DOI: 10.18372/2310-5461.68.20280
УДК 656.71.06:656.7.08 (045)

С. Ю. Бєлов, аспірант
Державний університет "Київський авіаційний інститут"
orcid.org/0009-0006-7285-7021
e-mail: 8956599@stud.kai.edu.ua;

М. Ю. Заліський, д-р техн. наук, професор
Державний університет "Київський авіаційний інститут"
orcid.org/0000-0002-1535-4384
e-mail: maximus2812@ukr.net

ОГЛЯД МОДЕЛЕЙ НЕЙРОННИХ МЕРЕЖ ДЛЯ ЗАДАЧ ВИЯВЛЕННЯ 2D-ОБ'ЄКТІВ

Вступ

Бурхливий розвиток можливостей обчислювальної техніки за останні два десятиліття зробив можливою реалізацію практично будь-якої моделі нейронної мережі. Це призвело до широкого впровадження нейронних мереж і штучного інтелекту практично в усі галузі людської життєдіяльності: від розпізнавання обличчя користувача на смартфоні та індустрії розробки ігор до важкої промисловості. Сьогодні також активно розвиваються технології комп'ютерного зору, куди, зокрема, входять задачі виявлення і класифікації об'єктів на зображеннях.

Теоретичні основи функціонування нейронних мереж були започатковані в другій половині 50-х років двадцятого століття американським нейрофізіологом Френком Розенблатом. Його мотивувала фізіологія процесу людського сприйняття. Створену ним модель він назвав перцептроном. З усього різноманіття архітектури нейронних систем перцептрон є найпростішою. У 1960 році Розенблат представив реалізацію цієї моделі у вигляді першого нейрокомп'ютера «Марк-1», який міг розпізнавати деякі літери англійського алфавіту.

Отже, на практиці, слідуючи термінології біології, було продемонстровано принципово нову царину алгоритмів, де ключовим елементом є нейрон. У цьому можна вбачати ідею про можливість розкладання звичного нам алгоритму з його блок-схемою в нейронну мережу. З математичної точки зору, це чимось нагадує розкладення функції в тригонометричний ряд Фур'є (де основними елементами є тригонометричні функції косинуса і синуса) або в степеневий ряд (де основним елементом є степенева функція). Подібно до того, як результатом розкладання функції в ряд є значення коефіцієнтів ряду, так і результатом «розкладання» алгоритму в нейронну мережу є значення параметрів нейронів та їх конфігурація.

Аналіз останніх досліджень і публікацій

Аналіз показав, що кількість публікацій у популярній науково-метричній базі Scopus за останні 7 років за такими ключовими словами як: neural network, artificial intelligence, machine learning, deep learning та object detection, – стрімко зростає. Відповідні дані щодо ключових слів та кількості публікацій наведено в табл. 1.

Таблиця 1

Кількість публікацій у Scopus з штучного інтелекту та виявлення об'єктів за 2018–2024 роки

keywords	2018	2019	2020	2021	2022	2023	2024
neural network	42 217	59 400	75 997	91 731	105 241	114 901	129 041
artificial intelligence	23 321	23 319	30 246	34 161	36 429	47 565	68 263
machine learning	26 059	45 411	56 736	72 180	89 196	110 391	152 493
deep learning	17 174	32 952	46 799	63 536	82 719	99 852	113 857
object detection	4 902	6 756	7 877	9 839	12 447	15 387	17 094

Дані із табл. 1 свідчать про інтерес і плідну роботу з боку наукової спільноти щодо дослідження штучного інтелекту. Зрозуміло, що серед цієї величезної кількості наукових статей є власне статті, присвячені конкретним моделям виявлення об'єктів і побудови архітектури нейронних

мереж. У той же час необхідно виділити клас оглядових статей. Цього року вийшли дві ґрунтовні оглядові статті щодо виявлення об'єктів: Emanuele Malagoli та Luca Di Persio [1]; і друга – Paschalis Tsirtsakis et al [2]. Раніше в 2023р. вийшло також наукове дослідження Zhengxia Zou et al [3]. У цих роботах розповідається про постановку задачі виявлення об'єктів, наводяться великі відомості з таких досить фундаментальних речей, як загальноприйняті набори даних, показники якості функціонування технологій штучного інтелекту та розглянуто широкий каталог моделей детекторів об'єктів заданого типу під час обробки зображень.

Постановка завдання

Слід зазначити, що технології комп'ютерного зору знаходять своє втілення в інтроскопах, що застосовуються на доглядових пунктах, а саме в задачі виявлення та розпізнавання небезпечних і заборонених предметів. Науковий інтерес представляє визначення ефективності застосування цих технологій у цій конкретній задачі. Для цього необхідно розглянути, які саме технології комп'ютерного зору із застосуванням нейронних мереж існують на сьогоднішній день. Головною проблемою є недосконалість методів виявлення, що застосовуються в сучасних інтроскопах, у частині високої ймовірності хибної тривоги.

Мета і задачі дослідження

Метою цієї статті є опис існуючого на сьогоднішній день великого каталогу моделей виявлення 2D-об'єктів від традиційних методів, таких як детектор Віоли-Джонса, який не використовує згорткові нейронні мережі, до моделі нейронної мережі YOLOv11, випущеної в листопаді 2024 року. При цьому увага буде приділятися задачі визначення способу проведення порівняльного аналізу моделей виявлення.

Основний матеріал

Відомо, що задачі комп'ютерного зору включають задачі класифікації зображень (image classification), локалізації зображень (image localization), виявлення об'єктів (object detection) та сегментації зображень (image segmentation) (див. рис. 1).

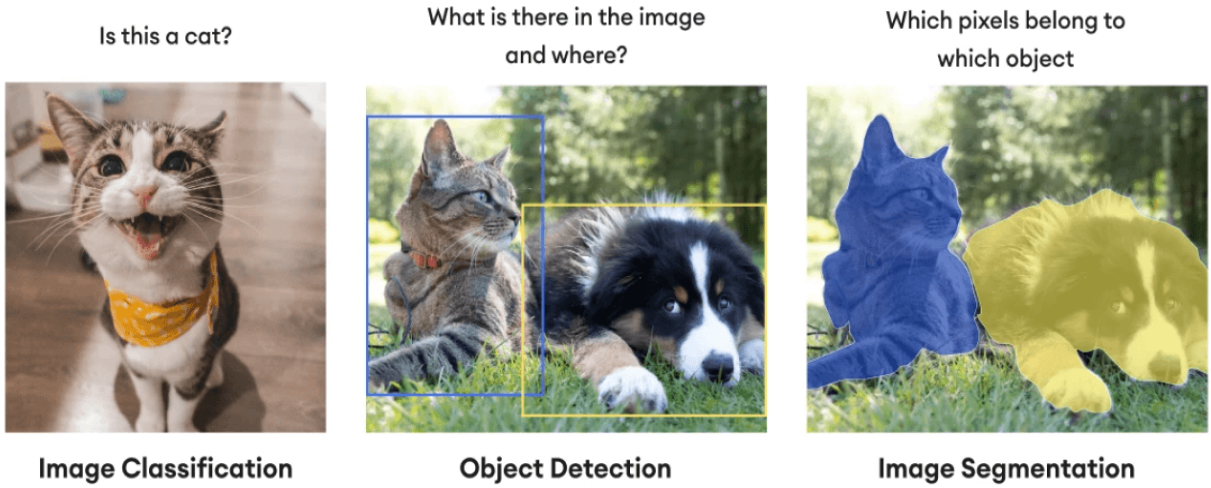

Рис. 1: Порівняння класифікації зображень, виявлення об'єктів і сегментації зображень.
Джерело: *https://www.ultralytics.com/blog/how-to-use-ultralytics-yolo11-for-image-classification*

У задачі класифікації зображення на вхід алгоритму подається зображення з одним об'єктом (наприклад, котом), а на виході очкується клас об'єкта (наприклад, кіт або собака). У задачі виявлення об'єктів на вхід алгоритму подається зображення з декількома об'єктами, а на виході очікується виявлення цих об'єктів у вигляді координат і розмірів обмежувальних рамок, а також визначення класу кожного виявленого об'єкта. У задачі сегментації аналогічний вхід, а на виході також вирішується задача класифікації для кожного виявленого об'єкта. Однак, замість обмежувальної прямокутної рамки алгоритм видає регіон з пікселями, що належать об'єкту. Ще виділяють задачу локалізації об'єкта на зображенні. Це окремий випадок задачі виявлення, коли на зображенні серед безлічі різних об'єктів знаходиться один і тільки один об'єкт заданого класу. У цій статті мова йде саме про задачу виявлення об'єктів, яка, втім, включає в себе і задачу класифікації. Каталог існуючих на сьогоднішній день методів виявлення 2D-об'єктів представлений в на рис. 2, каталог типових наборів даних (datasets) – в табл. 2.

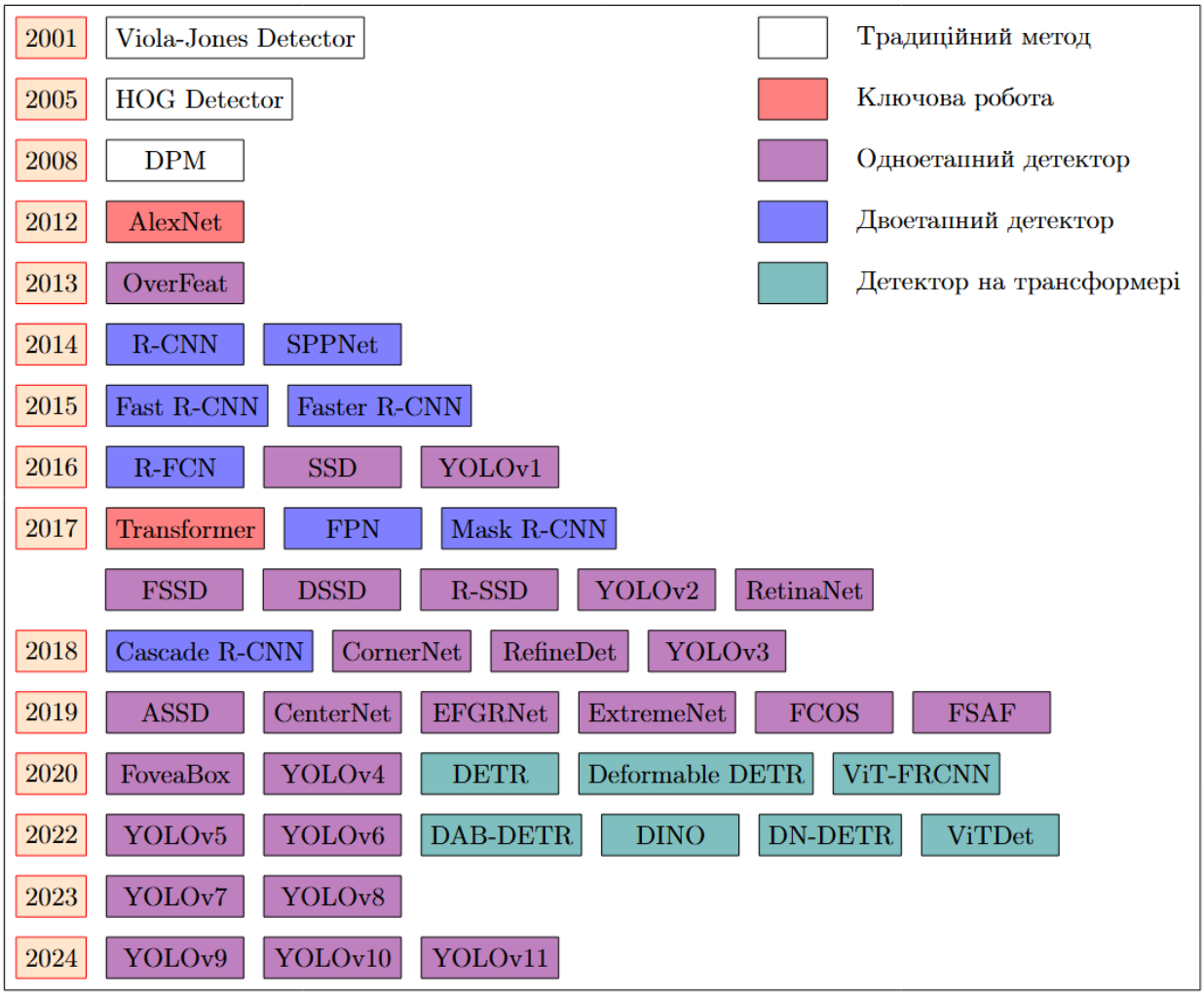

Рис. 2. Каталог методів виявлення 2D-об'єктів

Таблиця 2

Зведена інформація про широко використовувані набори даних для виявлення об'єктів. Для кожного набору даних вказано кількість класів і кількість зображень у тренувальних, валідаційних і тестових вибірках. Числа в дужках означають загальну кількість анотованих об'єктів

Dataset	Classes	Train	Validation	Test
PASCAL-VOC-2007	20	2501 (6301)	2510 (6307)	4952
PASCAL-VOC-2012	20	5717 (13 609)	5823 (13 841)	10 991
ILSVRC-2014	200	456 567 (478 807)	20 121 (55 502)	40 152
ILSVRC-2017	200	456 567 (478 807)	20 121 (55 502)	65 500
MS-COCO-2014	80	82 783	40 504	40 775
MS-COCO-2017	80	118 287	5000	40 670
OpenImages-v7	600	1 743 042 (14 610 229)	41 620 (303 980)	125 436 (937 327)
Objects365-2019	365	600 000 (9 623 000)	38 000 (479 000)	100 000 (1 700 000)

Як видно з рис. 2 та табл. 2, цей каталог досить різноманітний. Однак багато архітектур є частковим повторенням попередніх. При цьому виникає питання порівняльного аналізу зазначених алгоритмів та обґрунтування показників ефективності. Але, спершу розглянемо типові набори даних (datasets), на яких випробовуються зазначені моделі, і які найчастіше використовуються з методів виявлення. Ці набори даних взяті, що називається, з життя і в них розмічені досить звичайні класи об'єктів, такі як коти, собаки, велосипеди тощо. Варто зазначити про існування специфічних наборів даних, але в цьому огляді вони не зазначаються. Всі ці набори даних, по суті, знаходяться у вільному доступі (наприклад, на сайті компанії Ultralytics). Приклад такого типового набору даних наведено на рис. 3.

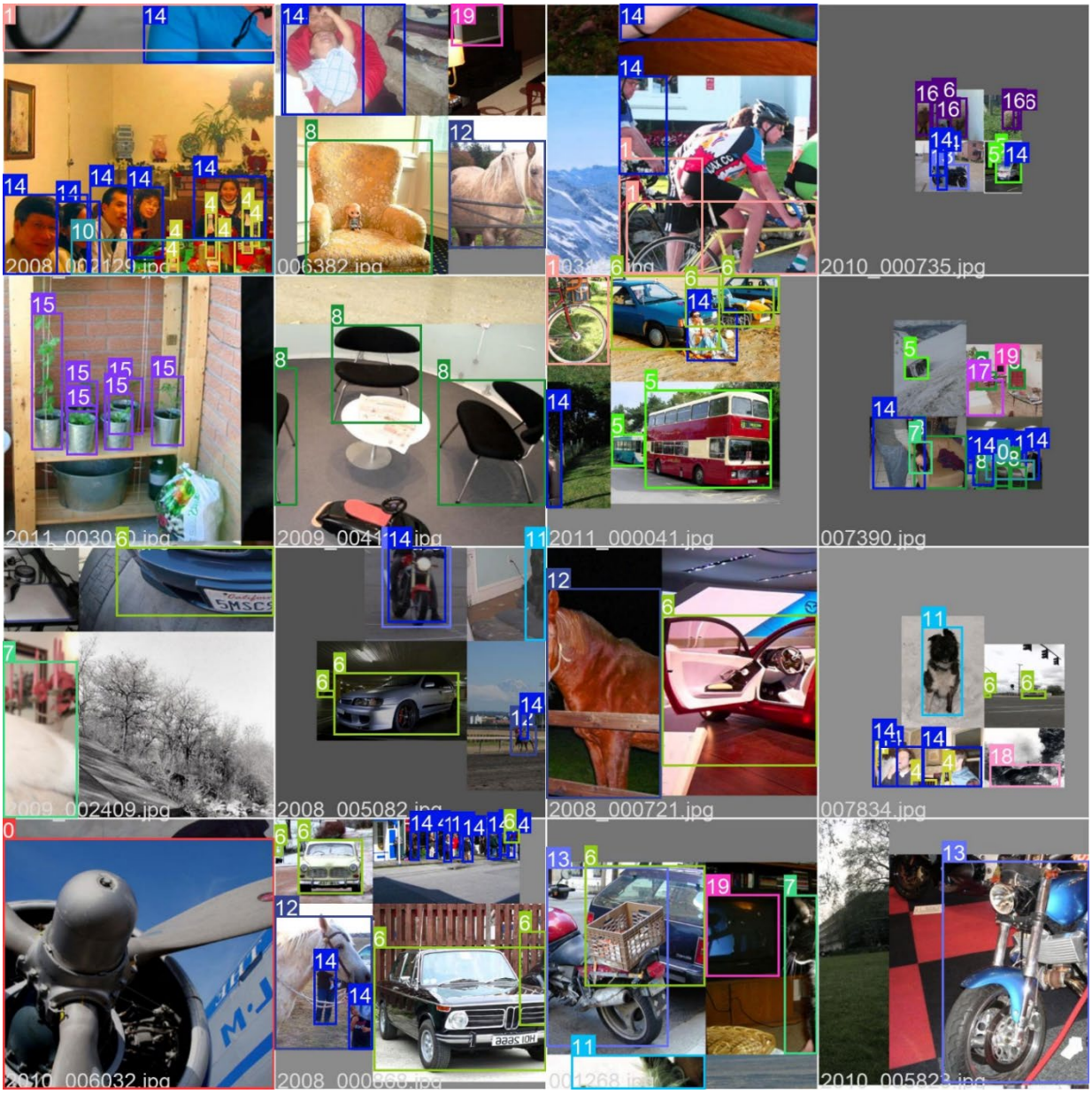

Рис. 3. Приклад зображень з набору даних PASCAL-VOC з анотаціями.
Джерело: *https://docs.ultralytics.com/datasets/detect/voc/*

Зазвичай, під показниками якості розуміються різні метрики (metrics). Як і у випадку з будь-яким алгоритмом, метрики можна розділити на дві групи: метрики якості та метрики швидкодії. Метрики якості характеризують, наскільки якісно працює алгоритм, а метрики швидкодії – наскільки швидко він працює. До метрик якості відносяться: Precision, Recall, Average Precision (AP), mean Average Precision (mAP), Average Recall (AR). До метрик швидкодії відносяться: Inference Time (час виведення), FPS (Frame Per Seconds), Latency (затримка) і Throughput (пропускна здатність). Далі в тексті будемо використовувати тільки англійські терміни.

Почнемо з метрик якості. Мабуть, найпопулярніша метрика – це mean Average Precision (mAP). Вона являє собою число в діапазоні [0;1]. Чим ближче значення цієї метрики до 1, тим краще працює модель. По суті, це значення агрегує Precision Recall. Алгоритм розрахунку цього показника наведено в роботі Padilla [4]. Зазначимо, що в записі mAP@0.5 число 0.5 – це порогове значення метрики IoU. Запис mAP@[0.5 : 0.95] означає, що даний mAP обчислено як середнє mAP для порогових значень IoU від 0.5 до 0.95 з кроком 0.05. На практиці метрика mAP залежить від набору даних, на якому вона обчислюється. Приклади метрик mAP для різних моделей виявлення представлено в табл. 3. На рис. 4 представлені mAP для різних наборів даних.

Також варто відзначити метрику Average Recall (AR). Це також число в діапазоні [0;1], і чим ближче до 1, тим краще працює модель. Вона показує, наскільки модель здатна виявити на зобра-

женні всі об’єкти заданого класу. Ця метрика доповнює AP з точки зору повноти прогнозів. Це важливо, якщо пропуск навіть одного об’єкта може мати тяжкі наслідки. Наприклад, пропуск забороненого предмета на знімку інтроскопа на пункті догляду.

Таблиця 3

Метрики mAP для різних моделей детекторів на наборі даних MS-COCO. Джерело: [1]

Year	Detector	Backbone	mAP@ [0.5 : 0.95]	mAP@0.5
Two-stage				
2015	Fast R-CNN [5]	VGG-16	19.7	35.9
2015	Faster R-CNN [6]	VGG-16	21.9	42.7
2016	R-FCN (multi-scale) [7]	ResNet-101	31.5	53.2
2017	Faster R-CNN + FPN [8]	ResNet-101	36.2	59.1
2017	Mask R-CNN [9]	ResNeXt-101-FPN	39.8	62.3
2018	Cascade R-CNN [10]	ResNet-101	42.8	62.1
One-stage				
2016	SSD512 [11]	VGG-16	28.8	48.5
2017	DSSD513 [12]	ResNet-101	33.2	53.3
2017	FSSD512 [13]	VGG-16	31.8	52.8
2017	RetinaNet-101-800 [14]	ResNet-101-FPN	39.1	59.1
2017	YOLOv2 [15]	Darknet-19	21.6	44.0
2018	RefineDet512 (multi-scale) [16]	ResNet-101	41.8	62.9
2018	CornerNet511 (multi-scale) [17]	Hourglass-104	42.1	57.8
2018	YOLOv3 [18]	Darknet-53	33.0	57.9
2019	EFGRNet (multi-scale) [19]	ResNet-101	43.4	63.8
2019	ASSD513 [20]	ResNet-101	34.5	55.5
2019	CenterNet511-104 (multi-scale) [21]	Hourglass-104	47.0	64.5
2019	ExtremeNet (multi-scale) [22]	Hourglass-104	43.7	60.5
2019	FCOS [23]	ResNeXt-64x4d-101-FPN	44.7	64.1
2019	FSAF (multi-scale) [24]	ResNeXt-101	44.6	65.2
2020	FoveaBox-align [25]	ResNeXt-101	43.9	63.5
2020	YOLOv4 [26]	CSPDarknet-53	43.5	65.7
2023	YOLOv7-E6E [27]	E-ELAN based	56.8	74.4
Transformbased				
2020	Deformable DETR (with TTA) [28]	ResNeXt-101 + DCN	52.3	71.9
2022	DINO (with TTA) [29]	SwinL	63.3	-

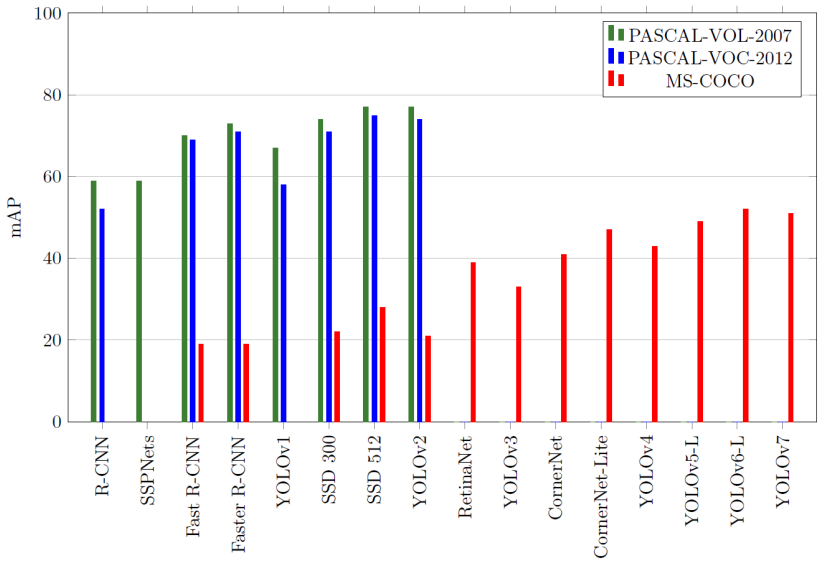

Рис. 4. Приклади метрики mAP на різних наборах даних

До метрик швидкодії відносять наступні:

1. Inference Time – це час, який необхідний моделі для виконання прямого проходу по вхідних даних.

2. Latency – це загальний час затримки при обробці одного пакета. До складу Latency входять: час попередньої обробки даних, Inference Time, час передачі даних (наприклад CPU -> GPU і назад), час постобробки даних (наприклад, NMS).

3. FPS – кількість зображень, які модель здатна обробити за одну секунду. Якщо система обробляє кадри послідовно, то FPS ≈ $(\text{Latency})^{-1}$.

4. Throughput – це загальна кількість оброблених даних за одиницю часу. Метрика може розраховуватися як для послідовно оброблюваних пакетів, так і в паралельному режиму. При вдалому розпаралелюванні буде більше ніж $(\text{Latency})^{-1}$.

Практично у всіх статтях щодо моделей виявлення наводяться графіки залежності AP або mAP від Latency. Приклад таких графіків для різних версій сімейства детекторів YOLO наведено на рис. 5. При порівняльному аналізі моделей детекторів немає еталонних, залізобетонних показників якості вільних від набору даних. Мається на увазі таких, як, наприклад, обчислювальна складність алгоритму. Ця характеристика залежить тільки від структури (блок-схеми) самого алгоритму і не залежить від архітектури процесора або вхідних даних. У статтях про моделі виявлення можна зустріти mAP, обчислені для різних наборів даних. Тобто, в такому випадку просто взяти і порівняти ці значення mAP буде явно неправильним підходом (рис. 4).

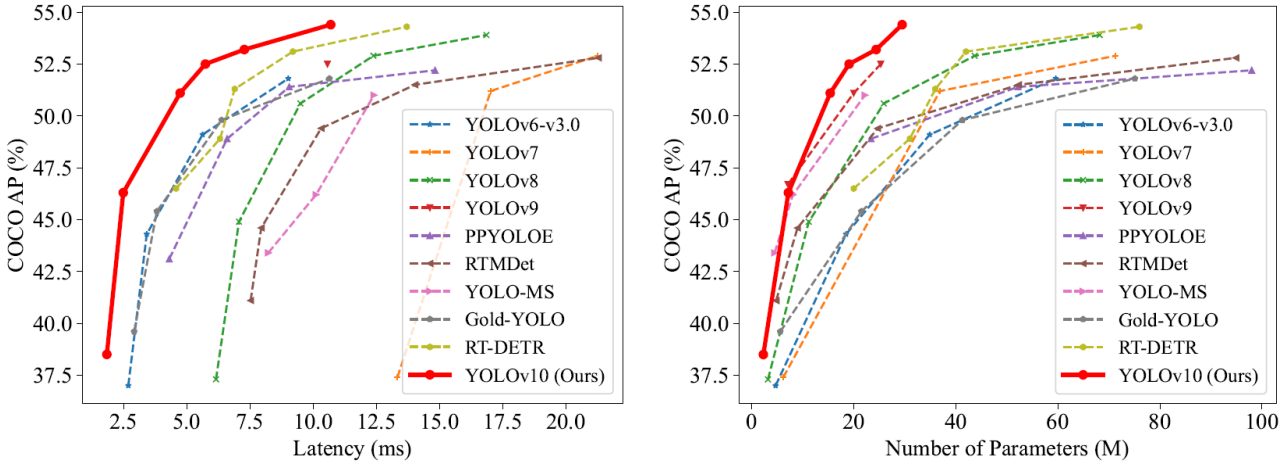

Рис. 5. Залежність AP від Latency (ліворуч) і від кількості параметрів (праворуч) для різних версій серії детекторів YOLO. Джерело: [30]

Щодо значень метрики швидкодії, такі як Latency, то вони безпосередньо залежать від характеристик комп'ютера (архітектура процесора, його частота, використання GPU тощо). Для проведення порівняльного аналізу моделей виявлення об'єктів необхідно мати на своєму комп'ютері спеціальній фреймворк, на вхід якого надходить каталог моделей виявлення і набір даних, а на виході фреймворка будуть обчислені показники якості (рис. 6).

Зрозуміло, що всі моделі повинні бути реалізовані і відповідати єдиному інтерфейсу. Популярними інструментами для експериментів з нейронними мережами, і в першу чергу з моделями виявлення, на сьогоднішній день є мова програмування Python і бібліотеки PyTorch, Keras/Tensorflow, NumPy. Слід зазначити, що бібліотеки PyTorch і Keras/Tensorflow виконують однакові функції, а саме допомагають синтезувати нейронні мережі різних конфігурацій, навчати їх, тестувати і використовувати за призначенням. Бібліотека PyTorch є більш новою. Існує ряд готових фреймворків, на які варто звернути увагу. Це, насамперед, реалізація моделей YOLO від компанії Ultralytics, MMDetection від компанії OpenMMLab, Detectron2 від компанії Meta AI тощо. Програмний код наведених фреймворків викладено на GitHub.

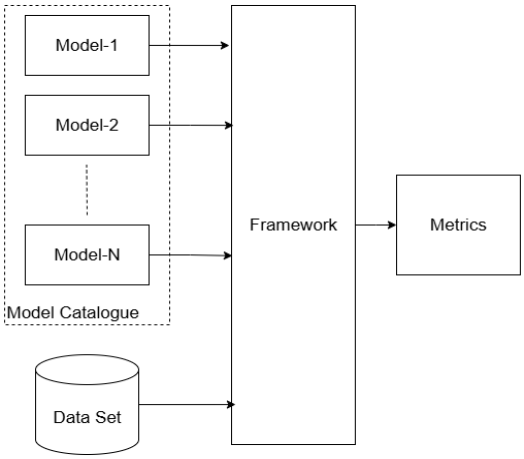

Рис. 6. Структурна схема фреймворка для проведення порівняльного аналізу алгоритмів виявлення

Як видно з рис. 2, всі методи виявлення 2D-об'єктів можна розділити на традиційні методи та методи, які базуються на глибокому навчанні, які в свою чергу поділяються на одноетапні, двоетапні та методи, які базуються на трансформерах. Далі наведено короткий опис для кожної групи.

1. Традиційні методи.

До застосування згорткових нейронних мереж було три основні методи для вирішення задачі виявлення. Детектор Віоли-Джонса базується на застосуванні гаароподібних ознак (набору невеликих чорно-білих фільтрів). Приклад застосування цього методу наведено на рис. 7.

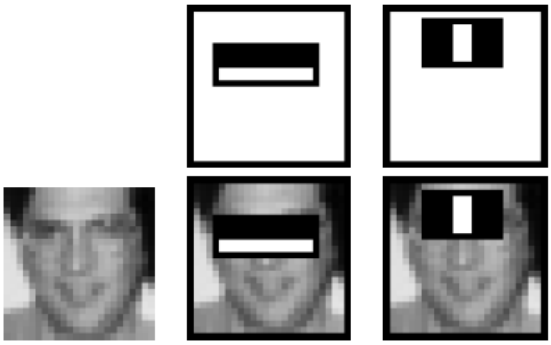

Рис. 7. Демонстрація детектора Віоли-Джонса.
Джерело: [31]

Метод HOG Detector (Histogram of oriented gradients) [32] аналізує орієнтовані градієнти. Тобто зображення розбивається на деякі невеликі регіони, для яких обчислюються градієнти напрямків, потім будується гістограма. В результаті знаходяться вектори ознак, які і подаються на вхід модулю на основі методу опорних векторів (SVM, Support Vector Machine) (див. рис. 8).

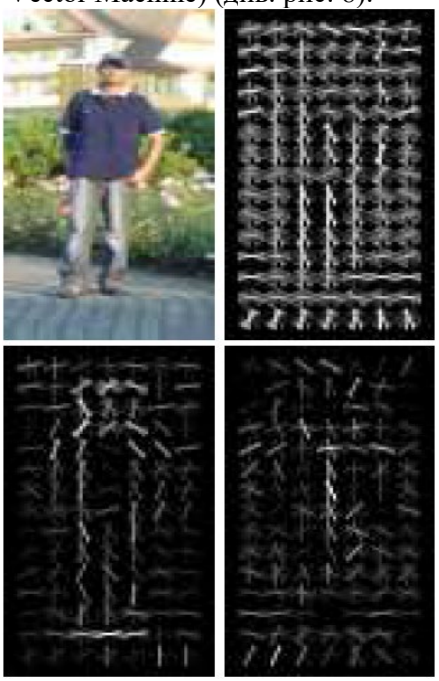

Рис. 8. Зображення та R-HOG градієнти.
Джерело: [32]

HOG ознаки вплинули на модель DPM (Deformable Part Model) [33]. Вона, по суті, також обчислює вектори ознак, але робить це набагато вдосконаленим способом.

2. Двоетапні детектори.

У 2012 році вийшла модель класифікації об'єктів AlexNet [34], яка ознаменувала кардинально новий підхід до вирішення задачі виявлення об'єктів із застосуванням згорткових нейронних мереж. Вона посіла перше місце на змаганні ILSVRC (ImageNet Large Scale Visual Recognition Challenge), де потрібно було класифікувати мільйон зображень, розподілених по тисячі категорій. Для моделі AlexNet помилка Top-5 склала 15,3 %, що майже в 2 рази менше в порівнянні з попередніми моделями, які не використовували згорткові нейронні мережі. Таким чином, цей результат показав, що глибокі згорткові нейронні мережі здатні перевершити традиційні алгоритми комп'ютерного зору, які ґрунтуються на ручному виділенні ознак. У цій мережі було близько 60 мільйонів параметрів, і вона була розгорнута на двох GPU процесорах. Потрібно було від п'яти до шести днів, щоб навчити модель AlexNet на наборі даних ImageNet розміром 1,2 мільйона зображень. У підсумку це зайняло 90 епох.

Першим двоетапним детектором став R-CNN [35, 36, 37]. У загальному вигляді робота двоетапного детектора складається, як випливає з назви, з двох етапів: генерація RoI (Region of Interest, регіонів інтересу) і класифікації об'єктів. На першому етапі модель сканує зображення з метою знайти регіони-кандидати RoI, в яких, найімовірніше, є об'єкт. Потім кожен запропонований регіон передається в ділянку класифікації та в ділянку регресії. На виході ділянки класифікації отримуємо клас об'єкта, а на виході ділянки регресії – уточнені координати обмежувальної рамки. При такому підході з аналізом заздалегідь відібраних регіонів досягається висока точність локалізації. Як видно з рис. 2, розвиток двоетапних детекторів зупинився в 2018 році з детектором Cascade R-CNN [10].

3. Одноетапні детектори.

Яскравим прикладом одноетапних детекторів є лінійка YOLO (You Only Look Once) [38, 39]. Одноетапний детектор відразу обробляє все зображення за допомогою однієї згорткової нейронної мережі, за один прохід, що є запорукою їх високої швидкості [40, 41]. Спочатку моделі відбувається вилучення ознак. Створюються багато представлень вхідного зображення в різних масштабах. Далі ці ознаки подаються на спеціальну ділянку виявлення, що відповідає за передбачення набору обмежувальних рамок, показник достовірності (confidence score) для кожної рамки і

ймовірність належності кожного об'єкта до певного класу. При навчанні варто відзначити використання спеціальної функції втрат, що поєднує в собі втрати при виявленні (точність визначення меж) і втрати при класифікації (точність передбачення класу). На виході, як правило, багато перекриваючих і дублюючих рамок з об'єктами. Для відкидання таких дублікатів проводиться ще постобробка. Наприклад, за допомогою алгоритму NMS (Non-Maximum Suppression).

4. Детектори, що базуються на трансформерах.

У 2017 році у роботі Ashish Vaswani et al [42] автори представили модель трансформера для обробки природної мови. Особливістю цієї моделі є здатність вловлювати далекі залежності в послідовних даних. Успіх цієї моделі надихнув дослідників на вивчення можливості застосувати цей метод у задачах комп'ютерного зору, зокрема для виявлення об'єктів. На відміну від згорткових мереж, трансформери використовують механізм самоуваги, який обчислює відношення між усіма елементами послідовності. Детектори цієї групи розглядають зображення як послідовність плям і використовують механізм самоуваги для моделювання зв'язків між ними. Це дозволяє ефективно моделювати глобальний контекст, що призводить до досить точного виявлення об'єктів, особливо в загромадженому середовищі [43, 44].

До переваг цієї групи моделей слід віднести ту особливість, що ці моделі добре піддаються розпаралелюванню. Це дозволяє навчати набагато більші моделі на величезних наборах даних.

Згорткові мережі ефективно виявляють локальні патери на зображенні, а трансформери – глобальні взаємозв'язки, хоча часто вимагають більше даних і обчислювальних ресурсів.

Гібридні моделі, що поєднують в собі згорткові нейронні мережі та шари трансформерів, прагнуть отримати найкраще з двох зазначених класів моделей.

Висновки

На сьогоднішній день існує великий каталог моделей для виявлення 2D-об'єктів на зображеннях. У той же час всі архітектури цих моделей досить схожі між собою. Серед існуючих моделей явними лідерами є одноетапні моделі (такі як YOLO) і група детекторів, які базуються на трансформерах. При сучасному розвитку обчислювальної техніки моделі обох груп можуть працювати в режимі реального часу.

Основною метрикою якості є mAP, однак вона залежить від набору даних, на якому була обчислена. Тому при вирішенні конкретної задачі для проведення порівняльного аналізу моделей виявлення все ж доведеться, керуючись напрацюваннями вчених і експертів в конкретній предметній області, вибрати кілька відповідних моделей нейронних мереж, мати їх програмну реалізацію, і проганяти їх через один і той самий набір даних.

ЛІТЕРАТУРА

[1] Malagoli E. and Di Persio L. 2D Object Detection: A Survey. *Mathematics*. 2025. Mar. Vol. 13, no. 6. P. 893. ISSN 2227-7390. DOI: 10.3390/math13060893.

[2] Tsirtsakis P., Zacharis G., Maraslidis G. S., and Fragulis G. F. Deep learning for object recognition: A comprehensive review of models and algorithms. *International Journal of Cognitive Computing in Engineering*. 2025. Dec. Vol. 6. P. 298–312. ISSN 2666-3074. DOI: 10.1016/j.ijcce.2025.01.004.

[3] Zou Z., Chen K., Shi Z., Guo Y., and Ye J. Object Detection in 20 Years: A Survey. Proceedings of the IEEE. 2023. Mar. Vol. 111, no. 3. P. 257–276. ISSN 1558-2256. DOI: 10.1109/jproc.2023.3238524.

[4] Padilla R., Netto S. L., and da Silva E. A. B. A Survey on Performance Metrics for Object-Detection Algorithms. 2020 International Conference on Systems, Signals and Image Processing (IWSSIP). IEEE. 2020. July. P. 237–242. DOI: 10.1109/iwssip48289.2020.9145130.

[5] Girshick R. Fast R-CNN. 2015 IEEE International Conference on Computer Vision (ICCV). IEEE. 2015. Dec. P. 1440–1448. DOI: 10.1109/iccv.2015.169.

[6] Ren S., He K., Girshick R., and Sun J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2017. June. Vol. 39, no. 6. P. 1137–1149. ISSN 2160-9292. DOI: 10.1109/tpami.2016.2577031.

[7] Dai J., Li Y., He K., and Sun J. R-FCN: Object detection via region-based fully convolutional networks. 2016. P. 379–387. DOI: 10.48550/arXiv.1605.06409.

[8] Lin T.-Y., Dollar P., Girshick R., He K., Hariharan B., and Belongie S. Feature Pyramid Networks for Object Detection. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE. 2017. July. DOI: 10.1109/cvpr.2017.106.

[9] He K., Gkioxari G., Dollar P., and Girshick R. Mask R-CNN. 2017 IEEE International Conference on Computer Vision (ICCV). IEEE. 2017. Oct. DOI: 10.1109/iccv.2017.322.

[10] Cai Z. and Vasconcelos N. Cascade R-CNN: High Quality Object Detection and Instance Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2021. May. Vol. 43, no. 5. P. 1483–1498. ISSN 1939-3539. DOI: 10.1109/tpami. 2019.2956516.

[11] Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu C.-Y., and Berg A. C. SSD: Single Shot-MultiBox Detector. Computer Vision – ECCV 2016. Springer International Publishing, 2016. P. 21–37. (ISSN 1611-3349). ISBN 9783319464480. DOI: 10.1007/978-3-319-46448-0_2.

[12] Fu C.-Y., Liu W., Ranga A., Tyagi A., and Berg A. C. DSSD : Deconvolutional Single Shot Detector. 2017. Jan. DOI: 10.48550/ARXIV.1701.06659.
[13] Li Z., Yang L., and Zhou F. FSSD: Feature Fusion Single Shot Multibox Detector. 2017. Dec. DOI: 10.48550/ARXIV.1712.00960.
[14] Lin T.-Y., Goyal P., Girshick R., He K., and Dollar P. Focal Loss for Dense Object Detection. 2017 IEEE International Conference on Computer Vision (ICCV). IEEE. 2017. Oct. DOI: 10.1109/iccv.2017.324.
[15] Redmon J. and Farhadi A. YOLO9000: Better, Faster, Stronger. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE. 2017. July. P. 6517–6525. DOI: 10.1109/cvpr.2017.690.
[16] Zhang S., Wen L., Bian X., Lei Z., and Li S. Z. Single-Shot Refinement Neural Network for Object Detection. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE. 2018. June. DOI: 10.1109/cvpr.2018.00442.
[17] Law H. and Deng J. CornerNet: Detecting Objects as Paired Keypoints. Computer Vision – ECCV 2018. Springer International Publishing, 2018. P. 765–781. (ISSN 1611-3349). ISBN 9783030012649. DOI: 10.1007/978-3-030-01264-9_45.
[18] Redmon J. and Farhadi A. YOLOv3: An Incremental Improvement. 2018. Apr. DOI: 10.48550/ARXIV.1804.02767.
[19] Nie J., Anwer R. M., Cholakkal H., Khan F. S., Pang Y., and Shao L. Enriched Feature Guided Refinement Network for Object Detection. 2019 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE. 2019. Oct. DOI: 10.1109/iccv.2019.00963.
[20] Yi J., Wu P., and Metaxas D. N. ASSD: Attentive single shot multibox detector. Computer Vision and Image Understanding. 2019. Dec. Vol. 189. P. 102827. ISSN 1077-3142. DOI: 10.1016/j.cviu.2019.102827.
[21] Duan K., Bai S., Xie L., Qi H., Huang Q., and Tian Q. CenterNet: Keypoint Triplets for Object Detection. 2019 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE. 2019. Oct. DOI: 10.1109/iccv.2019.00667.
[22] Zhou X., Zhuo J., and Krahenbuhl P. Bottom-Up Object Detection by Grouping Extreme and Center Points. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE. 2019. June. DOI: 10.1109/cvpr.2019.00094.
[23] Tian Z., Shen C., Chen H., and He T. FCOS: Fully Convolutional One-Stage Object Detection. 2019 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE. 2019. Oct. DOI: 10.1109/iccv.2019.00972.
[24] Zhu C., He Y., and Savvides M. Feature Selective Anchor-Free Module for Single-Shot Object Detection. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE. 2019. June. DOI: 10.1109/cvpr.2019.00093.
[25] Kong T., Sun F., Liu H., Jiang Y., Li L., and Shi J. FoveaBox: Beyound Anchor-Based Object Detection. IEEE Transactions on Image Processing. 2020. Vol. 29. P. 7389–7398. ISSN 1941-0042. DOI: 10.1109/tip.2020.3002345.
[26] Bochkovskiy A., Wang C.-Y., and Liao H.-Y. M. YOLOv4: Optimal Speed and Accuracy of Object Detection. 2020. Apr. DOI: 10.48550/ARXIV.2004.10934.
[27] Wang C.-Y., Bochkovskiy A., and Liao H.-Y. M. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE. 2023. June. P. 7464–7475. DOI: 10.1109/cvpr52729.2023.00721.
[28] Zhu X., Su W., Lu L., Li B., Wang X., and Dai J. Deformable DETR: Deformable Transformers for End-to-End Object Detection. 2020. Oct. DOI: 10.48550/ARXIV.2010.04159.
[29] Zhang H., Li F., Liu S., Zhang L., Su H., Zhu J., Ni L. M., and Shum H.-Y. DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection. 2022. Mar. DOI: 10.48550/ARXIV.2203.03605.
[30] Wang A., Chen H., Liu L., Chen K., Lin Z., Han J., and Ding G. YOLOv10: Real-Time End-to-End Object Detection. 2024. Vol. 37.
[31] Viola P. and Jones M. Rapid object detection using a boosted cascade of simple features. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001. IEEE Comput. Soc. 2001. P. I–511–I–518. (CVPR-01; Vol. 1). DOI: 10.1109/cvpr.2001.990517.
[32] Dalal N. and Triggs B. Histograms of Oriented Gradients for Human Detection. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). IEEE. 2005. Vol. 1. P. 886–893. DOI: 10.1109/cvpr.2005.177.
[33] Felzenszwalb P., McAllester D., and Ramanan D. A discriminatively trained, multiscale, deformable part model. 2008 IEEE Conference on Computer Vision and Pattern Recognition. IEEE. 2008. June. P. 1–8. DOI: 10.1109/cvpr.2008.4587597.
[34] Krizhevsky A., Sutskever I., and Hinton G. E. ImageNet classification with deep convolutional neural networks. Communications of the ACM. 2017. May. Vol. 60, no. 6. P. 84–90. ISSN 1557-7317. DOI: 10.1145/3065386.
[35] Girshick R., Donahue J., Darrell T., and Malik J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. 2014 IEEE Conference on Computer Vision and Pattern Recognition. IEEE. 2014. June. P. 580–587. DOI: 10.1109/cvpr.2014.81.
[36] He K., Zhang X., Ren S., and Sun J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2015. Sep. Vol. 37, no. 9. P. 1904–1916. ISSN 2160-9292. DOI: 10.1109/tpami.2015.2389824.
[37] Sermanet P., Eigen D., Zhang X., Mathieu M., Fergus R., and LeCun Y. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. 2013. Dec. DOI: 10.48550/ARXIV.1312.6229.

[38] Li C., Li L., Jiang H., Weng K., Geng Y., Li L., Ke Z., Li Q., Cheng M., Nie W., Li Y., Zhang B., Liang Y., Zhou L., Xu X., Chu X., Wei X., and Wei X. YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. 2022. Sep. DOI: 10.48550/ARXIV.2209.02976.
[39] Redmon J., Divvala S., Girshick R., and Farhadi A. You Only Look Once: Unified, Real-Time Object Detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE. 2016. June. P. 779–788. DOI: 10.1109/cvpr.2016.91.
[40] Jocher G. YOLOv5 release v7.0. https://github.com/ultralytics/yolov5/tree/v7.0. 2022. URL:https://github.com/ultralytics/yolov5/tree/v7.0, (Access date: 2025-08-10).
[41] Jocher G. Ultralytics YOLO. 2025. URL: https://github.com/ultralytics/ultralytics/tree/main, (Access date: 2025-08-10).
[42] Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser L., and Polosukhin I. Attention is all you need. 2017. Vol. 2017, December. P. 5999–6009. DOI: 10.48550/arXiv.1706.03762.
[43] Carion N., Massa F., Synnaeve G., Usunier N., Kirillov A., and Zagoruyko S. End-to-End Object Detection with Transformers. Computer Vision – ECCV 2020. Springer International Publishing, 2020. P. 213–229. (ISSN 1611-3349). ISBN 9783030584528. DOI: 10.1007/978-3-030-58452-8_13.
[44] Beal J., Kim E., Tzeng E., Park D. H., Zhai A., and Kislyuk D. Toward Transformer-Based Object Detection. 2020. Dec. DOI: 10.48550/ARXIV.2012.09958.

Бєлов С. Ю., Заліський М. Ю.

ОГЛЯД МОДЕЛЕЙ НЕЙРОННИХ МЕРЕЖ ДЛЯ ЗАДАЧ ВИЯВЛЕННЯ 2D-ОБ'ЄКТІВ

У даному оглядовій статті представлено каталог моделей нейронних мереж глибокого навчання для задач виявлення 2D-об'єктів на зображеннях. Ці моделі розділені на три групи: одноетапні, двоетапні та моделі, що базуються на трансформерах. Також згадані традиційні методи, розроблені до застосування глибокого навчання в задачах класифікації та виявлення. Розглядається постановка наступних задач комп'ютерного зору: класифікація зображення, локалізація зображення, виявлення об'єкта і сегментація зображення. Розглянуто також деякі фундаментальні компоненти виявлення 2D-об'єктів, такі як загальноприйняті та широко використовувані набори даних (PASCAL-VOC, ILSVRC, MS-COCO, OpenImages, Objects365) та наведено перелік показників якості з їх коротким описом. Останні розділені на метрики якості та метрики швидкодії. У метриках якості наведено Precision, Recall, Average Precision (AP), mean Average Precision (mAP), Average Recall (AR). Перелік метрик швидкодії включає Inference Time (час виведення), FPS (Frame Per Seconds), Latency (затримка) і Throughput (пропускна здатність). Наведено mAP для різних моделей і наборів даних, а також приклади залежності mAP від швидкодії та кількості параметрів для різних версій лінійки моделей YOLO. Висловлено спосіб проведення порівняльного аналізу різних моделей виявлення із зазначенням відповідних програмних інструментів. Для кожної групи моделей надано стислий опис особливостей її функціонування. Стаття також містить короткий історичний нарис застосування моделі AlexNet у 2012 році на змаганні ILSVRC (ImageNet Large Scale Visual Recognition Challenge). Даний огляд безсумнівно представляє інтерес для здобувачів та науковців, які не знайомі з тематикою виявлення 2D-об'єктів за допомогою нейронних мереж і методів машинного навчання, але при цьому бажають швидко отримати ознайомлювальну інформацію.

Ключові слова: виявлення об'єктів; нейронна мережа; згорткова нейронна мережа; машинне навчання; глибоке навчання; набір даних.

Bielov S., Zaliskyi M.

REVIEW OF NEURAL NETWORK MODELS FOR 2D OBJECT DETECTION

This review article presents a catalogue of deep learning neural network models for detecting 2D objects in images. These models are divided into three groups: single-stage, two-stage, and transformer-based models. Traditional methods developed prior to the application of deep learning in classification and detection tasks are also mentioned. The following computer vision tasks are considered: image classification, image localization, object detection, and image segmentation. Some fundamental components of 2D object detection are considered, such as commonly used data sets (PASCAL-VOC, ILSVRC, MS-COCO, OpenImages, Objects365), and a list of quality indicators with a brief description. The last ones are split into quality metrics and performance metrics. The list of quality metrics includes Precision, Recall, Average Precision (AP), Mean Average Precision (mAP), and Average Recall (AR). The list of performance metrics includes Inference Time, FPS (Frames Per Second), Latency, and Throughput. The mAP values for different models and datasets are presented. Also examples of mAP dependence on speed and number of parameters for different versions of the YOLO model family are presented. A method for conducting a comparative analysis of different detection models is presented, with an indication of suitable software tools. A brief description of the idea is provided for each group of models. The article contains a brief historical overview of the triumph of the AlexNet model in 2012 at the ILSVRC (ImageNet Large Scale Visual Recognition Challenge) competition. This overview is undoubtedly of interest to those who are unfamiliar with the topic of detecting 2D objects using neural networks and machine learning methods, but who want to quickly get up to speed.

Keywords: object detection; neural network; convolutional neural network; machine learning; deep learning; dataset.

Стаття надійшла до редакції 29.08.2025 р.
Прийнято до друку 10.12.2025 р.