

ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ  
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ  
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

**КВАЛІФІКАЦІЙНА РОБОТА**

на тему: «Методика автоматизованого розпізнавання на класифікації стрілецької зброї»

на здобуття освітнього ступеня магістра  
зі спеціальності 121 Інженерія програмного забезпечення  
освітньо-професійної програми «Інженерія програмного забезпечення»

*Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело*

\_\_\_\_\_ (підпис)

Владислав ГРЕБЕЩЕНКО

Виконав: здобувач вищої освіти групи ПДМ-62  
Владислав ГРЕБЕЩЕНКО

Керівник: Володимир САДОВЕНКО  
канд. фіз.-мат. наук, доц.

Рецензент: \_\_\_\_\_  
науковий ступінь, *Ім'я, ПРІЗВИЩЕ*  
вчене звання

Київ 2026

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**  
**Навчально-науковий інститут інформаційних технологій**

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти Магістр

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма «Інженерія програмного забезпечення»

**ЗАТВЕРДЖУЮ**

Завідувач кафедри

Інженерії програмного забезпечення

\_\_\_\_\_ Ірина ЗАМРІЙ

« \_\_\_\_ » \_\_\_\_\_ 2025 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Гребещенку Владиславу Едуардовичу

1. Тема кваліфікаційної роботи: «Методика автоматизованого розпізнавання на класифікації стрілецької зброї»

керівник кваліфікаційної роботи Володимир САДОВЕНКО, канд. фіз.-мат. наук, доц.

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «30» жовтня 2025 р. № 467.

2. Строк подання кваліфікаційної роботи «19» грудня 2025 р.

3. Вихідні дані до кваліфікаційної роботи: , методи попередньої обробки зображень, методика класифікації та розпізнавання об'єктів.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Дослідження інформації щодо використання машинного навчання в Збройних Силах.

2. Аналіз методик автоматизованого розпізнавання, опис алгоритму роботи, їх порівняння.

3. Розробка та тестування гібридної методики.
5. Перелік ілюстративного матеріалу: презентація
1. Мета, об'єкт та предмет дослідження.
  2. Порівняння аналогів.
  3. Алгоритм автоматизованого розпізнавання стрілецької зброї.
  4. Архітектура MobileNet
  5. Структура гібридної методики.
  6. Практичний результат.
  7. Порівняння результатів.
  8. Висновки.
  9. Публікації та апробації роботи.
6. Дата видачі завдання «31» жовтня 2025 р.

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз наявної науково-технічної літератури	31.10-12.11.25	
2	Вивчення матеріалів для аналізу розвитку технологій розпізнавання об'єктів	13.11-17.11.25	
3	Дослідження методів попередньої обробки зображень	18.11-22.11.25	
4	Аналіз особливостей існуючих методик автоматизованого розпізнавання	23.11-27.11.25	
5	Дослідження технологій автоматизованого розпізнавання	28.11-07.12.25	
6	Розробка системи автоматизованого розпізнавання	08.12-15.12.25	
7	Оформлення роботи: вступ, висновки, реферат	16.12-17.12.25	
8	Розробка демонстраційних матеріалів	18.12-19.12.25	

Здобувач вищої освіти

\_\_\_\_\_

(підпис)

Владислав ГРЕБЕЩЕНКО

Керівник

кваліфікаційної роботи

\_\_\_\_\_

(підпис)

Володимир САДОВЕНКО





## РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 74 стор., 1 табл., 26 рис., 26 джерел.

*Мета роботи* – підвищення точності автоматизованого розпізнавання на класифікації стрілецької зброї.

*Об'єкт дослідження* – процес автоматизованого розпізнавання на класифікації стрілецької зброї.

*Предмет дослідження* – методика автоматизованого розпізнавання на класифікації стрілецької зброї.

Наукова інновація полягає у розробці системи автоматизованого розпізнавання стрілецької зброї на класифікації на основі гібридного поєднання машинного навчання з детектором об'єктів. Ця система надає можливість виявити та класифікувати засіб ураження за оптимальний час з високим показником точності. Розглянуто та проаналізовано існуючі методики автоматизованого розпізнавання, визначені їх переваги та недоліки. Розроблена методика з урахуванням можливостей постійного машинного навчання може широко застосовуватись у військовій сфері.

**КЛЮЧОВІ СЛОВА:** КОМП'ЮТЕРНИЙ ЗІР, ГЛИБОКЕ НАВЧАННЯ, МАШИННЕ НАВЧАННЯ, АВТОМАТИЗОВАНЕ РОЗПІЗНАВАННЯ СТРЕЛЕЦЬКОЇ ЗБРОЇ, ДЕТЕКТОР.

## **ABSTRACT**

Text part of the master's qualification work: 74 pages, 26 pictures, 1 table, 26 sources.

The purpose of the work – increase the accuracy of automated recognition on the classification of small arms.

Object of research – process of automated recognition on the classification of small arms.

Subject of research – methodology of automated recognition on the classification of small arms.

The scientific innovation consists in the development of a system of automated recognition of small arms on the classification based on a hybrid combination of machine learning with an object detector. This system provides the opportunity to detect and classify a means of destruction in the optimal time with a high accuracy rate. The existing methods of automated recognition are considered and analyzed, their advantages and disadvantages are determined. The developed methodology, taking into account the capabilities of continuous machine learning, can be widely used in the military sphere.

**KEYWORDS: COMPUTER VISION, DEEP LEARNING, MACHINE LEARNING, AUTOMATED RECOGNITION OF SMALL ARMS, DETECTOR.**

## ЗМІСТ

ВСТУП.....	9
1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ .....	10
1.1 Автоматизоване розпізнавання.....	10
1.2 Застосування автоматизованого розпізнавання у Збройних Силах.....	16
1.3 Методики автоматизованого розпізнавання .....	19
2. АНАЛІЗ МЕТОДИК АВТОМАТИЗОВАНОГО РОЗПІЗНАВАННЯ СТРІЛЕЦЬКОЇ ЗБРОЇ НА КЛАСИФІКАЦІЇ.....	31
2.1 Аналіз існуючих методів автоматизованого розпізнавання.....	31
2.2 Аналіз переваг та недоліків методик автоматизованого розпізнавання .....	43
3. РОЗРОБКА ГІБРИДНОЇ МЕТОДИКИ АВТОМАТИЗОВАНОГО РОЗПІЗНАВАННЯ СТРІЛЕЦЬКОЇ ЗБРОЇ НА КЛАСИФІКАЦІЇ.....	47
3.1 Засоби розробки, бібліотеки та мови програмування, вимоги до них .....	47
3.2 Опис розробки методики.....	56
3.3 Програмна реалізація та результати порівнянь .....	59
ВИСНОВКИ.....	66
ПЕРЕЛІК ПОСИЛАНЬ .....	68
ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ .....	70

## ВСТУП

Автоматизоване розпізнавання на класифікації стрілецької зброї набуває все більшої актуальності в умовах зростання потреби у безпеці, оперативному моніторингу та підвищенні ефективності систем відеоспостереження. Сучасні технології комп'ютерного зору дають змогу значно покращити здатність технічних систем своєчасно фіксувати наявність небезпечних об'єктів, у тому числі зброї, що є критичним для запобігання правопорушенням, терористичним загрозам і несанкціонованому використанню озброєння.

У традиційних системах безпеки людський фактор залишається слабкою ланкою: оператори можуть пропустити небезпечний об'єкт через втому, високе інформаційне навантаження або погані умови спостереження. Автоматизація процесу розпізнавання зброї на основі нейронних мереж дає змогу підвищити точність, швидкість і надійність виявлення, а також забезпечити безперервний моніторинг у реальному часі.

Крім того, сучасні методи автоматизованого розпізнавання ґрунтуються переважно на використанні глибокого навчання та згорткових нейронних мереж, які здатні самостійно виділяти інформативні ознаки об'єктів і забезпечувати високу точність класифікації навіть у складних умовах освітлення, зашумлення чи часткового перекриття об'єктів та здатні працювати навіть на малопотужних пристроях — мобільних системах, дронах чи камерах відеоспостереження.

Архітектури типу MobileNet у поєднанні з одностадійними детекторами, зокрема Single Shot Detector, відкривають можливості для створення легких і швидкодіючих систем, придатних для роботи на обмежених обчислювальних платформах.

Таким чином, тема дослідження є актуальною як з наукової, так і з практичної точки зору, оскільки поєднує передові методи комп'ютерного зору з реальними потребами безпеки, дозволяючи створювати інтелектуальні системи раннього виявлення та класифікації стрілецької зброї.

# 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

## 1.1 Автоматизоване розпізнавання

Автоматизоване розпізнавання — це галузь штучного інтелекту та комп'ютерних наук, яка займається здатністю технічних систем самостійно ідентифікувати об'єкти, явища або закономірності на основі цифрових сигналів, зображень, звуку чи інших типів даних. Своєю суттю це процес, у якому алгоритми аналізують інформацію та роблять висновки без прямої участі людини, імітуючи здатність людини бачити, чути, сприймати структури та розуміти їх зміст.

Автоматизоване розпізнавання виникло як логічне продовження розвитку кібернетики, обчислювальної техніки та математичної статистики. Перші ідеї відслідковування та класифікації даних з'явилися ще в середині ХХ століття під час становлення цифрових комп'ютерів. Процеси, які раніше вимагали ручного аналізу, поступово передавалися машинам, що дозволило суттєво підвищити швидкість обробки інформації. Справжній прорив, однак, відбувся лише з появою машинного навчання та нейронних мереж, коли системи навчилися самостійно вдосконалюватись через досвід.

Основна мета автоматизованого розпізнавання — забезпечити можливість аналізу великих обсягів даних там, де людські ресурси обмежені, а точність і швидкість є критичними. Ці системи дозволяють вирішувати завдання, які потребують негайних рішень, працюють у місцях, недоступних або небезпечних для людини, та зменшують кількість людських помилок.

Сьогодні вони застосовуються в найрізноманітніших сферах: від медицини, де алгоритми розпізнають патології на знімках, до автомобільної промисловості, де автономні машини визначають об'єкти на дорозі. В промисловості автоматизовані системи аналізують якість продукції, у фінансах виявляють підозрілі транзакції, у сфері безпеки — ідентифікують обличчя та поведінкові патерни.

Методологічно автоматизоване розпізнавання ґрунтується на статистиці, оптимізації, теорії ймовірності та обчислювальних алгоритмах. Класична обробка зображень передбачала ручне створення ознак — людина-програміст визначала, що саме потрібно «бачити» системі. Але сучасні нейронні мережі, особливо глибоке навчання, дозволяють моделі самостійно виявляти важливі структури у даних. Завдяки цьому методи розпізнавання стали універсальними та здатними працювати з найскладнішими сигналами, де людське око не здатне помітити закономірностей.

Поява великих дата-сетів, швидких графічних процесорів та хмарних обчислень зробила автоматизоване розпізнавання доступним для широкого кола дослідників і компаній. Воно перестало бути сферою лише великих лабораторій — сьогодні ним займаються університети, технологічні корпорації, стартапи та незалежні розробники. Уряди багатьох країн активно впроваджують подібні системи у сферу безпеки, транспортну інфраструктуру, державний сектор.

Сучасне автоматизоване розпізнавання — це синтез математичних методів, програмної інженерії та нейронаукових концепцій, які разом створюють системи, здатні аналізувати світ у режимі реального часу. Воно спрямоване на розширення можливостей людини, підвищення ефективності процесів і створення інструментів, що працюють швидше, точніше та надійніше за традиційні методи. У майбутньому такі системи ставатимуть ще більш автономними, інтелектуальними та універсальними, інтегруючись у всі сфери взаємодії людини з технологіями.

Перші ідеї автоматичного розпізнавання з'явилися паралельно з розвитком кібернетики у 1950-х роках. У 1956 році відбулась знаменита Дартмутська конференція, яку вважають народженням штучного інтелекту як наукової дисципліни. На цьому етапі вчені намагалися навчати машини впізнавати прості символи, літери, цифри. Перші моделі мали обмежені обчислювальні можливості та використовували примітивні алгоритми — лінійні класифікатори, прості шаблонні методи й вручну розроблені правила.

У 1960–1970-х роках з'явилися перші серйозні підходи до зорового аналізу. Важливим внеском стали роботи Девіда Марра, який у 1970-х роках сформував

фундаментальні принципи комп'ютерного зору, запропонувавши багаторівневу модель аналізу зображення — від сировинної інформації до абстрактного 3D-представлення. У цей же час активно розроблялися алгоритми виділення контурів, фільтри (наприклад, фільтр Канні, 1986 рік), методи сегментації — все це створювало основу для подальшого розпізнавання.

Поява нейронних мереж стала першим великим етапом у напрямі автоматизованого розпізнавання. У 1957–1958 роках Розенблатт представив перцептрон, який формально є першою моделлю, що навчалася розпізнавати вхідні дані. Проте через обмеженість обчислювальних ресурсів розвиток сповільнився, і лише у 1986 році вихід роботи Румельхарта, Гінтона та Вільямса, які описали алгоритм зворотного поширення помилки, повернув нейронним мережам популярність. Це дозволило створювати багат шарові моделі, здатні вчитися складніших закономірностей.

У 1990-х роках з'явилися нейронні мережі зі згортками (convolutional neural networks). Найвизначніший внесок зробив Ян ЛеКун, який у 1989–1998 роках розробив LeNet-5 — одну з перших глибоких CNN, що застосовувалась для автоматичного розпізнавання рукописних цифр у банківській системі США. Це був перший великий прорив у практичному застосуванні автоматизованого розпізнавання.

Справжня революція відбулася у 2012 році, коли команда Гінтона, Сутьскевера та Крізевського створила AlexNet і перемогла на конкурсі ImageNet із величезним відривом. Цей рік вважають початком ери глибокого навчання, у якій автоматизоване розпізнавання стало домінуючою технологією в індустрії.

Після 2012 року розвиток пришвидшився. З'явилися нові архітектури: VGG у 2014 році, GoogLeNet у 2014 році, ResNet у 2015 році, DenseNet у 2016 році, та EfficientNet у 2019 році. Кожна з них покращувала якість автоматичного розпізнавання завдяки глибшим мережам, оптимізованій структурі або новим механізмам навчання. Глибокі CNN значно підвищили точність моделей у задачах класифікації, детекції, сегментації та інших видів комп'ютерного зору.

Автоматизоване розпізнавання перестало бути суто академічною дисципліною. Воно стало критично важливим компонентом у багатьох галузях. У медицині застосовується для аналізу медичних зображень, починаючи від рентгенів до МРТ; у транспорті — для автономного керування автомобілями; у безпеці — для відеоспостереження, ідентифікації об'єктів та осіб; у виробництві — для контролю якості продукції; у торгівлі — для систем рекомендацій і розпізнавання товарів; у побуті — в смартфонах, фотоапаратах, голосових помічниках.

Розвиток автоматизованого розпізнавання став можливим завдяки трьом ключовим факторам: зростанню обчислювальної потужності (GPU, TPU), появі великих масивів даних (ImageNet, COCO, OpenImages), а також новим теоретичним проривам у штучному інтелекті. У 2014 році було представлено Generative Adversarial Networks (GAN), у 2017 році — трансформери, які згодом, починаючи з 2020 року, стали застосовуватися не лише в NLP, але й у комп'ютерному зорі (Vision Transformers, ViT). Це позначило початок переходу від CNN-центричних методів до універсальних моделей розпізнавання.

На сьогодні автоматизоване розпізнавання є ключовою технологією сучасного світу. Воно працює автономно, часто в режимі реального часу, здатне навчатися на великих масивах даних і пристосовуватися до нових умов. Його мета — покращити ефективність, автоматизувати рутинні процеси, забезпечити точність там, де людське око або мозок мають обмеження, і врешті-решт сформувати інтелектуальні системи, здатні надійно взаємодіяти з реальним світом. Це не винайдено однією особою чи компанією — розвиток відбувався десятиліттями завдяки внеску тисяч науковців, інженерів та дослідницьких груп у всьому світі. Одним з основних засобів, що використовується для автоматизованого навчання – безпілотні літальні апарати.

Безпілотні літальні апарати (БПЛА), або дрони, мають довгу та складну історію, яка починається задовго до їх сучасної популярності у військовій, цивільній та науковій сферах. Їхнє створення стало результатом поєднання розвитку авіації, радіокерування, електроніки та автоматизованих систем

спостереження. Перші ідеї безпілотного польоту виникли ще на початку ХХ століття, коли інженери експериментували з радіокерованими літальними платформами. У 1916–1918 роках були створені перші прототипи радіокерованих літаків у США та Великобританії — вони використовувалися переважно як експериментальні цілі для тренування артилерії. Проте реальний розвиток дронів почався після Другої світової війни, коли технології радіокерування, двигуни та електроніка стали значно компактнішими й надійнішими.



Рис. 1.1 Розвідувально-ударний БПЛА MQ-9 Reaper

У 1950–1970-х роках БПЛА активно розвивалися для військових задач, особливо у сфері розвідки. Саме тоді з'явилися перші системи, здатні автономно виконувати польотні маршрути. В Ізраїлі, США та декількох інших країнах дрони стали ключовими інструментами спостереження, оскільки дозволяли отримувати дані без ризику для життя пілотів. У 1980-х роках Ізраїль продемонстрував ефективність дронів у бойових умовах, що привернуло увагу світу до їхнього потенціалу. У 1990-х роках розвиток GPS, цифрових камер та комп'ютерного обладнання пришвидшив появу нової генерації БПЛА. Приблизно в цей період безпілотники почали оснащувати системами автоматизованого аналізу зображень

і перших елементів комп'ютерного зору, що дозволяло автоматично виявляти об'єкти на місцевості.

На початку XXI століття дрони стали масовим інструментом. Вони почали застосовуватися не лише у військових цілях, але й у цивільних сферах, таких як картографія, агромоніторинг, рятувальні операції, будівництво та інспекції об'єктів інфраструктури. Розвиток комп'ютерного зору та глибокого навчання значно розширив можливості БПЛА. Завдяки потужним нейронним мережам дрони навчилися автоматично розпізнавати транспортні засоби, будівлі, техніку, людей, сільськогосподарські культури, зміни на місцевості, а також виконувати складніші завдання, наприклад, відстеження цілей чи виявлення небезпек у режимі реального часу.

Інтеграція автоматизованого розпізнавання стала одним із ключових етапів еволюції БПЛА. Якщо раніше дрони лише передавали відеосигнал оператору, то сьогодні вони здатні самостійно аналізувати потік даних, визначати важливі елементи сцени, приймати рішення про рух або попереджати оператора про виявлені об'єкти. Це стало можливим завдяки розвитку алгоритмів YOLO, Faster R-CNN, Mask R-CNN, DeepSORT, а також завдяки появі спеціалізованих процесорів для edge-обчислень, здатних працювати безпосередньо на борту дрона.

У сучасних умовах дрони перетворилися на повноцінні автономні системи спостереження й аналітики. Вони здатні працювати у координації з іншими безпілотниками, обмінюватися даними, будувати карти, проводити тепловізійний аналіз, визначати об'єкти приховані рослинністю, а також виконувати завдання розпізнавання навіть у складних погодних умовах. У військовій сфері БПЛА з автоматизованим розпізнаванням стали важливою частиною розвідувальних операцій, коригування вогню, пошуку техніки чи живої сили противника, а також оцінки пошкоджень. Водночас у цивільних сферах вони активно застосовуються для моніторингу довкілля, прогнозування врожаїв, виявлення лісових пожеж, пошукових операцій та складних інспекційних задач.

Таким чином, дрони стали результатом тривалої технологічної еволюції, що поєднала інженерію, авіацію, комп'ютерні науки та штучний інтелект. Їхня роль

у системах автоматизованого розпізнавання постійно зростає, а розвиток глибокого навчання робить ці апарати дедалі автономнішими, точнішими та кориснішими у найрізноманітніших сферах людської діяльності.

## **1.2 Застосування автоматизованого розпізнавання у Збройних Силах**

Автоматизоване розпізнавання у сучасних арміях стало одним із ключових технологічних напрямів, що визначають темпи та характер ведення війни. Воно охоплює широкий спектр застосувань — від аналізу відео з дронів до автоматичної ідентифікації техніки на супутникових знімках, підтримки бойових рішень, моніторингу полів бою й підвищення точності ударів. У Збройних Силах США, України та Великобританії технології візуального ШІ розвиваються паралельно, хоча масштаби, технічна база і способи впровадження різняться залежно від ресурсів, структури та досвіду війни.

У США застосування автоматизованого розпізнавання у військовій сфері охоплює значні масштаби та має системний характер. Наприклад, Project Maven, створена у 2017 році, представляє собою міжвідомчу ініціативу під егідою Міністерства оборони США, яка покликана впровадити машинне навчання та інтеграцію даних у розвідувальні робочі процеси.

Ця ініціатива використовує алгоритми машинного навчання і комп'ютерного зору для аналізу відео та зображень — зокрема, з дронів, супутників, камер спостереження. Завдання включають автоматичне виявлення, класифікацію, і відстеження об'єктів (техніка, споруди, інфраструктура, пересування) на полі бою.

Також США розвивають стратегію об'єднання різних сенсорів (камери, дрони, супутники, радары тощо) у спільну мережу під назвою Joint All-Domain Command and Control (JADC2). Це дозволяє оперативно збирати, обробляти та передавати розвідувальні дані між підрозділами — і тут автоматизоване розпізнавання відіграє ключову роль, забезпечуючи швидке інтерпретування великих потоків інформації.

Окрім того, підрозділи оборонного відомства США почали створювати спеціальні структури для роботи з генеративним штучним інтелектом, аналізом відео, плануванням операцій, логістикою, підтримкою командування.

Через це американські Збройні Сили отримують значні переваги: швидке оброблення розвідувальної інформації, автоматичне виявлення цілей, зменшення навантаження на аналітиків, можливість реагувати в реальному часі та приймати рішення швидше. Завдяки поєднанню аналітики, штучного інтелекту та великої кількості сенсорів, військові здатні підтримувати ситуаційну обізнаність на високому рівні та стрімко реагувати на зміни обстановки.

Незважаючи на те, що багато деталей залишаються невідомими (моделі, алгоритми, конкретна реалізація часто засекречені), загальна тенденція — масштабне залучення автоматизованого розпізнавання до розвідки, планування, командування і керування операціями.

Для України сучасна війна стала каталізатором активного впровадження технологій штучного інтелекту та комп'ютерного зору у військову діяльність. Після повномасштабного вторгнення з'явилися численні проєкти, покликані забезпечити ефективне використання ресурсів, підвищення точності уражень, аналіз розвідданих та автоматизацію рутинних задач.

Один із напрямів — це застосування штучного інтелекту для аналізу супутникових або аеророзвідувальних знімків, щоб виявляти переміщення ворожої техніки, розташування угруповань, логістичні ланцюги, бази тощо; це дозволяє пришвидшити процес від розвідки до фактичного бойового застосування.

Ще одне важливе застосування — дрони. Наприклад, за даними засобів масової інформації, українська компанія ZIR System розробила систему на базі штучного інтелекту, що дає змогу дронам автоматично розпізнавати різні типи цілей (піхота, машини, техніка, артилерія, танки тощо), навіть коли вони частково замасковані чи знаходяться в лісі.



Рис. 1.2 Супутниковий знімок міста Бахмут.

Завдяки таким системам, Україна може ефективніше використовувати недорогі безпілотники, навіть у ситуації, коли зв'язок із оператором переривається або за умов радіоефірного глушіння — штучний інтелект допомагає «завершити» місію автономно або з мінімальним контролем.

Крім того, автоматизовані системи допомагають прискорити процес прийняття рішень, зменшити завантаженість людей найрізноманітнішими рутинними задачами, обробити великі об'єми даних, комбінуючи розвідку, відео з дронів, супутникові знімки, метеоінформацію тощо — це дає змогу оперативно планувати дії, координувати підрозділи, підвищувати точність ударів і мінімізувати ризики.

Водночас, через брак часу, ресурсів і часто — через обмежений доступ до потужної обчислювальної інфраструктури, реалізація таких систем в Україні суттєво відрізняється від масштабів США. Часто використовують комерційні чи напівкомерційні рішення, адаптовані під бойові умови, або розробки приватних компаній. Проте у контексті війни навіть така «міцна середина» дає значну

перевагу, особливо у поєднанні з високою мобілізацією, гнучкістю та практичним досвідом реального бою.

Офіційні джерела про широкомасштабне застосування комп'ютерного зору або систем розпізнавання у Збройних Силах Великобританії дещо обмежені, особливо в публічній площині — з огляду на конфіденційність військових програм. Проте останні заяви вказують на те, що британські Збройні Сили активно планують інтегрувати штучний інтелект в майбутню структуру оборони. Наприклад, у 2025 році Міністр оборони Великобританії оголосив, що штучний інтелект стане центральним елементом майбутньої «стратегічної оборонної програми», з метою модернізації Збройних Сил, прискорення прийняття рішень, підвищення гнучкості та оперативності.

Серед можливих напрямів — аналітика з розвідувальних сенсорів, відео та супутникових даних, підтримка управління, логістики, планування операцій, а також інтеграція з безпілотними системами.

В британському контексті важливим є підхід до відповідальності та контролю: при впровадженні штучного інтелекту в оборону планується забезпечити належні політики та процедури, які гарантуватимуть, що рішення зі застосування сили (наприклад, ураження цілей) приймає людина — навіть якщо штучний інтелект надає рекомендації чи допомагає з цілевказанням.

Зараз британський оборонний сектор перебуває в стані трансформації: активно аналізуються можливості для інвестування в ШІ-проекти, проведуться дослідження, тести та розгортання пілотних систем. Швидше за все, за кілька років ми побачимо реальні приклади використання автоматизованого розпізнавання, інтегрованого з дронами, розвідкою, командуванням і підтримкою рішень.

### **1.3 Методики автоматизованого розпізнавання**

Автоматизоване розпізнавання ґрунтується на наборах алгоритмів і моделей, які дозволяють комп'ютеру аналізувати зображення, відео, аудіо або інші дані та робити висновки про те, які об'єкти, події або явища в них присутні. Ці методики

пройшли тривалий шлях розвитку — від простих порогових правил і фільтрів до сучасних глибоких нейронних мереж зі складною багаторівневою архітектурою.

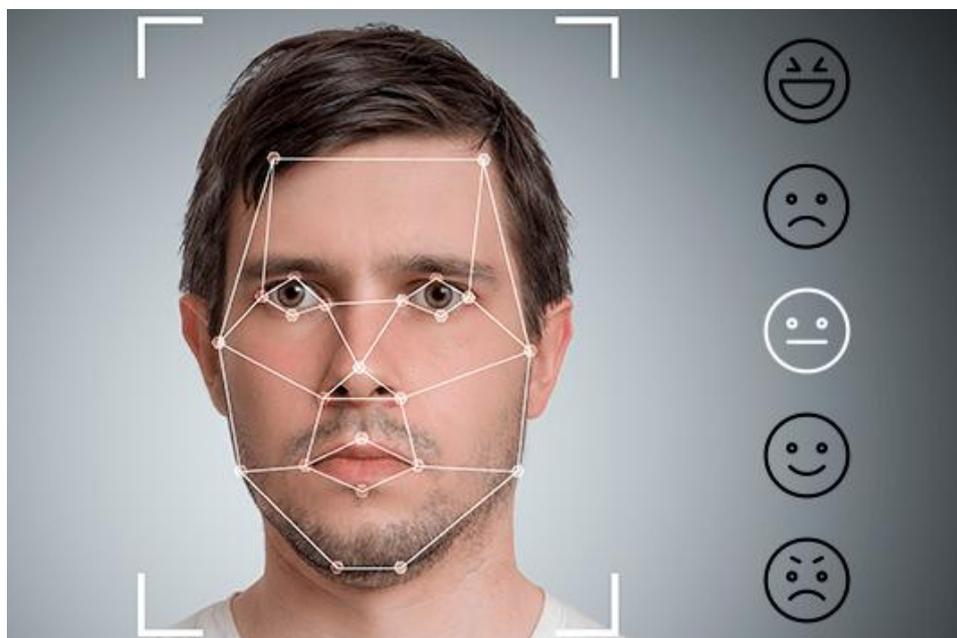


Рис. 1.3 Виділення ознак за допомогою комп'ютерного зору

У 1950–1960 роках основною ідеєю було те, що кожне зображення можна описати сукупністю наперед визначених ознак, які дослідник формулював вручну. Такий підхід — відомий сьогодні як ручне вилучення ознак — став фундаментом для перших чотирьох методик, що визначали розвиток комп'ютерного зору до появи глибокого навчання.

Одною з ранніх та найвпливовіших методик стало шаблонне розпізнавання, яке сформувалося в 1960–1970-х роках у контексті розвитку цифрових обчислень. Цей підхід передбачав накладання зразка-шаблону на зображення та пошук максимального збігу через кореляцію чи порівняння пікселів. Авторство шаблонних методів не можна приписати одній людині — це був природний етап еволюції цифрової обробки сигналів. Вони широко застосовувалися у ранніх системах OCR, перших промислових візуальних інспекціях та у системах аналізу друкованих символів. Попри свою простоту, шаблонне розпізнавання було надзвичайно обмеженим: воно не давало змоги враховувати зміну масштабу,

поворотів чи деформацій об'єкта, що згодом спричинило пошук більш стійких, інваріантних методик.

Великим проривом у 1990-х роках стали статистичні та структурні методи вилучення ознак, що дали початок таким підходам, як Haar-каскади та HOG-ознаки. Однією з найвідоміших ранніх робіт був алгоритм Haar Cascade Object Detection, створений Полом Віолою та Майклом Джонсом у 2001 році в лабораторії MIT. Їх стаття “Rapid Object Detection using a Boosted Cascade of Simple Features” стала революційною, оскільки вперше продемонструвала можливість детекції об'єктів у реальному часі на звичайних процесорах. Метод поєднував прості прямокутні ознаки Гаара, інтегральні зображення та каскадну структуру класифікаторів AdaBoost. Найбільш відомим застосуванням стала детекція облич, яка на початку 2000-х сформувала основу для камерних технологій у цифрових фотоапаратах, перших смартфонах і системах відеоспостереження. Попри свою ефективність, Haar-каскади потребували ручного визначення ознак і мали обмежені можливості в умовах складного фону, змін освітлення чи випадкових поворотів.

Ще одним ключовим етапом став розвиток Histogram of Oriented Gradients — методики, яку презентував Навніт Далал у співпраці з Біллом Трігтсом у 2005 році. Дослідження виконувалися у французькому INRIA й були сфокусовані на задачі детекції людей у міських умовах. HOG-ознаки виявилися надзвичайно стійкими до змін освітлення та локальних деформацій і швидко стали найпопулярнішою методикою для задач машинного зору до появи нейромереж. Детектори HOG+SVM десятиліттями використовувалися в автомобільній промисловості для пішохідної безпеки, у індустрії відеоспостереження, робототехніці та інших сферах. Однак ключовим недоліком залишалася залежність від ручного визначення ознак — модель бачила лише те, що інженер був здатний формально описати.

Попередньо визначені ознаки та класичні методи також включали широке розмаїття локальних дескрипторів, що з'являлися у 1990–2000-х роках. Серед найвідоміших — SIFT (Scale-Invariant Feature Transform), запропонований Девідом Лоу у 1999–2004 роках. SIFT став першим по-справжньому інваріантним до масштабу методом, здатним виявляти ключові точки та порівнювати

їх між зображеннями. Він був надзвичайно впливовим і відкрив нові можливості для 3D-реконструкції, SLAM-систем, мапування та ідентифікації об'єктів на складних сценах. Подібні дескриптори, такі як SURF, HESLAP чи BRIEF, домінували понад десять років і були основою комп'ютерного зору до епохи глибоких нейронних мереж.

Усі ці методики мали спільну характеристику — вони залежали від людського інженера. Саме дослідник визначав, яку ознаку вважати “важливою”, які структури слід шукати, як описувати контури, текстури чи градієнти. Це обмежувало гнучкість систем і не дозволяло їм масштабуватися на складні, неконтрольовані середовища.

Саме така криза традиційних методів і стала передумовою для появи глибокого навчання, яке змінило парадигму з «людина визначає ознаки» на «модель сама навчається ознакам». Але й попри це, перші чотири класичні методики залишаються історичним фундаментом автоматизованого розпізнавання, оскільки саме вони визначили розвиток галузі і стали відправною точкою для сучасних нейромережових технологій.

Методика автоматизованого розпізнавання на основі глибоких нейронних мереж, зокрема згорткових нейронних мереж (Convolutional Neural Networks, CNN), стала фундаментальною технологією в еволюції сучасних систем комп'ютерного зору. Її формування відбулося поступово, як результат десятиліть досліджень у галузі штучних нейронних мереж, математичного моделювання та обробки зображень. Хоча базові ідеї штучних нейронних мереж виникли ще у 1950-х роках, саме прориви кінця XX і початку XXI століття дали змогу сформуванню технології, яка сьогодні є центральною в автоматизованому аналізі візуальної інформації.

Історичним підґрунтям для появи CNN стали роботи Куніхіко Фукушіма, який на початку 1980-х створив архітектуру Neocognitron — першу модель, що застосовувала ідею локальних рецептивних полів та ієрархічного вилучення ознак. Ця архітектура вперше продемонструвала, що машина може навчитися виділяти інформативні патерни без ручного програмування детекторів. Однак

повноцінне становлення сучасних CNN стало можливим завдяки ЛеКуну та його команді в AT&T Bell Labs, які у 1989–1998 роках розробили мережу LeNet-5. Саме вона показала високі результати у задачах розпізнавання рукописних символів та стала практичним доказом того, що згорткові мережі здатні ефективно навчатися на реальних даних.

Подальший розвиток методики став можливим після 2012 року, коли відбулася знаменита подія — модель AlexNet під керівництвом Джеффри Гінтона та його учнів Алекса Крижевського і Іллі Суцкевера здобула переконливу перемогу у змаганні ImageNet ILSVRC. AlexNet довела, що глибокі CNN за умови достатнього обсягу даних і обчислювальних ресурсів значно перевершують усі попередні методи комп'ютерного зору. Ця перемога стала поворотним моментом, що запустив хвилю масового впровадження CNN у наукових, промислових і військових системах.

Глибокі CNN відрізняються від традиційних методів тим, що не покладаються на вручну створені дескриптори або фільтри. Натомість вони здатні самостійно, у процесі тренування, формувати багаторівневі представлення візуальної інформації — від примітивних градієнтів і текстур до складних семантичних патернів. Завдяки цьому CNN демонструють високу стійкість до варіацій освітлення, положення об'єкта у кадрі, масштабу та шумів, що робить їх надзвичайно ефективними для автоматизованого розпізнавання.

Методика глибоких CNN не є статичною — вона розвивається щороку завдяки появі нових архітектур, алгоритмів оптимізації та технік регуляризації. Після AlexNet з'явилися такі моделі, як VGGNet (2014), що продемонструвала важливість глибини мережі; ResNet (2015), що ввела концепцію залишкових зв'язків і стала справжнім проривом у можливості тренувати надзвичайно глибокі моделі; Inception (2014–2016), яка оптимізувала використання параметрів і обчислень; EfficientNet (2019), що запропонувала системний підхід до масштабування мереж. Кожне з цих удосконалень зміцнювало методику й розширювало сферу її застосування — від класифікації до сегментації, від детекції до аналізу динамічних сцен.

Сучасні CNN використовуються не лише у класичній класифікації зображень, а й у складних комп'ютерних задачах: ідентифікації об'єктів у реальному часі, аналізі відеопотоків, розпізнаванні поведінкових патернів, медичній діагностиці, геопросторовій аналітиці, автономному водінні, системах безпеки, робототехніці та оборонних технологіях. Головною перевагою CNN є їхня здатність забезпечувати високу точність і автономність у процесі інтерпретації візуальної інформації, що робить їх основою сучасних методик автоматизованого розпізнавання.

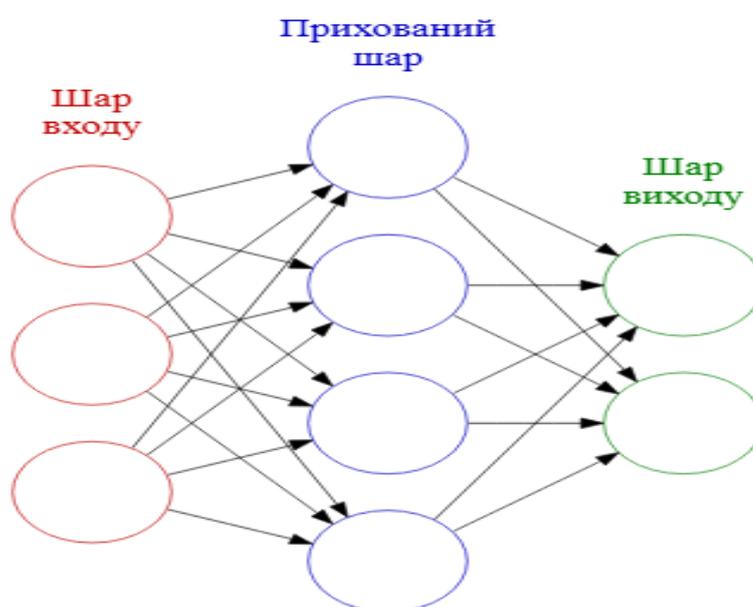


Рис 1.4 Робота методики на основі глибоких нейронних мереж

Одностадійні та двостадійні детектори є двома фундаментальними підходами до задачі виявлення об'єктів у комп'ютерному зорі. Обидва виникли як результат еволюції нейронних моделей і прагнення зробити детекцію об'єктів одночасно швидкою та точною. Хоча вони вирішують схожу задачу, ці методики мають різні історичні корені, архітектурні рішення, дослідницькі контексти та практичні застосування.

Одностадійні детектори з'явилися як спроба суттєво прискорити процес детекції, усунувши проміжний етап генерації регіонів-кандидатів. Перші ідеї зароджувалися ще на початку 2010-х років, коли класичні алгоритми на кшталт

HOG + SVM або DPM вже досягли своєї стелі продуктивності. Прорив стався у 2015 році, коли Джозеф Редмон (Joseph Redmon) у Facebook AI Research представив першу версію YOLO (You Only Look Once). Головна ідея була в тому, щоб подати завдання детекції як задачу одномоментного передбачення: мережа отримує зображення, розбиває його на сітку й одночасно виводить координати, класи та ймовірності об'єктів.

YOLO довів, що детекція може виконуватися у реальному часі без істотної втрати точності. У наступні роки з'явилися покращення YOLOv2, YOLOv3 (2016–2018), у яких удосконалили архітектуру, ввели anchor boxes, багатомасштабність та нові функції втрат. Паралельно народжувалася інша важлива гілка — SSD (Single Shot MultiBox Detector), представлена у 2016 році дослідниками з Google. SSD поєднав ідею anchor boxes із багаторівневими картами ознак, що дало змогу ефективно виявляти об'єкти різних розмірів.

Одностадійні моделі створювалися з чіткою метою — максимальна швидкість. Вони стали незамінними у системах реального часу: безпілотники, робототехніка, автономні автомобілі, відеоспостереження, edge-пристрої. Протягом 2020–2024 років розвиток продовжився завдяки YOLOv4, YOLOv5, EfficientDet, YOLOX, YOLOv7 та революційним YOLOv8 і YOLO-NAS, що зробили одностадійні детектори більш точними, ніж деякі двостадійні моделі. Екосистема стала особливо активною завдяки open-source спільнотам та дослідникам, які прагнули вирішити баланс між швидкістю, точністю та можливістю обробки великих потоків відео.

Суть одностадійного підходу проста: модель без проміжних кроків безпосередньо передбачає клас і координати для кожної комірки сітки. Це робить архітектуру компактною, ефективною та прекрасно придатною для апаратного прискорення. Одностадійні детектори фактично перевернули уявлення про те, що швидко означає неточно — сучасні представники перевершують конкурентів у багатьох категоріях.

Двостадійні детектори сформувалися раніше й стали першою справді точною парадигмою глибокої детекції. Їхня історія почалася у 2014 році, коли група

дослідників під керівництвом Росса Гіршіка (Ross Girshick) у Microsoft Research опублікувала роботу R-CNN. Ця архітектура комбінувала Selective Search для виділення областей-кандидатів та CNN для класифікації кожної області. Попри високу точність, R-CNN був повільним і незручним у використанні.

У 2015 році з'явився Fast R-CNN, який значно пришвидшив процес, а вже у 2016 Faster R-CNN став справжнім проривом. У ньому Selective Search замінили на Region Proposal Network (RPN) — першу модель, здатну сама генерувати області-кандидати. Структура двостадійного детектора стала завершеною: спершу мережа пропонує можливі регіони, а потім кожен регіон уточнюється і класифікується. Цей підхід забезпечує надзвичайну точність, особливо у складних сценах.

Метод був створений з академічною метою — поліпшення продуктивності в дослідницьких тестах (VOC, COCO) — і став базовим стандартом точності на багато років. Пізніше з'явилися його похідні: Mask R-CNN (2017, Facebook AI Research), Cascade R-CNN (2018), DetectoRS (2021). Усі вони зберігають двостадійну концепцію, розширюючи її масками сегментації, каскадними уточненнями чи вдосконаленими архітектурними блоками.

Двостадійні детектори були створені з метою отримати максимальну точність за будь-яку ціну. Вони використовуються у наукових дослідженнях, медичній діагностиці, супутниковій аналітиці, юридичній експертизі, системах з високими вимогами до точності класифікації та локалізації. Це методики, які вибирають там, де навіть невелика похибка може бути критичною.

Суть підходу полягає в поділі процесу на два окремі етапи: генерацію потенційних областей і подальше точне передбачення для кожної з них. Завдяки цьому мережа може концентрувати ресурси на важливих ділянках зображення, досягаючи вищої точності, ніж одностадійні аналоги, але жертвуючи швидкістю.

Одностадійні детектори стали символом реального часу, мобільності та практичного застосування. Двостадійні — символом наукової досконалості та високої точності. Обидві методики виникли від різних дослідницьких мотивацій, були створені різними групами науковців і у різні часові періоди, але сьогодні вони

існують як два рівноправні підходи, кожен з яких має свої унікальні переваги та варіанти застосування.

Хоча колись вони конкурували, сучасний комп'ютерний зір використовує обидва класи моделей у залежності від того, що важливіше — швидкість чи точність. Разом ці напрямки сформували сучасну екосистему детекції, заклали основи для подальших інновацій і забезпечили розвиток усього напрямку AI-зору.

Методика генеративного аналізу — це один із наймолодших і найдинамічніших напрямів у сучасному комп'ютерному зорі та машинному навчанні, що виріс із досліджень у сфері штучних нейронних мереж початку 2010-х років. Її виникнення пов'язане з переходом від традиційних статистичних методів до моделей, здатних не просто розпізнавати інформацію, а створювати нові дані, які ніколи не зустрічалися в навчальних наборах. Саме ця здатність — генерувати реалістичні зображення, відео, тексти й навіть нові варіації об'єктів — і дала назву методиці.

Перші ключові кроки до становлення генеративного аналізу закладалися в дослідженнях автоencoder-архітектур. У 2013 році автоенкодери використовувалися для стискання та реконструкції зображень, але вони ще не давали справжнього генеративного потенціалу. Прорив стався у 2014 році, коли Йєн Гудфеллоу та його колеги представили Generative Adversarial Networks (GANs) — модель, у якій два нейронні мережі протистоять одна одній, змагаючись у створенні та виявленні нових синтетичних даних. Це відкриття стало одним із найвпливовіших у галузі глибинного навчання за останнє десятиліття. GAN-мережі швидко довели свою здатність створювати зображення, майже невідмінні від фотографій, синтезувати об'єкти, змінювати стилі та навіть генерувати нові сцени на основі текстових описів. Поява GAN стала фундаментом для сучасного генеративного аналізу, оскільки дозволила моделювати складні розподіли реальних даних.

У паралельному напрямі, приблизно у 2013–2016 роках, активно розвивалися варіаційні автоенкодери (VAE), створені Дмитром Кінгмою та Максом Велінгом. VAE вперше дали спосіб математично описати простір можливих варіацій об'єктів

і генерувати їх із контрольованою випадковістю. На відміну від GAN, які працюють змагально, VAE пропонують суворо обґрунтовану ймовірнісну модель, що дозволяє генерувати плавні та керовані зміни між зображеннями, створювати узагальнені уявлення та модифікувати структуру об'єктів. Саме VAE вперше дозволили "заглянути" у латентний простір і керувати ним.

У 2018–2020 роках новою революцією стали дифузійні моделі, які започаткували дослідники DeepMind та інших провідних лабораторій. Ці моделі навчаються на процесі поступового "за-шумлення" та "роз-шумлення" зображень, виконуючи зворотний дифузійний процес. Основна ідея полягає в тому, що зображення перетворюється на шум, а модель вчиться відтворювати шлях назад — генеруючи реалістичну картину з випадкових точок. У 2020–2022 роках саме дифузійні моделі, такі як DDPM, Latent Diffusion та Stable Diffusion, стали світовим проривом, оскільки продемонстрували найвищу якість й реалістичність генерації зображень.

Генеративний аналіз не обмежується лише створенням зображень. Він має набагато ширший спектр застосувань. У комп'ютерному зорі його використовують для доповнення наборів даних, створення синтетичних варіацій об'єктів, моделювання складних сценаріїв, реконструкції зображень, редагування контенту й навіть побудови систем, що вміють пояснювати свою роботу. У медицині генеративні моделі відтворюють важкодоступні медичні знімки, покращують якість КТ та МРТ. У військових та безпекових задачах вони допомагають генерувати симулятивні дані для тренування детекторів, створювати варіанти небезпечних об'єктів, моделювати бойові сценарії або реконструювати низькоякісні кадри з відеоспостереження. У сфері автономного транспорту вони дозволяють синтезувати складні дорожні умови, яких немає в реальних наборах даних.

З появою великих трансформерів, таких як Vision Transformer (ViT) і тексто-візуальних моделей на зразок DALL·E та Imagen, генеративний аналіз почав інтегрувати семантику на новому рівні. Тепер генеративні моделі здатні працювати з контекстом, описом, структурою сцени та логікою світу. Сучасні моделі

уможливлюють перетворення текстової інформації в 3D-об'єкти, відео чи складні багатокадрові послідовності.

Методика генеративного аналізу стала фундаментом для побудови нових систем виявлення об'єктів, оскільки дозволяє навчати детектори навіть у випадках, коли реальних даних мало або вони недоступні. Завдяки здатності створювати контрольовані та реалістичні варіації об'єктів, генеративні моделі відіграють ключову роль у сучасних дослідницьких і промислових системах. Вони поєднують у собі теоретичну строгість, інноваційні ідеї та практичну користь, демонструючи, що штучний інтелект може не лише аналізувати, а й творчо продукувати дані нового покоління.

Важливе місце займають статистичні методи розпізнавання, що базуються на теорії ймовірностей і математичній статистиці. У межах цього підходу об'єкти або класи описуються ймовірнісними моделями, які відображають розподіли їхніх ознак. Розпізнавання здійснюється шляхом обчислення ймовірності належності спостереження до певного класу. До таких методів належать баєсівські класифікатори, гаусівські суміші, приховані марковські моделі. Вони широко застосовуються для розпізнавання сигналів, мовлення, часових рядів і процесів, де важливими є послідовність подій та їх імовірнісна природа.

Окремим напрямом є методи розпізнавання образів на основі правил і логічного висновку. У таких системах знання про предметну область формалізуються у вигляді набору правил, умов і логічних зв'язків. Розпізнавання здійснюється шляхом перевірки відповідності об'єкта цим правилам. Подібні підходи активно використовувалися в експертних системах і системах підтримки прийняття рішень. Їх перевагою є прозорість логіки та можливість прямого врахування експертних знань, однак складність масштабування і обмежена гнучкість знижують ефективність у задачах із великою різноманітністю даних.

Суттєвий внесок у автоматизоване розпізнавання роблять методи обробки сигналів. Вони застосовуються тоді, коли об'єкти представлені у вигляді часових або просторово-часових сигналів, наприклад аудіо, радарних або сенсорних даних. Для аналізу використовуються перетворення Фур'є, вейвлет-аналіз, фільтрація,

спектральний аналіз, автокореляція. Отримані параметри сигналів слугують основою для подальшої класифікації або ідентифікації. Ці методики є фундаментальними у задачах розпізнавання мовлення, біометрії, радіолокації та технічної діагностики.

Ще одним напрямом є структурні методи розпізнавання, у яких об'єкти описуються не набором числових ознак, а структурою та взаємозв'язками між елементами. Для цього використовують графи, дерева, граматики та формальні мови. Розпізнавання полягає у порівнянні структур або пошуку ізоморфізмів між ними. Такі методи особливо ефективні для аналізу складних об'єктів із чіткою ієрархією або топологією, наприклад схем, символів, молекулярних структур, документів із формалізованою будовою.

Також застосовуються методи нечіткої логіки, які дозволяють працювати з невизначеністю та нечіткими поняттями. У цих підходах об'єкти можуть частково належати до кількох класів із різним ступенем впевненості. Нечіткі системи добре підходять для ситуацій, де складно задати чіткі межі між класами або де рішення має прийматися на основі наближених, експертних оцінок. Вони часто використовуються в інтелектуальних системах керування, діагностики та підтримки рішень.

Окрему групу становлять еволюційні та біоінспіровані методи, такі як генетичні алгоритми, роїння частинок, мурашині алгоритми. В автоматизованому розпізнаванні вони зазвичай застосовуються для оптимізації параметрів моделей, відбору ознак або налаштування структури системи. Ці методи імітують природні процеси еволюції та колективної поведінки, що дозволяє знаходити ефективні рішення у складних багатовимірних просторах.

## 2. АНАЛІЗ МЕТОДИК АВТОМАТИЗОВАНОГО РОЗПІЗНАВАННЯ СТРІЛЕЦЬКОЇ ЗБРОЇ НА КЛАСИФІКАЦІЇ

### 2.1 Аналіз існуючих методів автоматизованого розпізнавання

Методика автоматизованого розпізнавання на основі комп'ютерного зору — це сукупність алгоритмів, моделей та технологій, які дозволяють цифровим системам інтерпретувати, аналізувати та осмислювати візуальну інформацію подібно до того, як це робить людський зір. Її розвиток охоплює кілька десятиліть наукових досліджень та інженерних проривів, починаючи від класичних математичних методів аналізу зображень і закінчуючи сучасними глибокими нейронними мережами.

Основна мета методики полягає в тому, щоб перетворити необроблені піксельні дані на структуровану інформацію, яку комп'ютер може інтерпретувати: класи об'єктів, їхню поведінку, просторове розташування, геометричні параметри, динаміку сцени та інші семантичні ознаки. Це робить можливими не лише прості задачі розпізнавання, а й складніший аналіз середовища, включно з прогнозуванням, відстеженням, ідентифікацією та генерацією нових візуальних даних.

Ранні етапи розвитку методики базувалися на строгих математичних моделях: перетворенні Фур'є, фільтрах Габора, морфологічній обробці, методах виділення контурів і шаблонному зіставленні. У 1980–1990-х роках розвиваються локальні дескриптори, такі як SIFT, SURF, HOG, які дозволили вперше досягти стійкого розпізнавання об'єктів у різних умовах освітлення та масштабу. Проте всі ці підходи мали істотний недолік — вони вимагали вручну спроектованих ознак, а їх ефективність залежала від досвіду інженера та конкретних характеристик задачі.

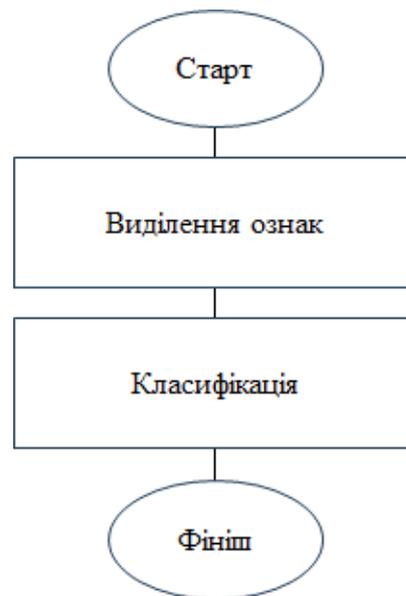


Рис. 2.1 Схеми алгоритму методики на основі комп'ютерного зору

Комп'ютерний зір, як методика автоматизованого розпізнавання, і сьогодні продовжує розвиватися. Додано механізми самопояснюваності моделей, підвищується їхня стійкість до шуму та атак, зменшуються обчислювальні витрати, удосконалюється генеративний напрям, що дозволяє системам не лише впізнавати, а й створювати зображення, прогнозувати розвиток сцени або заповнювати пропущені елементи. При цьому методика стає більш універсальною: вона застосовується в медицині, автомобільній промисловості, оборонній сфері, робототехніці, сільському господарстві, безпеці, торгівлі та інформаційних технологіях.

У своїй сучасній формі методика автоматизованого розпізнавання — це синтез математичних засобів, машинного навчання, нейронних мереж, оптимізаційних технік і апаратних прискорювачів. Вона охоплює повний цикл обробки: від збору та підготовки даних до побудови моделей, їх оптимізації, масштабування в реальні системи та постійного вдосконалення. Її сила полягає в тому, що комп'ютерний зір здатен не просто «бачити», а й «розуміти» контекст, логіку та структуру візуального світу, роблячи його аналіз точним, швидким і автоматизованим.

Глибоке навчання — це напрям штучного інтелекту, який базується на використанні багатосарових нейронних мереж, здатних самостійно вчитися представленням даних різної складності. Його фундаментальна ідея полягає в тому, що комп'ютерна модель може навчитися знаходити закономірності у даних без ручного формулювання правил, як це робилося в класичних алгоритмах машинного навчання. Замість того, щоб програміст визначав, які саме ознаки важливі, модель сама їх відкриває, будуючи дедалі складніші рівні абстракції — від простих граней та контурів до комплексних структур, таких як об'єкти або семантичні поняття.

Глибоке навчання дозволяє створювати системи, здатні бачити, чути, розуміти текст, генерувати новий контент, прогнозувати тенденції та робити складні рішення на основі великих обсягів даних. Воно лежить в основі сучасного комп'ютерного зору, розпізнавання мовлення, перекладу, робототехніки, автономного транспорту, біоінформатики, фінансового аналізу та інших інтелектуальних систем. Найбільша його цінність полягає в тому, що такі моделі стають точнішими зі зростанням даних, а з кожним поколінням архітектур вони демонструють нові можливості, що перевершують людські здібності у багатьох вузьких задачах.

Методи навчання в глибокому навчанні розвивалися паралельно з еволюцією штучних нейронних мереж і сьогодні охоплюють три великі групи: кероване, напівкероване та некероване навчання. Кожен із цих підходів був сформований зі своєю філософією, обсягом необхідних даних, рівнем людської участі та типами завдань, які вони найкраще розв'язують. Їхнє застосування безпосередньо впливає на ефективність систем комп'ютерного зору, обробки мови, генеративного моделювання та аналітики великих даних, тому розуміння цих методів є фундаментальним для досліджень у галузі штучного інтелекту

Кероване навчання є найстарішою і найпоширенішою парадигмою глибокого навчання. Воно формувалося ще з 1950–1960-х років у рамках класичної статистики та перших моделей розпізнавання образів, але справжній прорив відбувся після появи великих датасетів і потужних графічних процесорів

у 2000–2010-х роках. Суть керованого навчання полягає в тому, що нейронна мережа отримує великий набір прикладів, кожен із яких має правильну відповідь – мітку класу, координати об'єкта, значення для регресії або іншу форму істинного результату. Мережа вчиться відтворювати ці мітки, порівнюючи власні передбачення з правильними значеннями та поступово мінімізуючи помилку. Перевага цього методу полягає в його точності та передбачуваності, адже модель навчається на чітко визначених задачах. Він став основою таких проривів як ImageNet 2012, коли глибока мережа AlexNet продемонструвала перевагу над класичними алгоритмами. Недоліком керованого навчання є висока потреба в великих обсягах анотованих даних, а створення анотацій часто є дорогим, тривалим і потребує участі експертів.

Алгоритм керованого навчання має початковий набір, який складається з  $N$  прикладів, кожен з яких навчається вказується в такій формі:  $\{(x_1, y_1), \dots, (x_N, y_N)\}$ , де  $x_i$  – є вектором вхідних символів  $i$ -го прикладу, а  $y_i$  – вхідні значення для  $i$ -го прикладу. У наступному кроці алгоритм шукає функцію  $g: X \rightarrow Y$ , де  $X$  – вхідний простір моделі даних, а  $Y$  – вихідний простір внесених даних. Функцію  $g$  можна розглядати як частину простору функцій  $G$ , який також називають простором гіпотез. Взагалі функцію  $g$  можна представити іншою функцією  $f: X \times Y \rightarrow \mathbb{R}$ , де  $f$  – функція, що обчислює значення  $u$ , що оптимізує вираз  $g(x) = \operatorname{argmax}_y f(x, y)$ , де  $f$  – функція  $\Phi$ .

Напівкероване навчання виникло як відповідь на проблему нестачі розмічених даних і почало активно розвиватися після 2010 року, коли стало очевидно, що більшість інформації у світі не має міток. Ця парадигма поєднує невелику кількість мічених прикладів із великими масивами немічених даних. Модель використовує обмежену розмітку як «орієнтир», а щодо незмічених даних вчиться визначати структури, шаблони та закономірності самостійно. Значні прориви відбувалися під час появи методів consistency regularization, pseudo-labeling, MixMatch та FixMatch, які дозволили значно скоротити потребу в ручному маркуванні. Напівкероване навчання особливо ефективно в тих областях, де анотації дорого створювати, наприклад у медицині, відеоаналітиці

або робототехніці. Завдяки цьому методіві глибокі мережі змогли навчатися з мінімальним людським втручанням, зберігаючи при цьому близький рівень точності до керованих підходів.

Некероване навчання є найбільш «автономною» формою навчання і бере свій початок ще з досліджень когнітивних процесів у 1980-х роках, коли виникли перші автоасоціативні мережі та методи кластеризації. Його основна ідея полягає в тому, що алгоритм не отримує жодних міток і намагається самостійно знайти структуру у даних: групувати схожі об'єкти, стискати інформацію, вилучати приховані характеристики, виявляти аномалії або закономірності. Поява автоенкодерів у 2006 році, а пізніше генеративних моделей, зокрема Variational Autoencoder (2013) і Generative Adversarial Networks — GAN (2014), зробила некероване навчання надзвичайно потужним. Завдяки цим методам системи навчилися створювати нові дані, реконструювати зображення, відновлювати пропуски, генерувати контент і навчатися без людського нагляду. Некероване навчання особливо важливе в ситуаціях, коли структура даних невідома, коли потрібно скоротити розміри даних або коли мітки не можуть бути визначені однозначно.

Навчання з підкріпленням є методом, у якому штучний агент вчиться взаємодіяти з навколишнім середовищем через систему винагород і покарань. Замість того щоб отримувати набір готових прикладів із правильними відповідями, модель поступово досліджує середовище, виконує певні дії та спостерігає наслідки. Метою агента є максимізація сумарної винагороди у довгостроковій перспективі. Такий підхід походить із психології поведінки та біології, де тварини вчать через наслідки власних дій. У машинному навчанні ця концепція стала формалізованою завдяки Марковським процесам прийняття рішень та роботам Річарда Саттона і Ендрю Барто, які систематизували фундаментальні основи у 1990-х роках. За допомогою цього методу створюються алгоритми, здатні самостійно знаходити оптимальні стратегії: від автономного керування транспортом до навігації роботів, оптимізації систем, і, звичайно, відомих прикладів таких як AlphaGo, де агент навчився перевершувати людей шляхом мільйонів симулювань. Навчання з підкріпленням відрізняється тим, що знання формується не через статичний набір

даних, а через активний процес взаємодії й адаптації, де щоразу дія формується на основі власного досвіду.

Трансдукція є іншим методом машинного навчання, який фокусується не на побудові універсальної моделі, а на безпосередньому передбаченні результатів для конкретного набору тестових прикладів. На відміну від індукції, що прагне узагальнити знання на всі можливі майбутні дані, трансдукція працює точково, виходячи лише з наявних навчальних прикладів та конкретних тестових точок, для яких потрібно зробити прогноз. Цей підхід запропонував Володимир Вапнік — один із творців теорії статистичного навчання та опорних векторів. Трансдукція ґрунтується на принципі мінімізації ризику в умовах реального розподілу даних, а не створення абстрактної моделі, яка повинна працювати для будь-яких ситуацій. У практичному сенсі трансдуктивні алгоритми прагнуть зберегти структуру даних, виявити залежності між навчальними та тестовими зразками й одразу використовують ці взаємозв'язки для передбачення. Зокрема, трансдукція застосовується у завданнях класифікації з малою кількістю міток, у графових моделях, у рекомендаційних системах та методах напівкерованого навчання, де важливо максимально ефективно використати локальну структуру даних.

Сучасні системи глибокого навчання часто комбінують три підходи. Кероване навчання відповідає за точність і контрольованість результатів. Напівкеровані методи дають змогу працювати з великими наборами даних, у яких міток небагато. Некероване навчання формує фундамент розуміння даних і забезпечує можливість навчатися без участі людини. Разом ці методи формують повноцінну екосистему навчання, яка дає змогу штучним нейронним мережам діяти ефективно в реальному середовищі, зменшувати залежність від ручної праці та адаптуватися до складних задач, характерних для комп'ютерного зору, мовних систем, аналізу поведінки та генеративного моделювання.

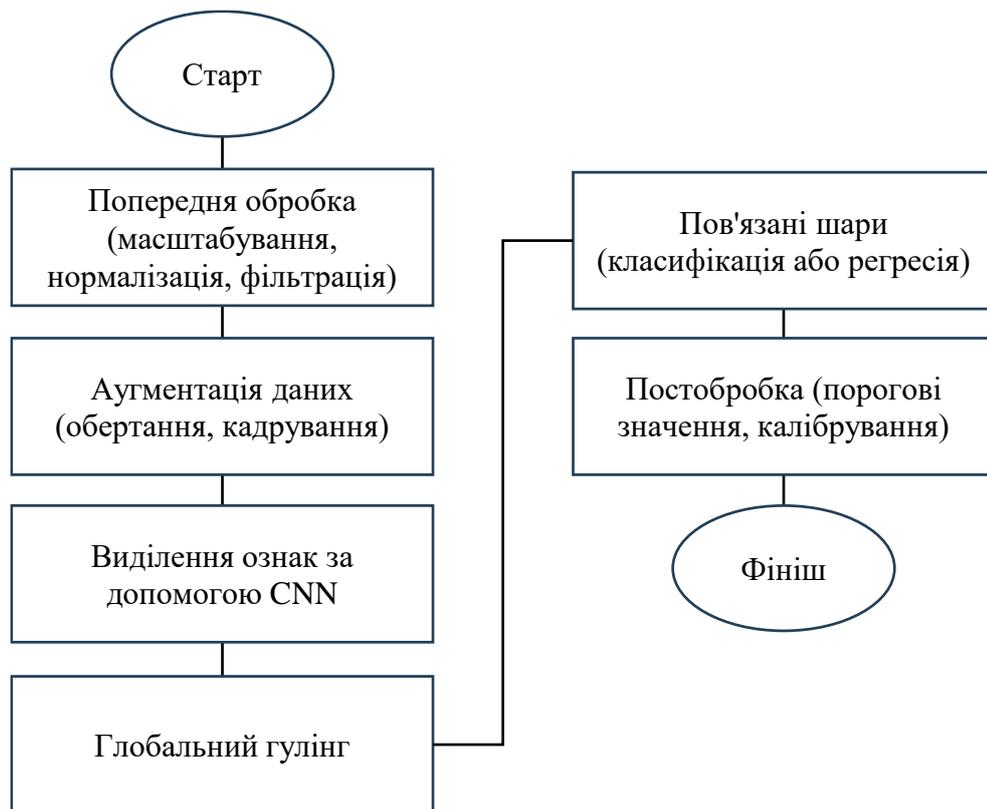


Рис. 2.2 Схеми алгоритму методики на основі глибоких нейронних мереж

Процес автоматизованого розпізнавання на основі глибоких нейронних мереж починається з отримання вихідного вхідного сигналу, найчастіше — цифрового зображення або відеокадру. На цьому початковому етапі система нічого не знає про зміст даних, тому зображення є лише набором числових значень пікселів. Перед тим як передати ці дані до моделі, відбувається попередня обробка. Вона включає нормалізацію яскравості, масштабування до потрібного розміру, усунення шумів, корекцію кольорів і перетворення у формат, оптимальний для обчислень. Цей етап критично важливий, оскільки він усуває випадкові спотворення та підвищує якість інформації, яку модель буде аналізувати.

Після підготовки даних зображення надходить у згорткову нейронну мережу. CNN складається з кількох рівнів, кожен із яких виконує певну функцію в аналізі та перетворенні інформації. Першою працює згорткова підсистема — вона застосовує фільтри, які "ковзають" по зображенню та виявляють найпростіші

закономірності: краї, кути, зміну текстури. На цьому рівні мережа ще не «розуміє», що саме зображено, але вже навчається розрізняти базові структурні елементи.

Далі працюють шари субдискретизації (пулінгу). Вони зменшують розмірність зображення, водночас зберігаючи найважливіші ознаки. Це робиться для зменшення обчислювальної складності, стабілізації навчання та формування інваріантності до невеликих зсувів або масштабних змін. Мережа поступово переходить від простих ознак до все більш складних — від ліній до патернів, від патернів до форми, від форми до окремих частин об'єкта.

Після проходження кількох рівнів згорток і пулінгу мережа вже сформувала глибокий набір високорівневих ознак. На цьому етапі вступають у роботу повнозв'язні шари. Вони діють як традиційний класифікатор: на основі отриманих ознак визначають, до якого класу може належати об'єкт. Ці шари працюють із абстрактними характеристиками, уже не з пікселями, і створюють внутрішню модель того, як виглядає об'єкт кожного класу.

У фіналі цього процесу мережа видає ймовірнісні оцінки для кожного класу. Це означає, що система не просто говорить «це об'єкт А», а визначає ступінь упевненості у кожному можливому варіанті. Такий підхід дозволяє гнучко реагувати на неоднозначні або складні випадки. Клас із найбільшою ймовірністю і є результатом класифікації.

Попри високу точність, результат мережі не сприймається як абсолютна істина. Він проходить додаткову перевірку, яка може включати логічний контроль, аналіз контексту, відсікання аномальних прогнозів або порівняння з порогоми достовірності. Якщо рівень упевненості недостатній, система може позначити об'єкт як невідомий або вимагати додаткові дані.

Коли ж модель демонструє достатній рівень точності та стабільності, вона інтегрується у реальні додатки. Це можуть бути системи безпеки, медичні засоби діагностики, автономні транспортні засоби, промислові роботи чи програмні інструменти аналітики. У таких системах CNN працює безперервно та автоматично, перетворюючи вхідні зображення на структуровану, осмислену

інформацію, яка допомагає людям приймати рішення або дозволяє техніці діяти самостійно.

Одностадійні детектори об'єктів — це підхід у комп'ютерному зорі, при якому процес виявлення і класифікації об'єктів виконується в єдиному проході нейронної мережі. Вони відмовляються від розділення задачі на два етапи (спочатку пошук регіонів, а потім класифікація), як це роблять двостадійні моделі на кшталт Faster R-CNN, і замість цього одразу прогнозують як координати рамок, так і класи кожного об'єкта. Такий підхід зменшує обчислювальну складність, робить систему набагато швидшою й дозволяє працювати у режимі реального часу навіть на слабших пристроях.

Ключовою ідеєю є те, що замість пошуку потенційних областей об'єктів модель обробляє увесь кадр одночасно. Зображення проходить через згорткову мережу, яка генерує багатовимірні карти ознак. На цих картах кожен осередок прогнозує можливу наявність об'єкта певного класу та відповідні координати рамки. Для цього використовуються попередньо визначені прив'язані рамки (anchors) або механізм прогнозування без якорів (anchor-free), де модель безпосередньо передбачає центри об'єктів та їхні розміри.



Рис. 2.3 Схема алгоритму методики на основі одностадійного детектору

Одностадійні детектори часто працюють на багатомасштабних картах ознак, завдяки чому вони можуть виявляти і великі, і дуже малі об'єкти. У раних моделях ці карти були фіксованими, але пізніше з'явилися архітектури з пірамідами ознак, де кожен рівень відповідає за свій масштаб. Такий підхід підвищує якість без значної втрати швидкості.

Під час тренування модель навчається одночасно двом задачам: класифікації та регресії рамок. Функції втрат комбінуються, щоб збалансувати їхній вплив. Оскільки одностадійні детектори роблять багато прогнозів одночасно, важливо правильно обробити фонові області та уникати дисбалансу між позитивними й негативними прикладами — для цього застосовуються спеціальні техніки, такі як фокальна функція втрат (Focal Loss), яка допомагає приділити більше уваги складним і рідкісним об'єктам.

На виході система генерує значну кількість рамок і ймовірностей, тому фінальний результат вимагає постобробки. Найчастіше використовується алгоритм нефінального придушення (NMS), який залишає лише найкращі і ненакладні рамки. Також можливе застосування калібрування ймовірностей або фільтрації за порогоми впевненості.

Одностадійні детектори стали популярними завдяки своїй швидкості, простоті та ефективності. Вони дозволяють реалізувати детекцію на мобільних пристроях, у вбудованих системах та автономних комплексах. Поява перших ефективних моделей, таких як SSD (2016) та YOLO (2015), стала важливою віхою, що забезпечила можливість працювати з відеопотоком у реальному часі й відкрила шлях до широкого застосування детекції в робототехніці, автомобільних системах, безпеці та військових технологіях.

Одностадійний підхід дав змогу перейти від повільних і громіздких систем до швидких, гнучких та універсальних моделей, які можуть працювати на різноманітному обладнанні й забезпечувати високу якість при значно меншій затримці. У сучасних системах саме одностадійні детектори найчастіше лежать в основі реальних застосувань комп'ютерного зору.

Двостадійний детектор об'єктів являє собою підхід у комп'ютерному зорі, який виконує розпізнавання у два послідовні, але логічно пов'язані етапи: генерацію регіонів-кандидатів і подальшу класифікацію цих регіонів із уточненням координат. Його поява була зумовлена бажанням поєднати високу точність класичних алгоритмів локалізації з можливостями глибоких згорткових мереж. Перший прорив у цьому напрямі зробила робота R-CNN Росса Гіршика та його колег у 2014 році, яка вперше застосувала глибоку нейронну мережу для обробки виділених регіонів. У подальші роки методика була суттєво вдосконалена: Fast R-CNN у 2015 році оптимізувала обчислення, поділивши екстракцію ознак для всього зображення, а не для кожного регіону окремо; той самий рік дав появу Faster R-CNN, де регіональні пропозиції почала генерувати власна нейронна мережа — RPN (Region Proposal Network), що фактично зробило підхід повністю нейромережевим.

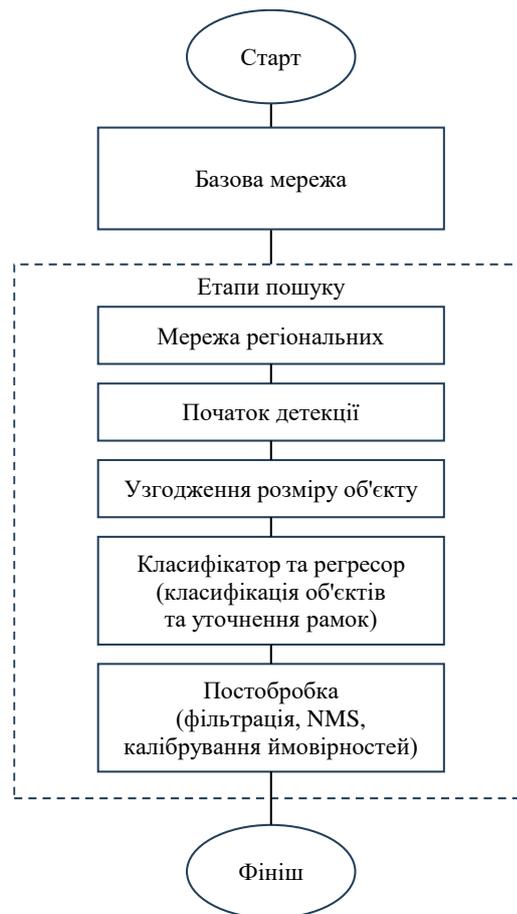


Рис. 2.4 Схема алгоритму методики на основі двостадійного детектору

Постановка задачі в двостадійному детекторі починається з того, що вхідне зображення проходить через базову згорткову мережу — зазвичай це глибока архітектура на кшталт VGG, ResNet або їхніх більш компактних модифікацій. Мережа витягує багатовимірні карти ознак, які відображають текстури, контури, форми та складні об'єктні патерни. На основі цих карт працює перша стадія — генерація регіонів-кандидатів. У класичному Faster R-CNN цю функцію виконує RPN, яка переглядає кожну позицію сітки ознак і на основі набору попередньо визначених "якорів" пропонує області, де потенційно може бути об'єкт. Ці пропозиції оцінюються на предмет «об'єкт / фон» і одночасно отримують первинні координати, що уже є грубою локалізацією.

Після генерації регіонів друга стадія детектора відповідає за точну класифікацію та уточнення меж об'єктів. Кожен запропонований регіон вирізається з карти ознак, але не шляхом прямого масштабування, а за допомогою спеціального механізму RoI Pooling або його вдосконаленої версії RoI Align, що дозволяє акуратно отримати фіксоване представлення області незалежно від її початкових розмірів. Ці фіксовані ознаки надходять у кінцеві повнозв'язні або згорткові прошарки, де визначається клас об'єкта, його точні координати та рівень упевненості. Уся система навчена кінцевим способом за допомогою функцій втрат, де одночасно зменшується похибка локалізації та похибка класифікації.

Алгоритм роботи двостадійного детектора можна уявити як послідовний рух інформації: зображення перетворюється у карту ознак; карта ознак генерує регіони-кандидати; регіони нормально вирівнюються; мережа класифікує, уточнює й відкидає зайві припущення за допомогою механізмів постобробки на кшталт NMS. У цій схемі перша стадія відповідає за пошук найбільш вірогідних місць появи об'єктів, а друга — за їх точний аналіз. Такий підхід забезпечує дуже високу точність розпізнавання і локалізації, хоча й має ціну у вигляді більших обчислювальних витрат порівняно з одностадійними методами. Завдяки гнучкій архітектурі й багатоступеневій обробці двостадійні детектори часто застосовують у системах, де точність критичніша за швидкодію, включно з медичними

аналізаторами, промисловою інспекцією, безпековими платформами та науковими дослідженнями.

## **2.2 Аналіз переваг та недоліків методик автоматизованого розпізнавання**

Методики на основі комп'ютерного зору, глибокого навчання та аналізу відеопотоку мають свої унікальні переваги.

Переваги методик, що базуються на комп'ютерному зорі, полягають у їхній здатності інтерпретувати зображення за формальними ознаками без необхідності у великих масивах даних. Класичні алгоритми, такі як аналіз контурів, сегментація, фільтри Габора та методи на основі градієнтів, дозволяють швидко й ефективно виділяти геометричні та текстурні характеристики. Їхньою ключовою перевагою є пояснюваність: кожен крок алгоритму зрозумілий, чіткий та контрольований. Такі методики добре працюють там, де умови контрольовані, а об'єкти мають передбачувані форми, контури або структурні закономірності. Крім того, вони потребують значно менше обчислювальних ресурсів, можуть працювати на слабких пристроях та легко адаптуються до специфічних завдань через точне налаштування параметрів.

Переваги методик на основі глибокого навчання, особливо згорткових нейронних мереж (CNN), випливають із їхньої здатності самостійно витягувати високорівневі ознаки без попереднього ручного програмування. Завдяки цьому CNN демонструють значно вищу точність, оскільки навчання базується на реальних прикладах, а не на спрощених математичних моделях. Їхньою головною сильною стороною є універсальність: вони здатні розпізнавати складні та нестабільні об'єкти, працювати у різних середовищах та адаптуватися до мінливих умов. З часом моделі стають тільки кращими, оскільки зростання наборів даних та вдосконалення архітектур (ResNet, EfficientNet, Vision Transformers) посилюють їхню здатність до узагальнення. CNN добре справляються з задачами,

де класичний комп'ютерний зір неефективний: низька якість зображення, змінне освітлення, часткове закриття об'єктів, складні текстури чи асиметричні форми.

Переваги методик, що працюють з відеопотоком, пов'язані з аналізом динаміки та контексту. На відміну від аналізу одиничного знімка, відео дає змогу відстежувати рух, зміну положення, швидкість та взаємодію об'єктів у просторі. Такий підхід дозволяє суттєво зменшувати кількість помилкових спрацьовувань і підвищувати надійність розпізнавання завдяки часовій послідовності кадрів. Для об'єктів, які можуть з'являтися та зникати, або змінюють свою форму у русі, аналіз відеопотоку забезпечує контекст, що значно покращує інтерпретацію. Крім того, методики роботи з відео дозволяють будувати складні системи відстеження, прогнозування траєкторій та аналітики поведінки. Це особливо важливо для безпекових систем, автономного транспорту, військових застосувань і глобального моніторингу. У поєднанні з нейронними мережами відеоаналіз стає інструментом, який не просто „бачить”, а й „розуміє” події в реальному часі.

Недоліки методики на основі комп'ютерного зору мають суттєві обмеження, пов'язані насамперед із чутливістю до якості вхідних даних. Вони можуть некоректно працювати при низькій роздільній здатності зображень, сильному шумі, зміні освітлення або нестандартних умовах, що часто зустрічається в реальних сценаріях. Крім того, традиційні алгоритми комп'ютерного зору потребують ручного виділення ознак, тобто інженеру доводиться визначати, які характеристики зображення важливі для аналізу. Це значно ускладнює адаптацію системи до нових типів завдань або об'єктів. Також такі методи обмежено узагальнюють інформацію, тому можуть помилково розпізнавати об'єкти при зміні їхньої форми, орієнтації або фону.

Методики на основі глибокого навчання, зокрема CNN, здатні ефективно виділяти складні ознаки зображень, але водночас вони вимагають великих обсягів даних для навчання і значних обчислювальних ресурсів. Процес навчання моделей є тривалим і енергомістким, а результати часто важко інтерпретувати, що створює проблему прозорості рішень у критичних сферах, таких як медицина або безпека. CNN також схильні до перенавчання, коли навчальна вибірка не репрезентує всі

можливі варіанти, і можуть давати помилкові результати при зміні умов або появі нових типів об'єктів. Крім того, моделі іноді надмірно реагують на дрібні деталі, які не важливі для задачі, що знижує їхню стабільність.

Методики на основі аналізу відеопотоку стикаються з проблемами високих вимог до обчислювальних ресурсів та пропускної здатності. Обробка відео в реальному часі є складною, особливо якщо камера високо роздільна або встановлено багато камер одночасно. Алгоритми відстеження об'єктів можуть помилятися при перекритті об'єктів, швидкому чи хаотичному русі, змінах освітлення або погодних умовах. Крім того, системи часто потребують тонкого налаштування під конкретні сценарії, що обмежує їх універсальність і ускладнює масштабування на різні середовища.

Отже, підсумовуючи усі переваги та недоліки вищевказаних методик можемо скласти таку порівняльну таблицю:

Таблиця 2.1

## Аналіз переваг та недоліків методик

Методика	Переваги	Недоліки
Комп'ютерний зір (Computer Vision)	Швидка обробка зображень у реальному часі, автоматичне виділення патернів і аномалій, зменшення ручної роботи	Чутливість до якості даних (шум, освітлення, низька роздільна здатність), потреба у ручному конструюванні ознак, обмежене узагальнення
Глибоке навчання (CNN)	Автоматичне виділення ознак, висока точність розпізнавання навіть у складних умовах, здатність узагальнювати та працювати з великими наборами даних	Потреба у великих обсягах даних та потужних обчислювальних ресурсах, складність інтерпретації рішень, схильність до перенавчання, нестабільність при зміні умов
Аналіз відеопотоку	Виявлення динамічних процесів, контекстний аналіз ситуацій, можливість інтеграції даних з кількох камер	Високі вимоги до ресурсів та пропускної здатності, помилки відстеження при перекритті або швидкому русі, потреба у тонкому налаштуванні під конкретні сценарії

### 3. РОЗРОБКА ГІБРИДНОЇ МЕТОДИКИ АВТОМАТИЗОВАНОГО РОЗПІЗНАВАННЯ СТРІЛЕЦЬКОЇ ЗБРОЇ НА КЛАСИФІКАЦІЇ

#### 3.1 Засоби розробки, бібліотеки та мови програмування, вимоги до них

Python є однією з найвпливовіших мов програмування сучасності, і його історія безпосередньо пов'язана з розвитком наукових обчислень, автоматизації та штучного інтелекту. Мову створив Гвідо ван Россум у 1989 році в Центрі математики та інформатики (CWI, Нідерланди). Первинною метою розробника було створення мови, яка поєднає в собі простоту читання, мінімалізм синтаксису та високу продуктивність розробки. Уже у 1991 році з'явився перший публічний реліз Python, який привернув увагу завдяки своїй інтуїтивній структурі та потужним абстракціям.



Рис. 3.1 Мова програмування Python

Python з самого початку проєктувався як інтерпретована, динамічно типізована мова, що дозволяє працювати з об'єктами, структурами даних і функціональними конструкціями без надлишкової системної складності. Завдяки цьому він став основою у сферах, де важливі швидка розробка, гнучкість та експресивність коду. З часом мова набула широкої підтримки в академічних

та наукових колах. Розвиток бібліотек NumPy, SciPy, matplotlib, а потім і фреймворків глибокого навчання TensorFlow, PyTorch та Keras зробив Python ключовим інструментом для машинного навчання, моделювання нейронних мереж та обчислювальної математики.

Python активно застосовується для написання алгоритмів комп'ютерного зору, зокрема через інтеграцію з OpenCV, MediaPipe та сучасними моделями на основі глибинних CNN. Його архітектура дозволяє ефективно поєднувати високорівневий код із низькорівневими оптимізованими модулями на C/C++ або CUDA, що критично важливо для обчислень у нейронних мережах, операцій згортки, лінійної алгебри та GPU-прискорення.

Мова також використовується в робототехніці, системах автоматичного керування, великих веб-платформах, аналітиці даних, створенні симуляцій, прогнозуванні, кібербезпеці та автоматизації інженерних процесів. Простота синтаксису забезпечує швидке прототипування моделей, а розвинена екосистема пакетів дозволяє реалізовувати складні наукові концепції з мінімальною кількістю коду. Це робить Python не просто мовою програмування, а фундаментальним інструментом науково-технічного прогресу, особливо в епоху інтенсивного розвитку штучного інтелекту й автоматизованих систем розпізнавання.

```

20
21 response = self.client.get(reverse('polls:index'))
22 self.assertEqual(response.status_code, 200)
23 self.assertContains(response, "No polls are available.")
24 self.assertQuerysetEqual(response.context['latest_question_list'], [])
25
26 @test
27 def test_index_view_with_a_future_question(self):
28     """
29     test_index_view_with_future_question_and_past_question
30     test_index_view_with_no_questions
31     test_index_view_with_two_past_questions
32 """
33 cr = unittest.TestCase
34 re = unittest.TestCase
35 se = unittest.TestCase
36 de = unittest.TestCase
37
38 def test_index_view_with_a_future_question(self):
39     """
40     Questions with a pub_date in the future should not be displayed on
41     the index page.
42 """
43 create_question(question_text="Future question.", days=30)
44 response = self.client.get(reverse('polls:index'))
45 self.assertContains(response, "No polls are available.")
46 self.assertEqual(response.status_code, 200)
47 self.assertQuerysetEqual(response.context['latest_question_list'], [])
48
49 def test_index_view_with_future_question_and_past_question(self):
50     """
51     Even if both past and future questions exist, only past questions
52     should be displayed.
53 """
54 create_question(question_text="Past question.", days=-30)
55 create_question(question_text="Future question.", days=30)
56 response = self.client.get(reverse('polls:index'))
57 self.assertQuerysetEqual(
58     response.context['latest_question_list'],
59     ['<question: Past question.>']
60 )
61
62 def test_index_view_with_two_past_questions(self):
63
64
65 """ Statement seems to have no effect. Unresolved attribute reference 'test' for class 'QuestionViewTests' """

```

Рис. 3.2 Середовище розробки PyCharm

PyCharm — це інтегроване середовище розробки Python, створене компанією JetBrains у 2010 році. Воно виникло як професійний інструмент для розробників, яким потрібна була не просто текстова редакторська підтримка Python-коду, а повноцінна платформа з аналітикою, статичним аналізом, навігацією проєктами, системою запуску, налагодженням та інструментами для роботи з фреймворками. Уже з перших версій PyCharm відзначався глибокою інтеграцією з механізмами інспекції коду JetBrains, що дозволило програмістам бачити помилки, попередження, недоліки стилю та потенційні проблеми ще до запуску програми.

PyCharm складається з декількох ключових модулів, які роблять його одним із найпотужніших IDE для Python. Його основою є розвинений парсер Python AST, що дозволяє аналізувати структуру коду, будувати індекси, визначати типи змінних та підказувати автодоповнення з урахуванням контексту. Завдяки цьому PyCharm підтримує складні середовища, включно з проєктами на основі Django, Flask, FastAPI, а також застосунки з машинного та глибокого навчання. Інструменти роботи з Jupyter Notebook і віртуальними середовищами дають змогу організовувати наукові експерименти, тестувати моделі нейронних мереж і контролювати залежності різних проєктів.

Розробники PyCharm інтегрували у середовище потужний відлагоджувач, який підтримує покрокове виконання, брейкпоїнти, аналіз значень змінних у реальному часі та навіть роботу з асинхронним кодом. Для аналітиків і фахівців зі штучного інтелекту особливо важливими є вбудовані інструменти відлагодження у віртуальних середовищах та підтримка наукового стека — NumPy, SciPy, Matplotlib, TensorFlow, PyTorch. PyCharm також має механізм інтеграції з Git, GitHub, GitLab, Bitbucket, що робить його зручним вибором для командних проєктів і колаборативної розробки.

Середовище надає інструменти для автоматичного форматування коду, аналізу продуктивності, роботи з базами даних, а також можливість запускати завдання безпосередньо з IDE — від локальних скриптів до Docker-контейнерів і віддалених серверів. Підтримка підключення через SSH дозволяє тренувати нейронні мережі на кластерних GPU-серверах без виходу з середовища розробки.

PyCharm використовується у наукових центрах, університетах, дослідницьких лабораторіях і комерційних компаніях, особливо там, де Python є основною мовою для побудови штучного інтелекту та обробки даних. Завдяки своїй інтелектуальній системі аналізу, гнучкості та інтегрованим інструментам PyCharm став стандартом де-факто серед професійних Python-розробників, що працюють над масштабованими моделями, складними алгоритмами та високонавантаженими системами комп'ютерного зору.

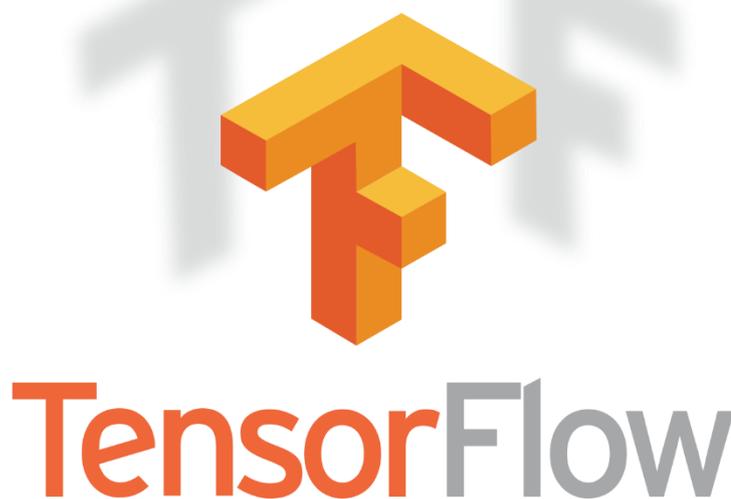


Рис. 3.3 Програмна бібліотека TensorFlow

TensorFlow — це високорівнева та водночас низькорівнева науково-обчислювальна платформа для побудови, навчання та розгортання моделей машинного й глибокого навчання, створена дослідниками Google Brain у 2015 році. Її поява була частиною еволюції інструментів для машинного навчання: спочатку Google створив DistBelief (2011), внутрішню систему для масштабованого глибинного навчання, але вона була надто жорсткою та монолітною. TensorFlow став наступним поколінням — відкритим, модульним та універсальним інструментом, орієнтованим на наукові дослідження, розвиток інтелектуальних систем і масове виробниче використання.

TensorFlow базується на концепції обчислювальних графів, у яких кожна операція є вузлом, а дані у вигляді тензорів рухаються між цими вузлами. Тензори — це багатовимірні масиви, що дозволяють ефективно описувати будь-які математичні об'єкти, необхідні в машинному навчанні: матриці ваг, набори зображень, послідовності, об'ємні карти ознак. Така структура дозволяє платформі оптимізувати виконання моделі на різних типах апаратури: CPU, GPU, TPU, а також у вбудованих системах і мобільних пристроях.

З наукової точки зору TensorFlow став одним із перших доступних інструментів, який дозволив виконувати автоматичне диференціювання у складних модельних архітектурах. Саме завдяки цьому стало можливим легко будувати та тренувати глибокі нейронні мережі з багатьма сотнями шарів, складними механізмами з'єднань і нелінійностями. Механізм автограду в TensorFlow аналізує граф обчислень і автоматично формує градієнти для подальшої оптимізації за допомогою алгоритмів типу Stochastic Gradient Descent, Adam, RMSProp та інших сучасних методів.

TensorFlow підтримує широкий спектр наукових задач: комп'ютерний зір, обробку природної мови, генеративні моделі, прогнозування часових рядів, оптимізаційні та симуляційні завдання. Це стало можливим завдяки модульності платформи та величезній кількості вбудованих математичних операцій, оптимізованих під паралельні обчислення. Важливою частиною його екосистеми є TensorFlow Extended (TFX) для побудови виробничих ML-пайплайнів, TensorFlow Lite для мобільних і вбудованих систем, а також TensorFlow Serving, що дозволяє деплоїти моделі з високою пропускнуою здатністю.

Наукове та технічне значення TensorFlow полягає в тому, що він суттєво спростив експериментування з нейронними мережами: від простих CNN до складних трансформерних архітектур, від класичних автокодерів до сучасних генеративних моделей. Платформа підтримує роботу з тензорними операціями на низькому рівні, але одночасно має високорівневі API, такі як Keras, які дозволяють швидко створювати моделі з мінімальним кодом. З переходом до TensorFlow 2.x модель стала більш дружньою до наукових досліджень: з'явився

режим `eager execution`, що дозволяє виконувати операції негайно, без необхідності попереднього складання графу.

TensorFlow перетворився на один із ключових інструментів сучасної науки про дані, ставши платформою, що дозволяє як дослідникам, так і інженерам працювати з нейронними мережами на рівні від фундаментальних математичних операцій до розгортання високопродуктивних промислових систем штучного інтелекту.



Рис. 3.4 Датасет COCO

COCO (Common Objects in Context) — це один із найвідоміших і найбільш уживаних відкритих датасетів, призначений для навчання, тестування та порівняння алгоритмів автоматизованого розпізнавання об'єктів, сегментації зображень і аналізу сцен. Датасет був створений дослідниками з Microsoft Research, Каліфорнійського університету в Берклі та інших наукових установ і вперше представлений у 2014 році в межах наукової ініціативи, спрямованої на подолання обмежень попередніх наборів даних, таких як PASCAL VOC чи ImageNet Detection. Основною ідеєю COCO стало моделювання «реальних сцен», у яких одночасно присутні декілька об'єктів різних класів у складному контексті, з частковими перекриттями, різними масштабами та фонами.

COCO застосовується як еталонний датасет у наукових дослідженнях і промислових розробках, пов'язаних із задачами `object detection`, `instance segmentation`, `semantic segmentation`, `keypoint detection` та `image captioning`. Його активно використовують для навчання та оцінювання моделей на основі глибокого

навчання, зокрема Faster R-CNN, YOLO, SSD, RetinaNet, Mask R-CNN, EfficientDet і MobileNet-SSD. У науковому середовищі COCO став стандартом для порівняння якості алгоритмів, а в прикладних системах — базою для попереднього навчання моделей, які потім донавчаються під конкретні прикладні задачі, включно з відеоспостереженням, автономними системами, робототехнікою та системами безпеки.

З технічної точки зору COCO є масштабним і добре структурованим набором даних. Він містить сотні тисяч зображень, поділених на тренувальні, валідаційні та тестові підмножини. Основна версія датасету охоплює понад 80 категорій об'єктів повсякденного використання, таких як люди, транспорт, тварини, побутові предмети та інструменти. Кожне зображення супроводжується докладною розміткою у форматі JSON, що включає координати обмежувальних рамок (bounding boxes), точні контури об'єктів у вигляді полігонів для instance segmentation, а також додаткові атрибути, такі як площа об'єкта, рівень складності та інформація про перекриття з іншими об'єктами.

Важливою технічною особливістю COCO є підтримка багаторівневої анотації. Для детекції об'єктів використовуються прямокутні рамки, для сегментації — точні маски на рівні пікселів, а для задачі визначення поз людини — ключові точки скелета. Така різноманітність розмітки дозволяє будувати універсальні моделі, здатні одночасно виконувати декілька задач комп'ютерного зору. Формат даних оптимізований для використання в популярних фреймворках глибокого навчання, таких як TensorFlow, PyTorch та їхніх високорівневих бібліотеках.

Окрему роль у COCO відіграє система метрик оцінювання. Замість простих показників точності використовується середня точність (Average Precision, AP), обчислена для різних порогів перекриття IoU (Intersection over Union). Це дозволяє більш об'єктивно оцінювати не лише факт виявлення об'єкта, а й точність локалізації. Такий підхід суттєво вплинув на розвиток алгоритмів детекції, стимулюючи створення моделей, які краще працюють із дрібними об'єктами та складними сценами.

MobileNet — це сімейство легковагових згорткових нейронних мереж, розроблених для ефективної роботи на пристроях з обмеженими обчислювальними ресурсами. Архітектура MobileNet була представлена у 2017 році дослідниками Google (Ендрю Говард та ін.) у науковій роботі “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications”. Основною метою створення MobileNet було поєднання достатньо високої точності комп’ютерного зору з мінімальними вимогами до пам’яті, обчислювальної потужності та енергоспоживання.

MobileNet широко використовується у задачах автоматизованого розпізнавання, таких як класифікація зображень, детекція об’єктів, сегментація, аналіз відеопотоку, а також у вбудованих системах, мобільних застосунках, безпілотних платформах і системах реального часу.

Ключовою ідеєю MobileNet є використання глибинно-роздільних згорток (depthwise separable convolutions) замість класичних згорткових шарів. У традиційній згортковій нейронній мережі кожен фільтр одночасно виконує просторову та каналну обробку, що призводить до значної кількості параметрів і операцій множення. У MobileNet ця операція розділяється на два послідовні етапи. Спочатку застосовується depthwise-згортка, яка виконує просторову фільтрацію окремо для кожного каналу вхідного тензора. Потім використовується pointwise-згортка з ядром  $1 \times 1$ , яка поєднує інформацію між каналами. Такий підхід різко зменшує обчислювальну складність і кількість параметрів, зберігаючи при цьому здатність мережі виділяти інформативні ознаки.

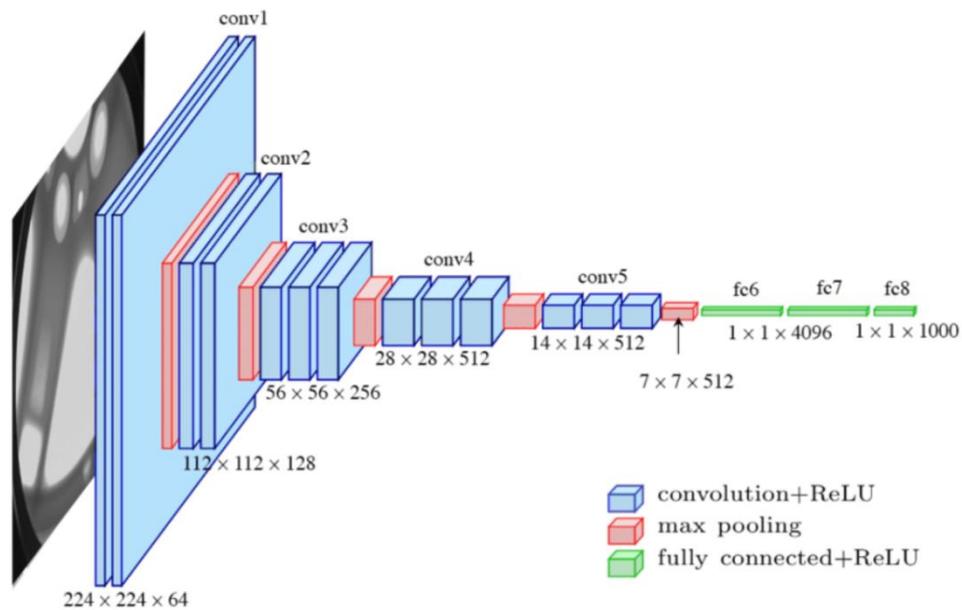


Рис. 3.5 Архітектура MobileNet

Архітектурно MobileNet складається з послідовності глибинно-роздільних згорткових блоків, кожен з яких включає depthwise convolution, нелінійну активацію (зазвичай ReLU або ReLU6), нормалізацію (Batch Normalization) і pointwise convolution. Початкові шари мережі навчаються виявляти прості низькорівневі ознаки, такі як краї та текстури, тоді як глибші шари формують абстрактні високорівневі представлення об'єктів. Наприкінці архітектури зазвичай використовується глобальний середній пулінг і повнозв'язний або  $1 \times 1$ -згортковий шар для отримання фінальних передбачень.

Важливою особливістю MobileNet є можливість керування компромісом між точністю та продуктивністю за допомогою двох гіперпараметрів. Перший — width multiplier, який зменшує або збільшує кількість каналів у кожному шарі, контролюючи ширину мережі. Другий — resolution multiplier, який задає розмір вхідного зображення, впливаючи на кількість обчислень у просторовому вимірі. Завдяки цьому MobileNet легко адаптується під конкретні апаратні обмеження або вимоги реального часу.

З наукової точки зору MobileNet є прикладом оптимізації глибоких нейронних мереж шляхом факторизації згорткових операцій, що дозволяє суттєво

знизити обчислювальну складність без радикальної втрати точності. Цей підхід вплинув на подальший розвиток компактних архітектур та інші моделі, орієнтовані на ефективність. Таким чином, MobileNet відіграє важливу роль у сучасних системах автоматизованого розпізнавання, поєднуючи теоретичні досягнення глибокого навчання з практичними вимогами реальних застосувань.

### 3.2 Опис розробки методики

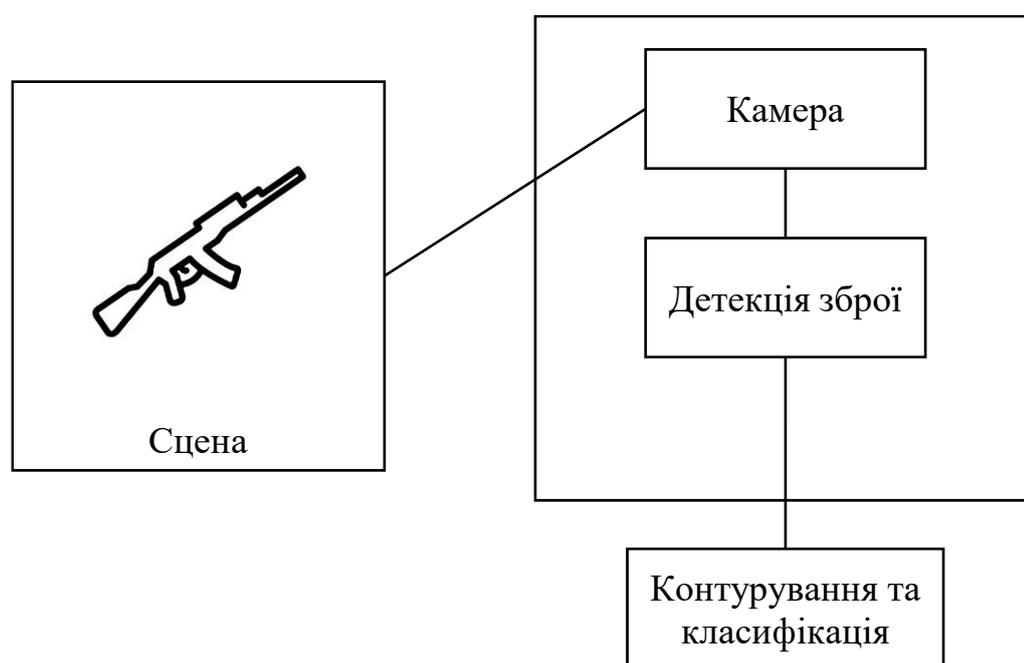


Рис. 3.6 Схема процесу автоматизованого розпізнавання зброї

На зображенні подано узагальнений процес автоматизованого розпізнавання об'єктів (зокрема зброї), який відображає послідовність перетворення візуальної інформації від реальної сцени до отримання семантично інтерпретованого результату.

Процес починається зі сцени, в якій присутній об'єкт інтересу — у даному випадку стрілецька зброя. Сцена є реальним фізичним середовищем, що містить різні об'єкти, освітлення, фони та можливі завади. На цьому етапі інформація ще не оцифрована й існує у вигляді світлового відбиття від об'єктів.

Далі використовується камера, яка виконує функцію сенсора. Вона перетворює оптичну інформацію зі сцени на цифрове зображення або відеопотік. Камера фіксує просторові та колірні характеристики сцени, формуючи матрицю пікселів, що є вхідними даними для системи комп'ютерного зору. Якість цього етапу істотно впливає на точність подальшого аналізу, оскільки роздільна здатність, шум та умови освітлення визначають інформативність даних.

Наступним етапом є детекція зброї. На цьому рівні застосовуються алгоритми автоматизованого розпізнавання, зазвичай побудовані на основі глибоких згорткових нейронних мереж. Система аналізує вхідне зображення або кадр відео з метою виявлення областей, у яких імовірно присутня зброя. Результатом цього етапу є визначення координат потенційного об'єкта (обмежувальні рамки) та оцінка ймовірності його наявності. Детекція зосереджується саме на пошуку об'єкта, а не на його детальній інтерпретації.

Після цього виконується етап контурування та класифікації. Контурування полягає в уточненні форми виявленого об'єкта, виділенні його меж та просторової структури, що дозволяє відокремити об'єкт від фону. Класифікація, у свою чергу, визначає, до якого класу належить знайдений об'єкт, наприклад тип або категорія зброї. На цьому етапі система переходить від простого виявлення до семантичного розуміння об'єкта.

Для реалізації вищевказаного задуму буде запропонована гібридна методика, що поєднує в собі застосування одностадійного детектору – Single Shot Detector, як засіб детекції та легковагову згорткову нейронну мережу – MobileNet, що дозволяє суттєво знизити обчислювальну складність без радикальної втрати точності.

Отже, складемо алгоритм роботи запропонованої методики, для її подальшої програмної реалізації.

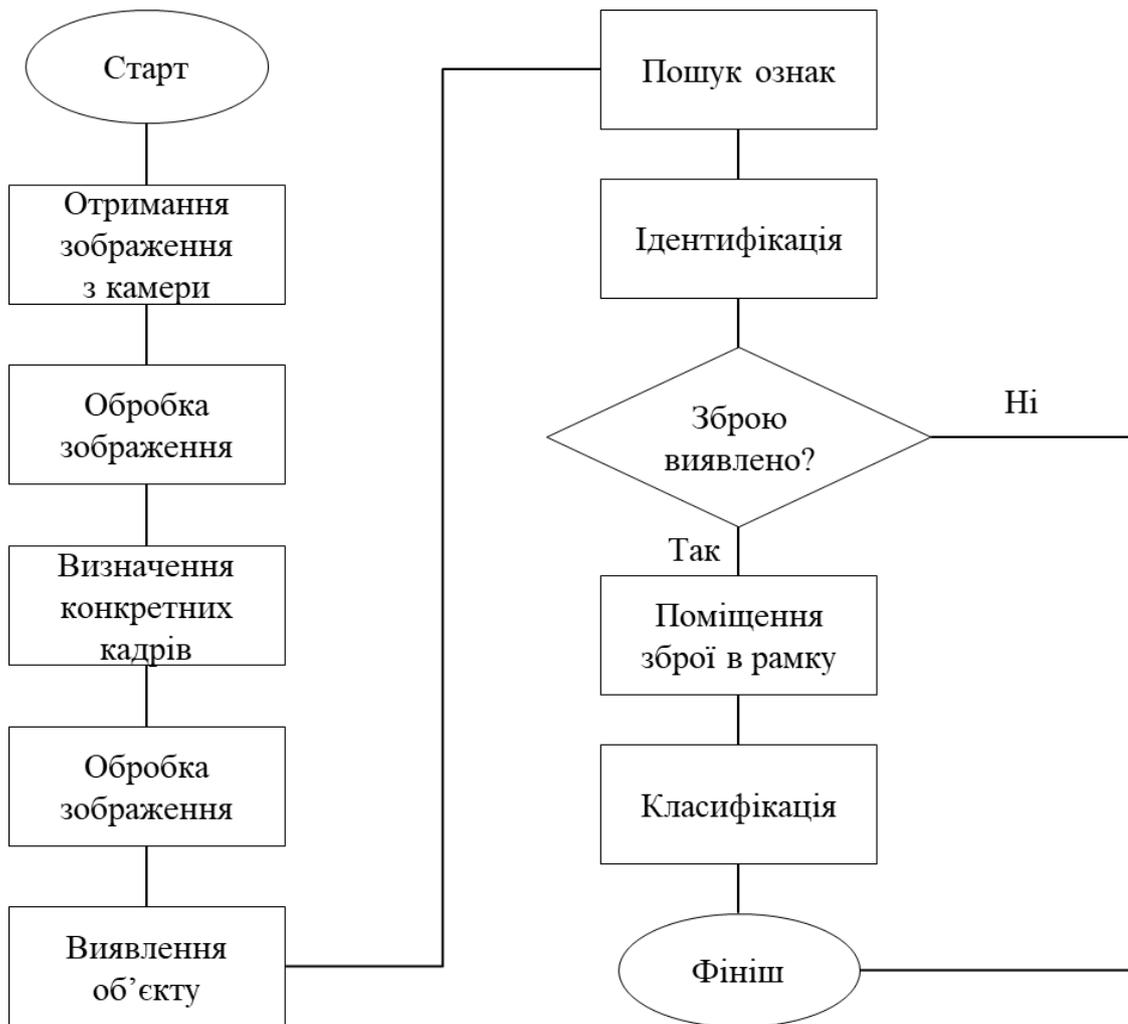


Рис. 3.7 Алгоритм автоматизованого розпізнавання стрілецької зброї

Згідно з алгоритмом, коли система починає працювати, вона спочатку завантажує відео з камер спостереження, зняте камерою. Але відео спостереження зазвичай містить багато безкорисних фрагментів. В результаті це призводить до збільшення обсягу обчислень та уповільнення швидкості виявлення. Таким чином, система не переходить до кожного кадру для виявлення відео. Натомість, система прийматиме зображення дійсного кадру як вхідні дані, щоб пришвидшити виявлення.

Коли зображення вводиться в систему, система виконує перетворення у градації сірого, щоб видалити фонові елементи зображення та перетворити його на чорно-біле. Після завершення обробки система виявляє об'єкти на зображенні. За допомогою алгоритму детектора Single Shot MultiBox (SSD) MobileNet система може витягувати ознаки з об'єктів.

Після вилучення ознаки система може вирішити проблему ідентифікації та визначити, до якої категорії належить об'єкт. Коли об'єкт виявляється як належний до зброї, система знаходить та маркує зброю.

### 3.3 Програмна реалізація та результати порівнянь

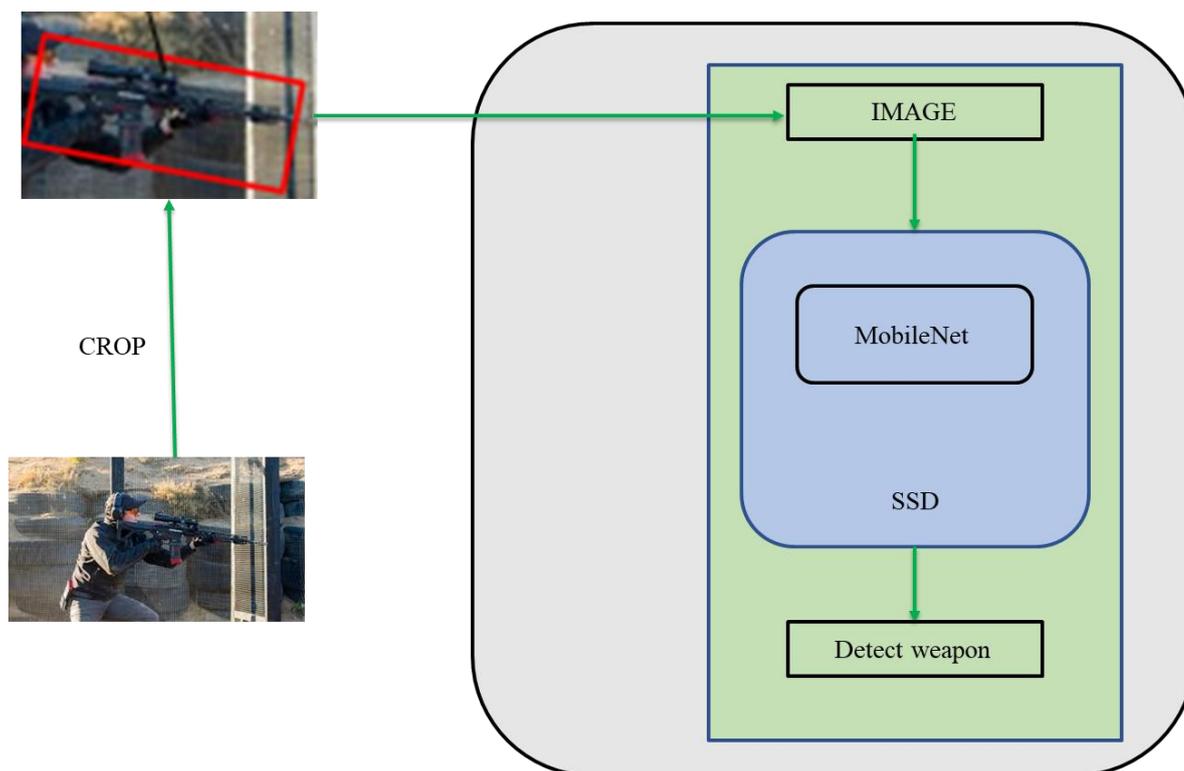


Рис. 3.8 Послідовність дій запропонованої методики

На рисунку показано послідовність дій детектора об'єктів. Спочатку система отримує зображення, потім обрізає це зображення та вводить його в систему виявлення об'єктів. Після цього зображення надходить до моделі SSD-MobileNet, також через TensorFlow, параметри розгортаються в навченій моделі в SSD-MobileNet. Нарешті, буде отримано результат виявлення зброї.

У цьому проєкті нейронна мережа навчається за допомогою контрольованого класифікаційного навчання у два основні кроки:

Перший – створюється набір даних з набору даних COCO для кожної зброї.

Другий – використовуючи вищезазначений набір даних, навчається детектор зброї шляхом точного налаштування існуючої попередньо навченої моделі.

Проект набору даних СОСО - це велика візуальна база даних для дослідження програмного забезпечення для візуального розпізнавання об'єктів. У цьому проекті дані - це зображення зброї з набору даних СОСО. Результатом точного налаштування є ефективніша модель, яка може виявляти більше видів зброї та підвищувати її ефективність.

Отже, завантажуюмо базові зображення стрілецької зброї для подальшого використання їх у програмі.

Для коректної роботи TensorFlow нам потрібно ввести специфічний формат даних Google, тому нам потрібно конвертувати наші дані у формат TFRecords. Використовуємо `gun_xml_to_csv` для конвертації XML-файлу у CSV-файл, потім генеруємо файл TFRecord з CSV-файлу за допомогою `generate_TRF`. Створюємо файл для карти позначок, що містить назву зброї та відповідний ідентифікатор.

```
item {
  name: "Sniper rifle"
  id: 1
}
item {
  name: "Automatic Rifle"
  id: 2
}
item {
  name: "Submachine gun"
  id: 3
}
item {
  name: "Shotgun"
  id: 4
}
item {
  name: "Handgun"
  id: 5
}
```

Рис. 3.9 Програмна реалізація у середовищі розробки

Назва та ідентифікатор зброї в `generate_TRF` повинні узгоджуватися з картою міток.

```
def class_text_to_int(row_label):
    if row_label == 'Sniper rifle':
        return 1
    elif row_label == 'Automatic Rifle':
        return 2
    elif row_label == 'Submachine gun':
        return 3
    elif row_label == 'Shotgun':
        return 4
    elif row_label == 'Handgun':
        return 5
```

Рис. 3.10 Програмна реалізація у середовищі розробки

Завантажуємо та підключаємо необхідні програмні засоби.

```
train_input_reader: {
  tf_record_input_reader {
    input_path: "PATH_TO_BE_CONFIGURED/\\models\\research\\object_detection\\data\\gun_train.record"
  }
  label_map_path: "PATH_TO_BE_CONFIGURED/\\models\\research\\object_detection\\data\\gun.pbtxt"
}
```

Рис. 3.11 Програмна реалізація у середовищі розробки

```
eval_input_reader: {
  tf_record_input_reader {
    input_path: "PATH_TO_BE_CONFIGURED/\\models\\research\\object_detection\\data\\gun_test.record"
  }
  label_map_path: "PATH_TO_BE_CONFIGURED/\\models\\research\\object_detection\\data\\gun.pbtxt"
  shuffle: false
  num_readers: 1
}
```

Рис. 3.12 Програмна реалізація у середовищі розробки

Після попереднього навчання на заздалегідь завантажених моделях, є необхідність протестувати систему, та скласти порівняльну таблицю між запропонованою методикою та існуючими методами автоматизованого розпізнавання стрілецької зброї на класифікації.

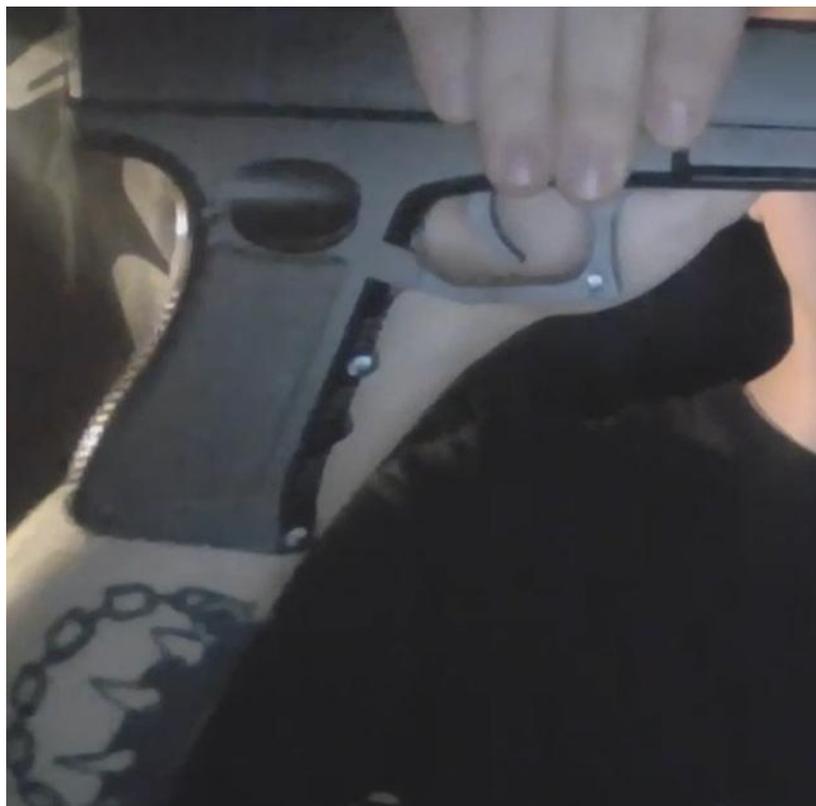


Рис. 3.13 Демонстрація практичних результатів



Рис. 3.14 Демонстрація практичних результатів



Рис. 3.15 Демонстрація практичних результатів

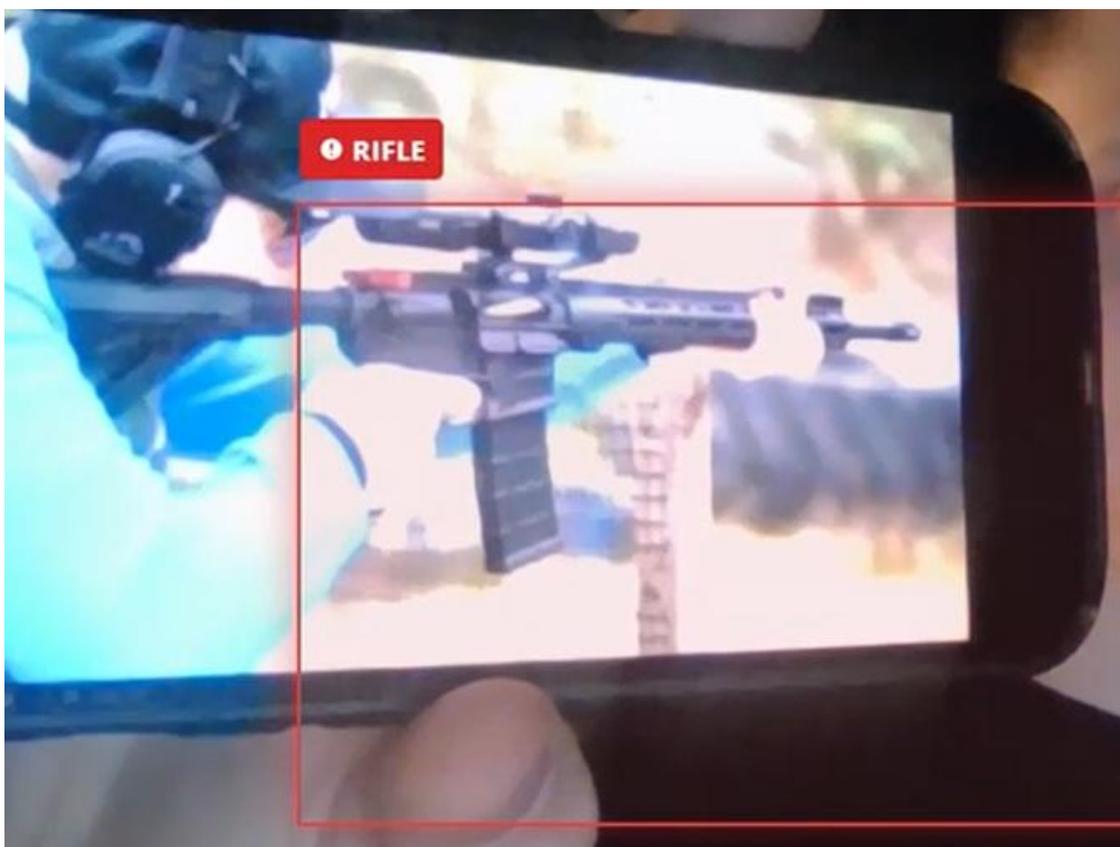


Рис. 3.16 Демонстрація практичних результатів

Враховуючи отримані результати складаємо діаграми порівняння з існуючими аналогами.

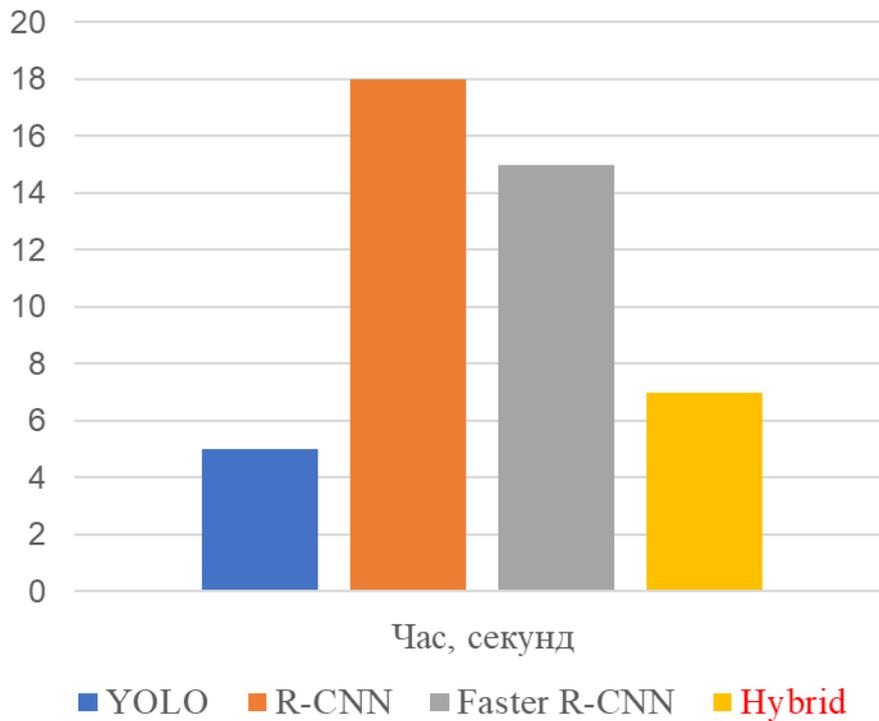


Рис. 3.17 Діаграма порівняння аналогів за часовим показником

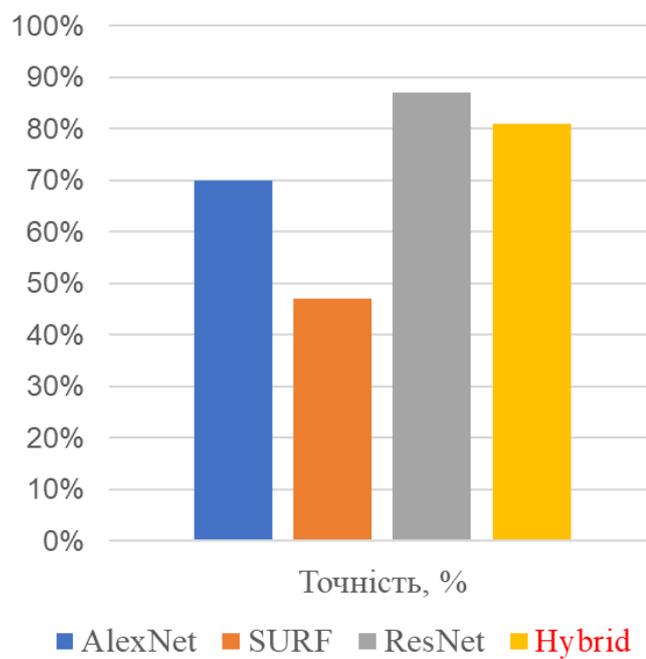


Рис. 3.18 Діаграма порівняння аналогів за показником точності

Під час випробувань особлива увага приділялась здатності системи розпізнавати об'єкти в умовах шуму, нечіткої картинки або зв'язку, відбивання або скаження світла від поверхностей та об'єктів та в умовах поганого освітлення, що дозволило нам переконатися та перевірити систему в стійкості до умовних перешкод.

З урахуванням отриманих результатів тестування можна зробити висновок, що розроблена гібридна методика має потенціал для подальшого впровадження в реальних умовах експлуатації.

## ВИСНОВКИ

У процесі виконання дипломної роботи була здійснена розробка системи автоматизованого розпізнавання об'єктів, а саме стрілецької зброї, з використанням методики поєднання машинного навчання та детектора об'єктів. Запропонована методика демонструє високу актуальність, оскільки дозволяє автоматично виявляти та класифікувати об'єкт озброєння на зображенні або відеопотоці без постійного залучення уваги людини, що є надзвичайно важливим для підвищення рівня безпеки, зменшення часу реагування та мінімізації людського фактору в критичних ситуаціях.

Автоматизоване розпізнавання стрілецької зброї є необхідним у сферах відеоспостереження, охорони громадського порядку, військових і правоохоронних застосувань, а також у системах моніторингу стратегічних об'єктів. Його використання дає змогу оперативно виявляти потенційні загрози, аналізувати великі обсяги візуальних даних у режимі реального часу та підтримувати прийняття рішень на основі об'єктивних алгоритмічних результатів.

Практична цінність розробленого підходу полягає у можливості інтеграції з сучасними камерами, безпілотними літальними апаратами та іншими сенсорними системами. Застосування згорткових нейронних мереж і сучасних детекторів об'єктів забезпечує достатню точність класифікації навіть за складних умов освітлення, часткових перекриттів або різних ракурсів зйомки. До ключових переваг таких систем належать масштабованість, адаптивність до нових типів даних, можливість подальшого навчання та оптимізації, а також висока швидкодія при обробці зображень і відео.

Тестуванням підтверджено високу точність та оптимальний час роботи системи при практичному використанні. Запропонована система має всі завдатки та перспективи розвитку, деталізації та вдосконалення, оскільки актуальність порушеного питання є високою. Проведене дослідження виконує практичну задачу вдосконалення систем автоматизації та класифікації, а отримані результати можуть

бути використані у практичних військових застосуваннях для підвищення ефективності та безпеки військових дій.

Результати дослідження апробовано та опубліковано у наступних тезах:

1. Гребещенко В.Е., Садовенко В.С. Сучасні інтелектуальні технології в Україні і світі з використанням мови програмування Java Script.

V Всеукраїнська науково-практична конференція «Сучасні інтелектуальні інформаційні технології в науці та освіті», 15 травня 2025 р., Збірник тез. К.: ДУІКТ, 2025. С. 193-194.

2. Гребещенко В.Е., Садовенко В.С. Новітні апаратні та програмні засоби інтелектуальних технологій у військовій топографії.

V Всеукраїнська науково-практична конференція «Сучасні інтелектуальні інформаційні технології в науці та освіті», 15 травня 2025 р., Збірник тез. К.: ДУІКТ, 2025. С. 271-273.

3. Гребещенко В.Е., Садовенко В.С. Парадигми програмної інженерії в автоматизації розпізнавання об'єктів.

II Всеукраїнська науково-технічна конференція «Виклики та рішення в програмній інженерії», 26 листопада 2025 р. Подано до друку.

4. Гребещенко В.Е., Садовенко В.С. Бази даних та знань, інтелектуальний аналіз та обробка інформації в автоматизації розпізнавання на класифікації стрілецької зброї.

II Всеукраїнська науково-технічна конференція «Виклики та рішення в програмній інженерії», 26 листопада 2025 р. Подано до друку.

5. Гребещенко В.Е., Садовенко В.С. Системи штучного інтелекту та нейронні мережі в автоматизації розпізнавання на класифікації стрілецької зброї.

II Всеукраїнська науково-технічна конференція «Виклики та рішення в програмній інженерії», 26 листопада 2025 р. Подано до друку.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Goodfellow I., Bengio Y., Courville A. Deep Learning. Cambridge : MIT Press, 2016. 775 p.
2. Szeliski R. Computer Vision: Algorithms and Applications. 2nd ed. Cham : Springer, 2022. 979 p.
3. Prince S. J. D. Computer Vision: Models, Learning, and Inference. Cambridge : Cambridge University Press, 2019. 451 p.
4. Zhang R., Isola P., Efros A. A. Colorful Image Colorization // European Conference on Computer Vision (ECCV). Cham : Springer, 2016. 649 p.
5. Redmon J., Divvala S., Girshick R., Farhadi A. You Only Look Once: Unified, Real-Time Object Detection // Proceedings of the Computer Vision and Pattern Recognition (CVPR), 2016. 340 p.
6. Ren S., He K., Girshick R., Sun J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks // IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017. 436 p.
7. Liu W., Anguelov D., Erhan D. et al. SSD: Single Shot MultiBox Detector // European Conference on Computer Vision (ECCV). Cham : Springer, 2016. 40 p.
8. Howard A. G., Zhu M., Chen B. et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, 2017. 236 p.
9. Sandler M., Howard A., Zhu M., Zhmoginov A., Chen L.-C. MobileNetV2: Inverted Residuals and Linear Bottlenecks // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2018. 197 p.
11. Tan M., Le Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks // Proceedings of the International Conference on Machine Learning (ICML), 2019. 173 p.
12. Lin T.-Y., Maire M., Belongie S. et al. Microsoft COCO: Common Objects in Context: Springer, 2014. 203 p.
13. He K., Zhang X., Ren S., Sun J. Deep Residual Learning for Image Recognition // Proceedings of the Computer Vision and Pattern Recognition (CVPR), 2016. 336 p.

14. Girshick R. Fast R-CNN // Proceedings of the IEEE International Conference on Computer Vision (ICCV). 2015. P. 1440–1448.
15. Zhao Z.-Q., Zheng P., Xu S.-T., Wu X. Object Detection with Deep Learning: A Review // IEEE Transactions on Neural Networks and Learning Systems. 2019. Vol. 30, No. 11. P. 3212–3232.
16. Voulodimos A., Doulamis N., Doulamis A., Protopapadakis E. Deep Learning for Computer Vision: A Brief Review // Computational Intelligence and Neuroscience. 2018.
17. LeCun Y., Bengio Y., Hinton G. Deep Learning // Nature. 2015. Vol. 521. P. 436–444.
18. Dosovitskiy A., Beyer L., Kolesnikov A. et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale // International Conference on Learning Representations (ICLR). 2021.
19. Silver D., Schrittwieser J., Simonyan K. et al. Mastering the Game of Go without Human Knowledge // Nature. 2017. Vol. 550. P. 354–359.
20. Szeliski R. Computer Vision: Algorithms and Applications. 2nd ed. Cham : Springer, 2022. 935 p.
21. Prince S. J. D. Computer Vision: Models, Learning, and Inference. Cambridge : Cambridge University Press, 2016. 592 p.
22. Rosebrock A. Deep Learning for Computer Vision with Python. PyImageSearch, 2019. 486 p.
23. Chollet F. Deep Learning with Python. 2nd ed. Shelter Island : Manning Publications, 2021. 504 p.
24. Li F.-F., Ramanan D., Fergus R. Visual Recognition and Understanding. Cham : Springer, 2017. 412 p.
25. R. Szeliski. Computer Vision: Algorithms and Applications. New York : Springer, 2010. 832 p.
26. C. M. Bishop. Pattern Recognition and Machine Learning. New York : Springer, 2006. 738 p.

## ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО -  
КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ  
ТЕХНОЛОГІЙ



КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### Магістерська робота

**«Методика автоматизованого розпізнавання на класифікації  
стрілецької зброї»**

Виконав: студент групи ПДМ-62 Владислав ГРЕБЕЩЕНКО

Керівник: кандидат фіз.-мат. наук, доцент, професор кафедри ІІЗ  
Володимир САДОВЕНКО

Київ - 2025

### МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

**Мета роботи:** підвищення точності автоматизованого розпізнавання на класифікації стрілецької зброї.

**Об'єкт дослідження:** процес автоматизованого розпізнавання на класифікації стрілецької зброї.

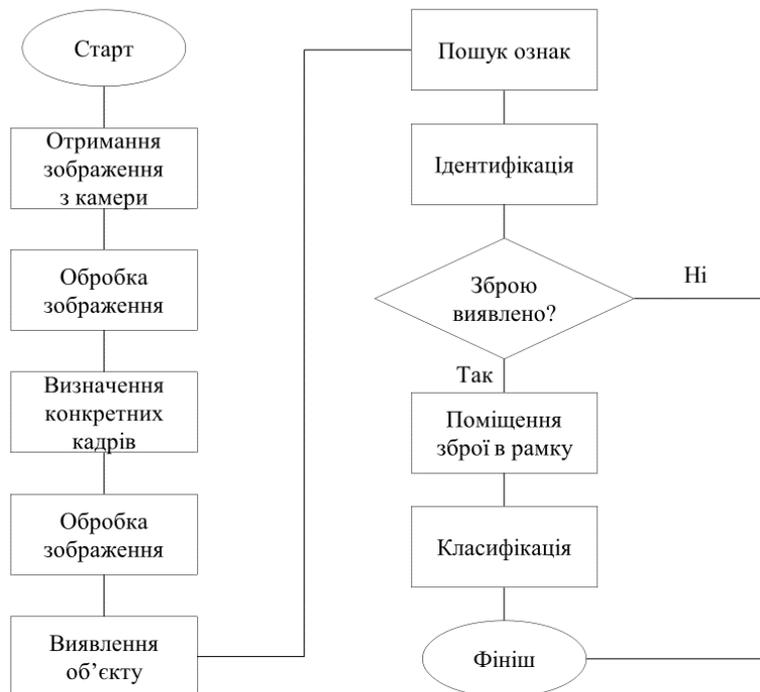
**Предмет дослідження:** методика автоматизованого розпізнавання на класифікації стрілецької зброї.

### ПОРІВНЯННЯ АНАЛОГІВ

Методики	Переваги	Недоліки
Класичний комп'ютерний зір	Автоматичне виділення патернів і аномалій	Низька точність Чутливість до якості даних (шум, освітлення, низька роздільна здатність), потреба в мануальному конструюванні ознак
Методики на основі глибокого навчання (CNN)	Висока точність Автоматичне виділення ознак об'єкту	Високі вимоги до обчислювальної техніки
Методика аналітики відеопотоку	Виявлення динамічних процесів	Складні архітектури для розробки Дуже високі вимоги до обчислювальної техніки Низька точність
Hybrid	Висока точність Автоматичне виділення ознак об'єкту Автоматизоване навчання Узагальнення та опрацювання великих наборів даних	Високі вимоги до обчислювальної техніки

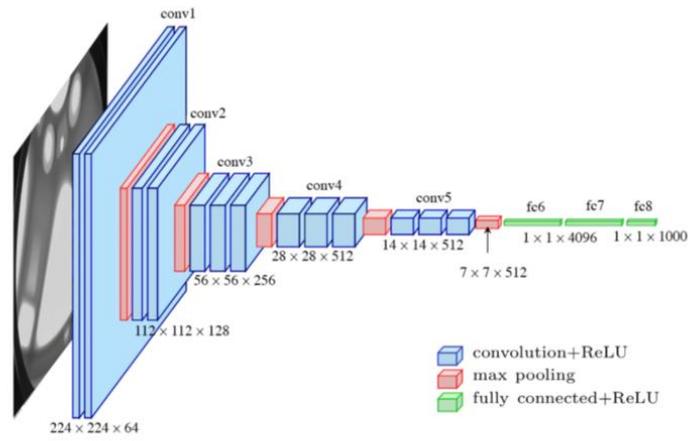
3

### АЛГОРИТМ АВТОМАТИЗОВАНОГО РОЗПІЗНАВАННЯ СТРЕЛЬЦЬКОЇ ЗБРОЇ



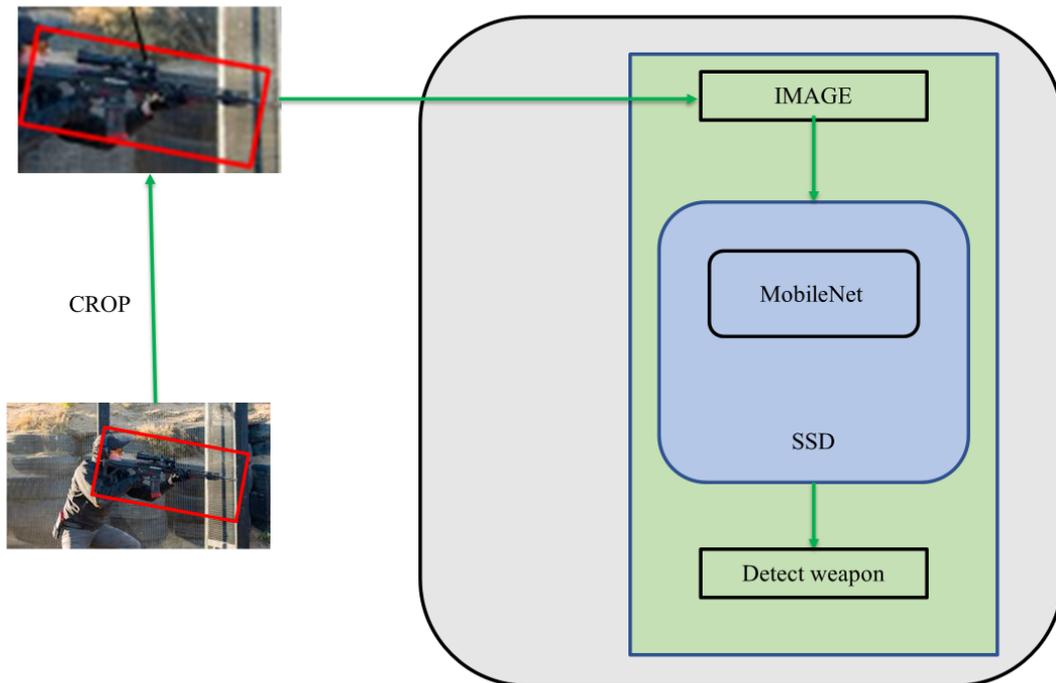
4

## АРХИТЕКТУРА MOBILENET



5

## СТРУКТУРА ГІБРИДНОЇ МЕТОДИКИ



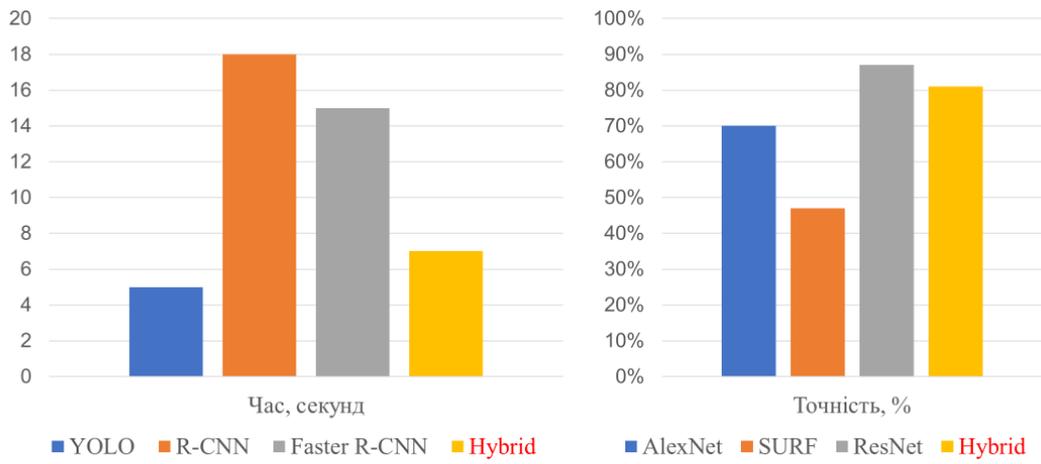
6

### ПРАКТИЧНИЙ РЕЗУЛЬТАТ



7

### ПОРІВНЯННЯ РЕЗУЛЬТАТІВ



8

## ВИСНОВКИ

1. Проведено аналіз існуючих методик та моделей автоматизованого розпізнавання на класифікації стрілецької зброї.
2. Розроблена інтелектуальна система, що забезпечує підвищення ефективності автоматизованого розпізнавання стрілецької зброї в режимі реального часу.
3. Визначено та обрано оптимальні середовища розробки, мови програмування, бібліотеки та інструменти для виконання поставлених завдань.
4. Проведено тестування розробленої гібридної методики, зібрано статистику щодо ефективності автоматизованого розпізнавання стрілецької зброї, встановлено середнє значення точності даних розпізнавання стрілецької зброї у межах 81%, зменшивши час розпізнавання в 1,5 рази.

9

## ПУБЛІКАЦІЇ ТА АПРОБАЦІЯ РОБОТИ

### **Тези доповідей:**

1. Гребещенко В.Е., Садовенко В.С. Сучасні інтелектуальні технології в Україні і світі з використанням мови програмування Java Script.  
V Всеукраїнська науково-практична конференція «Сучасні інтелектуальні інформаційні технології в науці та освіті», 15 травня 2025 р., Збірник тез. К.: ДУІКТ, 2025. С. 193-194.
2. Гребещенко В.Е., Садовенко В.С. Новітні апаратні та програмні засоби інтелектуальних технологій у військовій топографії.  
V Всеукраїнська науково-практична конференція «Сучасні інтелектуальні інформаційні технології в науці та освіті», 15 травня 2025 р., Збірник тез. К.: ДУІКТ, 2025. С. 271-273.
3. Гребещенко В.Е., Садовенко В.С. Парадигми програмної інженерії в автоматизації розпізнавання об'єктів.  
II Всеукраїнська науково-технічна конференція «Виклики та рішення в програмній інженерії», 26 листопада 2025 р. Подано до друку.
4. Гребещенко В.Е., Садовенко В.С. Бази даних та знань, інтелектуальний аналіз та обробка інформації в автоматизації розпізнавання на класифікації стрілецької зброї.  
II Всеукраїнська науково-технічна конференція «Виклики та рішення в програмній інженерії», 26 листопада 2025 р. Подано до друку.
5. Гребещенко В.Е., Садовенко В.С. Системи штучного інтелекту та нейронні мережі в автоматизації розпізнавання на класифікації стрілецької зброї.  
II Всеукраїнська науково-технічна конференція «Виклики та рішення в програмній інженерії», 26 листопада 2025 р. Подано до друку.

10