

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

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Розробка інтелектуальної системи розпізнавання
зображень та геолокації військової техніки за допомогою
технологій машинного навчання»

на здобуття освітнього ступеня магістра
зі спеціальності 121 Інженерія програмного забезпечення
(код, найменування спеціальності)
освітньо-професійної програми «Інженерія програмного забезпечення»
(назва)

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

_____ Вячеслав САВІЦЬКИЙ
(підпис)

Виконав: здобувач вищої освіти групи ПДМ-64

Вячеслав САВІЦЬКИЙ

Керівник: _____ Олесь ДІБРІВНИЙ
доктор філософії (PhD)

Рецензент: _____ Віктор ВИШНІВСЬКИЙ
д.т.н., професор

Київ 2024

**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

« _____ » _____ 2023 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

_____ Савіцькому Вячеславу Андрійовичу _____

1. Тема кваліфікаційної роботи: Розробка інтелектуальної системи розпізнавання зображень та геолокації військової техніки за допомогою технологій машинного навчання

керівник кваліфікаційної роботи Олесь ДІБРІВНИЙ доктор філософії (PhD)

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «19» 10.2023р. № 145.

2. Строк подання кваліфікаційної роботи «29» грудня 2023 р.

3. Вихідні дані до кваліфікаційної роботи: Дані про військову техніку. Комп'ютерний зір, як складова систем розпізнавання типів військової техніки та їх геолокації. Наукова література у сфері розпізнавання об'єктів та геолокації, яка стосується визначення об'єктів, створення наборів даних, обробки зображень та навчання нейронних мереж.

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

1. Дослідження даних про машинне навчання в різних арміях світу.

2. Аналіз технологій машинного, вимоги якості системи та опис математичних моделей

3. Опис проектування розробки.

5. Перелік графічного матеріалу: *презентація*

1. Види проведення розвідувальних заходів за допомогою машинного навчання.

2. Характерні ознаки військової техніки.

3. Типи алгоритмів машинного навчання.

4. Математична модель.

5. Блок-схема роботи інтелектуальної системи.

6. Інтерфейс інтелектуальної системи.

7. Тестування розробленої інтелектуальної системи.

6. Дата видачі завдання «19» жовтня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз наявної науково-технічної літератури	19.10-05.11.23	
2	Вимоги до системи	06.11-12.11.23	
3	Оцінка якості тестування розробленої системи	13.11-19.11.23	
4	Побудова методу	20.11-26.11.23	
5	Навчання моделей машинного навчання	27.11-03.12.23	
6	Розробка інтелектуальної системи	04.12-10.12.23	
7	Оформлення роботи: вступ, висновки, реферат	11.12-20.12.23	
8	Розробка демонстраційних матеріалів	21.12-29.12.23	

Здобувач вищої освіти

_____ (підпис)

Вячеслав САВІЦЬКИЙ

Керівник

кваліфікаційної роботи

_____ (підпис)

Олесь ДІБРІВНИЙ

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

Навчально-науковий інститут інформаційних технологій

**ПОДАННЯ
ГОЛОВІ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ
ЩОДО ЗАХИСТУ КВАЛІФІКАЦІЙНОЇ РОБОТИ
на здобуття освітнього ступеня магістра**

Направляється здобувач САВІЦЬКИЙ В.А. до захисту кваліфікаційної роботи

за спеціальністю 121 Інженерія програмного забезпечення

освітньо-професійної програми «Інженерія програмного забезпечення»

на тему: «Розробка інтелектуальної системи розпізнавання зображень та геолокації військової техніки за допомогою технологій машинного навчання».

Кваліфікаційна робота і рецензія додаються.

Директор ННІ ІТ

_____ (підпис)

Андрій БОНДАРЧУК

Висновок керівника кваліфікаційної роботи

Здобувач Савіцький Вячеслав Андрійович провів глибокий аналіз методів розв'язання поставленої задачі, обравши оптимальний варіант та вивчив існуючі реалізації. Робота актуальна через можливість допомоги Силам Оборони та покращення технологічного вмісту наявних методів автоматизованих систем управління. Дипломна робота відрізняється системністю та послідовністю, використанням передових технологій у розробці інтелектуальних систем.

Все це дозволяє оцінити виконану кваліфікаційну роботу здобувача Савіцькому В.А. на оцінку «відмінно» та присвоїти йому кваліфікацію магістр з інженерії програмного забезпечення.

Керівник кваліфікаційної роботи _____

(підпис)

Олесь ДІБРІВНИЙ

« ____ » _____ 20__ року

Висновок кафедри про кваліфікаційну роботу

Кваліфікаційна робота розглянута. Здобувач Савіцький В.А. допускається до захисту даної роботи в Екзаменаційній комісії.

Завідувач кафедри ІІЗ

_____ (підпис)

Ірина ЗАМРІЙ

ВІДГУК РЕЦЕНЗЕНТА **на кваліфікаційну магістерську роботу**

здобувача вищої освіти Савіцького Вячеслава Андрійовича

на тему «Розробка інтелектуальної системи розпізнавання зображень та геолокації військової техніки за допомогою технологій машинного навчання»

Актуальність.

Актуальність магістерської роботи полягає в необхідності вдосконалення систем військового моніторингу та безпеки через розробку інтелектуальної системи, яка використовує технології машинного навчання для ефективного розпізнавання зображень та геолокації військової техніки. Це дозволить підвищити точність та оперативність військового аналізу та прийняття рішень.

Позитивні сторони.

1. Логічність і послідовність викладеного матеріалу.
2. Розроблений мобільний застосунок на якому протестована інтелектуальна система.
3. Протестовані показники розпізнавання типу та геолокації військової техніки.

Недоліки.

1. Недостатньо чітко описано математичну модель інтелектуальної системи.
2. Доцільно було б збільшити кількість випробувань за допомогою інтелектуальної інформаційної системи для репрезентативної оцінки похибки розпізнавання типу та геолокації військової техніки.

Відзначені зауваження не впливають на загальну позитивну оцінку кваліфікаційної магістерської роботи.

Висновок: *кваліфікаційна магістерська робота заслуговує оцінку "відмінно", а здобувач **Савіцький Вячеслав Андрійович** заслуговує присвоєння кваліфікації: магістр з інженерії програмного забезпечення.*

Рецензент:
д.т.н., професор

 підпис

Віктор ВИШНІВСЬКИЙ

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 60 стор., 29 рис., 30 джерел.

Мета роботи – реалізація інтелектуальної системи розпізнавання зображень та геолокації військової техніки за допомогою технологій машинного навчання.

Об'єкт дослідження – система розпізнавання зображень та геолокації військової техніки за допомогою машинного навчання.

Предмет дослідження – методологія побудови та реалізація інтелектуальної системи розпізнавання зображень та геолокації військової техніки.

Наукова інновація поляє у розробці технологій виявлення військової техніки на основі машинного навчання з використанням Swift і CoreML. Ця розробка сприяє ефективному виявленню ворожої техніки і покращує систему управління даними. За допомогою цієї розробки можна швидко визначити місце розташування і клас ворожої техніки. Були розглянуті існуючі методи визначення та визначення їх переваг при формулюванні еталонних умов. Аналізується можливість навчання нейронної мережі з використанням алгоритму YOLO, а також мови програмування Swift і технології CoreML, і результати аналізуються далі. Розроблена технологія може бути застосована у військових системах управління для виявлення та відстеження військової техніки.

КЛЮЧОВІ СЛОВА: РОЗПІЗНАВАННЯ ОБРАЗІВ ВІЙСЬКОВОЇ ТЕХНІКИ, РОЗПІЗНАВАННЯ ГЕОЛОКАЦІЇ НАВКОЛИШНЬОГО СЕРЕДОВИЩА, МАШИННЕ НАВЧАННЯ, ІДЕНТИФІКАЦІЯ ОБ'ЄКТІВ, CoreML, CoreLocation, YOLO

ABSTRACT

Text part of the master's qualification work: 60 pages, 29 pictures, 30 sources.

The purpose of the work: implementation of an intelligent system of image recognition and geolocation of military equipment using machine learning technologies.

Object of research – a system of image recognition and geolocation of military equipment using machine learning.

Subject of research – methodology of construction and implementation of an intelligent system of image recognition and geolocation of military equipment.

Summary of the work – Scientific innovation abounds in the development of machine learning-based military equipment detection technologies using Swift and CoreML. This development contributes to the effective detection of enemy equipment and improves the data management system. With the help of this development, you can quickly determine the location and class of enemy equipment. The existing methods of definition and determination of their advantages in the formulation of reference conditions were considered. The ability to train a neural network using the YOLO algorithm as well as the Swift programming language and CoreML technology is analyzed, and the results are analyzed further. The developed technology can be applied in military control systems to detect and track military equipment.

KEYWORDS: MILITARY IMAGE RECOGNITION, ENVIRONMENTAL GEOLOCATION RECOGNITION, MACHINE LEARNING, OBJECT IDENTIFICATION, CoreML, CoreLocation, YOLO

ЗМІСТ

ВСТУП.....	9
РОЗДІЛ 1 ТЕОРЕТИЧНІ ВІДОМОСТІ.....	11
1.1 Що таке машинне навчання.....	11
1.2 Історія розвитку машинного навчання.....	12
1.3 Машинне навчання у повсякденному житті.....	13
1.4 Машинне навчання у військовій сфері України.....	14
1.5 Машинне навчання у збройних силах різних держав.....	15
1.5.1 Машинне навчання у збройних силах Великої Британії.....	15
1.5.2 Машинне навчання у збройних силах США.....	17
1.5.3 Машинне навчання у збройних силах Німеччини.....	20
1.6 Огляд типів алгоритмів.....	21
1.6.1 Кероване навчання.....	21
1.6.2 Неконтрольоване навчання.....	23
1.6.3 Напівкероване навчання.....	23
1.6.4 Навчання з підкріпленням.....	25
1.6.5 Трансдукція.....	26
РОЗДІЛ 2 НАУКОВО-ДОСЛІДНА ЧАСТИНА.....	29
2.1 Методи розпізнавання об'єктів на зображеннях.....	29
2.1.1 Методи засновані на ознаках(Feature-based methods).....	29
2.1.2 Методи засновані на виборі(Template-based methods).....	31
2.1.3 Методи глибокого навчання(Deep Learning methods).....	32
2.1.4 Методи з використанням дескрипторів ключових точок (Key Point Descriptors).....	35
2.2 Згортковані нейронні мережі.....	38
2.2.1 R-CNN.....	38
2.2.2 YOLO.....	42
2.2.3 SSD detector.....	44
2.2.4 Feature Pyramid Networks.....	48
РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА.....	52
3.1 Огляд на CoreML моделі.....	52
3.2 Створення CoreML моделі.....	58
3.3 Інтеграція CoreML моделі з мовою програмування Swift.....	64
ВИСНОВКИ.....	67
ПЕРЕЛІК ПОСИЛАНЬ.....	68
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація).....	71

ВСТУП

Обґрунтування вибору теми та її актуальність: в умовах сьогодення, військові конфлікти характеризуються високою мобільністю та динамікою бойових дій, існує нагальна потреба у швидкому та точному розпізнаванні геолокації та типі військової техніки противника. Швидке отримання та обробка оперативної інформації про переміщення та розташування техніки противника – є наймовірно важливим фактором для прийняття рішень у сфері військового управління.

Актуальність теми обумовлена такими факторами:

- висока динаміка бойових дій, яка потребує швидких рішень. Сучасні бойові машини та інша техніка має високі характеристики мобільності та можливості пересування на великі відстані не втрачаючи мобільність.

- велике поширення аеротехнічної розвідки та інших технологічних апаратів, що виконують функцію розвідки. Дрони відіграють одну з ключових ролей сучасних воєн, адже використання дронів не потребує ризикувати життям людини, людина тепер керує на відстані, у безпеці.

- швидка обробка великої кількості розвідувального матеріалу. Сучасні методи машинного навчання дозволяють створювати високоточні алгоритми розпізнавання типу військової техніки та геолокації по фото.

- можливість розпізнавання типу та геолокації військової техніки потенційного агресора для забезпеченні національної безпеки. Дана технологія дає можливість отримувати оперативну інформацію про перебазування техніки та розташування потенційних загроз.

Наукова новизна: під-час війни, технології розвиваються наймовірно стрімкими темпами. У 21 столітті перевагу має та сторона, яка має технологічну перевагу над супротивником. Тому портативність, легкість транспортування, точність результату і зручність використання засобів, які має видають інформацію про противника – є надзвичайно важливим засобом прискорення перемоги.

Програмні засоби використовуються у сучасній війні на повну, але потрібно покращувати вже наявні засоби та створювати нові.

Для реалізації та демонстрації дипломної роботи, був створений мобільний застосунок на платформі iOS, написаний на мові програмування Swift , застосовані бібліотеки CoreLocation та Vision.

Мета – підвищення ефективності пошуку геолокації та класифікації типу військової техніки зображеної на фотографії за допомогою методів машинного навчання.

Об'єкт дослідженні – автоматизація пошуку та розпізнавання об'єктів зображення та їх геопозиція на основі технологій машинного навчання.

Предмет дослідження – аналіз існуючих підходів розпізнавання об'єктів, пошук можливостей та технологій для підвищення точності результатів, розробка набору даних для навчання системи, впровадження системи у мобільний застосунок.

Методи дослідження: методи машинного навчання, методи розпізнавання об'єктів, методи розпізнавання геолокації, методи класифікації точності результату.

1 ТЕОРЕТИЧНІ ВІДОМОСТІ

1.1 Що таке машинне навчання

Машинне навчання (ML) – це один з розділів штучного інтелекту, який дає можливість комп'ютеру навчатись без явного програмування. Системи машинного навчання можуть навчитись виконувати задачі аналізуючи данні, на базі яких, комп'ютер був навчений. Якщо задача комп'ютера розпізнати певну людину на фото, то систему потрібно навчити, для цього потрібно завантажити системі фото людини, яку потрібно розпізнати, чим різноманітніші фото і чим їх більше – тим більша вірогідність точного розпізнавання.

Машинне навчання доволі розповсюджена технологія у сучасному житті, вона використовується у таких сферах як: медицина, транспорт, освіта, правоохоронна, бізнес та інші. У медичній сфері, ML використовується для розробки систем, які використовуються для діагностування хвороб, особливо для розпізнавання хвороб, які є рідкими діагнозами. У сфері транспорту машинне навчання використовується для систем керування. У освіті – використовується для адаптації навчання під індивідуальні потреби людини. У правоохоронній системі, це може бути розпізнавання номеру авто, розпізнавання обличчя, розпізнавання відбитку пальці, та розпізнавання стилю ходьби людини. Бізнес використовує машинне навчання багатьма способами: аналіз продажів, прогнозування на наступну декаду, рекомендації товару клієнту в інтернет магазині та багатьма іншими способами.

Навчаючи комп'ютер, ми даємо йому масив даних, яку він обчислює та аналізує, це можуть бути: звук, відео, зображення, текст та інше. Маючи дані, модель починає знаходити закономірності у масиві даних та використовує для закономірності для прогнозування результату на основі наданих даних.

1.2 Історія розвитку машинного навчання

Першою визначальною подією у світі штучного інтелекту стала Дартмутська конференція 1956 року. На цій конференції зібрались в одному місці дослідники з галузей – математики, інформатики, фізики та інших. Однією з цілей конференції була мета обговорити, чи має потенціал дослідження синтетичного інтелекту – саме таку назву мав штучний інтелект до отримання сучасної назви. Вклад цієї конференції був настільки великим, що була розроблена мова програмування для ШІ – LISP, також були створенні дослідницькі лабораторії ШІ у таких університетах, як MIT, Stanford та Carnegie Mellon.

У другій половині XX століття, у галузі штучного інтелекту був період значного прогресу, створювались нові методи та нейронні мережі, які могли виконувати прості завдання як розпізнавання рукописних цифр.

Важливим викликом стала обробка великих даних. Великі дані характеризуються трьома атрибутами: швидкість, різноманітність та обсяг кількості даних. Швидкість – характеризує швидкість генерації та обробки даних, різноманітність – характеризується різноманітністю обробки типів даних, а обсяг – характеризує розмір даних, з якими працює комп'ютер.

Наступним кроком розвитку машинного навчання після великих даних стала поява глибинного навчання. Глибинне навчання – це один з типів машинного навчання, який застосовує штучні нейронні мережі, які в свою чергу, є модуляцією людського мозку. Ці мережі є однією мережею шарів взаємопов'язаних вузлів, кожен з яких виконує певний математичний функціонал отриманими даними для навчання. Вихідні дані одного рівня служать входом для наступного, дозволяючи мережі отримувати дедалі складніші характеристики з даних.

Однією з найбільших переваг глибинного навчання – є те, що він може вивчати ієрархічне представлення даних. Мається на увазі, що комп'ютер автоматично розпізнає шаблони та абстракції. Наприклад, мережа спочатку може навчитись розпізнавати лише окремі букви, потім загалом букви, потім слова, а вже потім через якусь кількість абстракцій і цілі речення.

У ХХІ столітті машинне навчання стало невід'ємною частиною звичайного життя. У комп'ютерних іграх можна позмагатися з противником, яким керує комп'ютер, який підстроюється під рівень гравця, камера на смартфонах підлаштовується під умови навколишнього середовища для покращення якості знімку, мікрофон в телефоні відсіює звуки на другому плані при дзвінку.

1.3 Машинне навчання у повсякденному житті

Машинне навчання є дуже розповсюдженим у сучасному світі, його приклади можна зустріти у різних сферах життя. Коли робимо запит у пошуковому полі, то Google видає нам відповіді на наші пошукові запити. Наприклад, Youtube, Netflix, Megogo видає нам рекомендований відеоконтент навчаючись на наших попередніх запитах та контенті який і скільки ми дивились. Соціальні мережі рекомендують нам людей, яких ми можемо знати. Також, Youtube генерує субтитри для відео на різних мовах. Поштові сервіси навчилися фільтрувати спам у вхідних листах.

Також віртуальні помічники використовують технологію машинного навчання. У Google – це Google Assistant, у Apple – Siri, у Amazon – це Alexa. Віртуальні помічники можуть виконувати доволі прості на перший погляд функції: поставити будильник, додати нагадування, запустити/зупинити музику або відео, увімкнути/вимкнути ліхтарик, зателефонувати потрібному користувачу, ввести повідомлення голосом, та навіть викликати таксі.

Доволі популярним машинне навчання є у бізнесі, розпізнає мову, жести, образи предметів чи людей, проводить медичну та технічну діагностики, допомагає підставити коректне слово при написанні повідомлення на пристроях, допомагає аналізувати фінансовий ринок, ринок цінних паперів, цінних металів та криптовалют.

У бізнес середовищі машинне навчання можна використовувати наступним чином:

1. Оцінювати ефективність рекламних компаній.

2. Прогнозування залученості аудиторії.
3. Прогнозувати виконання фінансового плану.
4. Досліджувати ефективність відділів та окремих працівників.
5. Оцінювати ризики.

1.4 Машинне навчання у військовій сфері України

Джеймс Ставрідіс, колишній адмірал ВМС США і колишній командувач силами НАТО в Європі, вважає, що війна в Україні може стати полігоном для випробування штучного інтелекту. «ШІ стане потужним інструментом для тих, хто ухвалює рішення на полі бою на всіх рівнях» – сказав Джеймс Ставрідіс в інтерв'ю The Washington Post.

Далі в інтерв'ю, колишній командувач силами НАТО в Європі, розповів про випадок, який стався у 1988 році. У центрі бойової інформації, на крейсері “Вінсеннес”, офіцер неправильно ідентифікував ціль та помилково збив іранський комерційний лайнер, близько 300 цивільних загинули у ту трагедію. На авторитетну думку колишнього адмірала, якби вони на той момент мали допомогу у вигляді штучного інтелекту, який би вмів обробляти велику кількість даних і який би міг порівняти радіолокаційну картину з великою кількістю подібних сценаріїв, що сталося, то об'єкт був би ідентифікований як цивільний, то люди залишились би цілі. Машинне навчання тоді, могло би значно зменшити кількість втрат серед цивільного населення [1].

У війні 21 століття є одним з найважливіших чинників – це заміна людини на машину. Це може проявлятися від розвідки, де звичайних піших розвідників замінили БПЛА, до пункту АСУ(автоматизованої системи управління), де офіцеру командної ланки буде допомагати машина без емоцій, яка аналізує всю надану їй інформацію[2]. Сучасні методи машинного навчання могли би майже миттєво видавати детальну інформацію про цілі для застосування, до прикладу, високоточної зброї по найбільш оптимальним цілям противника. Також, машинне навчання може проаналізувати дані про логістичні ланцюги та надати детальну інформацію про слабкі точки противника, або ж вказати на наші слабкі сторони в

логістиці, які не є очевидними. Машинне навчання також можна залучити для психологічно-інформаційної війни. Наприклад, навчена модель могла видавати в результаті плакати з агітацією, або, навіть, повністю згенеровані пропагандистські ролики. У складській чи логістичній сфері, машинне навчання може допомогти контролювати кількість обладнання чи продовольства в пункті зберігання і в наступні замовлення збільшити кількість для недопускання дефіциту. При підготовці особового складу, як у фізичному, у інтелектуальному і в психологічному плані, машинне навчання допомагало б інструкторам оцінювати прогрес підрозділів, а спецпідрозділів допомагати відбирати потрібних кандидатів, серед великої кількості претендентів. Мабуть, одним з найголовніших плюсів навчання машини – є можливість аналітики і можливості вгадування наступних кроків супротивника при кібератаках[3].

1.5 Машинне навчання у збройних силах різних держав

Варто розуміти, що хоча війна і дорога справа, штучний інтелект і машинне навчання є також дорогими цінностями у армії. Бо потрібно зростити спеціалістів, надати їм гідну оплату праці, надати їм дороговартісне обладнання та оновлювати доволі часто матеріально-технічну базу для підрозділу розробки. Лідерами серед військової інтеграції машинного навчання є, доволі очевидно, США та Китай. Велика Британія, Канада, Іспанія, Німеччина, Франція, країни Скандинавії та деякі азійські країни також потужно розвивають цей напрямок але все ще не можуть конкурувати з двома лідерами у цьому питанні. Потрібно пам'яті, що підготувати, наприклад, пілота винищувача також є дорогим заняттям, але методи штучного інтелекту, зокрема машинне навчання.

1.5.1 Машинне навчання у збройних силах Великої Британії

Солдати Королівського логістичного корпусу Великої Британії для підвищення навичок водіння та здатності подавляти вогонь супротивника

використовують систему модуляцій та репетицій різних сценаріїв поведження супротивника користуються програмою Virtual Combat Convoy Trainer (VCCT).



Рис. 1.1. Віртуальний бойовий конвойний тренажер

Віртуальний бойовий конвойний тренажер (VCC Trainer) – це система віртуальної реальності (VR), яка розроблена армією Британії для підготовки конвоїв Королівських сил. Система розроблена так, що вона використовує машинне навчання (ML) для створення реалістичного навколишнього середовища, де солдати можуть відпрацьовувати навички управління колоною.

VCC Trainer використовує технології машинного навчання для створення динамічного середовища, що змінюється постійно, тобто, кожне нове тренування – новий маршрут і ландшафт. Це дозволяє солдатам покращувати свої навички, а не звикати до однієї і тієї самої ситуації, маршруту та ландшафту місцевості, також дозволяє вчитись відражати атаки противника, уникати дорожньо-транспортні пригоди та вправно поводитись у погані погодні умови. Система використовує машинне навчання для аналізу та оцінки дій солдатів. Що дозволить інструкторам, після аналізу даних, давати індивідуальні відгуки для солдатів, що покращить їх слабкі сторони. Для створення унікального ландшафту і поля дій у симуляторі – система машинного навчання (ML) використовує дані з реального світу.

1.5.2 Машинне навчання у збройних силах США

У США проводяться дослідження з тактичної допомоги пораненим в бою для всіх учасників бойових дій з використанням адаптивної навчальної платформи Cerego, яка використовує штучний інтелект і машинне навчання, що дозволяє скоротити час навчання з 12 до 6 годин. 110 стажерів з 321-ї навчальної ескадрильї на об'єднаній базі Сан-Антоніо-Лекленд, що у штаті Техас, мали необмежений доступ до усієї платформи. Програма аналізувала успіхи та давала аналітичну інформацію для інструкторів про успішність стажерів.

CereGo – це адаптивна платформа для тренувань, розроблена для американських військових задля підвищення ефективності їх навчання. Платформа використовує машинне навчання (ML) для пристосування навчання до потреб кожного солдата.

Система складається з двох основних частин:

- система машинного навчання: система збирає дані та за допомогою машинного навчання, аналізує їх та використовує для адаптації навчання;
- платформа для навчання: платформа, що забезпечує солдатам доступ до навчання.

Система Cerego працює наступним чином:

1. Спочатку солдат має пройти вступний тест для оцінки його початкового рівня.
2. Система за допомогою машинного навчання(ML) створює план навчання індивідуальний для кожного солдата, після проходження початкового тесту.
3. Солдат проходить навчання на платформі за матеріалом, який йому згенерував штучний інтелект.
4. Машинне навчання відстежує прогрес та адаптує навчальний план відповідно до проблемних тем солдата.

Переваги Cerego системи за машинним навчанням перед традиційними способами навчання:

- персоналізація: система пристосовує навчання до потреб кожного окремого солдата. Отже, це означає, що солдати можуть навчатись у власному темпі та зосередити навчання на закриття своїх слабких сторін.,

- ефективність: дозволяє солдатам засвоювати наданий матеріал більш ефективно, ніж традиційні методи навчання.,

- відстеження процесу: ML відстежує прогрес солдатів, тому інструктори можуть надавати індивідуальний зворотній зв'язок.

Advanced Targeting and Lethality Aided System (ATLAS) – це експериментальна система, розроблена американськими військовими для підвищення ефективності ураження цілей. Система застосовує ML для підвищення ефективності ураження цілей. Також, система використовує машинне навчання для автоматизації системи за для пасивного виявлення цілей, це дозволяє екіпажу одночасно виявляти до трьох цілей за той самий час, що зазвичай необхідний для виявлення тільки однієї цілі.

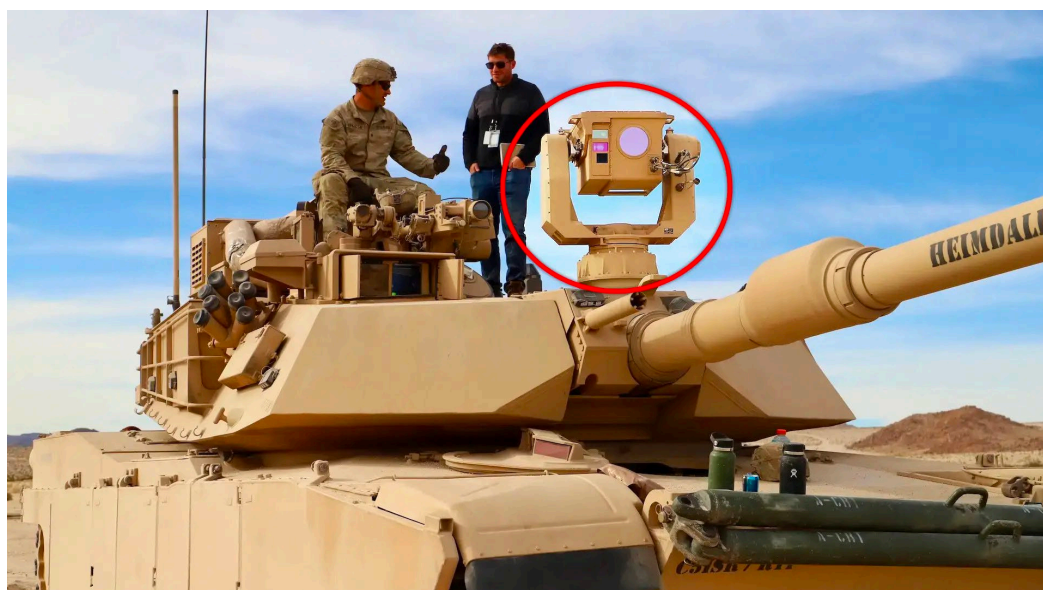


Рис. 1.2. Частина комплекту ATLAS на танку Abrams

На рис. 1.2 червоним колом зображено частину комплексу ATLAS, а саме сенсорний блок який розташований на башті танку Abrams. Трохи нижче розташований блок, який виявляє та розраховує влучання за допомогою лазерів і використовується для імітації бою та для оцінки втрат під час бою.



Рис. 1.3. Блок для виявлення цілі системи ATLAS

Основні блоки датчиків та інше суміжне обладнання розташоване у задній частині танку “Абрамс”[4]. Також на задній частині танку є кондиціонер для комп’ютера, ймовірно, який необхідний для охолодження і безперебійної роботи автоматизованої системи управління системи ATLAS.

Система ATLAS складається з двох основних компонентів:

- датчики: вони збирають інформацію про навколишнє середовище, а саме: зображення, звуки, теплові сигнатури.
- машинне навчання: ML аналізує з датчиків дані для виявлення, ідентифікації та відстеження цілей.

Система ATLAS може бути встановлена на різні транспортні засоби, включаючи: танки, бронетранспортери і гелікоптери. ATLAS інтегрований з бортовою системою управління вогнем і надає відповідальному на автоматизовану систему управління інформацію про цілі та вогневі рішення.

ATLAS може використовуватися для різних цілей, зокрема:

- виявлення і супроводження цілей: система ATLAS може виявляти і супроводжувати навіть у надскладних умовах бою, таких як: погана видимість, туман, а також дим.

- ідентифікація цілей: ATLAS може ідентифікувати такі цілі як бронетранспортери, літаки, танки і піхоту.

- прицілювання і ведення вогню: ATLAS може аналізувати і видавати інформацію про ціль екіпажу, що дозволяє екіпажу приймати рішення про ведення вогню.

Переваги ATLAS:

- автоматизація: система ATLAS автоматизує велику кількість задач екіпажу, які пов'язані з пасивним виявленням цілей. Це дозволяє екіпажу не втрачати концентрацію та займатись управлінням вогнем.

- швидкість: система може виявляти, ідентифікувати та відстежувати цілі незрівнянно швидше за людину.

- точність: ATLAS надає екіпажу точну інформацію про ціль, що збільшує вірогідність точного ураження цілі[5].

1.5.3 Машинне навчання у збройних силах Німеччини

Німеччина розробляє свій метавсесвіт під назвою GhostPlay, у якому планує тестувати свої військові винаходи.



Рис. 1.4. Демонстрація метавсесвіту GhostPlay

В основі платформи, яку німці називають “метавсесвітом Бундесверу”, лежить рушій штучного інтелекту, який побудований на моделі самонавчання завдяки машинному навчанню[6]. GhostPlay є унікальним, тому що він використовує алгоритм, так званої, “3 хвили”, це означає, що учасники бою зможуть приймати рішення схоже на людське мислення, але можливі непередбачувані рішення, вони також увімкненні в налаштуваннях. Якщо порівнювати “2 і 3 хвили”, то “2 хвиля” лише є допоміжною для людини, прискорює прийняття рішення та оптимізує його, замість запропонування нових варіантів рішення.

Переваги Ghost:

- реалістичність: система надає можливість солдатам отримувати досвід максимально схожий на бойовий;
- персоналізований зворотній зв'язок: система індивідуально, за допомогою машинного навчання, аналізує результати кожного бійця і тим самим боєць знає свої слабкі сторони, які потрібно покращити;
- ефективність та безпека: система дає змогу проходити навчання більш ефективно, а найголовніше, що більш безпечно[7].

1.6 Огляд типів алгоритмів

1.6.1 Кероване навчання

Кероване навчання – це тип машинного навчання, в якому система навчається з наданим набором певних даних, які мають містити вхідні і вихідні дані. Вхідні дані – це дані, які система аналізує і робить прогноз, а вихідні – це дані, які видає система після аналізу. Для того, аби алгоритм класифікувався, як контрольоване навчання, потрібно, аби він працював з прикладами, які містять не тільки вектор незалежних змінних, атрибутів й властивостей, але й значення, яке модель видасть після навчання – саме це значення називається цільовим. Уся різниця між цільовим і фактичним виходами моделі є похибкою навчання, або ще називають залишками, яка мінімізується в процесі навчання та вона виконує роль

того, хто навчає. Вихідний параметр помилок використовується для розрахунку коригувань параметрів моделі при кожній ітерації навчання[8].

Цей тип навчання найбільш розповсюджений для вирішення задач класифікації та регресії. Класифікація – це можливість визначення категорії вхідної моделі. Регресія – це значення, яке система передбачає на основі вхідних даних.

До числа алгоритмів керованого навчання для рішення задач класифікації відносяться:

- дерева рішення;
- машини опорних векторів;
- баєсовські класифікатори;
- лінійний дискримінантний аналіз;
- метод k-найближчих сусідів.

До алгоритмів керованого навчання з учителем для рішення задач регресії входять:

- лінійна регресія;
- логістична регресія;
- нейронні мережі.

Установка завдання машинного навчання алгоритму керованого навчання виглядає наступним чином: ми маємо початковий набір, який складається з N прикладів. Кожен приклад, що 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. Потім алгоритм спочатку має шукати 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 $f: X \times Y \rightarrow R$, де

f – функція, що обчислює значення u , що оптимізує вираз $g(x) = \operatorname{argmax}_u f(x, u)$, де f – функція Φ .

1.6.2 Неконтрольоване навчання

Ненаглядне навчання – це метод машинного навчання, який не використовує цільову функцію для встановлення параметрів навчальної моделі. В алгоритмах неконтрольованого навчання не обчислюється помилка виходу моделі по відношенні до вибірки навчання. Замість вищезгаданого, використовується інформація про параметри моделі та поточний стан навчальної вибірки. Прикладом є Евклідова відстань між векторами ознак та векторами ваг нейронів. Основним застосуванням неконтрольованого навчання є побудова моделей кластеризації. Цільові значення не можуть бути використані, оскільки кластерна структура даних не відома заздалегідь і визначається під час навчання моделі. Типовими прикладами моделей, що використовують неконтрольоване навчання, є когерентні мережі та карти, які широко використовуються в інтелектуальному аналізі даних. Ці моделі будуються на основі змагального навчання, де корекція вектора ваги нейрона контролюється відстанню між вектором ваги та вектором ознак навчальної вибірки. Інший метод – контрольоване навчання, яке встановлює цільове значення для кожного навчального набору, обчислює вихідну помилку і коригує параметри моделі, обчислюючи її величину і знак[9].

1.6.3 Напівкерowane навчання

Напівкерowane навчання (SSL) – це тип машинного навчання

(ML) техніка. Це щось середнє між контрольованим і неконтрольованим навчанням. Набір даних частково позначено, як показано на малюнку:

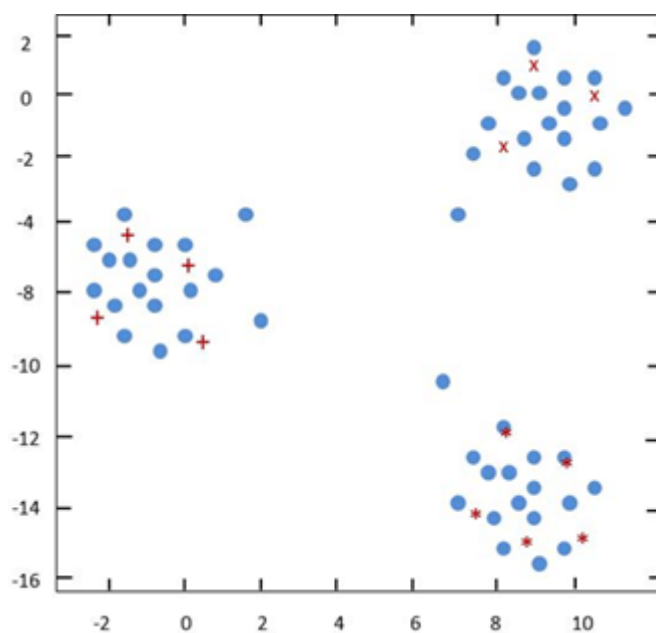


Рис. 1.5. Графічне зображення напівкерованого навчання

Основною метою SSL (напівкерованого навчання) є подолання недоліків як контрольованого, так і неконтрольованого навчання. Контрольоване навчання вимагає величезної кількості навчальних даних для класифікації тестових даних, що є доволі трудомістким процесом. З іншого боку, неконтрольоване навчання не потребує мічених даних, які групують дані на основі подібності в точках даних за допомогою методу кластеризації або підходу максимальної правдоподібності. Основним недоліком цього підходу є те, що він не може точно кластеризувати невідомі дані. Щоб подолати ці проблеми, дослідницьке співтовариство запропонувало напівкероване навчання (SSL), який може навчитися позначати невідомі тестові дані за допомогою невеликої кількості даних для навчання. SSL будує модель із кількома позначеними шаблонами як навчальні дані, а решта шаблонів розглядає як тестові дані. Напівкерована кластеризація є окремим випадком кластеризації. У звичайній кластеризації для кластеризації використовуються не мічені шаблони даних. Однак напівкерована кластеризація використовує як мічені, так і немічені дані, а також використовує додаткову

інформацію, також як парні обмеження (злиття і роз'єднання), щоб допомогти кластеризувати шаблони даних[10].

1.6.4 Навчання з підкріпленням

Навчання з підкріпленням (Reinforcement Learning) – це навчання тому, що робити, тобто як співвідносити ситуацію з поведінкою, щоб максимізувати числовий сигнал винагороди. Це процес навчання наступним діям. Як і в загальному машинному навчанні, учневі не кажуть, яку дію він повинен виконати. Замість цього, учень повинен фактично спробувати, яка поведінка принесе найбільшу винагороду. У більшості машин У найцікавіших і найскладніших ситуаціях дія може вплинути не лише на поточну винагороду, а й на наступну ситуацію і всі наступні винагороди. Вона також впливає на ці дві особливості – метод спроб і помилок і відкладена винагорода – дві найважливіші особливості навчання з підкріпленням.

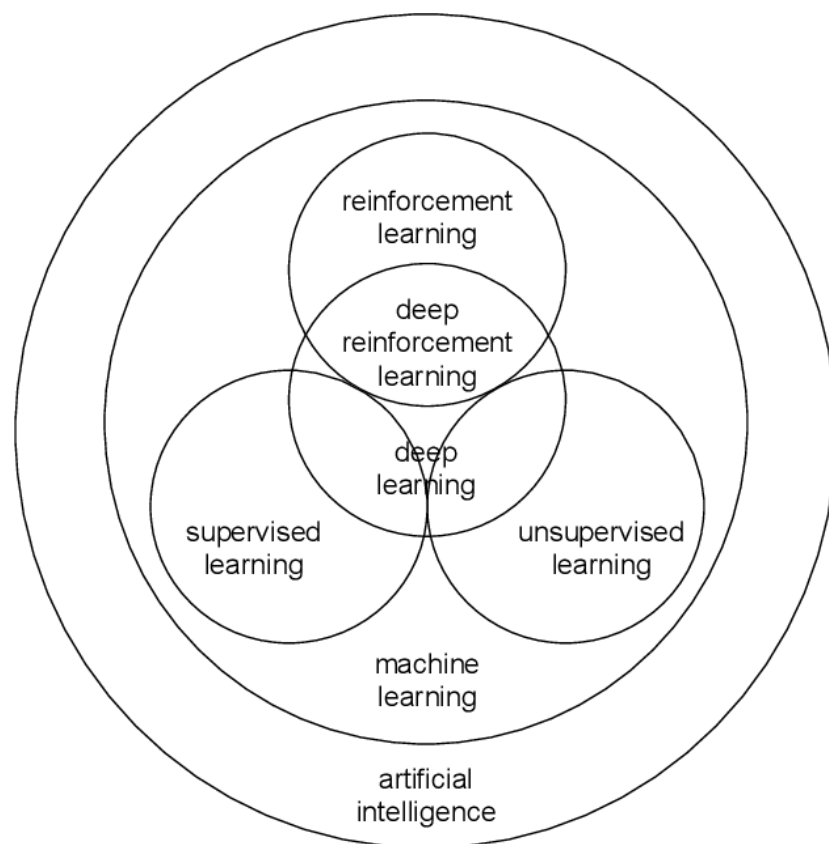


Рис. 1.6. Демонстрація залежності навчань один від одного

Навчання з підкріпленням відрізняється від навчання з контролем, яке широко вивчається в сучасному машинному навчанні, статистичному розпізнаванні образів і дослідженнях штучних нейронних мереж. Воно відрізняється від навчання під контролем, яке широко вивчається в сучасному машинному навчанні, статистичному розпізнаванні образів і дослідженнях штучних нейронних мереж.

Контрольоване навчання – це навчання на прикладах, наданих компетентними зовнішніми тренерами. Хоча це важливий тип навчання, його недостатньо для навчання на основі діалогу. Часто непрактично отримати приклади бажаної поведінки в інтерактивному завданні, які були б точними і репрезентативними для всіх ситуацій, в яких виконується інтерактивне завдання. Часто буває недоцільно отримувати такі приклади усередині невідомого. У ситуаціях, де навчання буде найбільш корисним, алгоритми повинні мати можливість навчатися самостійно.

Всі агенти навчання з підкріпленням мають чіткі цілі і можуть сприймати характеристики свого середовища, і можуть вибирати дії, які впливають на навколишнє середовище. Агенти повинні діяти, незважаючи на велику невизначеність середовища, з яким вони стикаються. Якщо навчання з підкріпленням передбачає планування, необхідно враховувати взаємодію між плануванням і вибором запасів у реальному часі. Якщо навчання з підкріпленням передбачає навчання під наглядом, необхідно розглянути питання про те, як агент буде діяти в тій чи іншій моделі середовища. Якщо навчання з підкріпленням передбачає навчання під наглядом, на це є причина. Вона, здебільшого, полягає у визначенні того, які компетенції є важливими, а які – неважливими[11].

1.6.5 Трансдукція

В індуктивному навчанні для навчання моделі використовуються немарковані набори тестів доступні. Ця установка загальні припущення і поки вона відхиляється від абсолютно безпрецедентного тестового набору використовується

в багатьох реальних сценаріях. Може використовуватися незважаючи на свої практичні переваги[12].

Індуктивне навчання, в галузі теорії статистичного навчання, відноситься до передбачення конкретних прикладів на основі конкретних прикладів у конкретній галузі. Його відрізняють від інших типів навчання, таких як індуктивне та дедуктивне навчання. Трансдукцію можна використовувати для класифікації нових даних, навіть якщо доступні лише обмежені набори позначених даних. Наприклад, модель машинного навчання можна навчити на наборі зображень автомобілів або людей. Цю модель можна використовувати для класифікації нових зображень, які не містять міток[13].

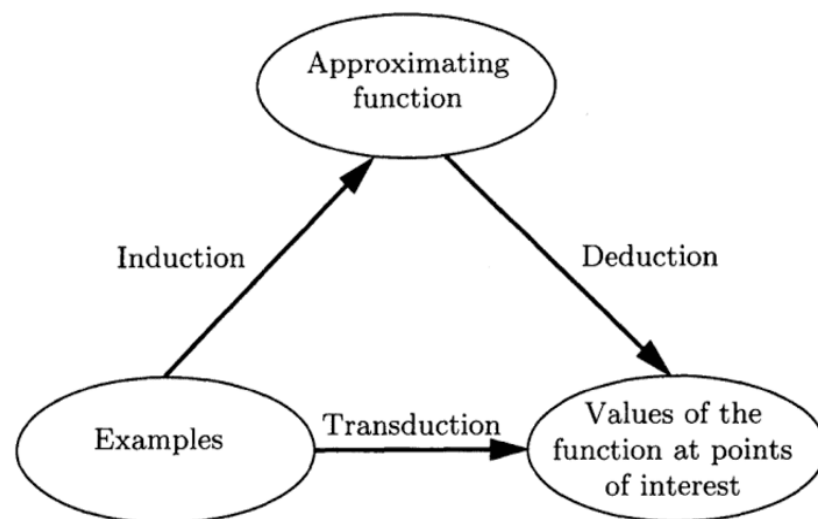


Рис. 1.7. Діаграма керованого навчання

Це цікава система керованого навчання, в якій класична задача "апроксимації функції відображення на основі даних і використання її для прогнозування" вважається складнішою, ніж вона повинна бути. Замість цього, конкретні прогнози робляться безпосередньо на основі реальних прикладів з предметної області. Апроксимація функції не є необхідною. Трансдукція природно асоціюється з

класом алгоритмів, відомих як навчання на основі прикладів або навчання на основі випадків. Найвідомішим з цього класу алгоритмів є, мабуть, алгоритм k-найближчих сусідів. Використання терміну трансдукції, коли йдеться про теорію та класичний машинний переклад, забарвлює використання терміну, коли йдеться про сучасне передбачення хвороби за допомогою рекурентних нейронних мереж у задачах обробки природної мови.

Трансдукцію можна використовувати для прогнозування нових значень даних, навіть якщо доступний лише обмежений набір позначених даних. Наприклад, модель машинного навчання можна навчити на даних про втрати техніки. Модель можна використовувати для прогнозування втрат техніки на основі нових даних, які не мають точних цифр[14].

2 НАУКОВО-ДОСЛІДНА ЧАСТИНА

2.1 Методи розпізнавання об'єктів на зображеннях

2.1.1 Методи засновані на ознаках (Feature-based methods)

Методи засновані на ознаках (FBM) – це підхід когнітивного моделювання, який будує модель, що пов'язує параметри контексту з відповідними діями шляхом визначення відповідних ознак. І контекст, і дії характеризуються різними властивостями, які створюють взаємозв'язки, що відображають когнітивну поведінку агента. Функція контексту охоплює такі властивості, як рівність від'ємника нулю в базовому відніманні, тоді як функція дії описує реакцію агента на результат, що дорівнює від'ємнику.

Асоціації, що позначаються як $X \rightarrow$, показують зв'язок між набором елементів контексту (X) і одним елементом дії (a). Ці асоціації визначають релевантність певних когнітивних патернів і розрізняють бажані та неправильні асоціації. Мережа ознак, центральний елемент FBM, допомагає структурувати зв'язки між ознаками і створювати зв'язки між узагальненням і спеціалізацією.

Модель FBM враховує толерантність до шуму, вважаючи взаємозв'язки дійсними лише за умови достовірності та точності даних. Модель враховує потенційні зміни в поведінці агентів через зміни в даних з плином часу, надаючи більшу вагу недавнім доказам, ніж минулим даним. Така динамічна адаптація має вирішальне значення для когнітивного моделювання в галузі, що розвивається, особливо в освітньому середовищі, де трансформація з часом є метою. Реалізація моделі включає такі параметри, як `min_evidence` та `min_assurance`, які визначають поріг прийняття взаємозв'язку. Інтеграція ставок дисконтування гарантує, що минула поведінка має менший вплив, ніж нещодавні спостереження. Підхід FBM до побудови моделі передбачає використання двовимірного масиву, ефективно організованого для представлення взаємозв'язків, що відповідають спостережуваним даним, що відповідає принципам простору версій.

FVM потужний інструмент когнітивного моделювання, який фокусується на складності взаємозв'язків між контекстом і поведінкою, враховуючи при цьому шум і динамічні зміни в когнітивних підходах акторів. Його структурований підхід, що поєднує доказовість і адаптивність, позиціонує його як цінну основу для розуміння і прогнозування когнітивної поведінки в різних сферах[15].

При адаптації когнітивних моделей важливо виокремити стислий набір асоціацій відповідно до цільового використання моделі. Замість того, щоб мати справу з усіма підтримуваними асоціаціями, часто практичніше зосередитися на обраній підмножині, що містить важливі аспекти поведінки агента. Процес стратегічного спрощення враховує специфіку та релевантність асоціацій і адаптує їх до використання моделі. Залежно від конкретного випадку використання виникають різні стратегії спрощення моделі. Якщо модель керує взаємодією з агентом, вибір найбільш конкретних асоціацій дозволяє розмежувати завдання до найдрібніших деталей. Для ефективної комунікації з людьми, необхідно вибирати асоціації, які роблять модель найбільш загальною. Корисним типом асоціації ж найконкретніша найвища підтримувана асоціація. Асоціація $C \rightarrow a$ є найвищою підтримуваною асоціацією. Немає прийнятої асоціації $X \rightarrow a$, що C являється узагальненням X і $\#(X \rightarrow a) = \#(C \rightarrow a)$, а також немає прийнятої асоціації $Y \rightarrow a$, щоб Y був узагальненням C та $\#(Y \rightarrow a) > \#(C \rightarrow a)$. Дотримання балансу між узагальненістю та конкретністю дозволяє уникнути поспішних висновків і надто складних пояснень. Асоціація $C \rightarrow a$ є найбільш загальною недостатньо доказовою асоціацією тільки тоді, коли $\#(C \rightarrow a) < \text{min_evidence}$. Не існує асоціації $X \rightarrow a$, де X є узагальненням C і $\#(X \rightarrow a) < \text{min_evidence}$.

Для багатьох конкретних взаємозв'язків дуже важливо розуміти суперечливий характер доказів. Це може вказувати на наявність шуму, потребу в більш точних моделях або фактичні зміни в підході агента до предметної області. Аналіз розподілу суперечливих доказів у часі може допомогти зрозуміти надійність і адаптивність когнітивних моделей. Таким чином, стратегічний вибір асоціацій виявляється нюансованим процесом, який забезпечує ефективну комунікацію і безперервний розвиток когнітивної моделі[16].

2.1.2 Методи засновані на виборі (Template-based methods)

Співставлення зразків, давня проблема комп'ютерного зору, включає алгоритми, які знаходять схожість між зображеннями, ускладненими потенційними спотвореннями, такими як масштабування, обертання та освітлення. Історично склалося так, що класичні алгоритми комп'ютерного зору вирішували цю складну задачу. Метод використовує згорткові нейронні мережі (CNN), навчені на шаблонах запитів і зображеннях для ідентифікації об'єктів. Мета полягає в тому, щоб зібрати достатньо просторової інформації для подолання проблем, притаманних пошуку шаблонів. Формулюючи проблему як пошук шаблону на зображенні, алгоритм має явну перевагу в тому, що генерує оцінку класу для кожного пікселя[17].

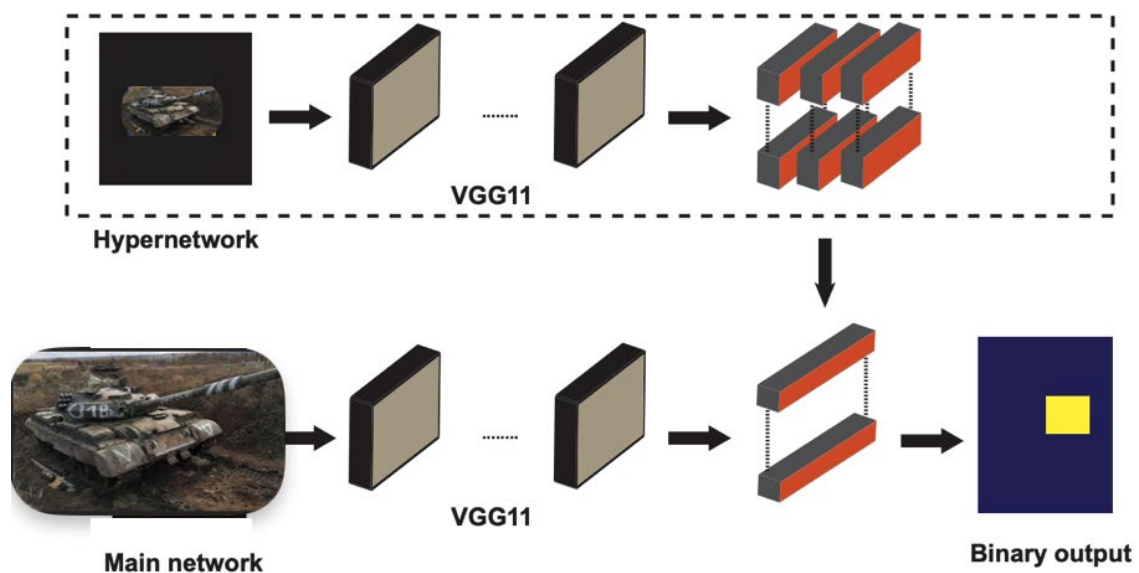


Рис. 2.1. Загальний огляд робочого процесу TBM

Показано, що навчений алгоритм успішно виконує завдання пошуку продуктів з контролем доступу. Показано порівняльний аналіз з використанням алгоритму SIFT.

CNN складається з декількох шарів, де кожен шар обробляє частину вхідного зображення, а універсальність CNN значно розширилася і охоплює різноманітні застосування, від підписів до зображень та ідентифікації. У цьому контексті

одночасно навчаються дві CNN, одна з яких обробляє шаблонні зображення, а інша – зображення запиту. Кінцевий згортковий шар обох мереж з'єднується за допомогою мостів, спочатку розроблених для зменшення параметрів великих CNN, таких як hyperneat. Мости полегшують передачу інформації від зображення-шаблону до зображення-запиту. Тензорне перетворення вимагає наявності повністю зв'язаного шару, щоб використовувати його виходи як ваги для кінцевого згорткового шару основної мережі. Вихідні канали гіперсітки перетворюються у вектори, а повністю зв'язаний шар використовується для побудови остаточного ядра[18].

$$a_i = W_i z_i + b_i$$

$$K_i = W_{out} a_i + B_{out}$$

$$K = (K_1 \quad K_2 \dots K_i \dots K_{Nin})$$

Цей процес включає гіперпараметри, глибину та довжину вхідних векторів. Отримані ядра об'єднуються для формування повного ядра.

2.1.3 Методи глибокого навчання(Deep Learning methods)

Основною метою RNN є розпізнавання послідовностей і шаблонів, таких як ті, що містяться в мові, рукописних текстах і пов'язаних з текстом програмах. Він використовує у своїй структурі циклічні зв'язки, одночасно використовуючи ітераційні обчислення для послідовної обробки вхідних даних. RNN, по суті, є розширенням стандартної нейронної мережі в часі, причому надходять на наступний часовий крок, а не на наступний рівень на тому ж часовому етапі. Вектори стану в прихованих одиницях зберігають усі попередні вхідні дані, і ці вектори стану використовуються для обчислення виходу. Архітектура RNN показана на нижче.

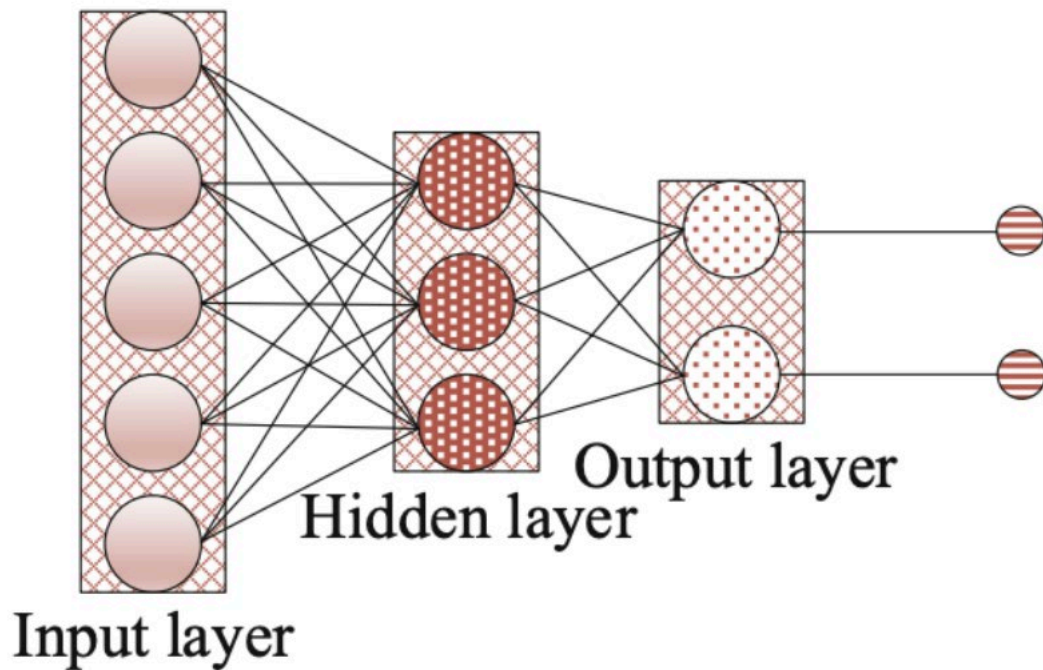


Рис. 2.2. Архітектура RNN

RNN є відносно новим доповненням до методів глибокого навчання. Тому його застосування все ще перебуває на ранніх стадіях, що залишає простір для подальших досліджень і розвідок.

DBN використовуються для дослідження різноманітних багатовимірних даних. Цей підхід передбачає наявність декількох шарів, між якими існують зв'язки, відмінні від зв'язків між одиницями в кожному шарі. DBN можна розглядати як гібридні багат шарові нейронні мережі, що охоплюють як спрямовані, так і неспрямовані зв'язки. Структура DBN включає метод жадібного навчання, в який включені обмежені машини Больцмана (RBM). Кожен рівень RBM взаємодіє як з попереднім, так і з наступним рівнем. Модель складається з мережі прямого поширення і декількох рівнів обмежених машин Больцмана, що діють як екстрактори ознак. DBN складається лише з двох шарів, прихованого і видимого[19].

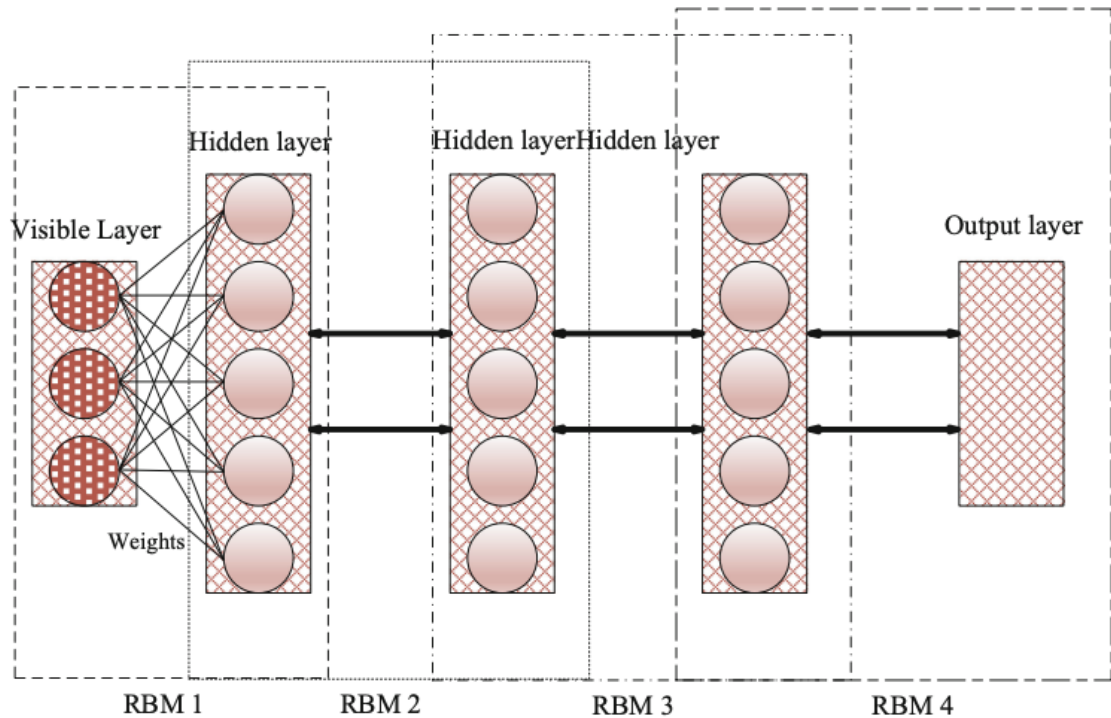


Рис. 2.3. Архітектура DBN

DBN виділяється як високонадійний метод глибокого навчання, що характеризується як високою точністю, так і обчислювальною ефективністю. У результаті області його застосування стали різноманітними, охоплюючи різноманітні захоплюючі способи використання в інженерних і наукових та військових задачах.

Як і RNN, LSTM використовує зв'язки зі зворотним зв'язком і слугує обчислювальним інструментом загального призначення. Цей підхід можна застосовувати для розпізнавання послідовностей, розпізнавання образів та обробки зображень. Загалом, система LSTM складається з трьох основних блоків: входу, виходу та зворотного зв'язку; помітною особливістю LSTM є їхня здатність визначати, коли приймати вхідні дані на нейрон, і продовжувати обчислення з попереднього часового кроку на основі поточних вхідних даних[20].

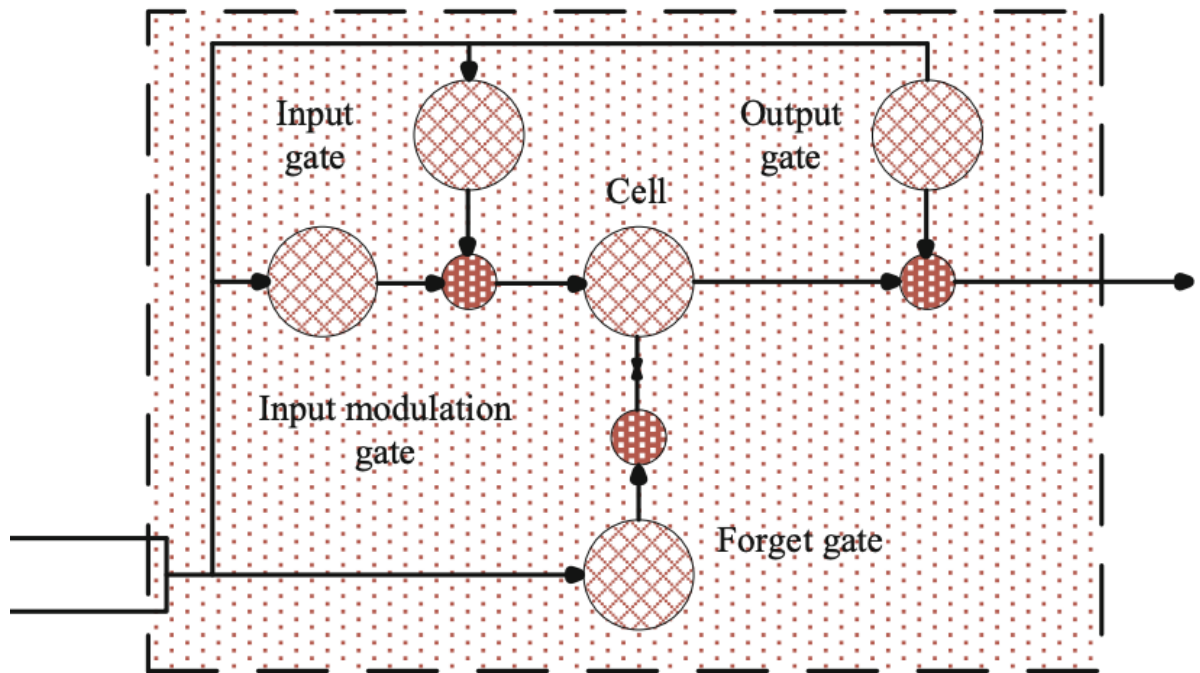


Рис.2.4. Архітектура LSTM

2.1.4 Методи з використанням дескрипторів ключових точок (Key Point Descriptors)

Основна ідея полягає в тому, щоб передбачити логічну відповідь, порівнюючи два пікселі поруч із ключовою точкою. Наприклад, BRIEF (Binary Robust Independent Elementary Features) працює без відповідного детектора; BRIEF використовує двоетапний процес, заснований на рандомізації відмінностей між пікселями навколо ключових точок. Спочатку зображення розмивається. Потім випадковим чином вибираються пари пікселів (x, y) поблизу ключової точки і відбираються відповідно до гаусівського розподілу з центром у ключовій точці. Потім кожна пара пікселів піддається порівняльному аналізу:

$$b = \begin{cases} 1 & \text{if } p(x) < p(y) \\ 0 & \text{else} \end{cases} \quad (2.1)$$

де x – пари пікселів.

BRISK (Binary Robust Invariant Scalable Key Points), вимагає, щоб інформація про масштаб була описана перед початком процесу. На відміну від BRIEF, де всі

пари вибираються випадковим чином, BRISK використовує рівномірний розподіл фіксованих пар навколо ключових точок. Ступінь охолодження пікселя змінюється в залежності від відстані від ключової точки. Набір пар ділиться на два набори, L і S, що відповідають великим і малим відстаням між двома елементами пари. Набір L використовується для визначення орієнтації ключової точки:

$$\begin{pmatrix} g_x \\ g_y \end{pmatrix} = \frac{1}{L} \sum_{(p_i, p_j) \in \mathcal{L}} \frac{I(p_i, \sigma_i) - I(p_j, \sigma_j)}{\|p_i - p_j\|^2} \begin{pmatrix} x_i - x_j \\ y_i - y_j \end{pmatrix} \quad (2.2)$$

де L – кардинал;

p – піксель;

(x, y) – його координати;

g – вектор результату рівняння.

Від g, орієнтація витягується. Розташування елементів набору S змінюється кутом оцінюється: $S\theta$. Елементи $S\theta$ порівнюються один з одним щоб отримати дескриптор.

Розмір зображення – від 1,5 до 5 з кроком 0,5. Щоб уникнути інтерполяції значень пікселів, зображення не збільшується. Тому розмір ключових точок на перетвореному зображенні буде скориговано відповідним чином. Для перетворення обертання ми застосовуємо наступну матриця для кожного кута між 10° і 350° з кроком 10° :

$$\begin{pmatrix} x_\theta \\ y_\theta \end{pmatrix} = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad (2.3)$$

де (x_θ, y_θ) і (x, y) – координати пікселів;

θ – кут повороту.

Початкова координата знаходиться в центрі зображення, де оцінюємо вплив оцінки кута на інваріантність дескриптора вимірювання виконується двічі, одне без будь-якої інформації про кут, другий з відомим кутом. Зсув проектує зображення так, ніби воно повернуте до осі глибини, з фіксованою лівою або нижньою стороною. Спостерігач проєкції встановлюється в центрі зображення. Що стосується повороту, то крок 10° від 10° до 80° . Матриці перетворень призначені для вертикального зсуву:

$$\begin{pmatrix} x_\theta \\ y_\theta \\ p_\theta \end{pmatrix} = \begin{pmatrix} D \cos(\theta) & 0 & \frac{DCol(\cos(\theta)-1)}{2} \\ 0 & D & 0 \\ \sin(\theta) & 0 & D + \frac{Col \sin(\theta)}{2} \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (2.4)$$

та для горизонтального зсуву:

$$\begin{pmatrix} x_\theta \\ y_\theta \\ p_\theta \end{pmatrix} = \begin{pmatrix} D & 0 & 0 \\ 0 & D \cos(\theta) & \frac{DRow(\cos(\theta)-1)}{2} \\ \sin(\theta) & 0 & D + \frac{Row \sin(\theta)}{2} \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (2.5)$$

Перевага надання інформації про азимут піднімає питання про оцінку азимута і про те, чи слід вимірювати азимут детектором або дескриптором. Дескриптори, розроблені Фан та ін., привернули увагу через їхню некутовість; SIFT і SURF показують інваріантність до повороту на 90° ; ORB 90° має періодичні помилки апроксимації на кожні 90° . Це підкреслює важливість врахування обертальної інваріантності при виборі дескрипторів зображень[21]. Результати щодо інваріантності обертання представлені на рисунку нижче:

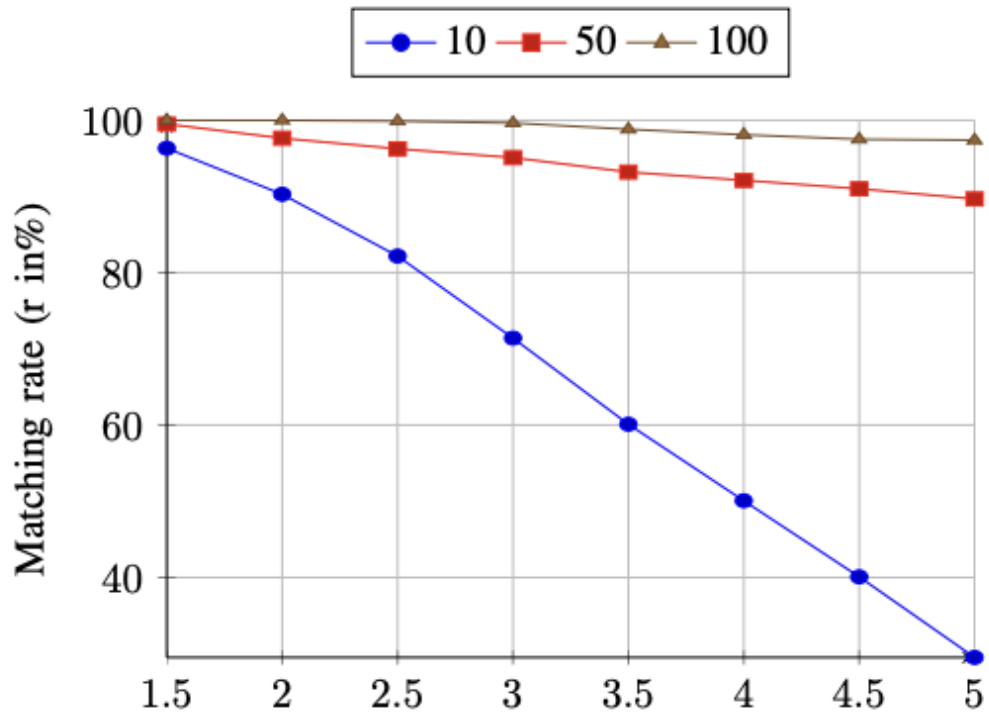


Рис. 2.5. Результати інвертивності обертання

2.2 Згортковані нейронні мережі

2.2.1 R-CNN

Підхід до виявлення об'єктів, який називається орієнтованим R-CNN, включає в себе орієнтовану мережу регіональних пропозицій (RPN) і орієнтовану головку R-CNN. Це двоступеневий детектор, у якому перший ступінь ефективно генерує високоякісні орієнтовані пропозиції, а другий етап включає в себе орієнтовану головку R-CNN для класифікації та регресії пропозицій. Магістраль мережі Feature Pyramid (FPN), дотримуючись архітектури, створює п'ять рівнів функцій $\{P_2, P_3, P_4, P_5, P_6\}$. Для простоти архітектура FPN і гілка класифікації в орієнтованому RPN опущені. Орієнтований RPN, маючи вхідне зображення будь-якого розміру, виробляє розріджений набір орієнтованих пропозицій, використовуючи легку повністю згорнуту мережу. Він приймає п'ять рівнів функцій FPN $\{P_2, P_3, P_4, P_5, P_6\}$ як вхідні дані та приєднує голівку однакового дизайну (що складається зі згорткового шару 3×3 і двох однорідних згорткових шарів 1×1) до кожного рівня. Три горизонтальні прив'язки зі співвідношенням

сторін {1:2, 1:1, 2:1} призначаються кожному просторовому розташуванню на всіх рівнях об'єктів. Ці прив'язки мають площі пікселів 32^2 , 64^2 , 128^2 , 256^2 , 512^2 на {P2, P3, P4, P5, P6} відповідно. Кожен якір представлений 4-вимірним вектором (a_x, a_y, a_w, a_h) , де (a_x, a_y) є центральною координатою, а a_w і a_h позначають ширину та висоту якоря. Один із двох однорідних згорткових шарів 1×1 служить гілкою регресії, виводячи зміщення $\delta = (\delta_x, \delta_y, \delta_w, \delta_h, \delta_\alpha, \delta_\beta)$ пропозицій відносно прив'язок. Для кожного розташування на карті об'єктів створюється пропозиція A (де A — кількість прив'язок, яка дорівнює 3 у цій роботі), що призводить до $6A$ виходів із гілки регресії. Декодування цих вихідних даних регресії дозволяє нам отримати орієнтовані пропозиції. Процес декодування виглядає таким чином:

$$\begin{cases} \Delta\alpha = \delta_\alpha \cdot w, & \Delta\beta = \delta_\beta \cdot h \\ w = a_w \cdot e^{\delta_w}, & h = a_h \cdot e^{\delta_h} \\ x = \delta_x \cdot a_w + a_x, & y = \delta_y \cdot a_h + a_y \end{cases} \quad (2.6)$$

де (x, y) – координати центру прогнозованої пропозиції;
 w і h – ширина і висота зовнішнього прямокутника прогнозованої пропозиції;
 $\Delta\alpha$ та $\Delta\beta$ – зміщення до центрів верхньої та правої сторін зовнішнього прямокутника.

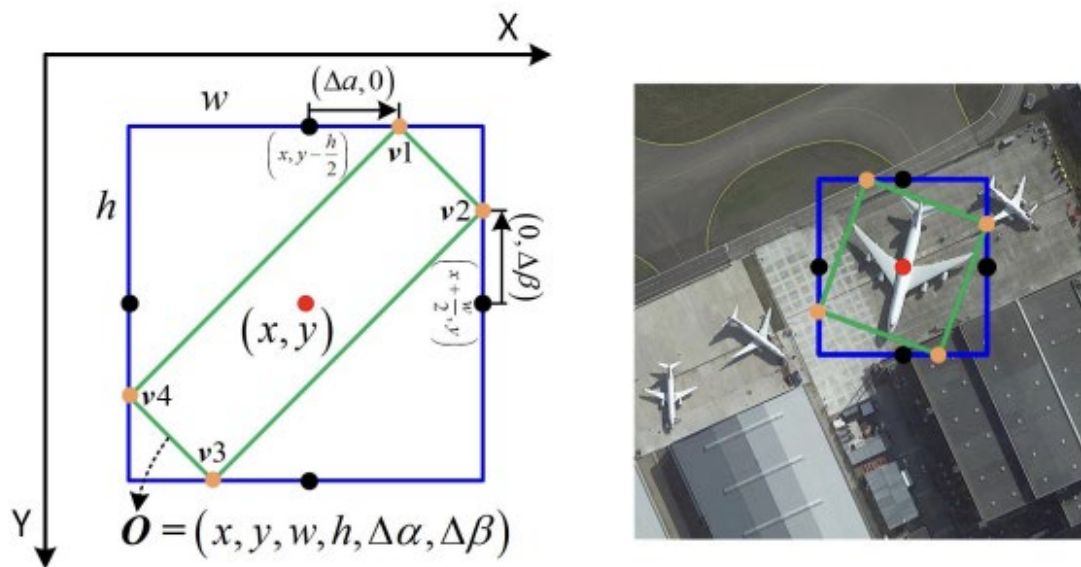


Рис. 2.6. Ілюстрація представлення зміщення середньої точки

Чорні крапки позначають середини кожної сторони горизонтальної рамки, що представляє зовнішній прямокутник орієнтованої обмежувальної рамки O . Помаранчеві крапки вказують на вершини орієнтованої обмежувальної рамки O . Для зображення об'єкта ми використовуємо орієнтовану обмежувальну рамку O з шістьма параметрами $O = (x, y, w, h, \Delta\alpha, \Delta\beta)$. Використовуючи ці шість параметрів, ми можемо отримати набір координат $v = (v_1, v_2, v_3, v_4)$, що представляє чотири вершини для кожної пропозиції. Тут $\Delta\alpha$ означає зміщення v_1 відносно середини $(x, y - h/2)$ верхньої сторони горизонтального прямокутника. Дотримуючись симетрії, $-\Delta\alpha$ позначає зсув v_3 відносно нижньої середньої точки $(x, y + h/2)$. Аналогічно, $\Delta\beta$ позначає зсув v_2 відносно правої середньої точки $(x + w/2, y)$, а $-\Delta\beta$ є зміщенням v_4 відносно лівої середньої точки $(x - w/2, y)$ [22]. Таким чином, координати чотирьох вершин можна виразити наступним чином:

$$\begin{cases} v_1 = (x, y - h/2) + (\Delta\alpha, 0) \\ v_2 = (x + w/2, y) + (0, \Delta\beta) \\ v_3 = (x, y + h/2) + (-\Delta\alpha, 0) \\ v_4 = (x - w/2, y) + (0, -\Delta\beta) \end{cases} \quad (2.7)$$

Використовуючи цей підхід представлення, виконується регресія для кожної орієнтованої пропозиції, прогножуючи параметри (x, y, w, h) для його зовнішнього прямокутника та виводячи параметри $(\Delta\alpha, \Delta\beta)$ для його середнього зміщення.

Щоб навчити орієнтований RPN, ми визначаємо позитивні та негативні зразки наступним чином. Кожному прив'язці присвоюється двійкова мітка $\{(p^* \in \{0, 1\})\}$, де 0 означає, що прив'язка належить до негативної проби, а 1 вказує на позитивну пробу. Зокрема, прив'язка вважається позитивним зразком за однієї з двох умов: (i) якщо прив'язка має перекриття перехрестя над з'єднанням (IoU) більше 0,7 з будь-яким блоком істинності землі, або (ii) якщо прив'язка має найвищий IoU перекривається з полем правдивості землі, і IoU перевищує 0,3.

Якірні зразки позначаються як негативні, якщо їхні IoU з базовими квадратами правдивості нижчі за 0,3. Анкери, які не підпадають ні в позитивні, ні в негативні категорії, розглядаються як недійсні зразки та ігноруються під час процесу навчання[23]. Важливо відзначити, що згадані прямокутники базової правди відносяться до зовнішніх прямокутників орієнтованих обмежувальних прямокутників.

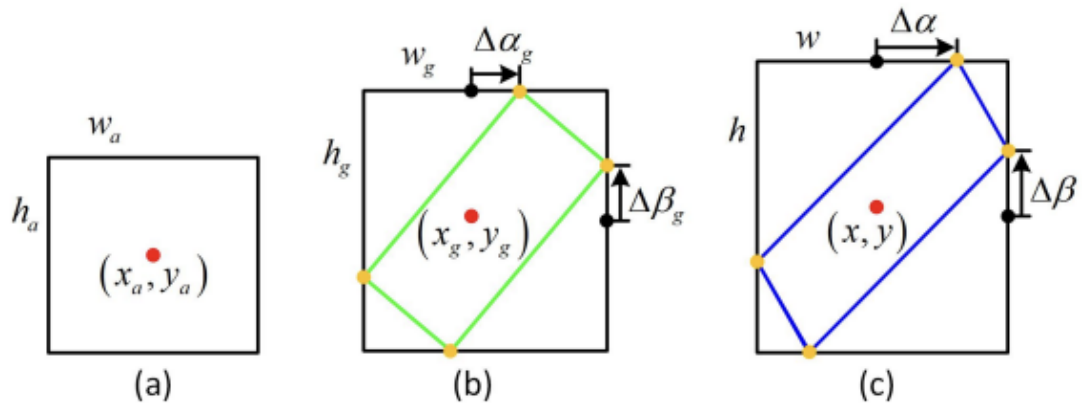


Рис. 2.7. Приклад параметризації регресії

На рисунку а зображено звичайний анкор, б – еталонна модель, с – передбачувана модель.



Рис. 2.8. Практичний приклад відображення параметрової регресії

2.2.2 YOLO

Алгоритм YOLO (You Look Only Once) – це метод виявлення об'єктів, який формулює виявлення об'єктів як регресійну задачу. Він має на меті оцінити обмежувальні рамки та пов'язані з ними ймовірності класів за один прохід через нейронну мережу, і дозволяє обробляти ці оцінки одночасно для зображення.

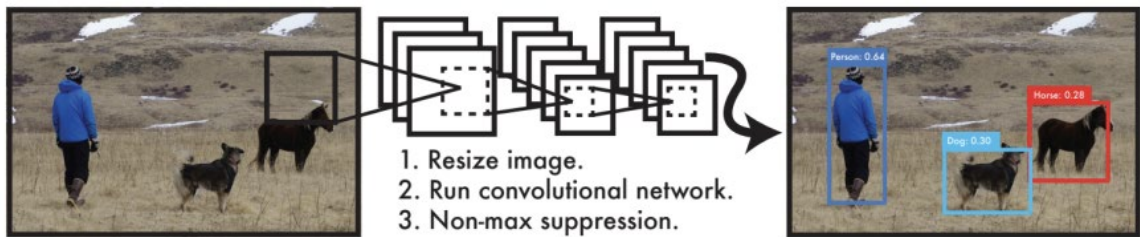


Рис. 2.9. Приклад роботи системи розпізнавання YOLO

У системі YOLO вхідні дані (зображення) піддаються зміні розміру з подальшим виконанням однієї згорткової нейронної мережі на зображенні. Процес виявлення згодом виконується з використанням немаксимального придушення. Відмінною рисою цієї системи є її здатність передбачати кілька обмежувальних прямокутників і ймовірностей класів одночасно через одну згорткову мережу[24]. Незважаючи на те, що система чудово швидко виявляє об'єкти на зображеннях, вона може зіткнутися з труднощами при точному локалізуванні менших об'єктів, особливо у випадках, коли точність має вирішальне значення. У YOLO вхідне зображення розбивається на сітки $S \times S$, і коли об'єкт знаходиться в центрі комірки сітки, ця комірка відповідає за виявлення об'єкта. Кожна комірка сітки використовує ознаки, витягнуті з усього зображення, для оцінки обмежувальних рамок і довірчих оцінок, пов'язаних з цими рамками. Ці довірчі оцінки вказують на ймовірність того, що об'єкт знаходиться всередині рамки, і на точність передбачення класу об'єкта. Довірчі оцінки визначаються наступним чином:

У цьому контексті $\Pr(\text{об'єкт})$ позначає ймовірність присутності об'єкта в комірці сітки, а $\Pr(\text{клас} \mid \text{об'єкт})$ означає ймовірність присутності певного класу об'єктів у комірці за умови, що комірка містить об'єкт. $\Pr(\text{об'єкт}) 0,6$ означає 60%

ймовірність того, що коробка містить об'єкт. $\text{IoU} - \text{truth/pred}$ — це метрика IoU для справжніх і передбачених обмежувальних рамок. Нульовий показник достовірності означає, що в цій клітинці немає об'єкта. Цей показник достовірності використовується для розрахунку середньої точності (mAP) при визначеному пороговому значенні. Обмежувальні рамки з оцінками достовірності нижче порогового значення не враховуються. Кожна обмежувальна рамка характеризується такими атрибутами, як b_x , b_y , w , h і достовірність для кожного об'єкта. Координати (b_x , b_y) позначають центр прямокутника відносно меж комірки сітки, тоді як ширина (w) і висота (h) передбачені відносно всього зображення. Крім того, кожна комірка сітки передбачає ймовірності класу ($P_r(\text{Class})$) об'єктів, які вона містить. Наприклад, якщо комірка сітки передбачає 60% ймовірність ($\text{pr}(\text{Car})$), це передбачає 60% шанс містити автомобіль і 40% шанс не містити автомобіль[25].

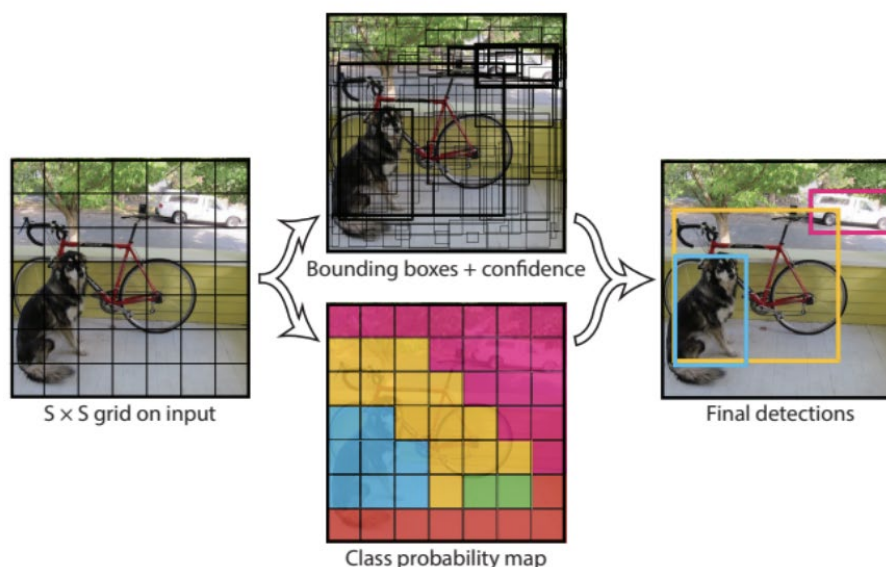


Рис. 2.10. YOLO модель

Як правило, сучасний детектор предметів складається з двох компонентів: хребта і голови. Магістраль попередньо навчена на ImageNet і включає такі опції, як VGG, ResNet і DenseNet для платформ GPU, а також SqueezeNet, MobileNet і ShuffleNet для платформ CPU. Детектори об'єктів можна класифікувати на два типи

на основі голови: одноступеневі детектори об'єктів і двоступеневі детектори об'єктів. Приклади двоступеневих моделей детекторів об'єктів включають серію R-CNN, швидші R-CNN, R-FCN і Libra R-CNN, тоді як одноступеневі моделі детекторів об'єктів включають YOLO, SSD і RetinaNet. Зараз детектори об'єктів містять додаткові шари між хребтом і головою, відомі як «шия» детектора об'єктів. Ці шари збирають карти функцій з різних етапів, і такі параметри, як Feature Pyramid Network (FPN), Path Aggregation Network (PANet), BiFPN і NAS-FPN, можуть служити шийкою мережі[26].

2.2.3 SSD detector

Сучасні системи виявлення об'єктів зазвичай використовують гіпотезу граничного поля, передискретизацію пікселів або об'єктів і підхід з використанням високоякісного класифікатора. Ці підходи домінують у тестах виявлення, найкращим прикладом яких є Faster R-CNN. Однак ці підходи часто вимагають значних обчислень і є надто повільними для роботи в реальному часі, навіть на висококласному обладнанні. Швидкість виявлення зазвичай вимірюється в секундах на кадр (SPF), а сучасні високоточні датчики, такі як Faster R-CNN, працюють зі швидкістю близько семи кадрів на секунду (FPS). Докладалися зусилля для прискорення роботи датчиків, спрямовані на різні етапи конвеєра виявлення, але значне збільшення швидкості часто відбувалося за рахунок зниження точності виявлення. Ключ до збільшення швидкості – запропонувати обмежувальну рамку, а потім усунути необхідність передискретизації пікселів або ознак. Незважаючи на схожі попередні спроби, автори запровадили кілька покращень, зокрема використання невеликого фільтра згортки для оцінки категорії об'єкта та зсувів обмежувальної рамки. Вони також використовують окремі оцінки для різних співвідношень і застосовують ці фільтри до декількох карт об'єктів, що дозволяє виявляти об'єкти на різних масштабах. Ці зміни, включаючи використання декількох шарів для прогнозування на різних масштабах, забезпечують високу точність при відносно низькій роздільній здатності і ще більше покращують показники виявлення. Хоча ці індивідуальні внески здаються

додатковими, разом вони значно покращують точність виявлення в реальному часі, перевершуючи YOLO і досягаючи 74,3% mAP для запропонованого SSD[27]. Крім того, вища якість виявлення розширює можливості застосування комп'ютерного зору в різних середовищах.

Метод Single Shot Multi-Box Detector (SSD) використовує мережу згортки, яка генерує послідовність обмежувальних рамок фіксованого розміру і оцінку асоціації, що вказує на наявність екземплярів класу об'єктів. Потім застосовується не максимальний крок придушення для отримання остаточного значення виявлення. В основі мережі лежить шар ініціалізації, що базується на стандартній архітектурі, яка зазвичай використовується для високоякісної класифікації зображень і називається базовою мережею. Ця базова сітка мінімізується перед шаром класифікації. Додаткові структури включаються в сітку для полегшення базових функцій виявлення: Багатомасштабні карти ознак для виявлення: згорнуті шари ознак додаються в кінці усіченої базової мережі. Розмір цих шарів поступово зменшується, щоб дозволити оцінювати виявлення на різних масштабах: на відміну від таких підходів, як Overfeat і YOLO, які мають справу з одномасштабними картами ознак, SSD має справу з картами ознак на різних масштабах.

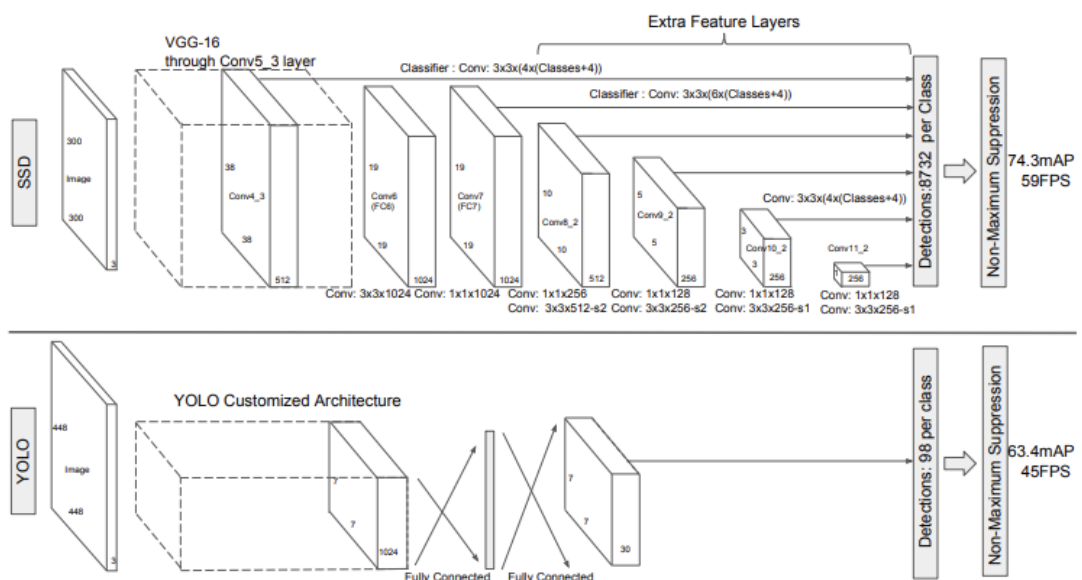


Рис. 2.11. Порівняння між двома моделями виявлення SSD і YOLO

Згортковий прогноз виявлення: кожен доданий шар ознак або, за бажанням, існуючий шар ознак у базовій мережі використовує згорткові фільтри для генерування фіксованого набору прогнозів виявлення. Архітектура мережі SSD, яка складається з невеликих ядер $3 \times 3 \times p$ для шару ознак $m \times n$ з p каналами. Ці ядра виробляють або оцінку категорії, або зсув форми відносно стандартних координат поля. Вихідне значення, оцінка або зсув, обчислюється в кожному місці, де застосовується ядро. Рамки за замовчуванням і співвідношення сторін: набір рамок за замовчуванням пов'язаний з кожною коміркою карти об'єктів, особливо на початку сітки. Ці рамки за замовчуванням є розбірними і забезпечують фіксоване положення відносно відповідної комірки. Прогноз для кожної комірки карти особливостей включає зміщення відносно форми рамки за замовчуванням і оцінку для кожного класу, що вказує на наявність екземплярів класу в межах цих рамок. Розрахунок включає s оцінок класів і 4 зміщення для кожної клітинки в даному місці, що в сумі дає $(s + 4)k$ фільтрів навколо кожного місця на карті особливостей. Це дає результат $(s + 4)kmp$ для карти об'єктів розміром $m \times n$. Коротше кажучи, ця архітектура є інноваційною в тому, що вона використовує вікно за замовчуванням, подібне до комбінованого блоку в Faster R-CNN, але застосовує його до декількох карт об'єктів з різною роздільною здатністю. Цей підхід ефективно охоплює діапазон можливих вихідних форм доменів, включаючи різні форми доменів за замовчуванням для різних карт об'єктів.

Основна відмінність у тренувальному SSD порівняно зі звичайним детектором, що використовує пропозиції регіонів, полягає в необхідності призначати правдиву інформацію конкретним виходам у межах попередньо визначеного набору виходів детектора. Подібна вимога існує для навчання YOLO та фази регіональних пропозицій Faster R-CNN і MultiBox. Після встановлення цього призначення функція втрат і зворотне поширення застосовуються наскрізним способом. Процес навчання також включає вибір полів і масштабів за замовчуванням для виявлення, а також стратегії жорсткого негативного аналізу та збільшення даних.

З точки зору стратегії відповідності під час навчання, мета полягає в тому, щоб визначити, які вікна за замовчуванням відповідають наземному виявленню істини, а потім навчити мережу відповідно. Кожне базове вікно істинності зіставляється з вікном за замовчуванням, що демонструє найбільше перекриття Jaccard, дотримуючись подібного підходу, як і в MultiBox. Однак, на відміну від MultiBox, у SSD ми додатково зіставляємо коробки за замовчуванням з будь-якою основною правдою з перекриттям Jaccard, що перевищує попередньо визначений поріг (0,5). Це спрощення в процесі навчання дозволяє мережі передбачати високі оцінки для кількох блоків за замовчуванням, що перекриваються, усуваючи обмеження вибору лише одного з максимальним перекриттям[28].

Загальна об'єктивна втрата функції є зваженою сумою втрати локалізації (loc) і втрати довіри (conf):

$$L(x, c, l, g) = \frac{1}{N} (L_{conf}(x, c) + \alpha L_{loc}(x, l, g)) \quad (2.8)$$

де N — кількість відповідних полів за замовчуванням.

Якщо $N = 0$, встановить втрату на 0. Локалізаційна втрата – це плавна втрата L1 між прогнозованою коробкою (l) і truth box (g). Подібно до Faster R-CNN, ми повертаємося до зсувів центрів (c_x, c_y) рамки за замовчуванням (d) і для її ширини (w) і висоти (h).

$$L_{loc}(x, l, g) = \sum_{i \in Pos} \sum_{m \in \{cx, cy, w, h\}} x_{ij}^k \text{smooth}_{L1}(l_i^m - \hat{g}_j^m) \quad (2.9)$$

$$\hat{g}_j^{cx} = (g_j^{cx} - d_i^{cx}) / d_i^w \quad \hat{g}_j^{cy} = (g_j^{cy} - d_i^{cy}) / d_i^h$$

$$\hat{g}_j^w = \log \left(\frac{g_j^w}{d_i^w} \right) \quad \hat{g}_j^h = \log \left(\frac{g_j^h}{d_i^h} \right)$$

Втрата довіри – це м'яка максимальна втрата для кількох класів довіри і ваговий член α встановлюється рівним одному, шляхом перехресної перевірки.

$$L_{conf}(x, c) = - \sum_{i \in Pos} x_{ij}^p \log(\hat{c}_i^p) - \sum_{i \in Neg} \log(\hat{c}_i^0) \quad \text{where} \quad \hat{c}_i^p = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)} \quad (2.10)$$

2.2.4 Feature Pyramid Networks

У контексті ієрархічної сегментації процес передбачає рекурсивне групування пікселів (або менших суперпікселів) у поступово більші, керуючись мірою подібності. Щоб досягти цього, ми використовуємо згорткові орієнтовані межі (COB) під час роботи із зображенням I , що призводить до ієрархічної сегментації, позначеної як набір розділів зображення (S_0, S_1, \dots, S_L) . Важливо розуміти, що кожен суперпіксель у (S_0) відповідає окремому пікселю у вхідному вхідному зображенні, (S_0) складається лише з одного суперпікселя, який представляє все зображення, а кількість суперпікселів у (S_L) і $(S_{\{l-1\}})$ відрізняється на одиницю (зокрема, один суперпіксель у (S_l) є об'єднанням двох суперпікселів у $(S_{\{l-1\}})$). Тут верхній індекс S вказує на рівень розділу в ієрархії сегментації. $S_{\{l-1\}}$ представляє найкращий набір суперпікселів в ієрархії, а суперпікселі в $S_{\{l+1\}}$ є об'єднанням суперпікселів в $S_{\{l\}}$. Щоб узгодити швидкість зниження дискретизації, характерну для згорткових нейронних мереж, рівні $\{l_1, l_2, l_3, l_4, l_5\}$ ретельно вибираються, щоб гарантувати, що кількість суперпікселів у $S_{\{l+1\}}$ становить $1/4$ цього в $S_{\{l\}}$. Отже, результуюча ієрархія суперпікселів S служить представленням структури частково-цілого у вхідному зображенні, полегшуючи відстеження зв'язків «предок-нащадок» між суперпікселями[29].

Кожен рівень відповідає рівню в ієрархії суперпікселів. Кожен суперпіксель в ієрархії суперпікселів відповідає вузлу графа на рівні відповідності піраміди графа. У результаті при переході від одного рівня до наступного більш високого рівня піраміди графа кількість вузлів зменшується в чотири рази. У піраміді діаграми визначено два типи ребер: контекстні ребра та ієрархічні ребра. Ребра контексту з'єднують сусідні вузли одного рівня, тоді як ієрархічні ребра з'єднують вузли різних рівнів, коли між відповідними суперпікселями існує зв'язок «предок-

нащадок». Контекстні межі сприяють поширенню контекстної інформації в межах одного рівня, тоді як ієрархічні межі заповнюють семантичні прогалини між різними рівнями. Важливо відзначити, що ієрархічні ребра є щільними, оскільки такі ребра існують між кожним вузлом і кожним із його предків і нащадків. Однак такі щільні з'єднання спричиняють значні витрати на обчислення та пам'ять. Таким чином, кожне ієрархічне ребро пов'язане з косинусною подібністю між вузловими характеристиками, а ієрархічні ребра організовано за косинусною подібністю ознак. Серед усіх ієрархічних ребер, пов'язаних із вузлом, ті, що знаходяться в нижніх 50%, систематично видаляються.

Графова нейронна мережа, яка називається GraphFPN, розроблена на основі графової піраміди. GraphFPN складається з двох типів шарів: контекстних та ієрархічних. Хоча ці шари мають однаковий набір вузлів у піраміді графа, вони використовують різні набори ребер графа. Контекстні рівні використовують виключно контекстні межі, тоді як ієрархічні рівні покладаються на скорочені ієрархічні межі. GraphFPN складається з контекстних шарів L1 на початку, ієрархічних шарів L2 у середині та контекстних шарів L3 у кінці. Важливо, що кожен рівень має власні параметри, які можна вивчати, відмінні від параметрів інших рівнів. Для простоти L1, L2 і L3 зберігають рівність у наших експериментах, а їхні конкретні значення обговорюються в дослідженнях абляції. Детальна конфігурація GraphFPN доступна в додаткових матеріалах. Незважаючи на те, що контекстні та ієрархічні рівні використовують різні межі, операції графової нейронної мережі (GNN) у цих рівнях залишаються ідентичними. Обидва типи шарів мають спільні механізми просторової та каналної уваги. Ми використовуємо механізм самоуважності з графічних мереж уваги як нашу просторову увагу.

$$\vec{h}'_i = \mathcal{M} \left(\vec{h}_i, \left\{ \vec{h}_j \right\}_{j \in \mathcal{N}_i} \right) \quad (2.11)$$

Механізм уваги каналу включає модуль уваги локального каналу на основі усереднення та модуль самоконтролю локального каналу. При локальній увазі каналу на основі середнього об'єднання вектори ознак вузла i та його сусідів усереднюються, щоб отримати вектор ознак $\vec{a}_i \in \mathbb{R}^C$. Усереднений вектор ознак пропускається через повністю зв'язний шар із сигмоїдальною активацією і виконується поелементне множення між виходом i та \vec{h}' .

$$\vec{h}_i'' = \sigma(\mathbf{W}_1 \vec{a}_i') \odot \vec{h}_i', \quad (2.12)$$

де σ — відноситься до сигмоїдної функції;

$\mathbf{W}_1 \in \mathbb{R}^{(c \times c)}$ — вагова матриця повнозв'язного шару, що вивчається;

\odot — означає поелементне множення.

Згорткові нейронні мережі мають здатність зберігати інформацію про місцезнаходження частин та об'єктів, що особливо корисно для виявлення об'єктів. З іншого боку, графові нейронні мережі краще вловлюють залежності між частинами та об'єктами на різних семантичних рівнях. Важливо відзначити, що в згорткових нейронних мережах функції магістральної та пірамідальної мережі (FPN) відповідають за різномасштабне кодування та декодування відповідно. На відміну від них, наш GraphFPN фокусується на багатомасштабному декодуванні. Отже, функції магістралі слугують вхідними даними для GraphFPN. Щоб використати переваги обох пірамідальних мереж ознак, злиття ознак поєднує кінцеві ознаки GraphFPN та згорткового FPN. Таким чином, для ініціалізації GraphFPN потрібно зіставити ознаки з кістяка, а кінцеві ознаки з GraphFPN потрібно зіставити зі згорнутою FPN перед процесом злиття. Багатомасштабні функціональні карти опорної ДФП та згорткової ДФП позначаються як $(C = \{C^1, C^2, C^3, C^4, C^5\})$ та $(P = \{P^1, P^2, P^3, P^4, P^5\})$, відповідно. Зауважте, що карта особливостей C являє собою остаточну карту особливостей п'яти етапів згортки в кістяку. Відображення згорткової нейронної мережі (CNN) в графову нейронну мережу (GNN): карта ознак хребта (" C^i ", карта ознак хребта (" $spine$ ") і

" S^i ", рівень " S^i " ("level"). Ця ознака складається з прямокутної сітки, де кожна комірка сітки відповідає прямокутній області оригінального вхідного зображення. На відміну від цього, суперпікселі в (S^i) зазвичай мають неправильну форму. Якщо декілька суперпікселів частково перекриваються, комірка сітки призначається найбільшому суперпікселю, що перекривається. В результаті такого призначення невелика колекція комірок сітки (C_{ik}) призначається тому самому суперпікселю в (S^i) (R^{ik}). Для цієї колекції виконується максимальне та мінімальне злиття, і результат злиття переноситься до повністю зв'язаного шару за допомогою активації ReLU[30].

3 ПРАКТИЧНА ЧАСТИНА

3.1 Огляд на CoreML моделі

CoreML – це високорівневий фреймворк, розроблений компанією Apple для ефективної та безпосередньої інтеграції моделей машинного навчання в застосунки, створені для різних платформ розробленими самою компанією Apple, включаючи iOS, iPadOS, macOS та watchOS. Відомий, у колі розробників, своєю простотою використання та продуктивністю, CoreML є чудовим інструментом для розробників, які працюють у екосистемі Apple. Серед ключових переваг CoreML – простота інтеграції, підтримка різних типів моделей, можливість робити висновки на пристрої, висока продуктивність та підтримка пакету Create ML, який дозволяє легко навчати моделі безпосередньо на пристроях компанії Apple. Фреймворк дозволяє розробникам створювати застосунки, які можуть виконувати розпізнавання образів, обробку природної мови та інші завдання машинного навчання, покращуючи користувацький досвід та функціональність застосунків. Простота інтеграції дозволяє розробникам швидко вбудовувати моделі в код, забезпечуючи велику гнучкість для роботи з різними типами моделей і алгоритмами машинного навчання. Ще однією важливою перевагою є можливість міркувати безпосередньо на пристрої, що підвищує швидкість і забезпечує конфіденційність даних. Висока продуктивність CoreML забезпечує зручний та ефективний спосіб обробки моделей на пристрої користувача, особливо в екосистемі Apple, що робить її ефективним інструментом для задач машинного навчання.

Переваги CoreML:

- простота інтеграції. Простота інтеграції є ключовою перевагою фреймворку CoreML. Розроблений Apple, CoreML забезпечує високий рівень простоти використання для розробників в екосистемах iOS, iPadOS, macOS і watchOS. Інтеграція моделей машинного навчання з CoreML є швидкою та безпечною. Розробники можуть легко

інтегрувати моделі у свої програми за допомогою зручного API CoreML. Фреймворк підтримує різні типи моделей, включаючи візуальне розпізнавання, обробку природної мови та інші завдання машинного навчання.

- підтримка різних типів моделей. Фреймворк CoreML вирізняється своєю універсальністю у підтримці різних типів моделей машинного навчання. Розробники можуть використовувати цей фреймворк для інтеграції різних моделей, таких як візуальне розпізнавання, обробка природної мови та системи рекомендацій.
- інференція на пристрої. Використання CoreML для машинного навчання дозволяє використовувати моделі машинного навчання безпосередньо на пристрої без необхідності постійного підключення до хмарних сервісів. Цей процес забезпечує швидку та ефективну обробку даних, гарантує високий рівень захисту даних і знижує навантаження на мережу. Такий підхід підвищує швидкість реагування та ефективність додатків і покращує користувацький досвід.
- висока продуктивність. CoreML забезпечує чудову продуктивність і дозволяє розробникам вбудовувати та використовувати моделі машинного навчання в програмах для платформ Apple. Ця структура дозволяє запускати оптимізовані операції висновку безпосередньо на пристрої, що забезпечує швидку обробку даних і ефективне використання ресурсів. Однією з головних переваг є вбудована оптимізація для чіпів Apple, таких як A14 Bionic і M1, що забезпечує максимальну продуктивність на пристроях Apple. Це дозволяє програмам миттєво реагувати на введення користувача та швидко виконувати завдання машинного навчання. Завдяки оптимізації завдань і паралельному виконанню CoreML значно підвищує ефективність використання обчислювальних ресурсів і забезпечує швидкий і точний висновок. Це стає ключовим фактором для додатків, де час відгуку та продуктивність мають вирішальне значення,

забезпечуючи високий рівень взаємодії з користувачем та ефективно використання ресурсів пристрою.

- підтримка пакету Create ML. CoreML інтегрований з Create ML і пропонує багато можливостей для розробників, які хочуть створити власні моделі машинного навчання. За допомогою Create ML можна легко створювати та навчати моделі без необхідності глибоких знань технічних деталей машинного навчання. Графічний інтерфейс користувача і можливість тренувати моделі на власних даних роблять побудову моделей інтуїтивно зрозумілою. Це дозволяє розробникам з будь-яким рівнем досвіду швидко і легко створювати моделі, оптимізовані для використання в їхніх додатках. Підтримка CoreML у Create ML ще більше спрощує розробникам створення моделей, оскільки параметри моделі можуть автоматично коригуватися під час навчання. Це робить машинне навчання доступним для розробників з різних галузей, а не лише для спеціалістів, і розширює коло людей, які можуть застосовувати методи машинного навчання у своїх проектах.

Недоліки CoreML:

- обмежена підтримка платформ. Обмежена підтримка платформ. CoreML розроблений для екосистеми Apple і має обмежену підтримку платформ. Це означає, що моделі, розроблені на CoreML, можна запускати безпосередньо на пристроях екосистеми Apple. Це обмеження є важливим при розгляді можливості використання моделей на інших платформах, таких як Android та веб-додатки.
- обмежена гнучкість. CoreML підтримує не всі типи моделей, а це означає, що розробники мають обмежену гнучкість у виборі архітектури. Фреймворк підтримує обмежену кількість алгоритмів машинного навчання та типів моделей. Це може впливати на вибір підходу до розв'язуваної задачі. Це може бути фактором для розробників, які використовують певну модель або потребують

певного алгоритму. Якщо обрана модель або алгоритм не входить до списку підтримуваних моделей, розробнику може знадобитися знайти альтернативне рішення або адаптуватися до можливостей CoreML.

- підтримуються не всі алгоритми. CoreML не підтримує всі алгоритми машинного навчання. Інфраструктура фреймворку обмежена і поки що не включає всі доступні алгоритми. Це може бути недоліком для розробників, які хочуть використовувати певні моделі або алгоритми, які наразі не підтримуються.

Щоб створити базову модель ML у Swift, потрібно вибрати або навчити модель машинного навчання. Якщо обрати стандартну модель, можна скористатися бібліотекою, такою як TensorFlow або PyTorch, або вибрати одну з моделей, наданих Apple. Якщо є потреба навчити власну модель, використовуйте відповідну структуру, таку як TensorFlow або PyTorch. Вивчіть правильний набір даних для свого завдання. Навчена модель повинна бути підготовлена до перетворення у формат CoreML. Наприклад, використовуйте інструмент CoreMLTools для перетворення моделі з TensorFlow на сумісну з CoreML. Використовуйте створений клас Xcode Swift, щоб інтегрувати отриману модель у свій проект Swift. Це дозволяє проектам Swift використовувати модель CoreML для прогнозування та виконання завдань машинного навчання.

Створення базової моделі ML у Swift – це складний процес вибору або навчання моделі машинного навчання та перетворення її в основний формат ML для інтеграції в проект Swift. З першого кроку потрібно вирішити, чи потрібно використовувати стандартну модель, чи хочете навчити свою власну. Якщо обрали рішення, можна використати бібліотеки, такі як TensorFlow або PyTorch, або вибрати одну з декількох моделей Apple. Якщо потрібно навчити власну модель, потрібно використовувати фреймворки, такі як TensorFlow або PyTorch, для навчання з набором даних, який відповідає конкретному завданню. Після завершення навчання, який перетворює модель у формат CoreML. Для переходу до наступного кроку потрібно використати інструмент "CoreMLTools". Наприклад,

якщо модель навчається за допомогою TensorFlow, використовуйте CoreMLTools, щоб перетворити її у формат CoreML. Це гарантує, що модель сумісна і може бути інтегрована в проект Swift. Після перетворення інтегрувати модель у проект Swift. Коли перетягується файл MLmodel в проект Xcode, Xcode автоматично створює клас Swift, що представляє модель. Цей створений клас дозволяє робити прогнози та вирішувати проблеми машинного навчання в Swift. Наприклад, можна використовувати згенерований клас Swift для створення екземпляра моделі, підготовки вхідних даних, прогнозування та виведення. Це дозволяє використовувати модель core ML для вирішення певних завдань у вашому проекті Swift. Таким чином, цей процес, від вибору та навчання моделей до їх перетворення та інтеграції в проекти Swift, є ключем до успішного управління моделями машинного навчання в середовищі Apple.

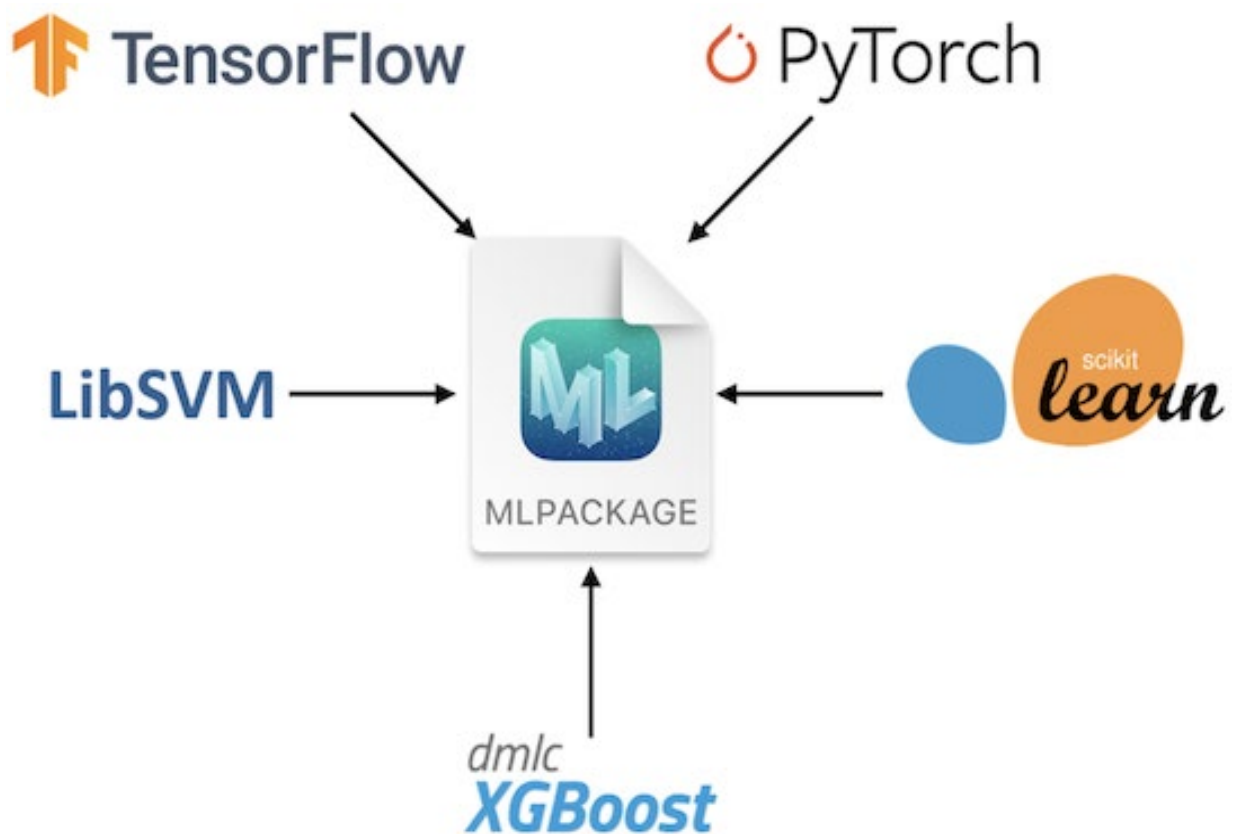


Рис. 3.1. Складові CoreML моделі

Створення базової моделі ML у Swift – це складний процес, який починається з вибору та навчання моделі машинного навчання і закінчується інтеграцією цієї моделі в проект Swift. Перший крок – вирішити, чи використовувати стандартні моделі, такі як моделі tensorflow, PyTorch або Apple, або тренувати власні. Важливо вибрати найкращий метод для конкретного завдання та врахувати деталі набору даних. Наступний крок полягає в тому, щоб перетворити та інтегрувати модель у проект Swift. Після навчання модель перетворюється у формат CoreML за допомогою інструменту CoreMLTools. Наприклад, якщо модель навчається в tensorflow, модель перетворюється у формат CoreML. В результаті виходить файл ML model інтегрується в проект Xcode, і автоматично створюється клас Swift, що представляє цю модель. Створені класи дозволяють ефективно керувати моделлю, готувати вхідні дані, робити прогнози і обробляти вихідні дані в коді Swift. Цей процес інтеграції дозволяє моделі CoreML коректно працювати в реальному середовищі програми і використовувати машинне навчання безпосередньо в середовищі Swift. При розробці програми для платформи Apple важливо враховувати деталі завдання і гнучкість CoreML для повної реалізації свого потенціалу.

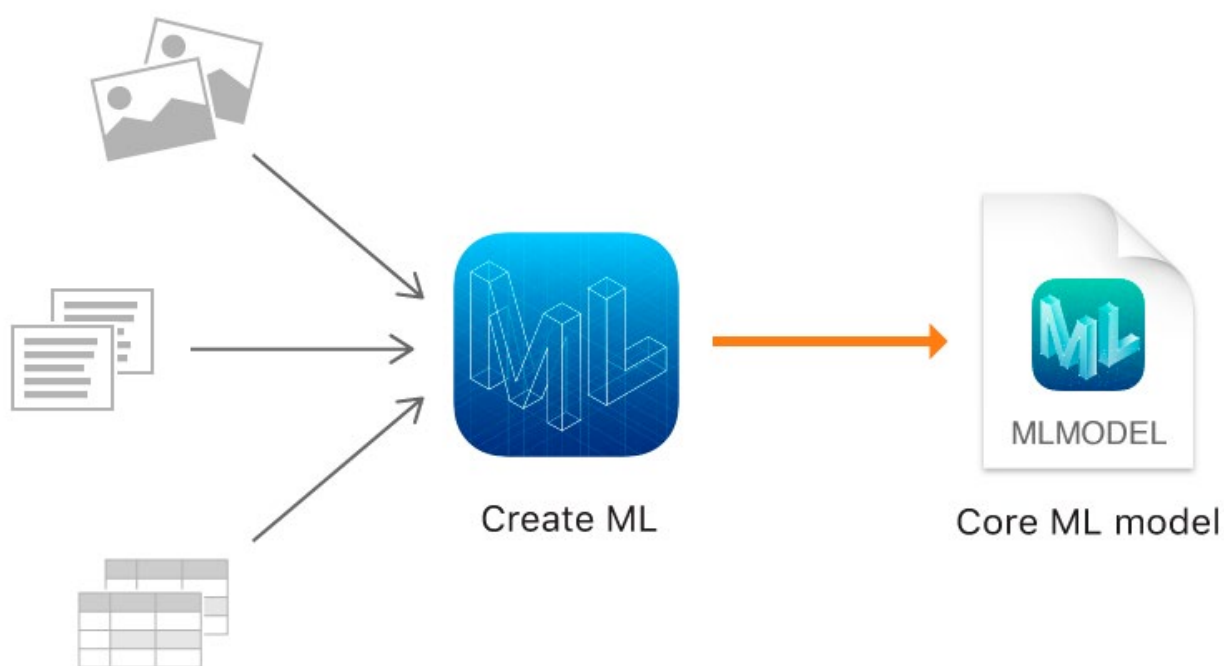


Рис. 3.2. Ілюстрація кроків до отримання готової CoreML моделі

Створення базової моделі CoreML у Swift – це багатоетапний процес, який передбачає вибір або навчання моделі та інтеграцію її у реальну програму. На останньому етапі (після перетворення) важливо підготувати модель до використання. Використовуйте такі інструменти, як CoreMLTools, щоб перевірити та перевірити свою модель. Переконайтеся, що він сумісний і правильний у різних сценаріях, і що немає перевантаження або неправильної відповіді. Коли перевірка пройшла успішно, далі можна перейти до поетапної інтеграції та розробки застосунку. Потрібно додати створений клас моделі Swift до проекту Xcode. Він використовується для запуску моделі, надсилання вхідних даних та отримання прогнозів. Для взаємодії потрібно створити інтерфейс для користувача. Під час інтеграції з Xcode необхідно переконатись, що програма може ефективно використовувати машинне навчання. Загальний процес створення базової моделі ML у Swift починається з розуміння завдання та обраної вами моделі, а потім переходить до навчання, перетворення, навчання та інтеграції у вашу програму. Використовуючи цей цикл, можна використовувати можливості машинного навчання, щоб надати застосунки Swift потужні інтелектуальні можливості, зробити їх привабливими і продуктивними для кінцевих користувачів.

3.2 Створення CoreML моделі

Створення моделей машинного навчання з використанням CoreML значно розширює можливості машинного навчання, такі як розуміння зображень та обробка природної мови. Щоб почати будувати модель CoreML, важливо чітко визначити цілі, яких потрібно досягти за допомогою моделі. Наступним важливим кроком – є збір відповідного та репрезентативного набору даних, який буде використовуватися для навчання моделі. Після того, як розробник визначив цілі та зібрав дані, можна обрати відповідний алгоритм машинного навчання для виконання завдання. Важливо ретельно підібрати найбільш підходящий алгоритм для конкретної ситуації, оскільки від цього буде залежати ефективність моделі. У світі розробки на платформі Apple існує потужний інструмент Create ML, який

спрощує процес побудови моделей на технології CoreML. Цей інтуїтивно-зрозумілий інтерфейс дозволяє розробникам без глибоких знань у галузі машинного навчання створювати та навчати моделі за допомогою простого візуального підходу. Create ML підтримує різні типи моделей. Інструмент дозволяє користувачам завантажити набір даних, вибрати відповідний алгоритм і навчити модель, перевіряючи результати в режимі реального часу.

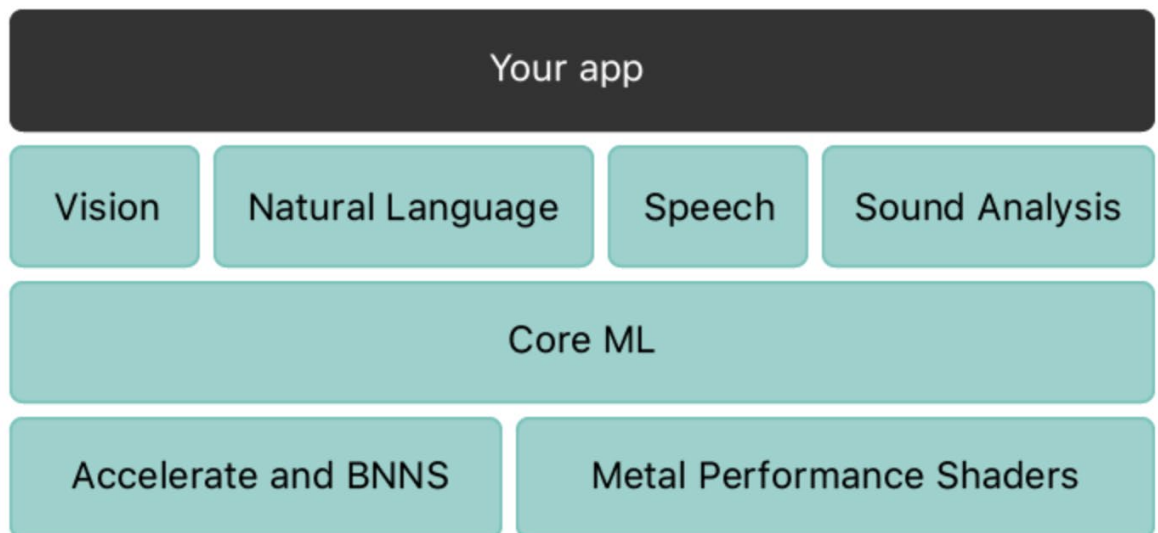


Рис. 3.3. Приклад CoreML API

Після вибору алгоритму та збору набору даних наступним кроком є навчання моделі в Create ML. Цей етап включає навчання моделі, визначення параметрів і передачу даних для аналізу результатів. Create ML дозволяє розробникам легко завантажувати набори даних і вибирати параметри навчання. Інтерфейс дозволяє вказати тип завдання (класифікація, регресія тощо), вибрати алгоритм і задати параметри моделі. Важливим кроком є ретельний вибір гіперпараметрів для оптимізації продуктивності. Під час навчання Create ML дозволяє візуально оцінювати результати на основі показників продуктивності та розподілу помилок. Це дозволяє розробникам швидше виявити проблеми і внести необхідні корективи для покращення моделі. Оптимізація моделі є важливим кроком після навчання. Create ML автоматично оптимізує моделі для використання в середовищі CoreML, але розробники також можуть вручну налаштувати різні параметри, такі як розмір

моделі та швидкість передачі даних. Це важливо для забезпечення продуктивності моделі на мобільних пристроях з обмеженими ресурсами. Таким чином, навчання та оптимізація моделей у Create ML характеризується високою зручністю використання та можливістю визначення найкращих параметрів для досягнення високої продуктивності в середовищі CoreML.



Рис. 3.4. Блок-схема CoreML

Щоб запустити програму, спочатку потрібно відкрити середовище розробки Xcode. Потім обрати у Xcode у рядку меню, щоб відкрити випадаючий список. У цьому випадаючому списку обрати пункт Відкрити інструменти розробника, а потім – Створити ML. З'явиться вікно пошуку. Потрібно обрати новий документ у нижній частині вікна, щоб розпочати новий проект CreateML.

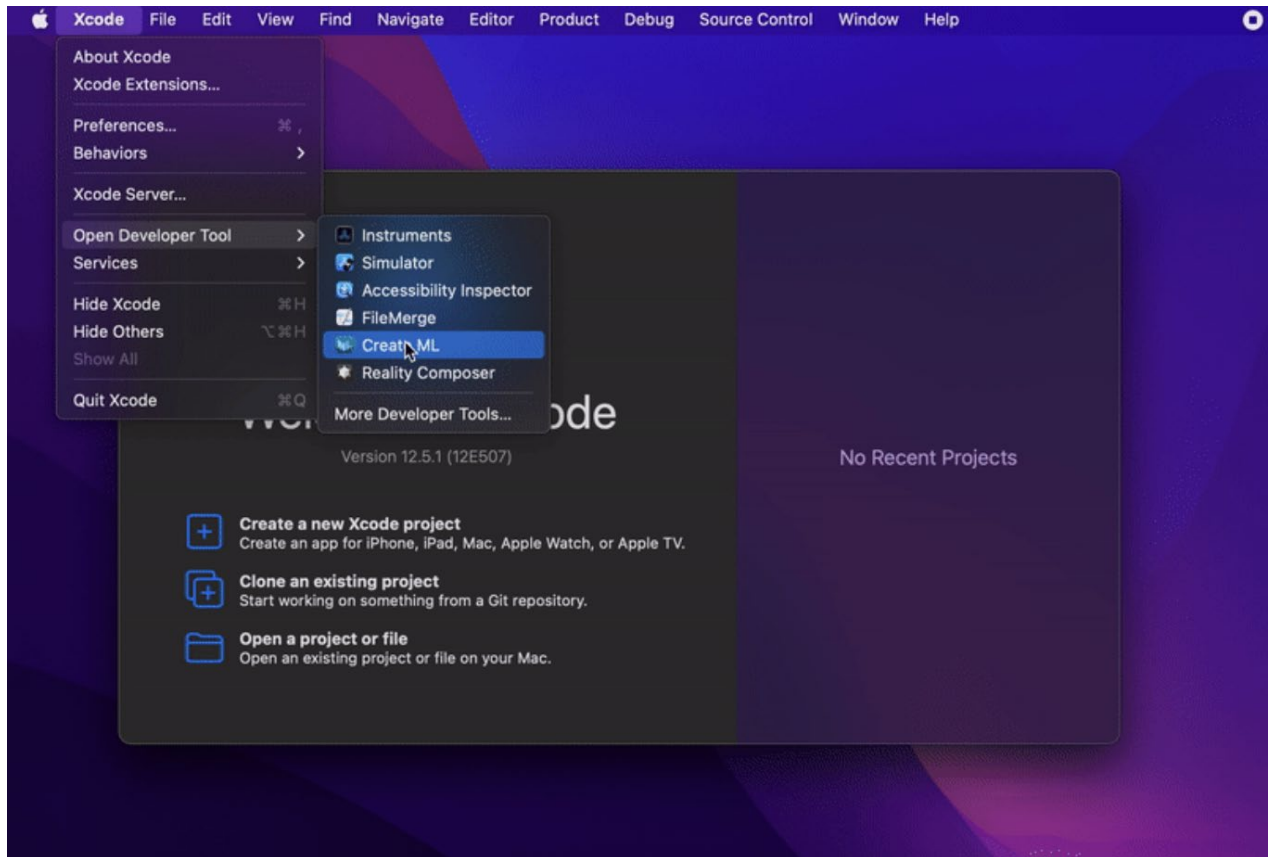


Рис. 3.5. Створення CreateML моделі у XCode

Відразу після створення моделі потрібно одразу ж обрати тип моделі. Інженери Apple попіклувались про рядових розробників, та створили вже готові шаблони моделей. Для цієї дослідної роботи були використані 2 шаблони: Image Classification та Object Detection.

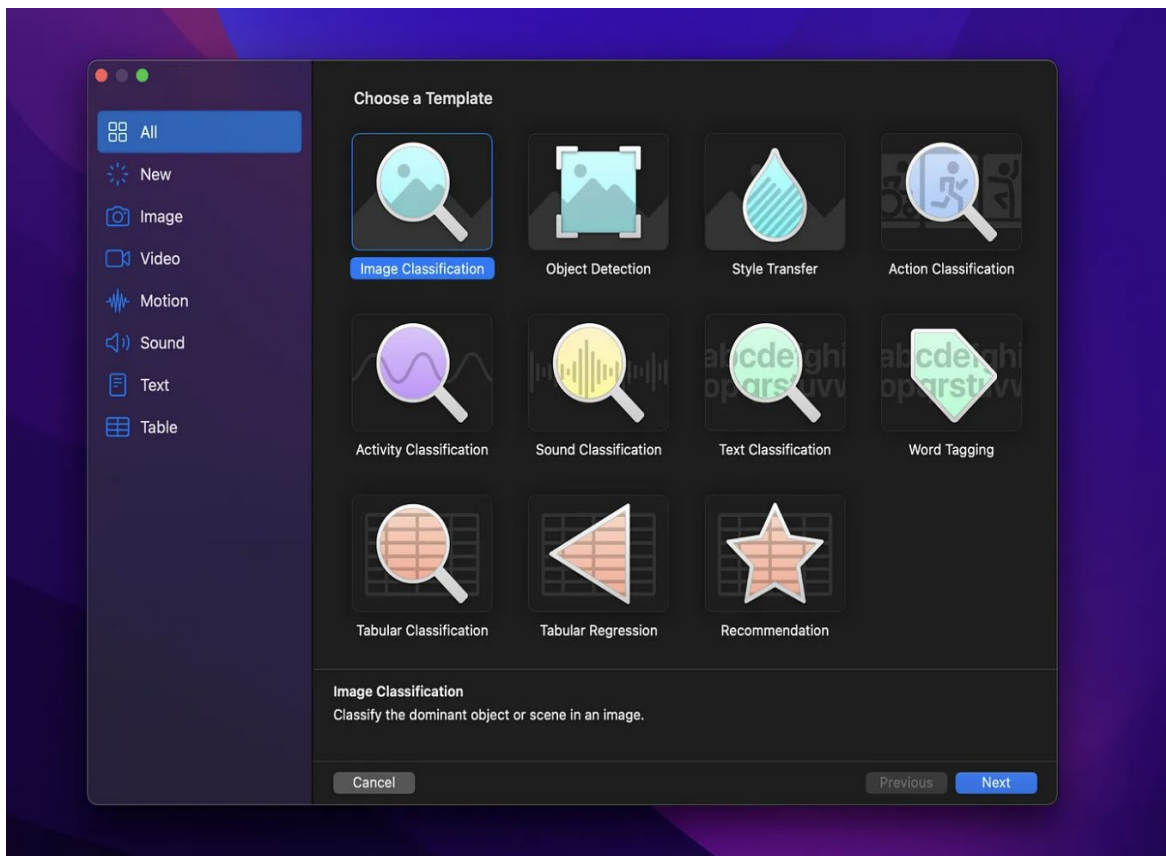


Рис. 3.6. Список усіх доступних шаблонів ML від Apple

Після вибору потрібного шаблону для подальших дій розробника перекидає на сторінку з навчанням моделі. Під заголовком "Дані" є три розділи: "Навчальні дані", "Перевірочні дані" та "Тестові дані". Навчальні дані використовуються для навчання моделі, тоді як перевірочні дані використовуються для забезпечення зворотного зв'язку про те, наскільки точно модель прогнозує невідомі дані під час навчання. Тестові дані не є обов'язковими, але можуть бути використані для перевірки точності моделі після завершення навчання. Наступний заголовок – "Параметри". Максимальна кількість ітерацій у цій моделі – лише одна. Це максимальна кількість разів, коли навчальні дані можуть бути передані моделі під час навчання. Заголовок "Доповнення" надає різні способи маніпулювання даними перед їх імпортуванням у модель. Маніпулювання даними під час виконання зменшує ймовірність того, що модель запам'ятає дані замість того, щоб їх вивчити.

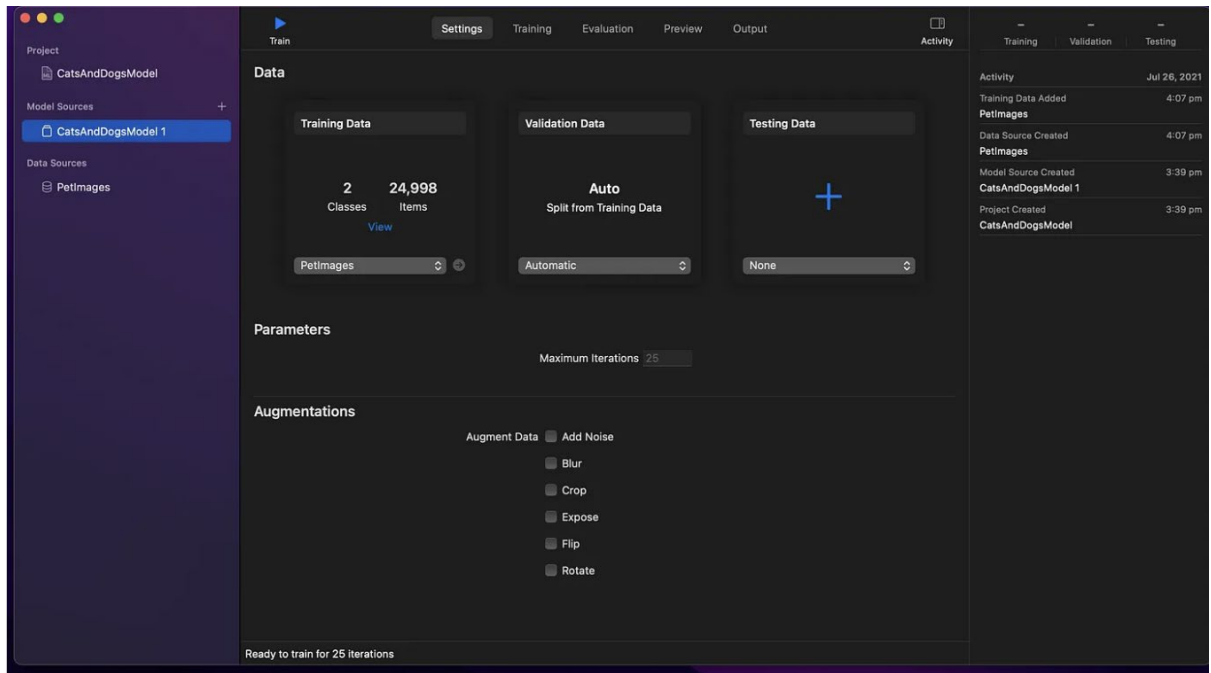


Рис. 3.7. Демонстрація доданих матеріалів у модель для навчання

Далі відбувається тренування моделі, яке займає якийсь час. Зазвичай, швидкість тренування залежить від кількості матеріалів для тренування моделі та від девайсу на якому і відбувається тренування.

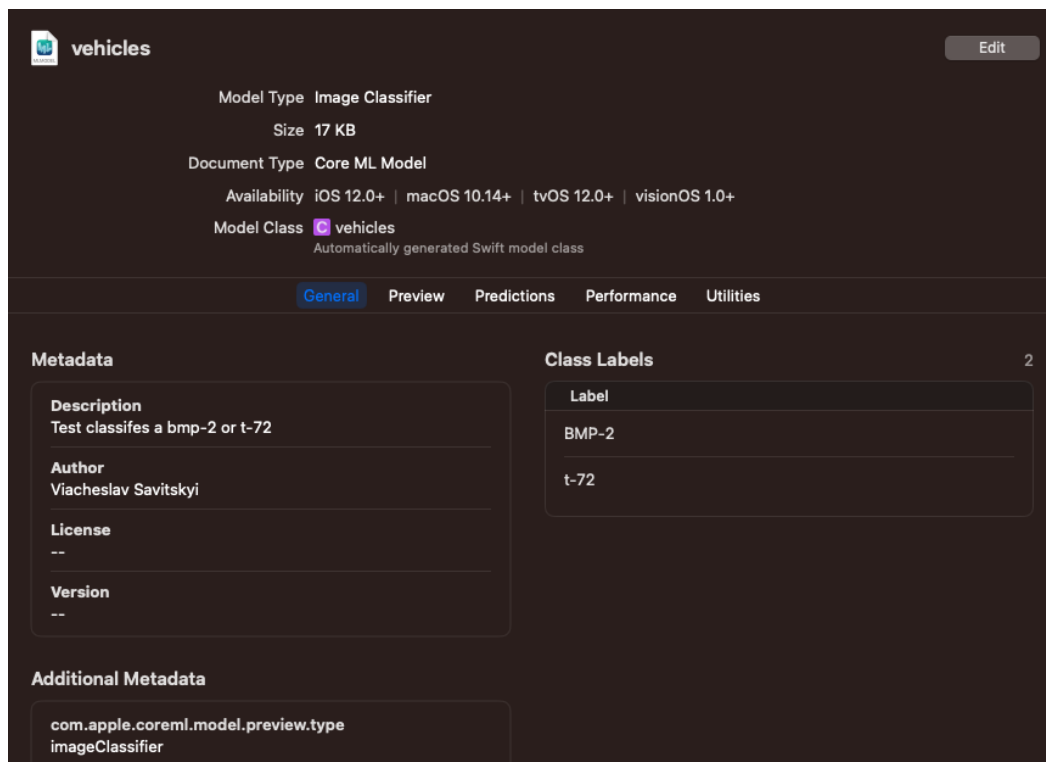


Рис. 3.8. Демонстрація навченої CoreML моделі

3.3 Інтеграція CoreML моделі з мовою програмування Swift

Насамперед потрібно додати саму створену модель з попереднього розділу у проект. Це робиться надзвичайно просто. Всього потрібно обрати декілька галочок, як показано на рисунку нижче.

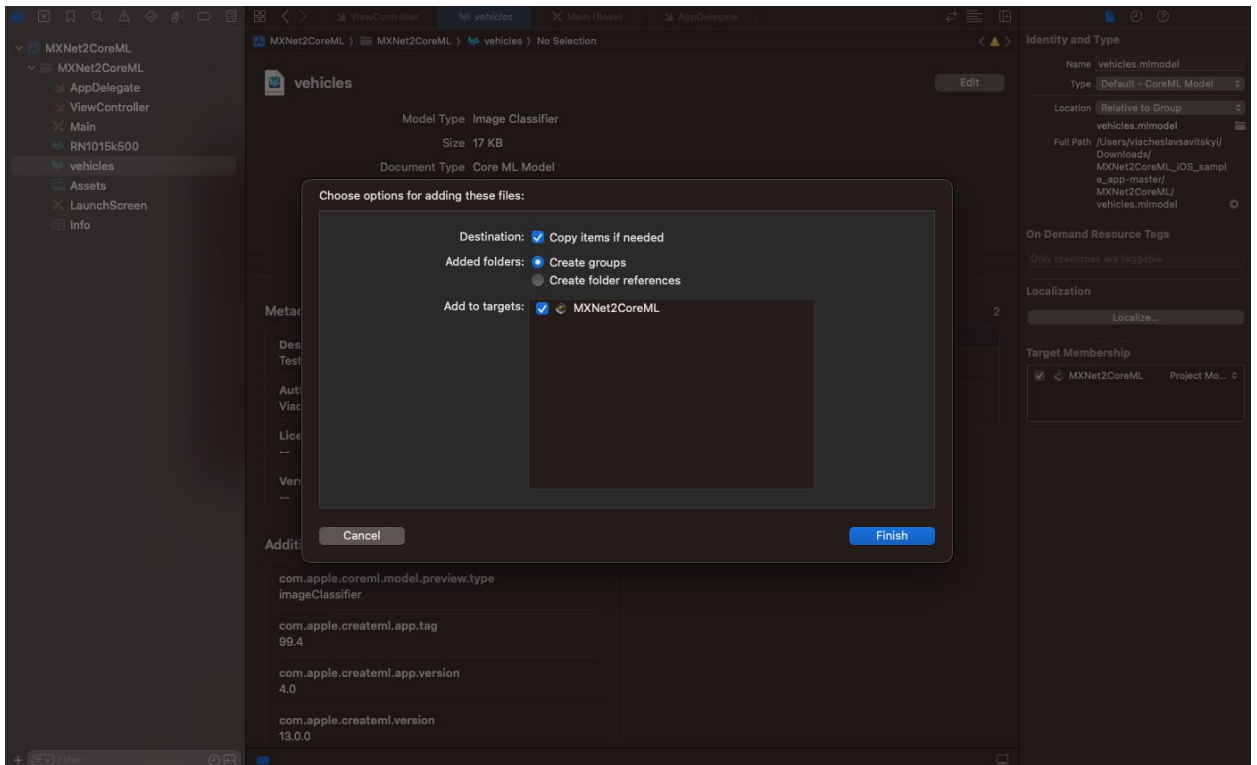


Рис. 3.9. Вибір опцій при додаванні моделі у проект

Для того аби застосунок міг визначити, який тип техніки зображений на фото та для того аби застосунок міг проаналізувати приблизне місце розташування техніки на фото – потрібно було реалізувати 2 CoreML моделі. Аби мати можливість коду звертатись до CoreML моделей потрібно імпортувати фреймворк Vision.


```

15
16 import UIKit
17 import Vision
18 import MapKit
19
20
21 class PredictionLocation: NSObject, MKAnnotation{
22     var identifier = "Prediction location"
23     var title: String?
24     var coordinate: CLLocationCoordinate2D
25
26     init(name:String,lat:CLLocationDegrees,long:CLLocationDegrees){
27         title = name
28         coordinate = CLLocationCoordinate2DMake(lat, long)
29     }
30 }
31
32 class PredictionLocationList: NSObject {
33     var place = [PredictionLocation]()
34     override init(){
35         place += [PredictionLocation(name:"1",lat: 0, long: 0)]
36         place += [PredictionLocation(name:"2",lat: 1, long: 1)]
37         place += [PredictionLocation(name:"3",lat: 2, long: 2)]
38     }
39 }

```

Рис. 3.10. Демонстрація імпорту фреймворку Vision

Фреймворк Vision – це фреймворк Apple, спеціально розроблений для обробки зображень і комп'ютерного зору на iOS і macOS. Vision дозволяє легко інтегрувати розпізнавання облич, розпізнавання об'єктів, аналіз тексту та інші технології обробки зображень у свої застосунки. Фреймворк надає набір інструментів для створення потужних застосунків з використанням технологій штучного інтелекту та машинного навчання. Vision включає алгоритми, які можуть ефективно розпізнавати та аналізувати великі обсяги візуальної інформації. Перевагою реалізації проекту саме методами, які надає компанія Apple, є те, що можна отримати відсоткове співвідношення впізнаваності місця розташування і типу військової техніки.



Рис. 3.11. Демонстрація впізнаваності моделі місця розташування військової техніки

ВИСНОВКИ

У процесі виконання дипломної роботи була здійснена комплексна розробка та впровадження інтелектуальної системи розпізнавання зображень та геолокації військової техніки з використанням передових технологій машинного навчання. Запропонована система є дієвим засобом підвищення ефективності виконання військових операцій та підвищення ефективності прийняття військових рішень. Однією з ключових особливостей розробленої системи є точне виявлення військової техніки за допомогою фотозйомки та комп'ютерного зору. Машинне навчання, застосоване під час роботи системи, дозволяє автоматизувати процес аналізу та класифікації зображень, що значно прискорює визначення місцезнаходження та типу техніки противника. Експерименти та випробування підтвердили високу точність і швидкість роботи системи при практичному використанні. Запропонована інтелектуальна система не тільки відповідає сучасним вимогам до засобів розвідки та управління бойовими діями, але й відкриває перспективи подальшого розвитку та вдосконалення такої техніки в майбутньому. У цілому дане дослідження вирішує практичну задачу вдосконалення військових систем управління та навігації, а отримані результати можуть бути використані у практичних військових застосуваннях для підвищення ефективності та безпеки військових дій.

ПЕРЕЛІК ПОСИЛАНЬ

1. James Stavridi - Ukraine War May Become a Proving Ground for AI. URL: https://www.washingtonpost.com/business/2023/05/31/ukraine-war-latest-us-can-bring-ai-quantum-computing-to-battle/63d221c4-ffa9-11ed-9eb0-6c94dcb16fcf_story.html
2. Ми Україна - Як штучний інтелект допомагає в сучасних війнах: чи зможе він замінити військових на фронті. URL: <https://weukraine.tv/yak-shtuchnyj-intelekt-dopomagaye-v-suchasnyh-vijnah-chy-zmozhe-vin-zaminyty-vijskovyh-na-fronti/>
3. Олександра Бодняк - Війна і роботи. URL: https://zaxid.net/viyna_z_rosiyeyu_yak_ukrayintsi_vikoristovuyut_shtuchniy_intelekt_n1549819
4. Military.com - How the Army and Air Force Integrate AI Learning Into Combat Training. URL: <https://www.military.com/military-life/how-army-and-air-force-integrate-ai-learning-combat-training.html>
5. Мілітарний - У США тестують систему цілевказання зі штучним інтелектом ATLAS на танку M1 Abrams. URL: <https://mil.in.ua/uk/news/u-ssha-testuyut-systemu-tsilevkazannya-zi-shtuchnym-intelektom-atlas-na-tanku-m1-abrams/>
6. Defense News - German military's metaverse promises virtual foes with an AI punch. URL: <https://www.defensenews.com/global/europe/2023/09/08/german-militarys-metaverse-promises-virtual-foes-with-an-ai-punch/>
7. GhostPlay. URL: <https://www.ghostplay.ai/>
8. Taiwo Ayodele - Types of Machine Learning Algorithms. URL: https://www.researchgate.net/publication/221907660_Types_of_Machine_Learning_Algorithms
9. IT-Enterprise - Machine Learning, ML. URL: <https://www.it.ua/knowledge-base/technology-innovation/machine-learning>
10. Віктор Синєглазов, Олена Чумаченко - Методи та технології напівкерованого навчання: курс лекцій URL: <https://ela.kpi.ua/bitstream/123456789/52775/1/Metody.pdf>

11. yaneya - Машинне навчання простими словами. Частина 2. URL: <http://mmf.com.ua/ar/1743>
12. Mehryar Mohri – Advanced Machine Learning. URL: https://cims.nyu.edu/~mohri/amls/aml_transduction.pdf
13. Olivier Bousquet - Transductive Learning: Motivation, Model, Algorithms. URL: https://is.mpg.de/fileadmin/user_upload/files/publications/pdf2527.pdf
14. Hiroki Ouchi, Jun Suzuk , Kentaro Inu - Transductive Learning of Neural Language Models for Syntactic and Semantic Analysis. URL: <https://aclanthology.org/D19-1379.pdf>
15. Geoffrey Webb - Feature Based Modelling. URL: https://www.researchgate.net/publication/2781458_Feature_Based_Modelling
16. Firoozeh Abolhasani Zadeh, Mohammadreza Vazifeh Ardalani, Ali Rezaei Salehi - An Analysis of New Feature Extraction Methods Based on Machine Learning Methods for Classification Radiological Images. URL: https://www.researchgate.net/publication/360862631_An_Analysis_of_New_Feature_Extraction_Methods_Based_on_Machine_Learning_Methods_for_Classification_Radiological_Images
17. Haicang Zhang, Yufeng Shen - Template-based prediction of protein structure with deep learning . URL: https://www.researchgate.net/publication/347993262_Template-based_prediction_of_protein_structure_with_deep_learning
18. Thomas Hossler - A Deep Learning approach to Template Matching. URL: <http://cs231n.stanford.edu/reports/2017/pdfs/817.pdf>
19. Amir Mosavi, Sina Ardabili, Annamaria R. Varkonyi-Koczy - List of Deep Learning Models. URL: https://www.researchgate.net/publication/338565420_List_of_Deep_Learning_Models
20. Russ Salakhutdinov - Deep Learning Supervised Learning. URL: https://www.cs.cmu.edu/~rsalakhu/talk_Simons_part1_pdf.pdf
21. Cuiyin Liu, Jishang Xu, Feng Wang - A Review of Keypoints' Detection and Feature Description in Image Registration. URL:

https://www.researchgate.net/publication/356709780_A_Review_of_Keypoints'_Detection_and_Feature_Description_in_Image_Registration

22. Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik - R-CNN for Object Detection. URL: https://courses.cs.washington.edu/courses/cse590v/14au/cse590v_wk1_rcnn.pdf

23. Sabari Prasad - DEEP LEARNING TERM WORK. URL: https://www.academia.edu/43405129/DEEP_LEARNING_TERM_WORK_1_R_CNN_2_FAST_R_CNN_3_FASTER_R_CNN_4_YOLO_by

24. Muhammad Tamjid Rahman - DRIVING-SCENE IMAGE CLASSIFICATION USING DEEP LEARNING NETWORKS: YOLOV4 ALGORITHM. URL: <https://www.diva-portal.org/smash/get/diva2:1689752/FULLTEXT01.pdf>

25. Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi - You Only Look Once: Unified, Real-Time Object Detection. URL: <https://arxiv.org/pdf/1506.02640.pdf>

26. Alejandro Aldaya, Antonio Monleon-Getino, Núria Flor Arnau, Joaquin Justel Pizarro - Object Detection with Machine Learning a YOLO review. URL: https://www.researchgate.net/publication/372419433_Object_Detection_with_Machine_Learning_a_YOLO_review

27. Wei Liu, Dragomir Anguelov, Dumitru Erha , Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg - SSD: Single Shot MultiBox Detector. URL: <https://www.cs.unc.edu/~wliu/papers/ssd.pdf>

28. Anurag Gupta, Akash Raj, Ayushman Pathak, Darshan Yadav - Real-Time Object Detection Using SSD MobileNet Model of Machine Learning. URL: https://www.researchgate.net/publication/371429419_Real-Time_Object_Detection_Using_SSD_MobileNet_Model_of_Machine_Learning

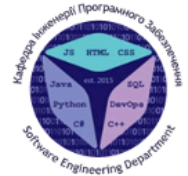
29. T A Chavan, Dheeraj Lokhande - USE OF A FEATURE PYRAMID NETWORK FOR OBJECT DETECTION-A DEEP LEARNING TECHNOLOGY. URL: https://www.researchgate.net/publication/368666393_USE_OF_A_FEATURE_PYRAMID_NETWORK_FOR_OBJECT_DETECTION-A_DEEP_LEARNING_TECHNOLOGY

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

(Презентація)



ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ



КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

МАГІСТЕРСЬКА РОБОТА

«РОЗРОБКА ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ ТА ГЕОЛОКАЦІЇ ВІЙСЬКОВОЇ ТЕХНІКИ ЗА ДОПОМОГОЮ ТЕХНОЛОГІЙ МАШИННОГО НАВЧАННЯ»

Виконав: студент групи ПДМ-64, Савіцький Вячеслав Андрійович

Керівник: PhD, доцент кафедри ІПЗ Дібрівний Олесь Андрійович

Київ - 2024

МЕТА, ОБ'ЄКТ, ПРЕДМЕТ ДОСЛІДЖЕННЯ

Мета роботи: підвищення якості розпізнавання зображень та геолокації військової техніки за допомогою технологій машинного навчання.

Об'єкт дослідження: процес розпізнавання зображень та геолокації військової техніки.

Предмет дослідження: методи та засоби розпізнавання зображень та геолокації військової техніки на основі технологій машинного навчання.

ВИДИ ПРОВЕДЕННЯ РОЗВІДУВАЛЬНИХ ЗАХОДІВ ЗА ДОПОМОГОЮ МАШИННОГО НАВЧАННЯ

Вид розвідувального заходу	Застосування машинного навчання	Зразки виявлення
Дані від сенсорів	1. Інфрачервона та лазерна технологія	1. Виявлення теплових слідів від техніки 2. Використання лазерів для вимірювання відстаней
Обробка даних в реальному часі	1. Системи реального часу	1. Виявлення та відстеження техніки в реальному часі
Використання радіоелектронних технологій	1. Розпізнавання сигналів 2. Спектральний аналіз	1. Визначення типу техніки за радіосигналами 2. Аналіз спектральних характеристик радіосигналів
Використання даних від дронів	1. Дрони для зйомки 2. Аналіз потоку даних	1. Зйомка високоякісних зображень військової техніки 2. Обробка даних в реальному часі для виявлення техніки
Обробка зображень та відео	1. Розпізнавання об'єктів 2. Обробка в реальному часі	1. Виявлення танків на фото з дронів 2. Аналіз відеопотоку для виявлення руху техніки
Аналіз супутникових знімків	1. Використання геоспеціальних даних 2. Картографування об'єктів	1. Виявлення техніки на супутникових знімках 2. Визначення розташування техніки на карті

3

ХАРАКТЕРНІ ОЗНАКИ ВІЙСЬКОВОЇ ТЕХНІКИ

Тип військової техніки	Характерні ознаки	Характерні габаритні ознаки
Бронетехніка	Захищена бронею, має колеса або гусениці.	Довжина: від 3 до 70 метрів. Ширина: від 2 до 10 метрів. Висота: від 2 до 5 метрів.
Артилерійські системи	Зазвичай має велику ствольну частину, розташовану на платформі.	Довжина ствола: від 2 до 100 метрів. Діаметр ствола: від 5 до 200 міліметрів. Ширина платформи: від 2 до 10 метрів.
Інженерна техніка	Зазвичай має велику вантажопідйомність, має спеціальне обладнання для виконання інженерних завдань.	Довжина: від 5 до 50 метрів. Ширина: від 2 до 10 метрів. Висота: від 2 до 5 метрів. Вантажопідйомність: від 10 до 1000 тонн.
Автомобільна техніка	Може мати колеса, гусениці або колісні ланцюги.	Довжина: від 2 до 20 метрів. Ширина: від 2 до 10 метрів. Висота: від 2 до 5 метрів. Вантажопідйомність: від 10 до 1000 тонн.
Зенітно-ракетні установки	Зазвичай має пускові установки для зенітних ракет, розташовані на платформі.	Довжина: від 5 до 20 метрів. Ширина: від 2 до 10 метрів. Висота: від 2 до 5 метрів. Кількість ракет: від 1 до 50.

4

ТИПИ АЛГОРИТМІВ МАШИННОГО НАВЧАННЯ

Тип вивчення	Приклади алгоритмів
Трансдукція	Meta-Learning Multi-Task Learning Zero-Shot Learning Knowledge Distillation
Некероване навчання	Агломеративна кластеризація Асоціативні правила K-Means Principal Component Analysis
Напівкероване навчання	Метод самонавчання Deep Convolutional Neural Networks Deep Recurrent Neural Networks Deep Autoencoders
Навчання з підкріпленням	Actor-Critic Trust Region Policy Optimization Asynchronous Advantage Actor-Critic Policy Gradient Methods
Кероване навчання	Q-навчання Deep Deterministic Policy Gradients Proximal Policy Optimization SVMs

5

МАТЕМАТИЧНА МОДЕЛЬ

$$M_p = \frac{m_p}{m_i} \frac{1 - P_x}{\sqrt{1 - P_x^2 + 0,25}}$$

M_p - чисельник масштабу;

m_p – середня квадратична похибка визначення положення межових точок;

m_i – точність фотограмметричних побудов (0,005 мм);

P_x – задане поздовжнє перекриття знімків у частках одиниці, найчастіше його приймають 0,6.

$$H(X) = -\sum_{j=1}^n P(x_j) \lg_2 P(x_j)$$

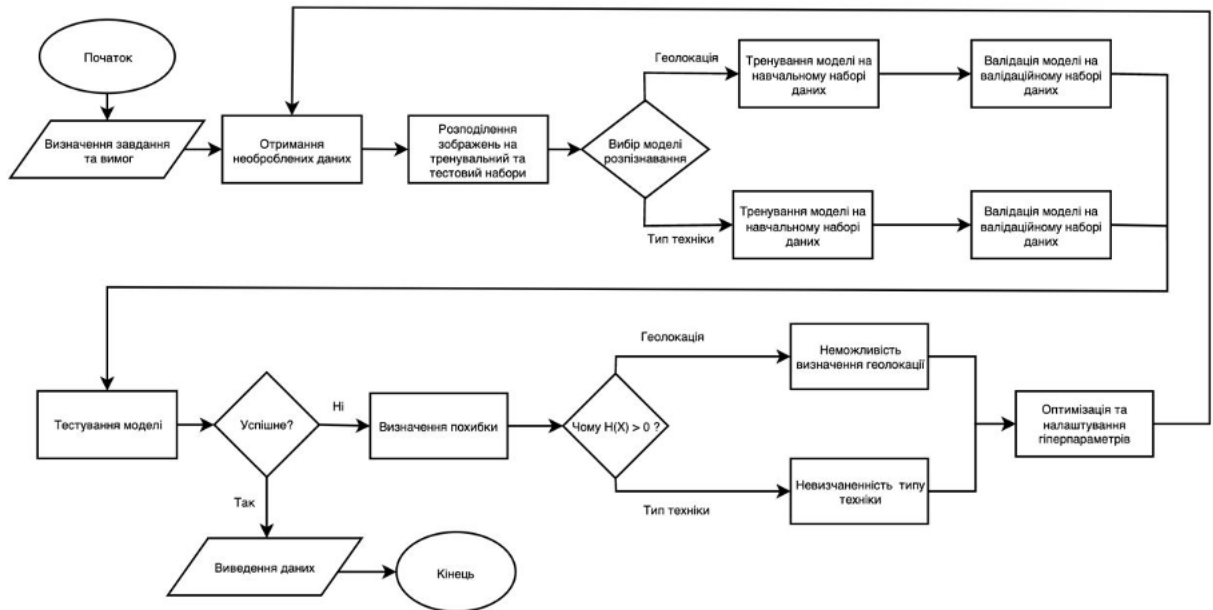
$H(X)$ – міра невизначеності (ентропія);

$P(x_j)$ - імовірність події;

x_j – випадкові величини (веденні дані).

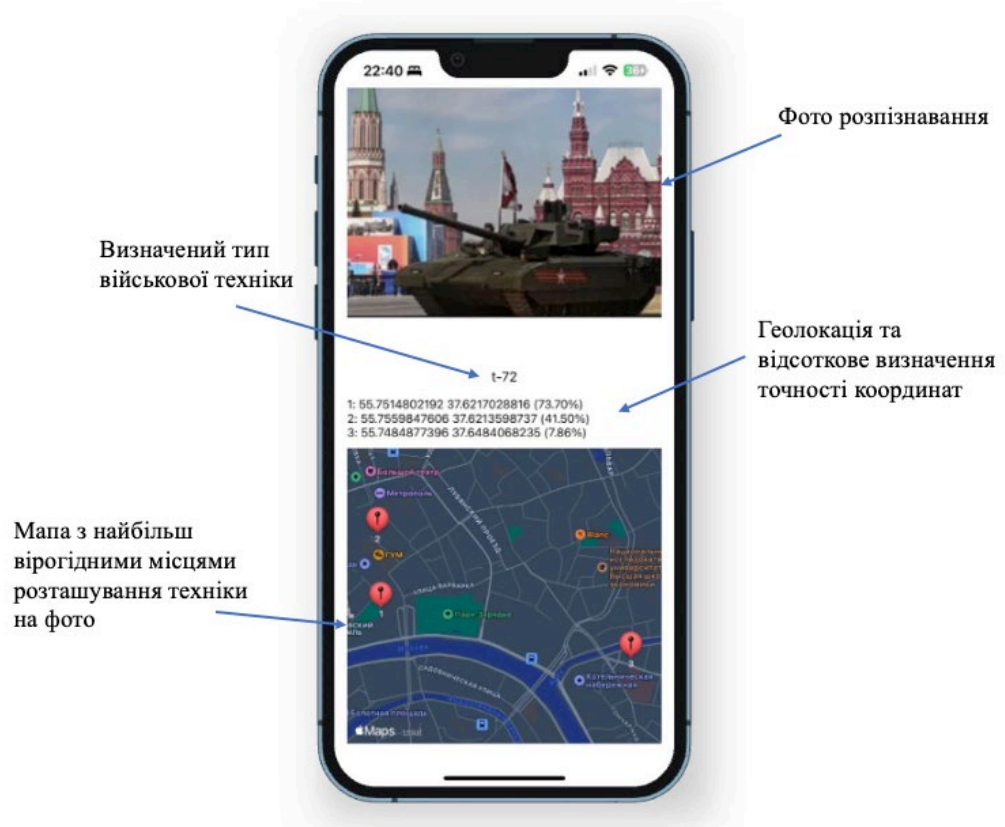
6

БЛОК-СХЕМА РОБОТИ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ



7

ІНТЕРФЕЙС ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ



8

ТЕСТУВАННЯ РОЗРОБЛЕНОЇ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ



ВИСНОВКИ

1. Проведено аналіз методів розпізнавання типу військової техніки та її геолокації. Обрано найоптимальніший алгоритм для забезпечення якості розпізнавання типу військової техніки та її геолокації.
2. Розроблена інтелектуальна система, що забезпечує підвищення точності розпізнавання типу військової техніки та її геолокації за фото.
3. Проведено тестування розробленої інтелектуальної системи на реальних даних, зібрано статистику щодо ефективності розпізнавання типу військової техніки та її геолокації, встановлено середнє значення достовірності даних інтелектуальної системи у межах 83%, що підвищило показник точності розпізнавання типу військової техніки та її геолокації на 11% в порівнянні з іншими методами.
4. Розроблено мобільний застосунок для розпізнавання типу військової техніки та її геолокації на основі фото на платформу iOS.

ПУБЛІКАЦІ ТА АПРОБАЦІЯ РОБОТИ

Тези доповідей:

1. Савіцький В.А., Дібрівний О.А. “Особливості оптимізації алгоритмів вибору відеострімінгового контенту з використанням машинного навчання” // I Всеукраїнська науково-технічна конференція «Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу» 28 листопада 2023 року, Державний університет інформаційно-комунікаційних технологій, Київ, Україна.
2. Савіцький В.А., Дібрівний О.А. “Особливості вдосконалення виробничих процесів за допомогою алгоритмів машинного навчання” // I Всеукраїнська науково-технічна конференція «Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу» 28 листопада 2023 року, Державний університет інформаційно-комунікаційних технологій, Київ, Україна.

11

ДЯКУЮ ЗА УВАГУ!

12