP50) YOLO-FIRI збільшується на 21 % на наборі даних KAIST, швидкість зменшується на 62 %, параметри знизилися на 89 %, розмір ваги зменшився більш ніж на 94 %, а витрати на обчислення – на 84 %. Порівняно з YOLO-FIR, YOLO-FIRI покращує артеріальний тиск приблизно на 5–20 %, AR (середнє відкликання), mAP50, F1 і mAP50:75. Крім того, через недоліки високого шуму і слабкі особливості злиття зображень можна застосувати до попередньої обробки зображень як метод покращення даних за допомогою злиття видимих та інфрачервоних зображень на основі згорткової нейронної мережі.

Результати дослідження та їх обговорення / Research results and their discussion

Контурний аналіз. Відомим методом визначення рухомих об'єктів є контурний аналіз [12]. Це метод описання, зберігання, розпізнавання, порівняння та пошуку графічних зображень (об'єктів) за їх контурами. Контур повністю визначає форму зображення і містить всю необхідну інформацію для розпізнавання зображень за формою. Такий підхід дає змогу не враховувати внутрішні точки зображення і цим самим істотно зменшити кількість інформації, що трансформується. Контур – крива, яка описує межу об'єкта на зображенні. Тому є можливість розглядати контури об'єктів, що зменшує складність алгоритмів. Головною перевагою контурного аналізу є незмінність щодо повороту, масштабу та зміщення контуру на зображенні.

Метод дає добрі результати у разі пошуку предмета заданої форми. Завдяки цьому часто можна забезпечити роботу системи в режимі реального часу. Однак у цього методу є істотні недоліки. Доволі часто трапляються розриви контуру. Отже, контурний аналіз характеризується невисокою стійкістю до перешкод, і будь-яке порушення цілісності контуру або погана видимість об'єкта призводить або до неможливості виявлення, або до помилкових спрацьовувань. Простота і швидкість аналізу контурів дають змогу успішно застосовувати цей підхід за умови наявності чітко окресленого об'єкта на контрастному фоні.

Метод порівняння на основі шаблонів [13] виконується за критерієм мінімуму (максимуму) деякої функції, що порівнює об'єкт та певний шаблон. Такий спосіб ідентифікації є одним із найпростіших. Вхідними параметрами методу є зображення, за яким необхідно шукати шаблон, зображення об'єкта, який потрібно знайти на тестованому зображенні. Мета методу – знайти на тестованому зображенні ділянку, що відповідає шаблону. Пошук шаблону виконується із його послідовним переміщенням на один піксель по зображенню та оцінюванням подібності кожної нової області до шаблону. За результатами перевірки вибирають ділянку з найбільшим коефіцієнтом збіжності. Це відсоток перекриття між областю зображення та шаблоном. Зіставлення за шаб-

лоном не дає змоги стверджувати, чи знайдено вихідний об'єкт, оскільки це імовірна характеристика, яка залежить від масштабу, кутів огляду, поворотів зображення та наявності фізичних перешкод [14]. Також можливі помилкові спрацювання алгоритму, коли шуканого об'єкта насправді немає, але є деякі загальні деталі в шаблоні та області на тестованому зображенні. Звичайно, цієї ситуації можна уникнути, перевірявши значення коефіцієнта збіжності (щоб воно було не меншим від певного порога).

Пошук за спеціальними точками. Часто алгоритми використовують ключові точки [15] (особливі точки) зображення для своєї роботи. Під ключовими точками розуміють окремі ділянки зображення, характерні для цього зображення. Існує велика кількість методів виявлення таких "особливих точок". Усі вони відрізняються швидкістю роботи, кількістю виділених точок, а також стійкістю до трансформацій зображення: повороту, зміни кутів огляду, зміни масштабу.

Для пошуку ключових точок на зображеннях та їх подальшого порівняння використовують три компоненти:

- детектор ознак [15] – шукає ключові точки на зображенні;
- дескриптор (екстрактор дескрипторів) – створює опис знайдених ключових точок, оцінюючи їх положення через опис навколишніх територій;
- matcher (матчер) – будує відповідності між двома наборами точок зображення.

На відміну від зіставлення шаблонів і аналізу контурів, алгоритми пошуку ключових точок стійкіші до перешкод, трансформацій і дають змогу знаходити об'єкти навіть за наявності фізичних перешкод.

Важливу роль у розпізнаванні рухомих об'єктів відіграє попереднє оброблення ключових кадрів. Від його налаштувань великою мірою залежить результат розпізнавання. Істотну роль відведено розпізнаванню кольорів на ключових кадрах, зокрема, розпізнаванню зеленого та брудно-зеленого кольору. Для цього здійснюємо фільтрацію у каналах кольору ключових кадрів відеопотоку.

Щоб був досягнутий максимально можливий рівень відстеження об'єкта (маркера), він повинен мати значну кількість унікальних (стабільних) ключових точок, які можна швидко виділити у відеопотоці та порівняти з наявним набором шаблонів. Для цього необхідно використовувати максимально швидкий детектор і дескриптор. Цей алгоритм дає змогу розпізнавати зображення під різними

кутами, на різних відстанях від камери, за різного освітлення та у разі часткового перекриття зображення [1].

Архітектура алгоритму YOLO. Модель YOLOv8 [16] використовують для виявлення об'єктів, цей алгоритм доступний у чотирьох основних версіях: малий (s), середній (m), великий (l) і надзвичайно великий (x). Кожна нова версія забезпечує все вищий рівень точності. Крім того, кожен варіант потребує певної кількості часу на тренувальний процес.

Метою графіки є створення високоефективної моделі детектора об'єктів із зазначенням продуктивності на осі Y і часу виконання на осі X. Початкові результати вказують на те, що YOLOv8 розпізнає із найвищою точністю.

Архітектура YOLO передбачає поділ вхідного зображення, передбаченого обмежувальними рамками, на сітку та ймовірності класу.

У [17] наведено огляд архітектури YOLO, в якому згортковий шар 448×448 пікселів, далі передбачено перетворення в основному блоці згорткової мережі для обчислювальних задач великої складності. Шар складається із фільтрів невеликого розміру через весь вхідний сигнал. Ідея методу полягає в тому, що нейромережа опрацьовує все вхідне зображення за один раз. Це забезпечує істотне збільшення швидкодії порівняно з іншими методами. Архітектура YOLO складається зі згорткового шару $3 \times 3 \times 192$, який об'єднує $2 \times 2 \times s-2$, далі іде шар $1 \times 1 \times 128$, потім згортковий шар $1 \times 1 \times 128$, відтак $1 \times 1 \times 512$, і завершується $3 \times 3 \times 1024$ [2]. Потім відбувається об'єднання шарів.

Вхідний шар: YOLO приймає вхідне зображення, як правило, фіксованого розміру.

Поділ на сітку: зображення розділено на сітку $S \times S$. Кожна клітинка сітки відповідає за обмежувальні рамки.

Прогноз обмежувальної рамки: кожна клітинка сітки передбачає кілька обмежувальних рамок. Ці рамки містять інформацію про координати (x, y) рамки, ширину (w), висоту (h) і показники достовірності.

Прогнозування класу: для кожної обмежувальної рамки модель передбачає ймовірності класу для різних категорій об'єктів. Це забезпечено за допомогою функції активації *softmax*.

Оцінка достовірності: кожне передбачення обмежувальної рамки пов'язане з оцінкою достовірності, яка вказує, наскільки ймовірно, що обмежувальна рамка містить об'єкт. Ця оцінка враховує як ймовірність присутності об'єкта, так і точність обмежувальної рамки.

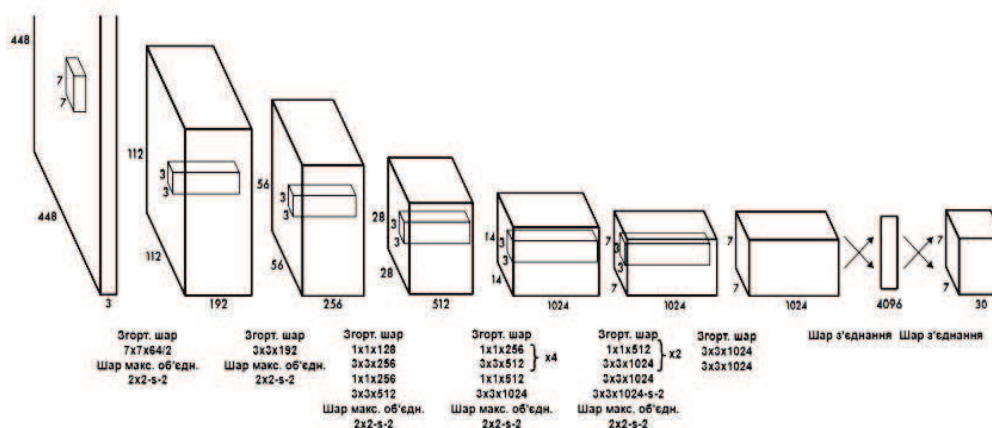


Рис. 1. Архітектура YOLO [17] / YOLO architecture [17]

Остаточний прогноз: остаточні прогнози отримано у результаті об'єднання координат обмежувальної рамки, ймовірностей класу та показників достовірності. Потім застосовано немаксимальне придушення, щоб відфільтрувати надлишкові та низьконадійні прогнози.

Вихід: остаточним результатом є набір обмежувальних рамок, кожна із яких пов'язана з міткою класу та оцінкою достовірності.

Функція втрат: YOLO використовує функцію втрат із кількох частин, яка містить терміни для наявності/відсутності об'єкта, координати обмежувальної рамки та ймовірності класу. Це дає змогу навчити модель для точного виявлення.

Алгоритм YOLO приймає зображення як вхідні дані та використовує глибоку згорткову нейронну мережу для виявлення об'єктів на зображенні.

Перші 20 згорткових шарів моделі попередньо навчено за допомогою ImageNet [18], із використанням шару об'єднання середнього часового значення та повністю зв'язаного шару. Після цього навчена модель трансформується для виконання завдання виявлення об'єктів, щоб підвищити продуктивність навченої мережі. Останній повністю пов'язаний рівень YOLO виконує передбачення як ймовірностей класу, так і координат обмежувальної рамки.

YOLO передбачає кілька обмежувальних рамок для кожної комірки сітки. Під час навчання кожен предиктор обмежувальної рамки відповідає за передбачення об'єкта із найвищим значенням перекриття вибірки з правильною інформацією.

Для підвищення точності виявлення об'єктів YOLO використовує метод немаксимального придушення [19]. Ця техніка дає змогу ідентифікувати та видаляти зайві або неправильні обмежувальні рамки, які можуть бути створені для одного об'єкта. Внаслідок застосування МНП для кожного об'єкта на зображенні вибирається тільки одна обмежувальна рамка, що поліпшує якість і підвищує ефективність виявлення [19].

Продуктивність алгоритму YOLO оцінюють за допомогою набору даних COCO [20] (Common Objects in Context). Набір даних COCO – це великий набір даних, що складається із 80 класів ознак. YOLOv1, базова версія, може розпізнавати 24 класи об'єктів і має 21,6 % mAP (середню точність), виміряну за допомогою набору даних COCO. YOLOv2 може розпізнавати 90 класів об'єктів із mAP набору даних COCO 30,2 %. YOLOv3 може розпізнавати 1000 класів функцій із mAP 57,9 %. YOLOv4 має вищу продуктивність порівняно з YOLOv3, але може розпізнавати 80 класів функцій із mAP 60,0 %. YOLOv8 істотно підвищує продуктивність порівняно з YOLOv4 і досягає mAP 83,5 % на наборі даних COCO. YOLOv6 досяг 84,4 % набору даних COCO mAP. YOLOv7 досяг mAP 85,4 %. YOLOv8 забезпечує додаткові підвищення продуктивності порівняно з YOLOv7 і досягає mAP 86,4 % [21].

Побудова моделі навчання. Навчання нейронної мережі полягає у налаштуванні параметрів мережі для отримання кращих результатів. Для більшості задач у нейронних мережах навчання виконують, подаючи у систему дані для навчання мережі, а правильна відповідь не відома заздалегідь. Поліпшення результатів і точності мережі досягають за допомогою мінімізації певної похибки, різниці між прогнозованими і фактичними виходами.

Точність моделі показує, наскільки добре модель передбачає правильний результат порівняно із фактичним результатом. Його розраховують, поділивши кількість правильних прогнозів на загальну кількість прогнозів. Хоча точність є корисним показником, іноді її недостатньо для оцінювання продуктивності складних моделей штучного інтелекту. Тому важливо враховувати інші метрики, такі як повторюваність і оцінка F1 [22].

Помилки класифікації – матриця плутанини [23] (matrix error). Якщо є два класи та алгоритм, який прогнозує кожен об'єкт до одного з класів, то матриця помилок класифікації виглядатиме так, як у табл. 1.

Табл. 1. Матриця плутанини / The confusion matrix

	$y = 1$	$y = 0$
$\tilde{y} = 1$	Позитив (TP)	Хибний позитив (FP)
$\tilde{y} = 0$	Хибний негатив (FN)	Негатив (TN)

де $\tilde{y}$ – відповідь алгоритму на об'єкті, а y – мітка класу на цьому об'єкті.

Точність вимірює відношення позитивних прогнозів до всіх позитивних прогнозів, які зробила модель. Точність показує, наскільки добре модель уникатиме помилкових спрацьовувань. З іншого боку, повторюваність вимірює співвідношення правильних позитивів до всіх фактичних позитивів. Це показує, наскільки успішно модель уникне помилкових негативів.

Отже, застосовуються помилки класифікації, серед яких хибно негативні (FN) і хибно позитивні (FP) [24].

$$P = \frac{TP}{TP + FP} \times 100\%, \quad (1)$$

$$R = \frac{TP}{TP + FN} \times 100\%. \quad (2)$$

Оцінка F1 – це комбінація точності та повторюваності, що забезпечує збалансовану оцінку продуктивності моделі

$$F1 = \frac{2 \times P \times R}{P + R} \times 100\%. \quad (3)$$

Для попереднього порівняння моделей традиційно використовують еталонні набори даних. Набір даних COCO [25] використовує спеціальну метрику усередненої точності AP та mAP.

$$AP = \int_0^1 P(R) dR, \quad (4)$$

$$mAP = \frac{1}{n} \sum_{i=1}^n \frac{TP}{FP + TP}. \quad (5)$$

Це середня міра точності, яку беруть з різних значень перетину над об'єднанням.

Нейронна мережа на основі багатошарового персептрона – система взаємопов'язаних шарів нейронів. Кожен нейрон характеризується функцією активації, яка перетворює вхідний сигнал нейрона на вихідний. Зв'язки нейронів з іншими нейронами характеризуються коефіцієнтами зв'язку – вагами. Важливим фактором у навчанні нейронної мережі є тип вхідних даних. Для досягнення найкращих результатів необхідно попередньо відобразити дані за допомогою операцій центрування та масштабування:

$$x_n = (x - m_x) m_x. \quad (6)$$

Навчання є ітераційною послідовністю операцій для обчислення вихідного сигналу мережі та подальшої зміни ваг з'єднань. Як алгоритм коригування ваги в мережах зазвичай використовують алгоритм зворотного поширення помилок. Цей алгоритм належить до класу градієнтних алгоритмів, тобто зміни вагових коефіцієнтів зв'язків здійснюються в напрямі мінімізації градієнта помилки. Похибка передбачення під час навчання дорівнює різниці між сигналом на виході мережі та вихідним опорним значенням, що відповідає вхідним даним:

$$e_i = (y_i - d_i). \quad (7)$$

Навчання мережі необхідно здійснюватиме доти, доки не зменшиться середнє значення похибки за одну епоху навчання. Подальше навчання зазвичай призводить до погіршення аналітичних можливостей нейронної мережі.

Мережа була навчена за допомогою алгоритму зворотного поширення помилок і алгоритму градієнтного спуску. В основі ідеї алгоритму – використання початкової похибки нейронної мережі для обчислення значень корекції ваг нейрона в її прихованих шарах:

$$E = \sum_{i=1}^{k=2} (y - y'), \quad (8)$$

де k – кількість вихідних нейронів мережі; y – цільове значення; y' – фактичне вихідне значення.

Цей алгоритм є ітераційним, він використовує покрокове навчання. Це працює так: на навчальний вхід подається один тестовий приклад. Потім значення ваг коригуються. На кожній ітерації відбуваються прямий і зворотний проходи мережі. На прямому вході вектор поширюється від входів до виходів мережі. Так формується певний вихідний вектор, який відповідає фактичному стану ваг. Похибку нейронної мережі розраховують як різницю між фактичним і цільовим значеннями.

Під час зворотного проходу ця помилка поширюється від виходу мережі до її входів, а ваги нейронів коригуються відповідно до формули:

$$w_{ji}'(n) = -\eta \partial E_{av} \partial w_{ji}, \quad (9)$$

де w_{ji} – вага i -го зв'язку j -го нейрона; η – параметр швидкості навчання, що дає змогу додатково контролювати розмір кроку корекції; Δw_{ji} з метою точнішого налаштування на мінімальну похибку підбирають експериментально під час навчання (зміни в інтервалі від 0 до 1).

Метод стохастичного градієнтного спуску. Метод стохастичного градієнтного спуску (SGD) використовують для коригування параметрів моделі. Градієнт зазвичай розглядають як суму градієнтів, зумовлених кожним елементом навчання. Вектор параметра змінюється в напрямку антиградієнта із заданим кроком. Отже, стандартний градієнтний спуск потребує одного проходу над навчальними даними, перш ніж він зможе змінити параметри. За стохастичного градієнтного спуску значення градієнта апроксимується градієнтом функції витрат, розрахованим лише на одному навчальному елементі. Потім параметри змінюються пропорційно до близького градієнта. Отже, параметри моделі змінюються після кожного об'єкта навчання. Для великих наборів даних стохастичний градієнтний спуск може забезпечити значну перевагу в швидкості над стандартним градієнтним спуском.

Градієнтний спуск – це метод оптимізації, який використовують для пошуку мінімуму (або максимуму) функції. Основна ідея полягає у тому, щоб крок за кроком рухатися в напрямку, протилежному до напрямку градієнта (або похідної) функції в кожній точці, зменшуючи значення функції. Цей метод часто використовують для навчання моделей машинного навчання, таких як лінійна регресія.

Нехай $f_\theta(x)$ – це досліджувана функція, $L(\theta)$ – функція витрат, а, власне, θ – оптимізаційний параметр.

$$L(\theta) = \frac{1}{2N} \sum_{i=1}^N (y^i - f_\theta(x^i))^2 \quad (10)$$

$$f_\theta(x) = \sum_{j=1}^D \theta_j x_j, \quad (11)$$

де N – кількість ітерацій; D – кількість вхідних параметрів; x^i – незалежна змінна, а y^i – результат.

Алгоритм градієнтного спуску виконується доти, доки не стає збіжним вираз:

Розв'язати $L(\theta)$ для θ_j , щоб отримати градієнт для кожного θ_j :

$$\frac{\partial L(\theta)}{\partial \theta_j} = -\frac{1}{2N} \sum_{i=1}^N (y^i - f_\theta(x^i)) x_j^i. \quad (12)$$

Оновити кожен θ_j у зворотному (негативному) напрямку, для того щоб мінімізувати втрати:

$$\theta_j = \theta_j + \eta \frac{1}{N} \sum_{i=1}^N (y^i - f_\theta(x^i)) x_j^i. \quad (13)$$

Метод градієнтного спуску може забезпечити найоптимальніше рішення у випадку, коли цільова функція має форму опуклої кривої. Проте, коли значення змінної наближається до оптимальних значень, спуск може відбуватися повільніше і потрібно більше ітерацій. Цей оптимізатор продемонстрував найвищу ефективність під час розпізнавання об'єктів з шумами.

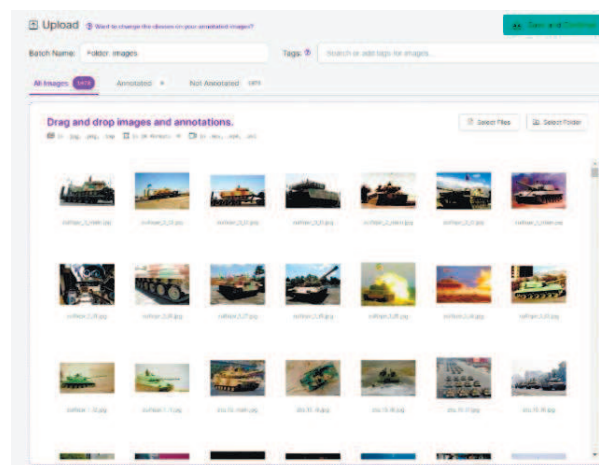


Рис. 2. Формування dataset / Dataset formation

Побудова класифікатора. Під час дослідження було використано власний набір даних, що містить багато зображень танків різних моделей і з різних ракурсів. Ці зображення взято з електронних книг і вебсайтів, із соціальних мереж [18]. Відповідно, необхідно було вручну додавати мітки до зображень для подальшого навчання моделі. Для ефективного завантаження та оброблення зображень ми вирішили використовувати платформу Roboflow.

Це точно призначало мітки кожному об'єкту на зображеннях, надаючи моделі достатньо інформації для розпізнавання та класифікації об'єктів.

Аналізуючи результати експерименту, бачимо, що оптимізатор стохастичного градієнтного спуску (SGD) виявився найефективнішим у більшості випадків. Наприклад, для роздільної здатності 640×640 SGD показав найвищу точність, відгук та F1-оцінку, а також найвищий показник середнього значення точності виявлення об'єктів (mAP@0.95). На інших роздільних здатностях результати SGD були такими самими. Оптимізатори Adam та AdamW також демонструють деяку ефективність, проте їх результати

виявилися нижчими порівняно з оптимізатором SGD у більшості випадків. Наприклад, в AdamW нижчі точність та відгук на роздільності 1024×1024 порівняно з SGD. Також помітно, що значення F1-оцінки для Adam та AdamW нижчі порівняно з SGD на всіх роздільних здатностях. Загалом результати свідчать про перевагу оптимізатора SGD у виявленні об'єктів за різних умов експерименту.

Обговорення результатів дослідження. Побудовано систему трекінгу об'єктів, які зняті з відеокамери і розпізнають бойові машини, що рухаються в польових умовах. Трекінг об'єктів здійснюється за допомогою YOLOv8 на основі алгоритмів машинного навчання.

Табл. 2. Результати навчання, отримані за допомогою технік машинного навчання *Матриця плутанини* /
Obtained learning results using the Confusion Matrix machine learning techniques

Optimizer	Precision	Recall	map@0.5	map@0.95	F1 score
512×512					
Adam W	0,340	0,873	0,352	0,248	0,488
Adam	0,835	0,842	0,811	0,489	0,839
SGD	0,965	0,950	0,977	0,200	0,950
640×640					
Adam W	0,930	0,810	0,878	0,681	0,865
Adam	0,821	0,765	0,788	0,433	0,792
SGD	0,969	0,960	0,982	0,833	0,964
1024×1024					
Adam W	0,921	0,722	0,808	0,622	0,810
Adam	0,820	0,764	0,787	0,432	0,791
SGD	0,963	0,967	0,972	0,825	0,965

Внаслідок навчання модель продемонструвала точність, досягши високого рівня правильної класифікації об'єктів у відеопотоці [26].

Крім того, навчена модель продемонструвала мінімальні втрати під час навчання, що свідчить про високу ефективність та оптимальну адаптацію до даних навчання. Графіки навчання та втрат чітко відображають стабільний процес навчання та відсутність значних флуктуацій втрат під час оптимізації моделі.

Отже, за результатами виконаної роботи можна сформулювати такі наукову новизну та практичну значущість результатів дослідження.

Наукова новизна отриманих результатів дослідження – розвинено надалі розпізнавання бойових машин у відеопотоці даних у режимі реального часу на основі методів машинного навчання YOLOv8 із наповненням власної бази даних бойових машин. Найкращих результатів досягнуто за допомогою оптимізатора стохастичного градієнтного спуску SGD.

Практична значущість результатів дослідження – досягнуто показників точності розпізнавання 96 %.

Висновки / Conclusions

Здійснено навчання для різних моделей YOLO із використанням оптимізаторів Adam W, Adam, SGD та роздільної здатності 512×512, 640×640, 1024×1024 px зображень. Не завжди вдавалося досягти оптимальної продуктивності, збільшуючи роздільну здатність зображення під час навчання.

Різні роздільні здатності зображень та оптимізатори неоднаково вплинули на продуктивність моделі, а стандартні метрики оцінки не забезпечують найточнішого вигляду.

Найефективнішим оптимізатором є SGD, найкращі показники точності для розпізнавання бойових машин.

Метрики оцінки середньої точності (mAP), точності ефективні в контексті виявлення об'єктів у військових застосуваннях. Під час тестування спостерігалось

обмежене виявлення об'єктів, незадовільне для військових цілей.

Для майбутніх досліджень рекомендовано проаналізувати вплив роздільної здатності зображення на значення оцінки моделі.

References

- [1] Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2021). A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems*.
- [2] Zhang, Y., Zhang, H., Huang, Q., Han, Y., & Zhao, M. (2024). DsP-YOLO: An anchor-free network with DsPAN for small object detection of multiscale defects. *Expert Systems with Applications*, 241, 122669. <https://doi.org/10.1016/j.eswa.2023.122669>
- [3] Patel, M., Liu, X. C., Yang, K., Tassone, C., Escott, B., & Thometz, J. (2024). 3D Back Contour Metrics in Predicting Idiopathic Scoliosis Progression: Retrospective Cohort Analysis, Case Series Report and Proof of Concept. *Children*, 11(2), 159. <https://doi.org/10.3390/children11020159>
- [4] Wang, Q., Lu, C., Gao, L., & He, G. (2024). Transformer-Based Multiple-Object Tracking via Anchor-Based-Query and Template Matching. *Sensors* (Basel, Switzerland), 24(1). <https://doi.org/10.3390/s24010229>
- [5] Lu, H., Nie, J. (2024). Coarse registration of point cloud base on deep local extremum detection and attentive description. *Multimedia Systems*, 30(1), 23. <https://doi.org/10.1007/s00530-023-01203-w>
- [6] Moksyakov A., Wu Y., Gadsden S. A., Yawney J., AlShabi M. Object Detection and Tracking with YOLO and the Sliding Innovation Filter. *Sensors*, 2024; 24(7):2107. <https://doi.org/10.3390/s24072107>
- [7] Diwan, T., Anirudh, G., & Tembhumne, J. V. (2023). Object detection using YOLO: Challenges, architectural successors, datasets and applications. *Multimedia Tools and Applications*, 82(6), 9243–9275. <https://doi.org/10.1007/s11042-022-13644-v>
- [8] Du, J. (2018, April). Understanding of object detection based on CNN family and YOLO. In *Journal of Physics: Conference Series* (Vol. 1004, p. 012029). IOP Publishing. <https://doi.org/10.1088/1742-6596/1004/1/012029>
- [9] Shafiee, M. J., Chywl, B., Li, F., & Wong, A. (2017). Fast YOLO: A fast you only look once system for real-time embedded

- object detection in video. *arXiv preprint arXiv:1709.05943*. <https://doi.org/10.15353/vsnl.v3i1.171>
- [10] Zhang, Z., Lu, X., Cao, G., Yang, Y., Jiao, L., & Liu, F. (2021). ViT-YOLO: Transformer-based YOLO for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 2799–2808). <https://doi.org/10.1109/ICCVW54120.2021.00314>
- [11] S. Li, Y. Li, Y. Li, M. Li and X. Xu. "YOLO-FIRI: Improved YOLOv5 for Infrared Image Object Detection", in *IEEE Access*, Vol. 9, pp. 141861–141875, 2021, <https://doi.org/10.1109/ICCVW54120.2021.00314>
- [12] Malik, J., Belongie, S., Leung, T., Shi, J. (2001). Contour and texture analysis for image segmentation. *International journal of computer vision*, 43, 7–27. <https://doi.org/10.1023/A:1011174803800>
- [13] Hashemi, N. S., Aghdam, R. B., Ghiasi, A. S. B., Fatemi, P. (2016). Template matching advances and applications in image analysis. *arXiv preprint arXiv:1610.07231*.
- [14] Cox, G. S. (1995). Template matching and measures of match in image processing. University of Cape Town, South Africa.
- [15] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60, 91–110. [tps://doi.org/10.1023/B:VISI.0000029664.99615.94](https://doi.org/10.1023/B:VISI.0000029664.99615.94)
- [16] Mukherjee, D., Jonathan Wu, Q. M., Wang, G. (2015). A comparative experimental study of image feature detectors and descriptors. *Machine Vision and Applications*, 26, 443–466. <https://doi.org/10.1007/s00138-015-0679-9>
- [17] Liu, Q., Ye, H., Wang, S., & Xu, Z. (2024). YOLOv8-CB: Dense Pedestrian Detection Algorithm Based on In-Vehicle Camera. *Electronics*, 13(1), 236. <https://doi.org/10.3390/electronics13010236>
- [18] Koga, S., Hamamoto, K., Lu, H., & Nakatoh, Y. (2024). Optimizing Food Sample Handling and Placement Pattern Recognition with YOLO: Advanced Techniques in Robotic Object Detection. *Cognitive Robotics*. <https://doi.org/10.1016/j.cogr.2024.01.001>
- [19] Wang, Y., Wang, B., Huo, L., & Fan, Y. (2024). GT-YOLO: Nearshore Infrared Ship Detection Based on Infrared Images. *Journal of Marine Science and Engineering*, 12(2), 213. <https://doi.org/10.3390/jmse12020213>
- [20] Wang, Z., Hua, Z., Wen, Y., Zhang, S., Xu, X., & Song, H. (2024). E-YOLO: Recognition of estrus cow based on improved YOLOv8 n model. *Expert Systems with Applications*, 238, 122212. <https://doi.org/10.1016/j.eswa.2023.122212>
- [21] Giudici, P., Centurelli, M., & Turchetta, S. (2024). Artificial Intelligence risk measurement. *Expert Systems with Applications*, 235, 121220. <https://doi.org/10.1016/j.eswa.2023.121220>
- [22] Shinde, S., Khoje, S., Raj, A., Wadhwa, L., & Shaikha, A. S. (2024). Artificial intelligence approach for terror attacks prediction through machine learning. *Multidisciplinary Science Journal*, 6(1), 2024011–2024011. <https://doi.org/10.31893/multiscience.2024011>
- [23] Dogan, A., Okatan, A., & Cetinkaya, A. (2021). Vehicle Classification and Tracking Using Convolutional Neural Network Based on Darknet Yolo with Coco Dataset. *AI and Big Data in Engineering Applications*, 179.
- [24] Nazarkevych, M., Oliarnyk, R., Troyan, O., & Nazarkevych, H. (2016, September). Data protection based on encryption using Ateb-functions. In 2016 XIth International Scientific and Technical Conference Computer Sciences and Information Technologies (CSIT) (pp. 30–32). IEEE. <https://doi.org/10.1109/STC-CSIT.2016.7589861>
- [25] Medykovskyy, M., Lipinski, P., Troyan, O., & Nazarkevych, M. (2015, September). Methods of protection document formed from latent element located by fractals. In 2015 Xth International Scientific and Technical Conference "Computer Sciences and Information Technologies" (CSIT) (pp. 70–72). IEEE. <https://doi.org/10.1109/STC-CSIT.2015.7325434>
- [26] Sheketa, V., Zorin, V., Chupakhina, S., Kyrsta, N., Pasyeka, M., & Pasioka, N. (2020, November). Empirical method of evaluating the numerical values of metrics in the process of medical software quality determination. In 2020 International Conference on Decision Aid Sciences and Application (DASA) (pp. 22–26). IEEE. <https://doi.org/10.1109/DASA51403.2020.9317218>

M. A. Nazarkevych, N. T. Oleksiv

Lviv Polytechnic National University, Lviv, Ukraine

OBJECT RECOGNITION SYSTEM BASED ON THE YOLO MODEL AND DATABASE FORMATION

A system for recognizing objects that are captured in real time on a video camera in a noisy environment that changes to the surrounding conditions has been built. The method of filling the database for mobile military objects was studied. For object recognition, the YOLO v8 neural network is used, which allows you to track moving and identify objects that fall into the video from the video camera. This neural network makes it possible to track objects with a change in scale, during movement with obstacles. It has been analyzed that the recognition of objects is carried out on the basis of contour analysis, comparison with a template and detection and matching of features. Artificial intelligence methods based on YOLO v8 were used to recognize military equipment. Trained for different YOLO models using Adam W, Adam, SGD optimizers and 512×512, 640×640, 1024×1024 px image resolution. Improved object recognition is achieved by analyzing contours, comparing patterns, and comparing entered special points. Different image resolutions and optimizers have shown different effects on model performance, and standard evaluation metrics do not provide the most accurate view. The most effective optimizer is gradient descent (SGD), which has shown the best accuracy for combat vehicle recognition. The gradient is usually considered as the sum of the gradients caused by each training element and is used to adjust the model parameters. As a result of the development of the system, indicators with recognition accuracy (accuracy) of 92 %, F1-estimate (F1 score) – 89 %, average indicator of accuracy (mAP) – 90 % were formed. A method of filling the data set and creating a classifier is proposed. A model of combat vehicle recognition was built. Graphs, results of recognition of moving objects in the Yolo8x neural network are presented.

Keywords: recognition, YOLO, artificial intelligence, decision support, neural network.

Інформація про авторів:

Назаркевич Марія Андріївна, д-р техн. наук, професор, кафедра інформаційних систем та мереж.

Email: mariia.a.nazarkevych@lpnu.ua; <https://orcid.org/0000-0002-6528-9867>

Олексів Назар Тарасович, аспірант, кафедра інформаційних систем та мереж. Email: nazar.oleksiv.mnsa.2020@lpnu.ua;

<https://orcid.org/0000-0001-7821-3522>

Цитування за ДСТУ: Назаркевич М. А., Олексів Н. Т. Система розпізнавання об'єктів на основі моделі YOLO. *Український журнал інформаційних технологій*. 2024, т. 6, № 1. С. 120–126.

Citation APA: Nazarkevych, M. A., & Oleksiv, N. T. (2024). Object recognition system based on the YOLO model. *Ukrainian Journal of Information Technology*, 6(1), 120–126 <https://doi.org/10.23939/ujit2024.01.120>