КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 1 (29), 2025
ISSN 2663 - 4023
DOI 10.28925/2663-4023.2025.29.912
УДК 004.89 (004.49)

Костюк Антон Юрійович
аспірант факультету електронних та інформаційних технологій,
Національний університет «Чернігівська політехніка», Чернігів, Україна
ORCID ID: 0009-0001-3107-9976
1999kostyukanton@gmail.com

Зайцев Сергій Васильович
доктор технічних наук, професор кафедри інформаційних та комп'ютерних систем,
Національний університет «Чернігівська політехніка», Чернігів, Україна
ORCID ID: 0000-0001-6643-917X
serza1979@gmail.com

Василенко Владислав Михайлович
кандидат технічних наук,
т.в.о. відділу інформаційних та комунікаційних технологій ІТГІП НАНУ,
Інститут телекомунікацій і глобального інформаційного простору
Національної академії наук України, Київ, Україна
ORCID ID: 0000-0001-8156-1894
vladvasilenko9@gmail.com

Зайцева Лілія Ігорівна
аспірант, Інститут телекомунікацій і глобального інформаційного простору
Національної академії наук України, Київ, Україна
ORCID ID 0000-0002-0668-711X
lili5990n@ukr.net

ІДЕНТИФІКАЦІЯ ВІЙСЬКОВИХ ОБ'ЄКТІВ НА ОСНОВІ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ

Анотація. У статті досліджуються сучасні підходи до розпізнавання військових об'єктів за допомогою нейронних мереж (ШНМ). У ній висвітлюється застосування алгоритму капсульної мережі, який покращує моделювання ієрархічних зв'язків за допомогою передових технологій. У цій статті досліджується проблема ідентифікації сучасних військових активів за допомогою штучних нейронних мереж (ШНМ). Фундаментальним аспектом автоматизованого виявлення цілей є здатність розпізнавати об'єкти на зображеннях, отриманих розвідувальними платформами, такими як дрони. У цьому контексті згорткові нейронні мережі (ЗНМ) відіграють вирішальну роль в аналізі та класифікації візуальних даних, отриманих під час аероспостереження. Хоча ЗНМ є високоефективними для ідентифікації об'єктів на основі зображень, їхня продуктивність значною мірою залежить від наявності великих навчальних наборів даних. Однак, через класифікований характер військової інфраструктури, отримання достатньої кількості навчальних даних залишається суттєвим обмеженням. Отже, недостатня кількість навчальних даних може значно знизити продуктивність ЗНМ. Для вирішення цієї проблеми було обрано багаторівневу платформу CapsNet, спеціально розроблену для розпізнавання військових об'єктів з невеликим навчальним набором. Набір даних, використаний у дослідженні, взято з https://universe.roboflow.com/robo-flow-woln1/military-object-detection-x7gfp., включає як військові, так і цивільні об'єкти. Запропонована структура демонструє значне покращення точності розпізнавання, досягаючи 96,54%. Експериментальні результати показують, що цей підхід перевершує багато інших алгоритмів за точністю розпізнавання, а також сприяє розробці інтерактивних інтерфейсів за допомогою шаблонів проектування та структур. Крім того, досліджуються автоматизовані методи перевірки інтерфейсів для виявлення потенційних проблем та помилок на ранніх стадіях розробки. Загалом, стаття пропонує детальний аналіз методів та алгоритмів, які підвищують ефективність розпізнавання

об’єктів з вибраних зображень. Вона також підкреслює важливість врахування потреб користувачів та зручності під час розробки програмних продуктів.

Ключові слова: інтерактивні інтерфейси; графічний інтерфейс користувача; розпізнавання об’єктів; штучна нейронна мережа.

ВСТУП

В останні десятиліття все більше країн інтегрують передові технології у військові операції, що призводить до переходу від звичайної війни до інформаційно-орієнтованої війни, яка зараз домінує в сучасних бойових стратегіях. Основною метою збору розвідувальних даних є швидка, точна та ефективна ідентифікація військових активів за допомогою БПЛА. Виробники постійно вдосконалюють моделі дронів. Це дозволяє їм непомітно збирати інформацію з повітря про цілі, що цікавлять конфліктуючі країни, будь то окремі особи, групи чи іноземні території. Зокрема, дрони часто з’являються поблизу аеропортів, місць високого рівня безпеки, таких як центри утримання під вартою та військові заводи, а також використовуються для відстеження осіб. Виявлення об’єктів є фундаментальним кроком як для процесів відстеження, так і для розпізнавання. Усі наступні операції залежать від якості розпізнавання. Наразі ідентифікація об’єктів спирається на кілька широко використовуваних стратегій, включаючи зіставлення ознак, моделювання фону, виявлення країв, методи на основі глибокого навчання та підходи на основі візуального сприйняття. Традиційні методи зіставлення ознак [2]–[4] пропонують високу точність, але вимагають фізичних опорних моделей та є обчислювально ресурсоємними. Методи ідентифікації на основі фону [5]–[7] дозволяють автоматично відокремлювати об’єкти від їхнього оточення, проте динамічні фони можуть суттєво впливати на надійність. Методи сегментації порогів [8], [9] добре працюють у середовищах зі стабільним фоном та чітко визначеними об’єктами, але мають труднощі у складних умовах, що обмежує їхню ефективність у реальних сценаріях. На завершення, традиційні методи обчислення ідентичності стикаються з труднощами в обробці складних та різноманітних реальних сценаріїв. Тим часом дослідження комп’ютерного зору стрімко розвиваються останніми роками, маючи значний потенціал для різних застосувань обробки відео, особливо в галузі охорони здоров’я та безпеки. Комп’ютерний зір дозволяє машинам сприймати та інтерпретувати своє оточення, подібно до людей. Виявлення військових об’єктів за допомогою багаторівневих капсульних мереж служить вирішальним візуальним інструментом. Цей прогрес зумовлений досягненнями у штучному інтелекті, розробкою нових методів візуалізації та вдосконаленням обчислювальних моделей для більш ефективного вирішення завдань комп’ютерного зору.

У багатьох системах комп’ютерного зору ідентифікація об’єктів відіграє вирішальну роль, надаючи додаткову інформацію як про виявлений об’єкт, так і про його місцезнаходження. Після ідентифікації об’єкта стає можливим подальший аналіз, такий як розпізнавання конкретних предметів або відстеження руху військової техніки. Розпізнавання об’єктів широко застосовується в різних галузях, включаючи взаємодію людини з комп’ютером, робототехніку, відстеження військових об’єктів, пошукові системи та автоматизацію транспортних засобів. Ці застосування вимагають ключових аспектів, таких як час обробки (наприклад, онлайн, реальні та офлайн моделі), виправлення дефектів та ідентифікація зміни пози. У той час як деякі системи зосереджені на ідентифікації одного об’єкта з однієї точки зору, іншим потрібно розпізнавати кілька об’єктів або один і той самий об’єкт з різних точок зору. У воєнних умовах зброя та

військова техніка часто маскуються за допомогою захисних сіток або приховуються в різноманітних ландшафтах. Як результат, камуфляж, разом з динамічними та непередбачуваними зонами бойових дій, значно ускладнює ідентифікацію військових цілей.

Аналіз досліджень та публікацій. Щодо стратегій нейронних мереж, LeNet [7] зазвичай використовується для виявлення об'єктів. Незважаючи на свою просту структуру, він забезпечує чудову продуктивність в автоматичному розпізнаванні об'єктів. З часом дослідники розширили його архітектуру, щоб покращити його можливості для нелінійного навчання, що призвело до розробки AlexNet [22]. Завдяки включенню функції активації ReLU та багатошарового розгалуження, AlexNet значно покращує розпізнавання малих об'єктів, досягаючи збільшення точності більш ніж у 20 разів порівняно з попередніми моделями. Після AlexNet мережеві структури стали центральними для покращення продуктивності [9]. Помітним прикладом є VGGNet [20], який замінює великі згорткові ядра 5×5 на менші 3×3, тим самим підвищуючи точність виявлення, зменшуючи при цьому кількість параметрів для більш ефективного підходу. Подальші вдосконалення представили більш складні нейронні шари для вилучення ознак. Серед найбільш широко використовуваних архітектур є GoogleNet [11] та залишкові мережі [2]. GoogleNet інтегрує кілька згорткових компонентів в єдиний шар, тоді як залишкові мережі використовують залишкові з'єднання замість прямих виходів, створюючи альтернативні зв'язки між вхідним та вихідним шарами підходу.

Ця структура використовує оптимальну фільтрацію Габора в поєднанні з мережами глибоких пірамід ознак для покращення продуктивності виявлення. Як попередній етап, інтеграція текстурних ознак військових об'єктів з вимогами завдання виявлення здійснюється за допомогою запропонованої мережі пропозицій з тонкими регіонами (FRPN). Фільтр Габора включено в конструкцію мережі, оптимізуючи вилучення ознак. Підхід включає побудову оптимального набору фільтрів Габора [19] шляхом аналізу розподілу енергії зображення після перетворення, тим самим мінімізуючи обчислювальне навантаження та підвищуючи ефективність обробки. На цьому етапі застосовується метод сегментації на основі ентропії Реньї для уточнення вибору областей та підвищення точності виявлення. Зрештою, для покращення розпізнавання ознак малого розміру була введена висококорисна пірамідальна мережа (HUFPN). На цьому етапі зверху вниз будується фундаментальна піраміда ознак. Шляхом перехресного посилання та вирівнювання відбиттів у різних масштабах уточнюється представлення компонентів малих об'єктів, що ефективно вирішує проблему ідентифікації малих об'єктів. Автор стверджує, що результати тестування підтверджують значні переваги запропонованого підходу з точки зору точності, ефективності та розпізнавання малих об'єктів. Порівняно з існуючими методами, ця методика виділяється як найсучасніше рішення, створюючи оптимальні умови для швидкої та точної ідентифікації військових об'єктів та точного націлювання на військові ділянки. Головною метою валідації [6] є допомога операторам дронів з інтелектуальними системами візуального інтелекту, які допомагають розрізняти та відстежувати сумнівні або підозрілі події на послідовних відеозображеннях. Система візуального спостереження вимагає швидких та інтелектуальних підходів для виявлення та відстеження рухомих об'єктів. Це дослідження розглядає методи ідентифікації та моніторингу об'єктів в автономних безпілотних апаратах. Характеристики руху витягуються за допомогою оптимізованого універсального методу виявлення фону, тоді як відстеження об'єктів досягається за допомогою алгоритму оптичного потоку Лукаса-Канаде в поєднанні з методом безперервного адаптивного відстеження середнього зсуву.

Для підвищення точності відстеження об'єктів відеопослідовності були скориговані за варіаціями тону, контрасту та інтенсивності. Після визначення області пошуку гістограма кольорів об'єктів була записана та збережена як орієнтир. Для виявлення цілей та сегментації зображення автори застосували метод зворотної проекції гістограми. Під час тестування гістограма була зворотно проектована на зображення для ідентифікації елемента з найвищою ймовірністю. Останній крок включав обчислення нового розміру та площі елемента за допомогою методу усереднення. Алгоритм CAM Shift, спочатку розроблений для відстеження людських облич у користувацьких інтерфейсах, використовує адаптивну стратегію відстеження зі зсувом середнього значення. На відміну від традиційного відстеження зі зсувом середнього значення, він динамічно налаштовує розмір вікна пошуку, покращуючи точність відстеження. У своєму дослідженні [17] автори представили детектор об'єднаних об'єктів зображень (IFOD), структуровану платформу, що складається з чотирьох основних модулів. По-перше, модуль об'єднання зображень інтегрує три різні зображення в одне RGB-зображення, поєднуючи різні джерела інформації. По-друге, до об'єднаного зображення застосовується екстрактор ознак на основі CNN, який витягує значні семантичні ознаки, що сприяють ідентифікації об'єктів. По-третє, модуль пропозиції області інтересу (ROI) використовує об'єднане зображення для створення сотень або навіть тисяч кандидатів обмежувальних рамок для кожної ROI на виділеній карті, наданій екстрактором ознак. Нарешті, регресія та класифікація ROI уточнюють ці обмежувальні рамки та визначають відповідні класи об'єктів, забезпечуючи точне виявлення та класифікацію.

Автори пропонують новий підхід до виявлення військових об'єктів за допомогою багатоканальних ЗНС. Їхній метод інтегрує просторові, часові та теплові дані для побудови 3D-зображень, які потім об'єднуються в карти покриття ЗНС без нагляду. Запропонована методологія виявлення базується на методі швидкої R-ЗНС та інтегрує стратегію міжпросторового перенесення для вдосконалення моделі ЗНС за допомогою багатоканальних зображень. Щоб оцінити ефективність свого підходу, автори провели експерименти з використанням зображень з набору даних SENSIAC (Центр аналізу військової сенсорної інформації), порівнюючи його продуктивність з передовими методами виявлення, щоб підкреслити його переваги.

Формулювання цілей статті. Мета дослідження — ідентифікувати військові об'єкти за зображеннями, отриманими за допомогою дрона. Це дослідження зосереджене на:

a) розробці нового підходу з використанням капсульних мереж (CapsNet) для виявлення військових об'єктів на зображеннях на основі глибокого навчання;
b) збиранні набору даних з достатньою кількістю військових об'єктів для перевірки запропонованої методології.

У розпізнаванні об'єктів ключовий процес включає визначення площі та розміру всіх об'єктів на зображенні, незалежно від таких факторів, як масштаб, перекриття або умови освітлення. Основною метою системи розпізнавання об'єктів є надійне виявлення та класифікація всіх об'єктів, що належать до заздалегідь визначених категорій. Ця технологія особливо корисна для таких застосувань, як моніторинг кількості людей у обмежених військових зонах, де вона допомагає забезпечити безпеку та ситуаційну обізнаність. Крім того, системи розпізнавання об'єктів можуть бути адаптовані для візуальних інструментів веб-пошуку, таких як Pinterest, які використовують розпізнавання образів, щоб допомогти користувачам знаходити певні елементи або контент у великих колекціях зображень. Щоб вирішити проблему розпізнавання військових об'єктів, у статті представлено метод глибокого трансферного навчання, який складається з двох ключових

компонентів: вбудовування знань через трансферне навчання та змішаний підхід для покращеного вилучення ознак. Дослідження підкреслює, що використання можливостей вилучення ознак з великих наборів даних значно покращує виконання пов'язаних завдань, забезпечуючи ефективну передачу знань між нейронними мережами. Трансферне навчання застосовується шляхом підтримки фіксованих ваг у певних шарах під час перенавчання інших. Ключовим аспектом глибокого трансферного навчання є вибір шарів для перенесення, а шарів для точного налаштування для нового застосування. Завдяки ретельному тестуванню автори виявили, що перенавчання останніх трьох шарів зі збереженням попередніх у різних масштабах інтегровано в кожну ітерацію розпізнавання, що дозволяє системі адаптуватися до відбиттів та варіацій об'єктів у різних масштабах. Поєднуючи ці дві стратегії, запропонований підхід значно покращує виявлення військових цілей, навіть у сценаріях з обмеженими навчальними даними.

МЕТОДИКА ДОСЛІДЖЕННЯ

Для полегшення ідентифікації військових об'єктів було розроблено багаторівневу мережу CapsNet, як показано на рис. 1.

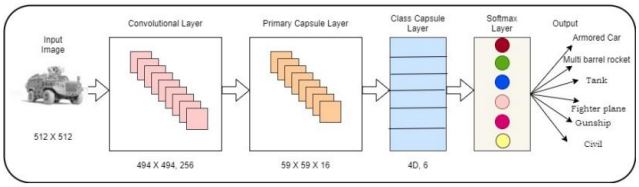

Рис. 1. Багаторівнева архітектура CapsNet для розпізнавання військових об'єктів

Реалізація багаторівневої архітектури CapsNet включає такі ключові операції:

1. Згортковий шар — витягує ознаки з вхідного зображення.
2. Первинний капсульний шар — обробляє витягнуті ознаки.
3. Шар капсули класів — застосовує динамічну маршрутизацію для уточнення ознак для ідентифікації та класифікації.
4. Шар SoftMax — перетворює вихідні дані шару капсули класів на значення ймовірності, що відповідають кожному класу військових об'єктів.

Перед подачею в мережу вхідні зображення проходять попередню обробку для покращення їхньої контрастності та зменшення шуму. Зображення спочатку перетворюються на градації сірого, що забезпечує однорідність вхідних даних для запропонованої багаторівневої моделі CapsNet. Крім того, всі зображення змінюються на стандартизований формат 512×512 після перетворення на градації сірого та попередньої обробки, що забезпечує стабільну якість вхідних даних. Цей крок попередньої обробки призводить до отримання висококонтрастних, гладких зображень у градаціях сірого, що покращує можливості розпізнавання архітектури CapsNet.

У згорткових шарах вхідні зображення або попередні карти ознак обробляються за допомогою невеликих фільтрів (ядер), які ковзають по вхідному сигналу. Ці фільтри виконують поелементне множення з подальшим підсумовуванням, генеруючи вихідне значення. Ця операція використовує 2D-ядро, яке взаємодіє з 2D-вхідною областю на кожному кроці. Фільтр систематично рухається зліва направо та зверху вниз, створюючи карту ознак, яка виділяє певні візерунки або текстури, присутні на зображенні, оптимізуючи здатність моделі розпізнавати відповідні ознаки.

Одне ядро на кожне зображення є ефективним, оскільки воно дозволяє виявляти певний візерунок по всьому зображенню, незалежно від його розташування. Однак, покладатися лише на одну ознаку недостатньо для точної ідентифікації об'єкта. Щоб охопити ширший діапазон відмінних характеристик, одночасно застосовуються кілька ядер, кожне з яких генерує власну окрему карту ознак. Такий підхід дозволяє мережі виявляти різні візерунки та ознаки на зображенні. Шляхом накладання згорткових шарів мережа поступово витягує більш абстрактні ознаки на кожному рівні. Ця поступова абстракція покращує здатність мережі розпізнавати складні об'єкти та заплутані візерунки, покращуючи її загальну точність та стійкість у завданнях виявлення об'єктів.

Під час проектування згорткової нейронної мережі (ЗНМ) вирішальну роль відіграють кілька гіперпараметрів. Одним із важливих гіперпараметрів є розмір ядра, який часто встановлюється на рівні 3×3, але його можна налаштувати на більші розміри, такі як 5×5 або 7×7, залежно від складності вхідних даних. Для зображень у градаціях сірого використовуються 2D-ядра, тоді як для RGB-зображень потрібні 3D-ядра для обробки глибини вхідних даних. Кількість ядер на шар, зазвичай степінь двійки (від 32 до 1024), визначає, скільки ознак мережа може витягти на кожному етапі. Збільшення кількості ядер підвищує здатність мережі розпізнавати ширший діапазон шаблонів, покращуючи її точність та стійкість у таких завданнях, як виявлення та класифікація об'єктів. Завдяки точному налаштуванню цих гіперпараметрів мережу можна оптимізувати для різних типів та складностей вхідних даних, забезпечуючи покращену продуктивність у різних завданнях.

Крок визначає, наскільки далеко рухається фільтр на кожному етапі під час операції згортки. Значення кроку, рівне 1, означає, що фільтр зміщується на один піксель за раз, але більші значення можна використовувати для регулювання рівня зниження дискретизації та зменшення обчислювальних витрат. Заповнення застосовується для збереження просторових розмірів вхідних даних, запобігаючи надмірному зменшенню розміру вихідних даних у міру поглиблення мережі. Це особливо важливо для підтримки достатньої просторової інформації для глибших шарів. Зі збільшенням кількості згорткових шарів також зростає кількість параметрів, що навчаються, у мережі, що може призвести до збільшення обчислювальної складності та вимог до пам'яті. Щоб вирішити цю проблему, використовується спільне використання параметрів, тобто один і той самий набір ваг використовується для всіх нейронів на карті ознак у заданому шарі. Це зменшує загальну кількість параметрів, які потрібно вивчити, роблячи мережу ефективнішою, водночас дозволяючи їй вивчати складні ознаки з вхідних даних. Цей метод значно зменшує обчислювальне навантаження, дозволяючи мережі ефективно узагальнювати, покращуючи її ефективність та масштабованість.

На рис. 2 проілюстровано конструкцію згорткового шару, що використовується в запропонованій багатошаровій архітектурі CapsNet.

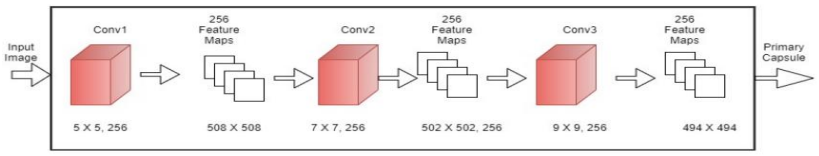

Рис. 2. Рівень згортки багаторівневої архітектури CapsNet

Попередньо оброблене вхідне зображення 512×512 подається на згортковий шар, який складається з трьох внутрішніх згорткових шарів: Conv1, Conv2, Conv3. Кожен з

цих шарів використовує 256 фільтрів зі стандартним кроком 1. Розміри фільтрів становлять 5×5 у Conv1, 7×7 у Conv2 та 9×9 у Conv3. У першому згортковому шарі (Conv1) до вхідного зображення 512×512 застосовується ядро 5×5 з кроком 1. Отримана карта ознак має розміри [508×508, 256], де 256 представляє кількість карт ознак, згенерованих цим шаром. Потім ця карта ознак передається на другий згортковий шар (Conv2), який використовує ядро 7×7. Вихідна карта ознак Conv2 має розміри [502×502, 256], де зменшення розміру відбувається завдяки застосуванню розміру ядра та кроку. Далі Conv3 застосовує ядро 9×9 до карти ознак з Conv2, в результаті чого отримується карта ознак з розмірами [494×494, 256]. Таким чином, кінцевий вихід згорткових шарів, який слугуватиме вхідними даними для наступного шару в архітектурі CapsNet, становить [494×494, 256]. Загальна кількість параметрів, що навчаються, для Conv1, Conv2 та Conv3 наведена в табл. 1, із загальною кількістю 40 448 параметрів, налаштованих на цих трьох згорткових шарах. Ці параметри є важливими для того, щоб модель вивчала ознаки під час навчання, і їхня кількість зростає з глибиною шарів та складністю використовуваних розмірів ядра.

Таблиця 1

Кількість параметрів, навчених на згортковому шарі

The name of the inner layer	Number of parameters for training
Conv1	256*(5*5+1)=6656
Conv2	256*(7*7+1)=12800
Conv3	256*(9*9+1)=20 992

Первинний капсульний шар, який слідує за згортковим шаром у мережі, включає три ключові процеси: згортку, переформування та стиснення. Спочатку шар отримує карти ознак, згенеровані згортковим шаром (у цьому випадку 36 карт ознак). Ці карти ознак потім переформовуються у чотири вектори по дев’ять вимірів кожен, що складають 36 елементів (36 = 4×9) для кожного місця на зображенні. Крок стиснення виконується за допомогою функції Squash, яка гарантує, що довжина кожного вектора залишається між 0 та 1, що представляє ймовірність присутності об’єкта в певному місці. Ця функція нормалізує вектори, спрощуючи їх інтерпретацію для виявлення об’єктів. У багатошаровій архітектурі CapsNet використовуються первинні капсули з 16 вимірами, причому кожна капсула представляє невелику просторову область на вхідному зображенні. Первинний капсульний шар структурований ієрархічно, що містить кілька рівнів. На першому рівні капсули генерують 16 капсул другого рівня, а ці капсули другого рівня додатково створюють 16 первинних капсул третього рівня. Ця багаторівнева ієрархічна структура дозволяє мережі фіксувати дедалі абстрактніші ознаки та зв’язки між об’єктами на різних рівнях гранулярності. Використання кількох шарів капсул дозволяє мережі підтримувати просторові зв’язки та краще обробляти складні шаблони та розпізнавання об’єктів.

На рис. 3 показано основний шар капсули в рамках запропонованої багатошарової архітектури CapsNet. Він отримує карту ознак [494×494, 256] від згорткового шару на вхід. Застосовується ядро 3×3 з кроком 1, що створює карту ознак [492×492, 256]. Цей шар складається з 16 первинних капсул, які об’єднують отримані ознаки в більш складні представлення. Кожна капсула функціонує подібно до згорткового шару, застосовуючи ядра 3×3 з кроком 2 до входу 492×492, що призводить до виходу 247×247 на капсулу. З 16 капсулами на першому рівні кінцевий розмір вихідного сигналу шару первинної

капсули становить [247×247, 16], який потім передається на другий рівень первинних капсул. На другому рівні застосовується згорткова операція з використанням ядра 3×3 з кроком 1, що створює вихідний сигнал розміром [245×245, 256]. Цей вихідний сигнал потім пересилається на 16 капсул, кожна з яких має 16-вимірне представлення. Подібно до першого рівня, ці 16 капсул працюють незалежно. Кожна капсула застосовує ядра 3×3 (з кроком 2) до карти ознак розміром 122×122. Вихідні дані первинних капсул на рівні 2 потім подаються на первинні капсули рівня 3. На третьому рівні застосовується інша згорткова операція з використанням ядра 3x3 з кроком 1, що створює вихідний сигнал розміром [122-3+1] = [120×120, 256]. Цей результат переноситься на 16 капсул, кожна з яких має 16-вимірне представлення. Як і на рівнях 1 та 2, ці капсули функціонують незалежно. Кожна капсула на третьому рівні застосовує 3×3 ядра з кроком 2 до карти ознак розміром 120×120, що призводить до вихідного розміру [120-3+1] = [59×59, 256] після згортки. Кінцевий вихідний розмір первинних капсул 3 рівня становить [59×59, 16], тобто кожна капсула містить 16 вимірів ознак.

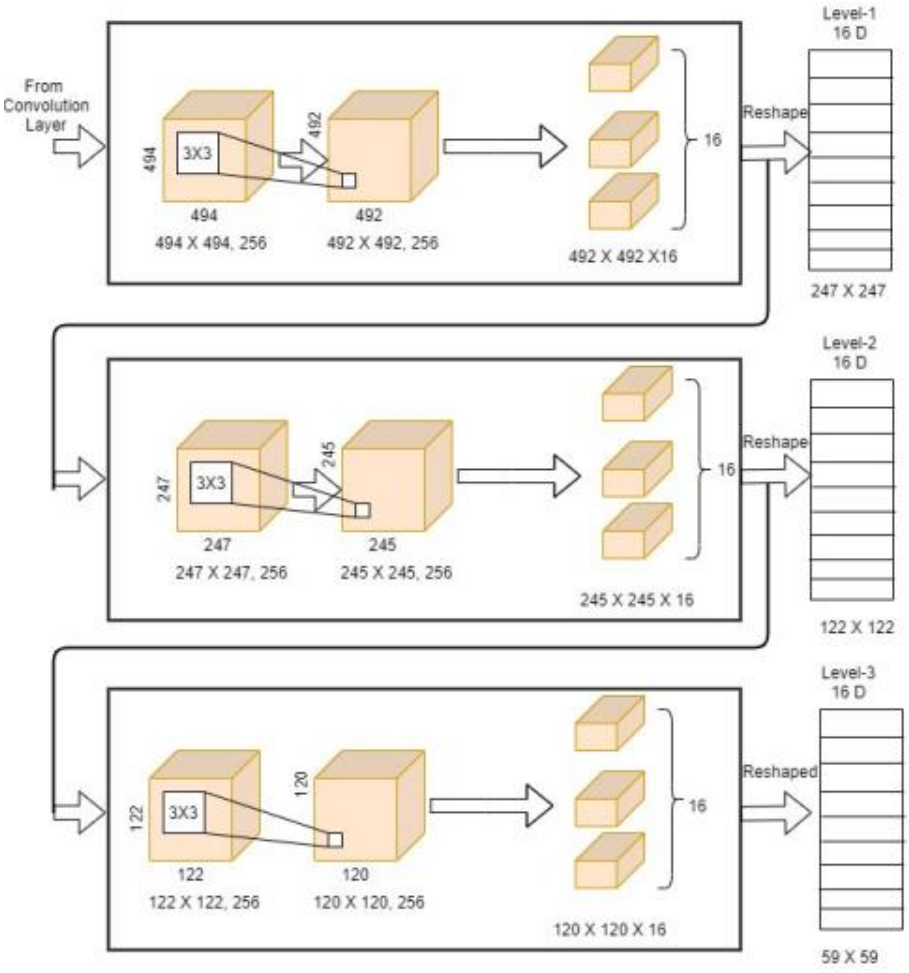

Рис. 3. Основний капсульний рівень багатошарової архітектури CapsNet

У багатошаровій архітектурі CapsNet рівень капсули класів замінює рівень максимального пулінгу звичайних CNN, використовуючи динамічну маршрутизацію за узгодженням. Як показано на рис. 4, цей рівень складається з двох підрівнів: перший рівень обробляє вихідні дані першого рівня для уточнення кінцевих капсульних представлень.

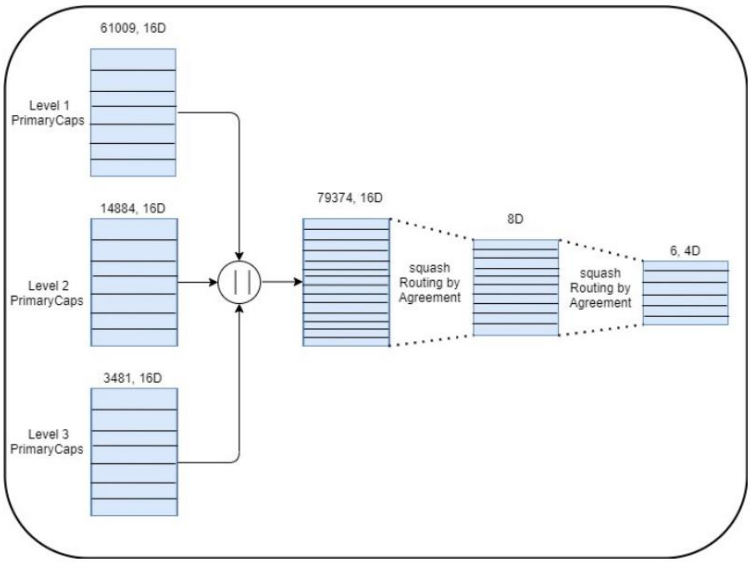

Рис. 4. Рівень капсули класів багаторівневої архітектури CapsNet

Входами для рівня капсули класів є оброблені виходи первинних рівнів капсули 1, 2 та 3, розміри яких становлять 61009×16D, 14884×16D та 3481×16D відповідно. Вони об'єднуються для формування одного входу, якщо розмір дорівнює 79374×16D. Обмеження класів з рівня 1 потім передаються на рівень 2, де між ними виконується динамічна маршрутизація для уточнення процесу класифікації.

Процес маршрутизації в CapsNet включає передачу виходу i-ї капсули з рівня l, позначеної як ui, до капсул у наступному рівні l+1. Кожна j-та капсула в рівні l+1 отримує ui як вхід, і ця взаємодія регулюється вагою wij, яка визначає зв'язок між i-тою капсулою в рівні l та j-тою капсулою в рівні l+1. Отримане перетворення, представлене як uj/I, відображає внесок i-ї капсули в шарі l до j-ї капсули в шарі l+1.

$$v_{j\vee i=W_{ij}\cdot v_j}$$

Зважена сума cj усіх прогнозів первинних капсул для класу капсул j обчислюється за допомогою функції стиснення, яка забезпечує відповідне масштабування внесків різних капсул. Ця функція допомагає регулювати вплив прогнозів окремих капсул та підтримує динамічний характер процесу маршрутизації.

$$C_j = \sum_{i=1}^{H} C_{ij} \cdot v_{j\vee i} \quad (1)$$

Щоб гарантувати, що результат знаходиться між 0 та `1, застосовується функція стиснення (також відома як функція стискання). Зазвичай вона обчислюється наступним чином.

Ось чіткіша та більш структурована версія алгоритму динамічної маршрутизації:

1. процедура Routing (vj/I, m, l)
2. для кожної капсули I в шарі та капсули j в шарі l+1
3. wij←0 (ініціалізація ваг)
4. для ітерації t = 1 до m
5. для кожної капсули I в шарі l
6. cij←softmax (wij) (обчислення коефіцієнтів зв'язку)
7. для кожної капсули j в шарі l+1
8. Sj←∑i cij·vj|I (обчислення зваженої суми прогнозів)

9. для кожної капсули j в шарі l+1
10. SQj←squash(sj) (застосування функції squashing)
11. для кожної капсули I в шарі l та капсули j в шарі l+1
12. wij←wij+vj/i*SQj (оновлення ваг маршрутизації)
13. повернення SQ (кінцеві виходи капсули після маршрутизації).

Шар класу капсули вимагає навчання а значна кількість параметрів. Навчені параметри Cij визначаються шляхом множення кількості векторів, отриманих з первинного шару капсули, на необхідні вихідні вектори. На рівні 1 три вхідні джерела об'єднуються, тобто (61009 + 14884 + 3481)×10^24) = 79374×10^24 = 81278976, тоді як на рівні 2 їх 10^24×6 = 6144.

Таким чином, загальна кількість параметрів на обох рівнях становить 81 285 120. Крім того, навчання параметрів потрібне при перетворенні скалярних величин у векторні величини між капсулами, що позначається Bij. Ці навчальні параметри обчислюються наступним чином: від 16D до 8D капсул — 81 278 976×16×8 = 10 403 708 928; від 8D до 4D капсул — 6 144××4 = 196 608. Обчислення цих значень призводить до 10 403 708 928-196 608 = 10 403 905 536. Отже, загальна кількість параметрів, що навчаються, у шарі капсули класу становить 10 485 190 656.

РЕЗУЛЬТАТИ ДОСЛДІЖЕННЯ

Ефективність запропонованої моделі оцінюється за допомогою набору даних зображень військових цілей, а її продуктивність порівнюється з продуктивністю методу опорних векторів (SVM) та традиційної архітектури згорткової нейронної мережі (CNN). Модель реалізована на Python з використанням потужних бібліотек глибокого навчання та фреймворків, таких як Keras та TensorFlow, для проектування та навчання архітектур нейронних мереж. Ці фреймворки пропонують ефективні інструменти для побудови та оптимізації моделей глибокого навчання, що дозволяє швидше розробляти та покращувати продуктивність.

Для оцінки можливостей алгоритму для тестування було обрано два зображення з навчального набору даних. На рис. 5 показано зображення військового літака, а на рис. 6 — зображення танка. Ці зображення слугують репрезентативними прикладами для оцінки здатності моделі точно виявляти та розпізнавати військові цілі, демонструючи ефективність запропонованого підходу до розпізнавання об'єктів у реальних сценаріях.

Рис. 5. Процедури розпізнавання літаків

Рис. 6. Зображення для розпізнавання танка

Рис. 7 та 8 ілюструють процес навчання моделі, а рис. 9 та 10 демонструють розпізнавання військових об’єктів. Цей експеримент було проведено з використанням запропонованої багаторівневої архітектури CapsNet як алгоритму класифікації.

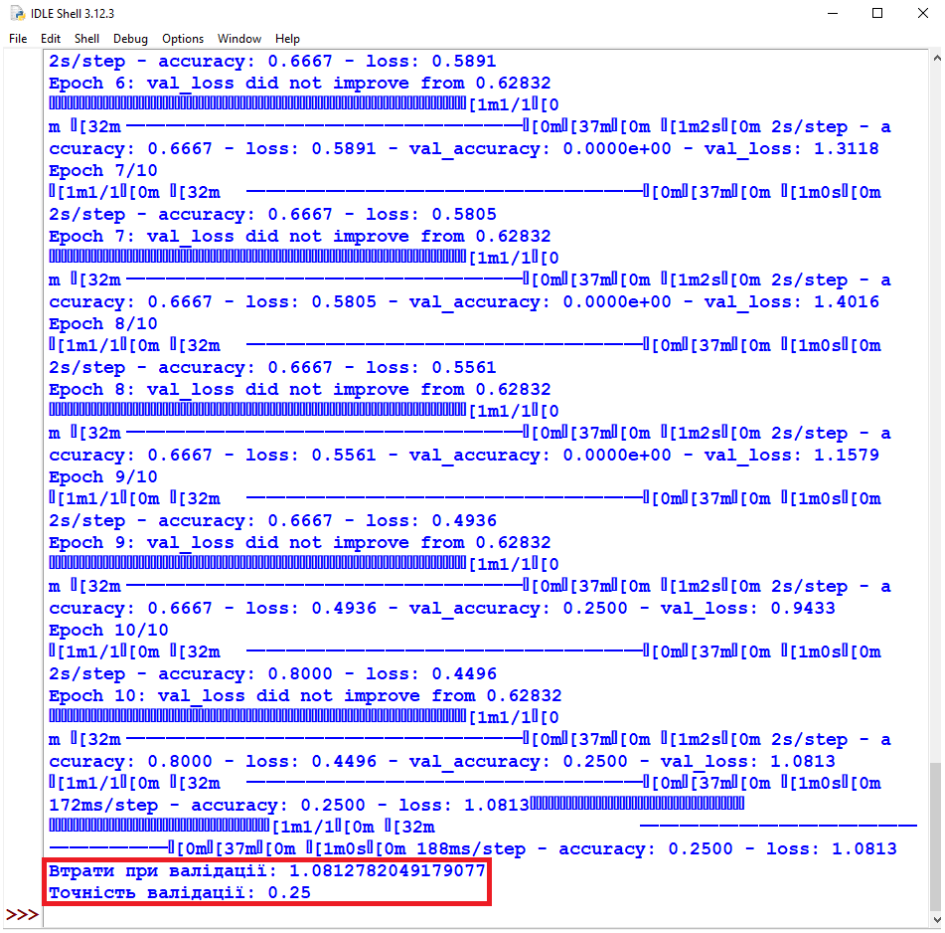

Рис. 7. Навчання моделі з меншою кількістю зображень для розпізнавання

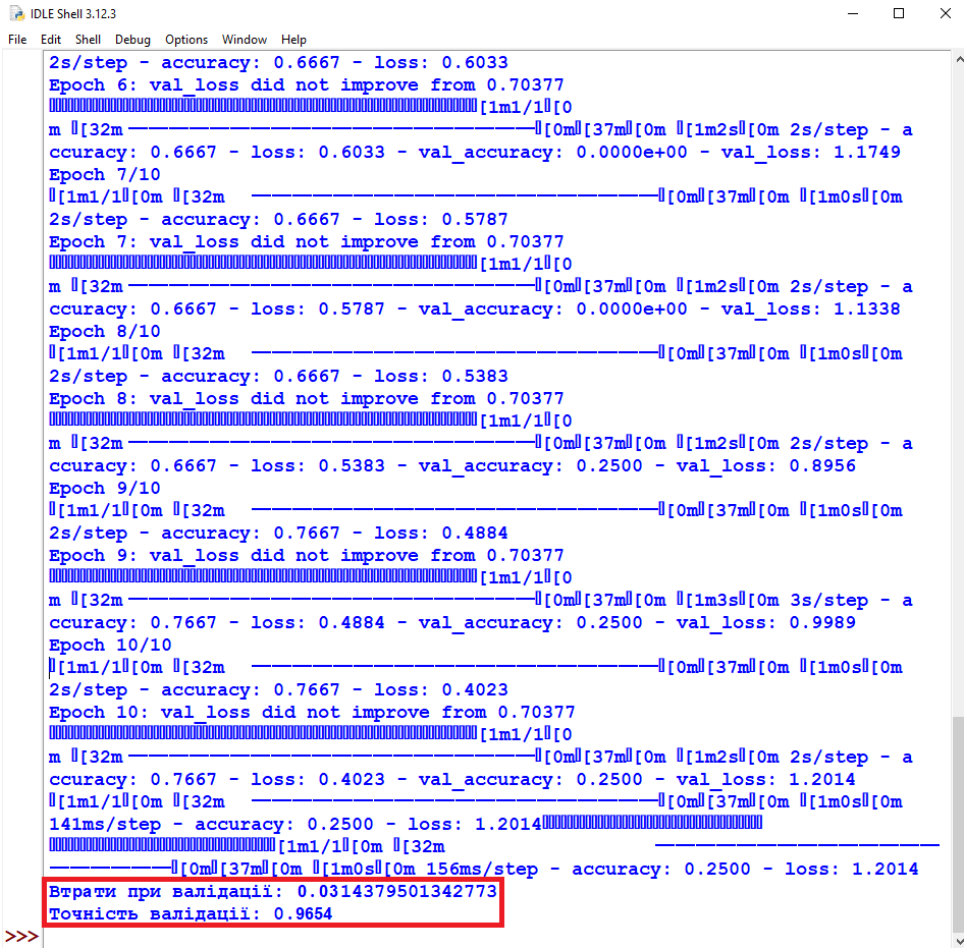

Рис. 8. Навчання моделі з додатковими зображеннями для розпізнавання

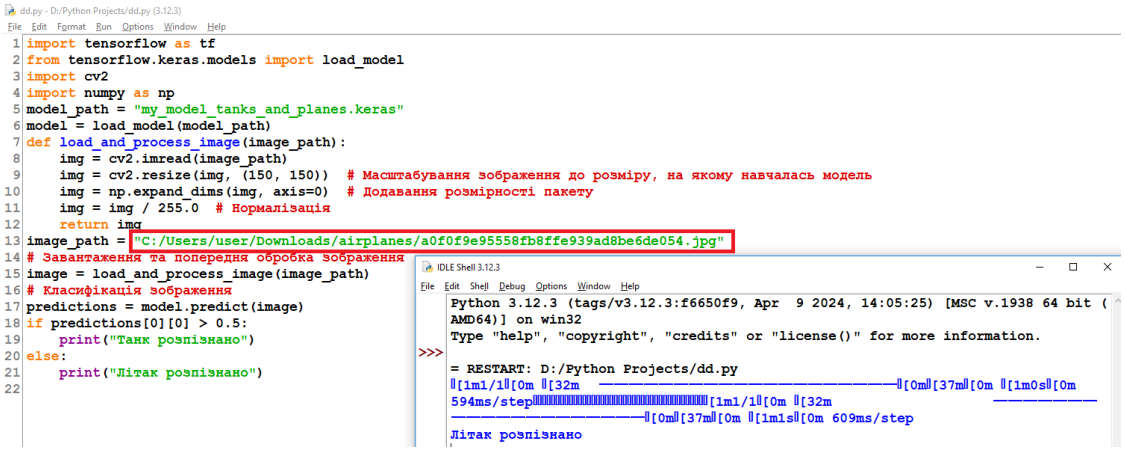

Рис. 9. Розпізнавання літаків

ISSN 2663 - 4023

```python
import tensorflow as tf
from tensorflow.keras.models import load_model
import cv2
import numpy as np
model_path = "my_model_tanks_and_planes.keras"
model = load_model(model_path)
def load_and_process_image(image_path):
    img = cv2.imread(image_path)
    img = cv2.resize(img, (150, 150))  # Масштабування зображення до розміру, на якому навчалась модель
    img = np.expand_dims(img, axis=0)  # Додавання розмірності пакету
    img = img / 255.0  # Нормалізація
    return img
image_path = "C:/Users/user/Downloads/archive/vt5/amx_32_main.jpg"
# Завантаження та попередня обробка зображення
image = load_and_process_image(image_path)
# Класифікація зображення
predictions = model.predict(image)
if predictions[0][0] > 0.5:
    print("Танк розпізнано")
else:
    print("Літак розпізнано")
```

```
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr  9 2024, 14:05:25) [MSC v.1938 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:/Python Projects/dd.py
1/1 [32m━━━━━━━━━━━━━━━━━━━━[0m [37m[0m [1m0s[0m
594ms/step[1m━━━━━━━━━━━━━━━━━━━━[0m [32m━━━━━━━━━━[0m
━━━━━━━━━━━━━━━━━━━━[0m [37m[0m [1m1s[0m 609ms/step
Літак розпізнано
>>>
========== RESTART: D:/Python Projects/dd.py ==========
1/1 [32m━━━━━━━━━━━━━━━━━━━━[0m [37m[0m [1m0s[0m
547ms/step[1m━━━━━━━━━━━━━━━━━━━━[0m [32m━━━━━━━━━━[0m
━━━━━━━━━━━━━━━━━━━━[0m [37m[0m [1m1s[0m 563ms/step
Танк розпізнано
```

Рис. 10. Розпізнавання резервуарів

У середньому, класифікація методом опорних векторів (SVM) досягає точності 85,16%, тоді як CNN з переносним навчанням (CNN-TL) та запропонована архітектура досягають 91,2% та 95,2% відповідно. Запропонована модель демонструє точність 86,12%, повноту 85,96% та F1-оцінку 85,77% для всіх класів. На рис. 11–14 представлено аналіз точності, прецизійності, повноти та F1-оцінки у вигляді коробчастої діаграми, що ілюструє розподіл даних, асиметрію та ключові статистичні квартилі. На рис. 11 показано, що запропонована архітектура досягає мінімальної точності приблизно 93% та максимальної 96,5% з медіаною 95%. На рис. 12 показано діапазон точності від 87% до 92%, з 50-м квартилем 89,5%. На рис. 13 показано діапазон повноти від 61% до 82%, тоді як на рис. 14 показано діапазон F1-оцінки від 85% до 90%.

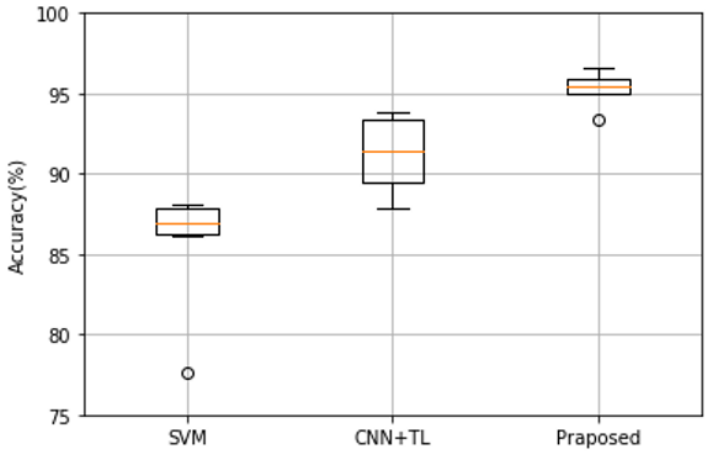

Рис. 11. Порівняння точності з аналізом Box Plot

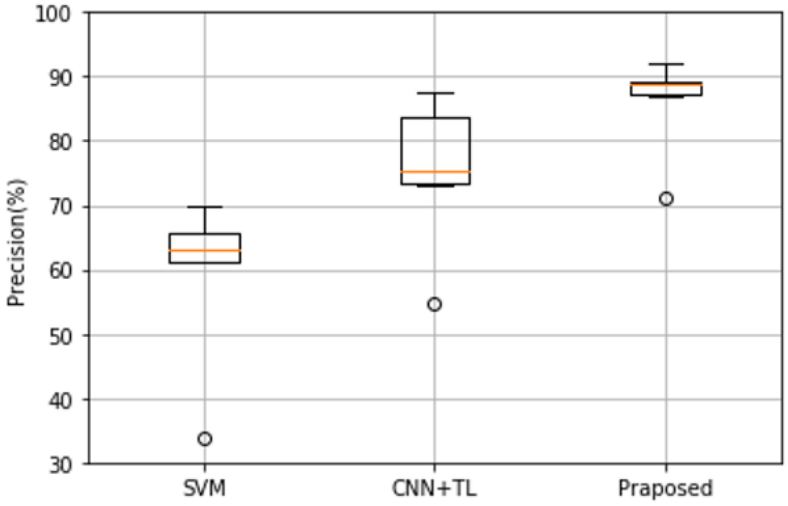

Рис. 12. Порівняння точності з аналізом Box Diagram

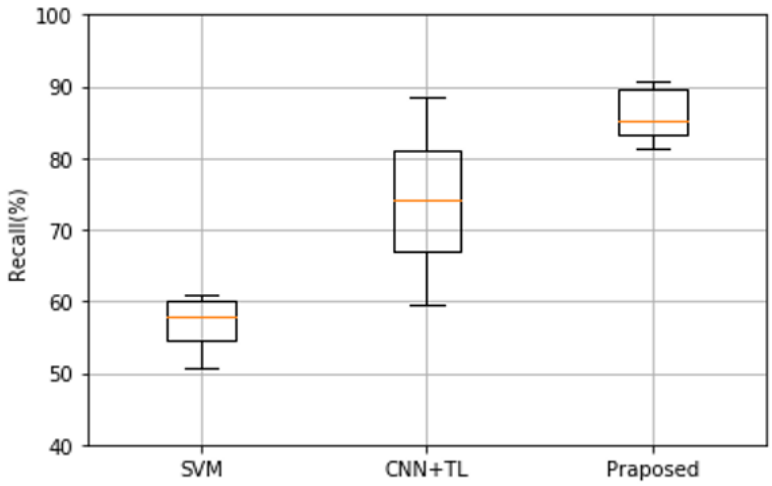

Рис. 13. Порівняння роздрібної торгівлі з аналізом Boch Plot

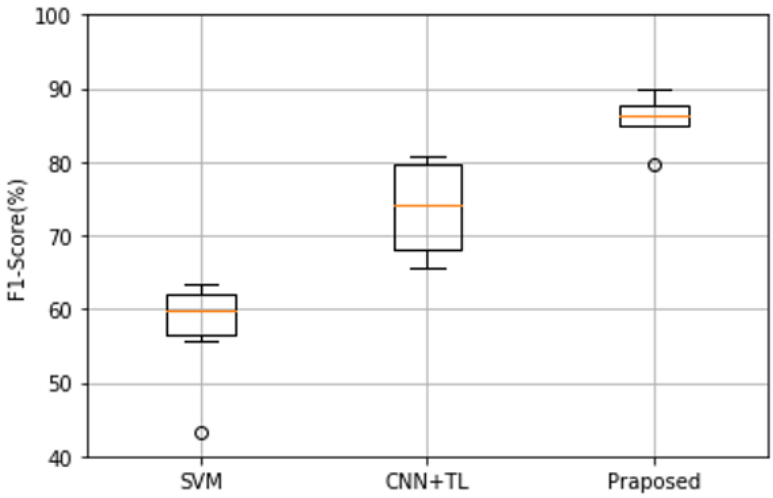

Рис. 14. Порівняння балів F1 з аналізом коробкової діаграми.

ВИСНОВКИ

У статті наведено теоретичний огляд та запропоновано інноваційне рішення наукової проблеми оптимізації інтерактивних інтерфейсів шляхом розробки нових моделей та методів. Зокрема, вдосконалений метод аналізу посилань ефективно знижує фізіологічні витрати, оскільки коротші відстані руху призводять до меншої втоми з часом та сприяють скороченню періодів виконання завдань, висновок, який вже підтверджено моїм дослідженням. Хоча вдосконалений метод аналізу посилань може не покращити всі аспекти зручності використання, він відіграє вирішальну роль в оптимізації функціональної близькості, забезпечуючи коригування фізичної відстані та компонування для мінімізації як часових, так і фізіологічних витрат, пов'язаних з частим використанням інтерфейсу.

Інші фактори, такі як концептуальна сумісність та доступність, наразі не враховуються в еталонному аналізі. Однак, вдосконалений підхід аналізу посилань все ще може допомогти в кількісній оцінці цих аспектів зручності використання. Експерти з зручності використання можуть проводити власний аналіз, але цей метод служить основою для порівняння різних альтернатив.

Вдосконалений підхід до проектування інтелектуальних інтерфейсів особливо ефективний для інтерфейсів, які є фіксованими та когнітивно простими, тобто тих, які не вимагають значних когнітивних зусиль від користувачів під час процедурних завдань. Цей підхід є найефективнішим, коли між користувачем та інтерфейсом відбуваються часті фізичні рухи. Завдяки оптимізації дизайну, цей метод може значно зменшити фізичні рухи, що, у свою чергу, знижує втому, зменшує пов'язані з цим ризики (такі як травми) та покращує продуктивність за рахунок зниження рівня помилок. Це особливо вигідно в середовищах, де поширена повторювана та безперервна взаємодія з інтерфейсами.

Ще одним цінним застосуванням цього методу є проектування схильних до помилок інтелектуальних інтерфейсів керування. Вони часто пов'язані з високими витратами на відновлення системи, оскільки помилки в роботі можуть бути дорогими та трудомісткими. Удосконалений метод проектування інтерфейсу враховує як зручність використання, так і користувацький досвід, що може бути включено в дизайн макета, щоб забезпечити більш інтуїтивну та ефективну взаємодію користувачів із системою. Це зменшує кількість помилок та підвищує загальну зручність використання інтерфейсу, що робить його особливо важливим у складних системах керування.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Kuang, C. (2020). *User friendly: How the hidden rules of design are changing the way we live, work, and play*. MCD.
2. Watan, S., & Shoger, S. (2018). *Refactoring UI.*
3. Unger, R., & Chandler, C. (2011). *UX design: A practical guide to designing interaction experience*. 336 p.
4. Li, S., Xu, L. D., & Zhao, S. (2018). 5G Internet of Things: A survey. *Journal of Industrial Information Integration, 10,* 1–9. https://doi.org/10.1016/j.jii.2018.01.005
5. Wroblewski, L. (2011). *Mobile first*. New York.
6. Cooper, A. (2009). *Psychiatric hospital in the hands of patients*.
7. Krug, S. (2017). *Don't make me think*. New Riders.
8. Li, Q. (2023). Deep learning based pavement crack detection system. *Journal of Physics: Conference Series*. https://www.researchgate.net/publication/373417689_Deep_Learning_based_Pavement_Crack_Detection_System

9. Hussain, M., & Khanam, R. A. (2024). Comprehensive review of convolutional neural networks for defect detection in industrial applications. *IEEE*. https://ieeexplore.ieee.org/document/10589380
10. Mueen, M., & Abbood, M. (2025). Investigation of IoT and deep learning techniques integration for smart city applications. *American Journal of Computing and Engineering*, *8*(1), 57–68.
11. Harel, J., Koch, K., & Perona, P. (2006). Graph-based visual saliency. *Proceedings of NeurIPS*. California Institute of Technology. https://proceedings.neurips.cc/paper_files/paper/2006/file/4db0f8b0fc895da263fd77fc8aecabe4-Paper.pdf
12. Achanta, R., Hemami, S., Estrada, F., & Susstrunk, S. (2009, June 20–26). Frequency-tuned salient region detection. *2009 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1597–1604). Miami, FL, USA.
13. Fassold, H., & Bailer, W. (2022). Few-shot object detection as a semi-supervised learning problem. *International Conference on Content-based Multimedia Indexing*. https://www.researchgate.net/publication/364245764_Fewshot_Object_Detection_as_a_Semi-supervised_Learning_Problem
14. Liu, M., & Yao, P. (2024). Robust classification of incomplete time series with noisy labels. *2024 27th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. https://www.researchgate.net/publication/382154918_Robust_Classification_of_Incomplete_Time_Series_with_Noisy_Labels
15. Stephens, B. (2011). Push recovery control for force-controlled humanoid robots. https://www.researchgate.net/publication/266907694_Push_Recovery_Control_for_Force-Controlled_Humanoid_Robots
16. Pontil, M., & Theodoros, E. (2001). Support vector machines: Theory and applications. *Lecture Notes in Computer Science*, 249–257.
17. Mikolov, T., & Chen, K. (2013). Efficient estimation of word representations in vector space. https://www.researchgate.net/publication/234131319_Efficient_Estimation_of_Word_Representations_in_Vector_Space
18. Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. https://arxiv.org/abs/1409.1556
19. He, K., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 770–778). https://www.cvfoundation.org/openaccess/content_cvpr_2016/papers/He_Deep_Residual_Learning_CVPR_2016_paper.pdf
20. Huang, G. (2018). Densely connected convolutional networks. https://arxiv.org/abs/1608.06993
21. Hays, J., & Maire, M. (2014). Microsoft COCO: Common objects in context. *Lecture Notes in Computer Science*. https://www.researchgate.net/publication/263002356_Microsoft_COCO_Common_Objects_in_Context
22. Talusani, H. (2020). Detection of military targets from satellite images using deep convolutional neural networks. *2020 IEEE 5th International Conference on Computing Communication and Automation*.
23. Zhou, L. (2023). A multi-scale object detector based on coordinate and global information aggregation for UAV aerial images. *Remote Sensing*, *15*(14), 3468. https://www.mdpi.com/2072-4292/15/14/3468
24. Sabur, S., Frost, N., & Hinton, J. (2017). Dynamic routing between capsules. *Advances in Neural Information Processing Systems (NeurIPS)*.
25. Kostiuk, Yu. V., Skladannyi, P. M., Bebeshko, B. T., Khorolska, K. V., Rzaieva, S. L., & Vorokhob, M. V. (2025). *Information and communication systems security*. [Textbook] Kyiv: Borys Grinchenko Kyiv Metropolitan University.
26. Kostiuk, Yu. V., Skladannyi, P. M., Hulak, H. M., Bebeshko, B. T., Khorolska, K. V., & Rzaieva, S. L. (2025). *Information security systems.* [Textbook] Kyiv: Borys Grinchenko Kyiv Metropolitan University.
27. Hulak, H. M., Zhyltsov, O. B., Kyrychok, R. V., Korshun, N. V., & Skladannyi, P. M. (2023). *Enterprise information and cyber security.* [Textbook] Kyiv: Borys Grinchenko Kyiv Metropolitan University.

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 1 (29), 2025
ISSN 2663 - 4023

Anton Kostiuk
Postgraduate student, Faculty of Electronic and Information Technologies,
National University "Chernihiv Polytechnic", Chernihiv, Ukraine
ORCID ID: 0009-0001-3107-9976
1999kostyukanton@gmail.com

Serhiy Zaitsev
Doctor of Technical Sciences,
Professor of the Department of Information and Computer Systems,
National University «Chernihiv Polytechnic», Chernihiv, Ukraine
ORCID ID: 0000-0002-7751-5956
Serza1979@gmail.com

Vladislav Vasylenko
Candidate of Technical Sciences,
t.v.o. Department of Information and Communication Technologies,
Institute of Telecommunications and Global Information Space of the
National Academy of Sciences of Ukraine, Kyiv, Ukraine
ORCID ID: 0000-0001-8156-1894
vladvasilenko9@gmail.com

Lilia Zaitseva
Postgraduate Student, Institute of Telecommunications and Global Information
Space of the National Academy of Sciences of Ukraine, Kyiv, Ukraine
ORCID ID: 0000-0002-0668-711X
lili5990n@ukr.net

IDENTIFICATION OF MILITARY OBJECTS BASED ON ARTIFICIAL NEURAL NETWORKS

Abstract. This paper explores modern approaches to military object recognition using neural networks (ANNs). It highlights the application of the capsule network algorithm, which improves the modeling of hierarchical relationships using advanced technologies. This paper explores the problem of identifying modern military assets using artificial neural networks (ANNs). A fundamental aspect of automated target detection is the ability to recognize objects in images acquired by reconnaissance platforms such as drones. In this context, convolutional neural networks (CNNs) play a crucial role in the analysis and classification of visual data acquired during aerial surveillance. Although CNNs are highly effective for object identification based on images, their performance is largely dependent on the availability of large training datasets. However, due to the classified nature of military infrastructure, obtaining sufficient training data remains a significant limitation. Therefore, insufficient training data can significantly reduce the performance of the NNN. To solve this problem, a multi-layer CapsNet platform was chosen, specifically designed for military object recognition with a small training set. The dataset used in the study is taken from https://universe.roboflow.com/robo-flow-woln1/military-object-detection-x7gfp., which includes both military and civilian objects. The proposed framework demonstrates a significant improvement in recognition accuracy, reaching 96.54%. Experimental results show that this approach outperforms many other algorithms in recognition accuracy, and also contributes to the development of interactive interfaces using design patterns and frameworks. In addition, automated methods for interface verification are investigated to identify potential problems and errors at the early stages of development. Overall, the paper offers a detailed analysis of methods and algorithms that improve the efficiency of object recognition from selected images. It also emphasizes the importance of considering user needs and usability when developing software products.

Keywords: interactive interfaces; graphical user interface; object recognition; artificial neural network.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Kuang, C. (2020). *User friendly: How the hidden rules of design are changing the way we live, work, and play*. MCD.
2. Watan, S., & Shoger, S. (2018). *Refactoring UI*.
3. Unger, R., & Chandler, C. (2011). *UX design: A practical guide to designing interaction experience*. 336 p.
4. Li, S., Xu, L. D., & Zhao, S. (2018). 5G Internet of Things: A survey. *Journal of Industrial Information Integration, 10*, 1–9. https://doi.org/10.1016/j.jii.2018.01.005
5. Wroblewski, L. (2011). *Mobile first*. New York.
6. Cooper, A. (2009). *Psychiatric hospital in the hands of patients*.
7. Krug, S. (2017). *Don't make me think*. New Riders.
8. Li, Q. (2023). Deep learning based pavement crack detection system. *Journal of Physics: Conference Series*. https://www.researchgate.net/publication/373417689_Deep_Learning_based_Pavement_Crack_Detection_System
9. Hussain, M., & Khanam, R. A. (2024). Comprehensive review of convolutional neural networks for defect detection in industrial applications. *IEEE*. https://ieeexplore.ieee.org/document/10589380
10. Mueen, M., & Abbood, M. (2025). Investigation of IoT and deep learning techniques integration for smart city applications. *American Journal of Computing and Engineering, 8*(1), 57–68.
11. Harel, J., Koch, K., & Perona, P. (2006). Graph-based visual saliency. *Proceedings of NeurIPS*. California Institute of Technology. https://proceedings.neurips.cc/paper_files/paper/2006/file/4db0f8b0fc895da263fd77fc8aecabe4-Paper.pdf
12. Achanta, R., Hemami, S., Estrada, F., & Susstrunk, S. (2009, June 20–26). Frequency-tuned salient region detection. *2009 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1597–1604). Miami, FL, USA.
13. Fassold, H., & Bailer, W. (2022). Few-shot object detection as a semi-supervised learning problem. *International Conference on Content-based Multimedia Indexing*. https://www.researchgate.net/publication/364245764_Fewshot_Object_Detection_as_a_Semi-supervised_Learning_Problem
14. Liu, M., & Yao, P. (2024). Robust classification of incomplete time series with noisy labels. *2024 27th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. https://www.researchgate.net/publication/382154918_Robust_Classification_of_Incomplete_Time_Series_with_Noisy_Labels
15. Stephens, B. (2011). Push recovery control for force-controlled humanoid robots. https://www.researchgate.net/publication/266907694_Push_Recovery_Control_for_Force-Controlled_Humanoid_Robots
16. Pontil, M., & Theodoros, E. (2001). Support vector machines: Theory and applications. *Lecture Notes in Computer Science*, 249–257.
17. Mikolov, T., & Chen, K. (2013). Efficient estimation of word representations in vector space. https://www.researchgate.net/publication/234131319_Efficient_Estimation_of_Word_Representations_in_Vector_Space
18. Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. https://arxiv.org/abs/1409.1556
19. He, K., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 770–778). https://www.cvfoundation.org/openaccess/content_cvpr_2016/papers/He_Deep_Residual_Learning_CVPR_2016_paper.pdf
20. Huang, G. (2018). Densely connected convolutional networks. https://arxiv.org/abs/1608.06993
21. Hays, J., & Maire, M. (2014). Microsoft COCO: Common objects in context. *Lecture Notes in Computer Science*. https://www.researchgate.net/publication/263002356_Microsoft_COCO_Common_Objects_in_Context
22. Talusani, H. (2020). Detection of military targets from satellite images using deep convolutional neural networks. *2020 IEEE 5th International Conference on Computing Communication and Automation*.
23. Zhou, L. (2023). A multi-scale object detector based on coordinate and global information aggregation for UAV aerial images. *Remote Sensing, 15*(14), 3468. https://www.mdpi.com/2072-4292/15/14/3468
24. Sabur, S., Frost, N., & Hinton, J. (2017). Dynamic routing between capsules. *Advances in Neural Information Processing Systems (NeurIPS)*.

25. Kostiuk, Yu. V., Skladannyi, P. M., Bebeshko, B. T., Khorolska, K. V., Rzaieva, S. L., & Vorokhob, M. V. (2025). *Information and communication systems security.* [Textbook] Kyiv: Borys Grinchenko Kyiv Metropolitan University.
26. Kostiuk, Yu. V., Skladannyi, P. M., Hulak, H. M., Bebeshko, B. T., Khorolska, K. V., & Rzaieva, S. L. (2025). *Information security systems.* [Textbook] Kyiv: Borys Grinchenko Kyiv Metropolitan University.
27. Hulak, H. M., Zhyltsov, O. B., Kyrychok, R. V., Korshun, N. V., & Skladannyi, P. M. (2023). *Enterprise information and cyber security*. [Textbook] Kyiv: Borys Grinchenko Kyiv Metropolitan University.