

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри інтелектуальних
інформаційних систем

_____Юрій КОНДРАТЕНКО

« ____ » _____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ МАГІСТРА
СИСТЕМА ВИЯВЛЕННЯ ВІЙСЬКОВИХ ОБ'ЄКТІВ НА
ДИНАМІЧНИХ ЗОБРАЖЕННЯХ НА ОСНОВІ
АРХІТЕКТУР НЕЙРОННИЙ МЕРЕЖ

Спеціальність 122 Комп'ютерні науки

Освітня програма «Інтелектуальні інформаційні системи»

Здобувач

_____ Денис АРХІПОВ

« ____ » _____ 2024 р.

Керівник канд. техн. наук, доцент

_____ Євген СІДЕНКО

« ____ » _____ 2024 р.

Чорноморський національний університет імені Петра Могили
(повне найменування закладу вищої освіти)

Факультет	Комп'ютерних наук
Кафедра	Інтелектуальних інформаційних систем
Рівень вищої освіти	Другий (магістерський)
Освітній ступень	Магістр
Спеціальність	122 Комп'ютерні науки
Освітня програма	Інтелектуальні інформаційні системи

ЗАТВЕРДЖУЮ

Завідувач кафедри інтелектуальних
інформаційних систем

_____ Юрій КОНДРАТЕНКО

« ____ » _____ 2024 р.

ЗАВДАННЯ
на кваліфікаційну роботу здобувача

Архіпова Дениса Владиславовича

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: «Система виявлення військових об'єктів на динамічних зображеннях на основі архітектур нейронних мереж».

Керівник роботи: Сіденко Євген Вікторович, доцент кафедри ІС, канд. техн. наук, доцент.

Затверджена наказом ЧНУ ім. Петра Могили від «03» червня 2024 р. № 140/1.

2. Строк представлення кваліфікаційної роботи « ____ » _____ 2024 р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні: система, яка автоматично виявляє військові об'єкти на зображеннях або відео, отриманих з дронів; Ідентифікацію військових об'єктів на основі набору зображень або відеоматеріалів, отриманих із дронів за допомогою архітектур нейронних мереж.

4. Перелік питань, що підлягають розробці: аналіз сучасних систем для задачі виявлення військових об'єктів на зображенні з дронів; огляд сучасних методів та

архітектур нейронних мереж; Порівняння ефективності різних архітектур для задачі розпізнавання об'єктів; вибір та налаштування оптимальної архітектури нейронної мережі; розробка та навчання моделі на основі анотованих даних; Тестування системи в умовах низької видимості та обмеженого шуму на динамічних зображеннях.

5. Перелік графічних матеріалів: презентація.

Керівник роботи

(Особистий підпис)

Євген СІДЕНКО

(Власне ім'я ПРІЗВИЩЕ)

Здобувач

(Особистий підпис)

Денис АРХІПОВ

(Власне ім'я ПРІЗВИЩЕ)

Дата видачі завдання «07» червня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

кваліфікаційної роботи

Тема: Система виявлення військових об'єктів на динамічних зображеннях на основі архітектур нейронних мереж

№	Найменування роботи	Початок	Закінчення	Примітки
1	Отримання завдання на виконання КР	03.06.2024	07.06.2024	
2	Аналіз предметної області та постановка задачі	10.06.2024	20.06.2024	
3	Огляд літературних джерел за темою кваліфікаційної роботи, зокрема аналіз публікацій та аналогічних систем розпізнавання військових об'єктів	21.06.2024	01.07.2024	
4	Огляд існуючих методів та архітектур нейронних мереж для вирішення задачі розпізнавання об'єктів на зображенні	01.09.2024	25.10.2024	
5	Реалізація системи виявлення військових об'єктів за допомогою нейронних мереж з аналізом отриманих результатів	26.10.2024	21.11.2024	
6	Перший попередній захист КР на засіданні комісії кафедри	22.11.2024	22.11.2024	
7	Корегування роботи за результатами попереднього захисту	23.11.2024	05.12.2024	
8	Другий попередній захист КР на засіданні комісії кафедри	06.12.2024	06.12.2024	
9	Доробка та остаточне оформлення КР	07.12.2024	10.12.2024	
10	Подання КР, її електронної копії та інших документів (відгуку, рецензії) до захисту	16.12.2024	17.12.2024	

Керівник роботи

(Особистий підпис)

Євген СІДЕНКО
(Власне ім'я ПРІЗВИЩЕ)

Здобувач

(Особистий підпис)

Денис АРХІПОВ
(Власне ім'я ПРІЗВИЩЕ)

Дата складання календарного плану
«19» червня 2024 р.

АНОТАЦІЯ

до кваліфікаційної роботи
здобувача групи 601м ЧНУ ім. Петра Могили

Архіпова Дениса Владиславовича

на тему: “СИСТЕМА ВИЯВЛЕННЯ ВІЙСЬКОВИХ ОБ’ЄКТІВ НА ДИНАМІЧНИХ ЗОБРАЖЕННЯХ НА ОСНОВІ АРХІТЕКТУР НЕЙРОННИЙ МЕРЕЖ”

Актуальність даного дослідження полягає у необхідності підвищення ефективності та точності виявлення військових об’єктів з дронів на полі бою, проектуванню гібридної системи на основі навчених моделей нейронних мереж для виконання поставленої задачі по виявленню об’єктів. Це дозволить автоматизувати процес ідентифікації різних військових об’єктів, що може стати критично важливим інструментом під час військових операцій для підвищення обізнаності та ефективності командування.

Об’єктом дослідження є процеси розпізнавання об’єктів на зображення та відео у реальному часі.

Предметом дослідження є моделі та архітектури нейронних мереж для задачі розпізнавання військових об’єктів.

Метою роботи є вдосконалення точності, ефективності та швидкості архітектур нейронних мереж для задачі розпізнавання військових об’єктів за рахунок оптимізації гіперпараметрів та створення гібридної системи на основі навчених моделей.

В результаті виконання роботи було досліджено сучасні архітектури нейронних мереж та виконано порівняльний аналіз їх ефективності у відповідній задачі. Створено унікальний датасет з зображеннями військових об’єктів та техніки отриманих з дронів. Проаналізовано вплив гіперпараметрів на ефективність та точність архітектур Faster R-CNN та YOLOv11, визначені основні їх переваги та недоліки, а також розроблено систему на основі цих моделей для підвищення загальної точності (mAP50) при роботі в реальному часі.

Дана робота складається з чотирьох розділів. Кожен розділ відповідно присвячений: огляду предметної області та постановці задачі; методам та архітектурам нейронних мереж для виконання поставленої мети; навчанню моделей нейронних мереж та порівнянню їх ефективності; використанню найбільш підходящих моделей для реалізації системи виявлення військових об’єктів на

динамічних зображення; аналізу результатів та виявлення недоліків. Загальний обсяг роботи – 96 сторінок. Кваліфікаційна робота містить 1 додаток, 35 рисунків, 5 таблиць і 49 джерел посилання.

Ключові слова: військові об'єкти, YOLO, Faster R-CNN, mAP, класифікація об'єктів, EfficientDet, машинне навчання, нейронні мережі, дрони.

ABSTRACT

to the qualification work by the student of the group 601m of Petro Mohyla Black Sea National University

Arhipov Denys

“SYSTEM FOR DETECTION OF MILITARY OBJECTS IN DYNAMIC IMAGES BASED ON NEURAL NETWORK ARCHITECTURES”

The relevance of this research lies in the necessity to improve the efficiency and accuracy of detecting military objects from drones on the battlefield. The goal is to design a hybrid system based on trained neural network models to accomplish the task of object detection. This will enable the automation of identifying various military objects, which could become a critically important tool during military operations, enhancing situational awareness and the efficiency of command decision-making.

An object of research is the processes of real-time object recognition in images and videos.

A subject of the research is the models and architectures of neural networks for the task of military object recognition.

A purpose of the is to improve the accuracy, efficiency, and speed of neural network architectures for military object recognition tasks by optimizing hyperparameters and creating a hybrid system based on trained models.

As a result of the work, modern neural network architectures were examined, and a comparative analysis of their efficiency in the task was conducted. A unique dataset of images featuring military objects and equipment captured by drones was created. The impact of hyperparameters on the efficiency and accuracy of the Faster R-CNN and YOLO11 architectures was analyzed, their key advantages and disadvantages were identified, and a system was developed based on these models to improve the overall accuracy (mAP50) during real-time operations.

This work consists of four sections. Each of them is devoted to: a review of the subject area and problem formulation; methods and neural network architectures for

achieving the research objectives; training neural network models and comparing their efficiency; utilizing the most suitable models for implementing a system for detecting military objects in dynamic images; analyzing the results and identifying shortcomings.

The overall scope of the work is 96 pages. Thesis contains 1 application, 35 figures, 5 tables and 49 references in it.

Key words: military objects, YOLO, Faster R-CNN, mAP, object classification, EfficientDet, machine learning, neural networks, drones.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	4
ВСТУП	5
1 ОГЛЯД ПРЕДМЕТНОЇ СФЕРИ ТА ПОСТАНОВКА ЗАДАЧІ.....	7
1.1 Огляд та аналіз останніх публікацій для задачі розпізнавання військових об'єктів на зображенні	8
1.2 Постановка задачі.....	16
Висновки до розділу 1	19
2 МЕТОДИ ТА АРХІТЕКТУРИ НЕЙРОННИЙ МЕРЕЖ ДЛЯ ЗАДАЧІ РОЗПІЗНАВАННЯ ВІЙСЬКОВИХ ОБ'ЄКТІВ	20
2.1 CNN в задачах розпізнавання об'єктів на зображеннях	20
2.2 Використання архітектури YOLO в задачах розпізнавання військових об'єктів	22
2.3 Архітектури Faster R-CNN та EfficientDet в задачах розпізнавання військових об'єктів	25
2.4 Використання нечіткої логіки в задачах розпізнавання об'єктів на зображенні.....	29
2.5 Гібридні нейронні мережі в задача розпізнавання об'єктів на зображеннях	30
2.6 Порівняння архітектур та методів, які використовуються для задачі розпізнавання об'єктів на динамічному зображенні	32
2.7 Загальна структура інформаційної системи для задачі виявлення військових об'єктів з динамічного зображення	35
Висновки до розділу 2	38
3 СТВОРЕННЯ НАБОРУ ДАНИХ. ПОРІВНЯЛЬНИЙ АНАЛІЗ МОДЕЛЕЙ	

НЕЙРОННИХ МЕРЕЖ.....	39
3.1 Збір та створення набору даних.....	39
3.2 Балансування набору даних	42
3.3 Метрики, які використовуються для перевірки ефективності та точності навчених моделей	47
3.4 Навчання моделей YOLO для задачі виявлення військових об'єктів на зображенні.....	50
3.5 Навчання моделі Faster R-CNN для задачі виявлення військових об'єктів на зображенні.....	58
3.6 Навчання моделі архітектури EfficientDet для задачі виявлення військових об'єктів на зображенні	61
3.7 Порівняльний аналіз ефективності моделей та архітектур нейронних мереж	64
Висновки до розділу 3	66
4 РЕАЛІЗАЦІЯ СИСТЕМА ВІЯВЛЕННЯ ВІЙСЬКОВИХ ОБ'ЄКТІВ НА ОСНОВІ ГІБРИДНОГО ПІДХОДУ	68
4.1 Навчання моделей YOLO_D та Faster_RCNN_D.....	68
4.2 Створення системи на основі моделей YOLO та Faster R-CNN	72
4.3 Аналіз результатів системи.....	77
Висновки до розділу 4	79
ВИСНОВКИ.....	80
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	82
ДОДАТОК А Апробація роботи.....	89

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

РЕБ – радіоелектронна боротьба.

ВО – військовий об'єкт.

БПЛА – безпілотний літальний апарат.

ІІ – штучний інтелект.

ГНМ – гібридні нейронні мережі

БМП – бойова машина піхоти

CNN – Convolutional Neural Network

YOLO – You Only Look Once

R-CNN – Region-based Convolutional Neural Network

Faster R-CNN – Faster Region-based Convolutional Neural Network

EfficientDet – Efficient Detection

BiFPN – Bidirectional Feature Pyramid Network

mAP – mean Average Precision

UAV – Unmanned Aerial Vehicle

MSE – Mean Squared Error

ResNet-50 – Residual Network with 50 layers

GPU – Graphics processing unit

LSTM – Long short-term memory

DenseNet – Densely Connected Convolutional Networks

Faster R-CNN – Region-based Convolutional Neural Network

RPN – Region Proposal Network

ROI – Region of Interest

FPN – Feature Pyramid Networks

IFV – Infantry fighting vehicle

ВСТУП

Сучасні військові конфлікти ставлять перед собою високі вимоги до оперативного збору та аналізу інформації для швидкого прийняття рішень. Одним із ключових аспектів успішних військових операцій є здатність швидко та точно виявляти важливі військові об'єкти, такі як танки, різна бронетехніка, артилерійські установки, ДОТи (довготривалі оборонні точки) та командні пункти, а також все частіше точки зв'язку, спостереження та РЕБ. Використання дронів для спостереження з повітря забезпечує ефективний і безпечний спосіб збору візуальної інформації на великих територіях. Проте для більш швидкого та точного отримання корисних даних з таких зображень необхідно автоматизоване виявлення об'єктів на відео та фото, що значно прискорює прийняття оперативних рішень на полі бою.

У зв'язку з цим, дана робота пропонує систему для виявлення військових об'єктів на зображеннях з дронів на основі різних архітектур нейронних мереж, таких як YOLO (You Only Look Once), Faster R-CNN, EfficientDet. Ці архітектури були обрані через їхню високу продуктивність у завданнях виявлення об'єктів на зображеннях та можливість працювати в реальному часі або близькому до нього. Пропонована система має на меті автоматизувати процес ідентифікації різних військових об'єктів, що може стати критично важливим інструментом під час військових операцій для підвищення обізнаності та ефективності командування.

Основою роботи системи є використання моделей нейронної мережі, навчених на зображеннях зібраних з дронів, Оскільки для таких специфічних задач не існує великих загальнодоступних наборів даних, у цій роботі буде застосовуватись самостійно зібраний набір даних з анотованими зображеннями військових об'єктів по типам об'єкта (танк, бронетехніка, ДОТ і т.п). Анотація таких зображень є важливим кроком для навчання моделей нейронних мереж, адже коректне позначення об'єктів дозволяє системі виявляти їх з високою точністю. Крім того, ця робота зосередиться на порівнянні різних архітектур нейронних

мереж з точки зору точності, швидкості обробки та продуктивності у складних реальних умовах. Також будуть розглянуті обмеження й переваги кожної моделі та запропоновані рекомендації для розробки більш ефективних систем розпізнавання військових об'єктів.

В результаті дослідження буде запропонована оптимальна архітектура нейронної мережі для завдань виявлення військових об'єктів на зображеннях з дронів, що зможе задовольнити вимоги до точності та часу обробки в умовах активних військових дій. Дана система може бути подальшим кроком у розвитку автоматизованих технологій для аналізу зображень у військових та цивільних цілях, а також може бути вдосконалена шляхом дослідження нових архітектур нейронних мереж та методів обробки зображень для подальшого підвищення точності й надійності системи.

1 ОГЛЯД ПРЕДМЕТНОЇ СФЕРИ ТА ПОСТАНОВКА ЗАДАЧІ

Виявлення об'єктів та їх класифікація на зображенні – є однією із найважливіших сфер дослідження у сучасному світі. В контексті військових операцій, такі задачі як, розвідка та збір інформації, все частіше покладаються на використання БПЛА та автоматизованих систем аналізу зображення. Ці технології використовуються для забезпечення ефективності, точності та зменшення ризиків втрати людських життів.

Завдяки високій роздільній здатності камер дронів та можливості працювати в широкому спектрі умов, БПЛА забезпечують збирання великої кількості зображень, що містять важливу інформацію про стан об'єктів на місцевості.

В умовах сучасних конфліктів, коли військові дії часто відбуваються в густонаселених або складних для доступу районах, автоматизовані системи аналізу даних з дронів стають незамінними для прийняття оперативних рішень. Однак це завдання пов'язане з низкою технічних та алгоритмічних викликів.

Використання різних підходів, архітектур та технологій для виконання цієї задачі, дозволяє створювати унікальні системи, які налаштовані на конкретні результати.

Машинне навчання (МН) є одним із таких підходів та активно розглядається у наукових дослідженнях останнім часом. МН знайшло широке застосування в різних сферах, таких як інтелектуальний аналіз текстів, виявлення спаму, рекомендації відео, звісно і в класифікації та виявленні об'єктів.

За останні кілька років вчені опублікували численні оглядові статті, присвячені різним аспектам машинного навчання [1-3].

Наприклад, деякі статті зосереджені на архітектурі згорткових нейронних мереж (CNN). Останнім часом CNN набули популярності в галузі глибокого навчання завдяки своїй простій архітектурі та здатності забезпечувати хороші результати. Нейромережі побудовані за допомогою архітектури CNN добре підходять для задачі класифікації та виявлення об'єктів на зображеннях, оскільки

вони можуть ефективно обробляти візуальні особливості, які є важливими для розпізнавання таких об'єктів, як військова техніка [4].

Однак, окрім CNN, варто розглянути популярні можливі підходи для вирішення цієї задачі, щоб мати достатньо інформації для постановки задачі і вибору підходів для її вирішення.

1.1 Огляд та аналіз останніх публікацій для задачі розпізнавання військових об'єктів на зображенні

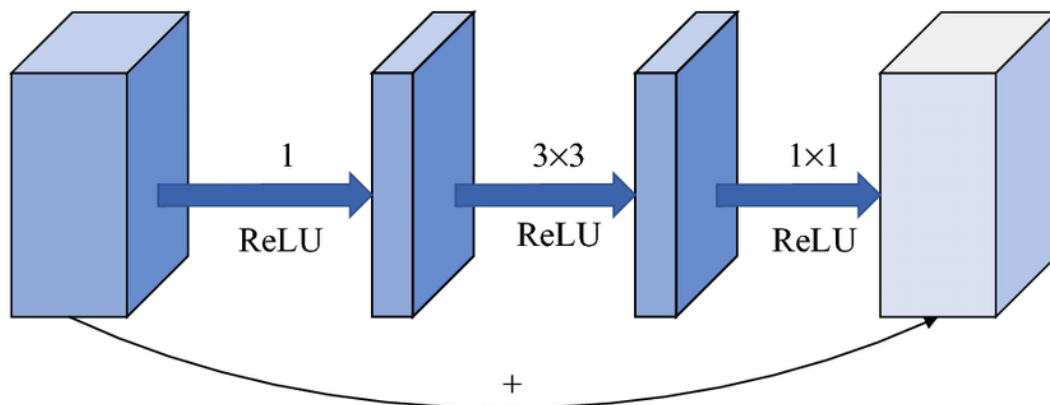
Дослідження задачі розпізнавання військових об'єктів набуло великої популярності за останні роки, досліджуються різні впливи параметрів, архітектур, навантаження на систему та інші показники. Для більшого розуміння задачі, її проблематики та підходів вирішення, варто не тільки проаналізувати деякі дослідження, а і визначити спільні риси, характеристики та результати.

Одне з досліджень [5] спрямоване на вирішення проблеми виявлення військових об'єктів в умовах обмежених ресурсів апаратної платформи. Основна мета – покращення продуктивності моделі при зменшенні обчислювальної складності та кількості параметрів. Для цього була використана модифікована архітектура YOLOv5, яка показала високі результати у військових завданнях [6].

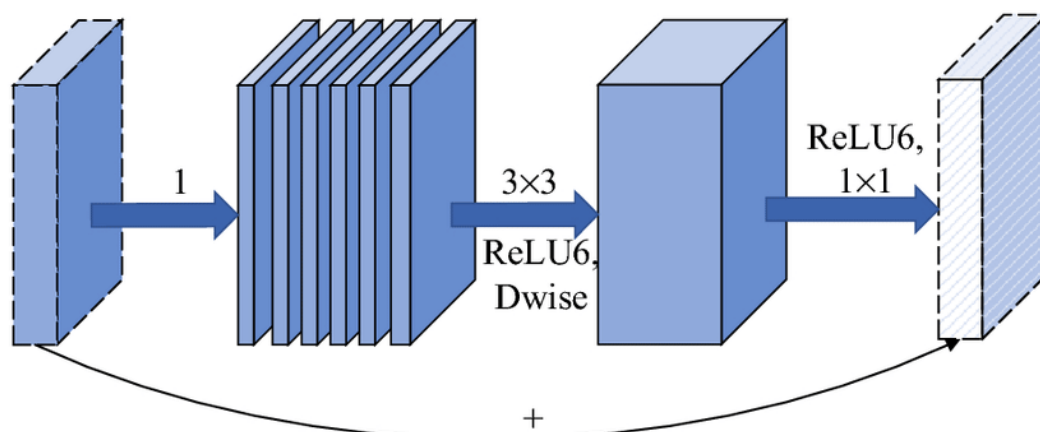
Дослідникам вдалося покращити виразність ознак і зменшило кількість параметрів і обчислень. Вони також інтегрували MobileNetV2 разом з механізмом уваги CBAM (Convolutional Block Attention Module), що дозволило досягти підвищення mAP на 1,3%, зменшення кількості параметрів на 67,45%, зниження обчислювальної складності на 73,17%.

Основним принципом MobileNetV2 є інвертовані залишки, це концепція яка на відмінність від традиційних залишків, спершу використовує вузький блок зі згорткою 1x1, потім застосовується згортка 3x3 для широкого блоку, яка відповідає за основну обробку функції і завершує згорткою 1x1. Це допомагає уникнути надлишкового обсягу обчислень та зберігання, зокрема в глибоких мережах. Ідея полягає в тому, щоб змінити класичний підхід до залишків, який використовується

у ResNet, де великі шари замінюються невеликими. В інвертованих залишках використовуються невеликі шари для переходу між різними представленнями шарів. Ілюстрована структура моделі MobileNetV2 зображена на рис. 1.1 [7, 8].



(a) Residual block



(b) Inverted residual block

Рисунок 1.1 – Залишковий блок та інвертований залишковий блок

Цей підхід був розроблений для використання на платформах зі скромними ресурсами, такі як дрони, що робить його застосовним у реальних бойових умовах для поліпшення розвідки та підвищення ефективності командирів на полі бою.

У дослідженні [9] для розпізнавання військової техніки, зокрема танків і прапорів, були використані архітектури YOLOv3, YOLOv4, та 4 модифіковані версії YOLOv5. Одна з версій YOLOv5x1 досягла найвищих результатів, модифікація

параметрів дозволила отримати mAP більше 90%.

Це дослідження показує, що YOLOv5 може бути ефективним інструментом для підвищення ефективності військових систем спостереження завдяки розпізнаванню особливостей по типу прапорів на техніці. Результати виявлення отримані в цьому дослідженні наведені на рис. 1.2.



Рисунок 1.2 – Результати виявлення об'єктів системою на основі YOLOv5

В одному з досліджень було перевірено модель Mask-RCNN із двома різними основними архітектурами ResNet-50 та MobileNet [10].

ResNet-50 забезпечує високу точність виявлення завдяки глибокій архітектурі, але потребує більше обчислювальних ресурсів та часу на обробку, у той час, як MobileNet може мати дещо нижчу точність, але її перевагою є швидкість та ефективність.

Отримана точність знаходиться на рівні 96.1% і краще себе показала архітектура ResNet-50, це робить Mask-RCNN на основі ResNet-50 високоефективною моделлю для виявлення об'єктів у реальних умовах. Це дослідження демонструє ефективність використання Mask R-CNN з різними базовими мережами для виявлення дронів у реальних умовах, підкреслюючи

баланс між точністю та обчислювальною ефективністю.

Інше дослідження [11] було зосереджене на застосуванні YOLOv8 для ідентифікації різних компонентів військової техніки. Застосування цієї технології дозволило досягти точності виявлення з використанням загальнодоступного набору даних, що включає 11800 зображень із одинадцятьма класами військових об'єктів. Модель показала результат 99,1% mAP50.

Дослідник Лю Хуаньхуа використовував модифіковану модель YOLOv5, для задачі визначення військових об'єктів дроном на малій висоті. Він стикнувся з проблемою обмеженого набору зображень з різноманітними військовими об'єктами та її низькою якістю. Для вирішення проблеми, було використано метод аргументації даних, а саме використання розмитих зображень для покращення точності в умовах туману та розмитих через рух зображень. Це підвищує стійкість моделі до умов, з якими часто стикаються дрони на низьких висотах. Це не тільки збільшило кількість тренувальних даних, а і покращило результати на 20,4%.

Для виявлення маленьких об'єктів було впроваджено карти ознак та оптимізовано функції втрат шляхом збільшення штрафу за помилки в розпізнаванні малих об'єктів або класів, де кількість зразків менша. Це зробило модель більш чутливою до малих об'єктів та рідкісних класів.

Загальний mAP моделі досяг 99.2%, що на 2.1% краще, ніж оригінальна версія YOLOv5. Швидкість детекції складає 40 кадрів за секунду, що робить її придатною для реального часу, що є критичним для військових операцій. Модель показала значний приріст продуктивності в умовах туману та руху, що є частими проблемами для зображень з дронів [12].

Окрім модифікації моделей архітектури YOLO, також були проведені порівняльні дослідження схожих моделей [13].

Для обробки даних в цьому дослідженні застосовуються архітектури глибокого навчання YOLO, Faster R-CNN та EfficientDet.

Дослідження проводить порівняння ефективності виявлення об'єктів за допомогою цих моделей на наборі даних UAVDT [14], що містить зображення,

зібрані дронами для різних цілей, таких як транспортні засоби, пішоходи тощо.

YOLOv5 показує найкращі результати за точністю та швидкістю виявлення об'єктів у реальному часі в порівнянні з іншими моделями.

Faster R-CNN та EfficientDet теж демонструють конкурентні результати за точністю, однак ці моделі потребують більше часу на обробку та більшу кількість епох для тренування, що робить їх менш оптимальними для застосування у режимі реального часу.

Висновком дослідження є те, що архітектура YOLOv5 є лідером серед розглянутих моделей за точністю та швидкістю, що робить його найкращим вибором для завдань з виявлення об'єктів з дронів в реальному часі. Незважаючи на високу точність моделей Faster R-CNN та EfficientDet, вони менш ефективні в плані швидкості, що може бути недоліком у динамічних середовищах, де потрібна швидка реакція.

Щодо систем на основі архітектури EfficientDet, є непов'язане з військовою сферою дослідження розпізнавання людей під час пошуково-рятувальних операцій [15].

У пошуково-рятувальних операціях дрони дозволяють швидко і з мінімальними витратами проводити пошук людей в небезпечних умовах, наприклад, у горах або інших складних середовищах. Людей на таких знімках важко виявляти через різні фактори: зміну пози, варіації масштабу, низьку видимість, камуфляж навколишнього середовища, несприятливі погодні умови, розмитість руху і високу роздільну здатність зображень.

У дослідженні було використано архітектуру EfficientDET та методи ансамблевого навчання для точного виявлення людей на аерознімках та використано відкритий датасет HERIDAL, який був створений для вирішення проблеми виявлення у складних умовах, таких як забруднене фоном середовище, різні погодні умови та денні або нічні сценарії.

Завдяки використанню архітектури EfficientDET у поєднанні з ансамблевим навчанням, було досягнуто mAP у 95.11%, що є одним з найвищих результатів на

цьому наборі даних для виявлення людей. Структура розробленої завдяки дослідженню системи наведена на рис. 1.3.

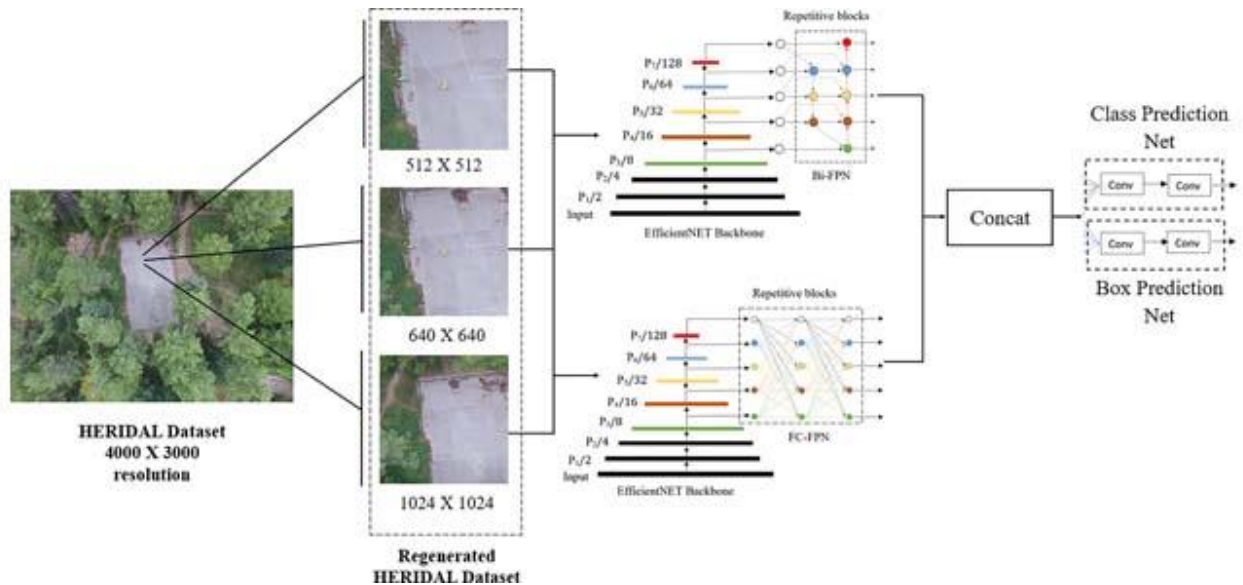


Рисунок 1.2 – Структура системи виявлення об'єктів на основі EfficientDET

Також можна зустріти дослідження, пов'язані з використанням нечіткої логіки в задачі розпізнавання об'єкта та його позиції на зображенні. Результати цього дослідження та розробленої системи наведено на рисунку 1.3 [16].

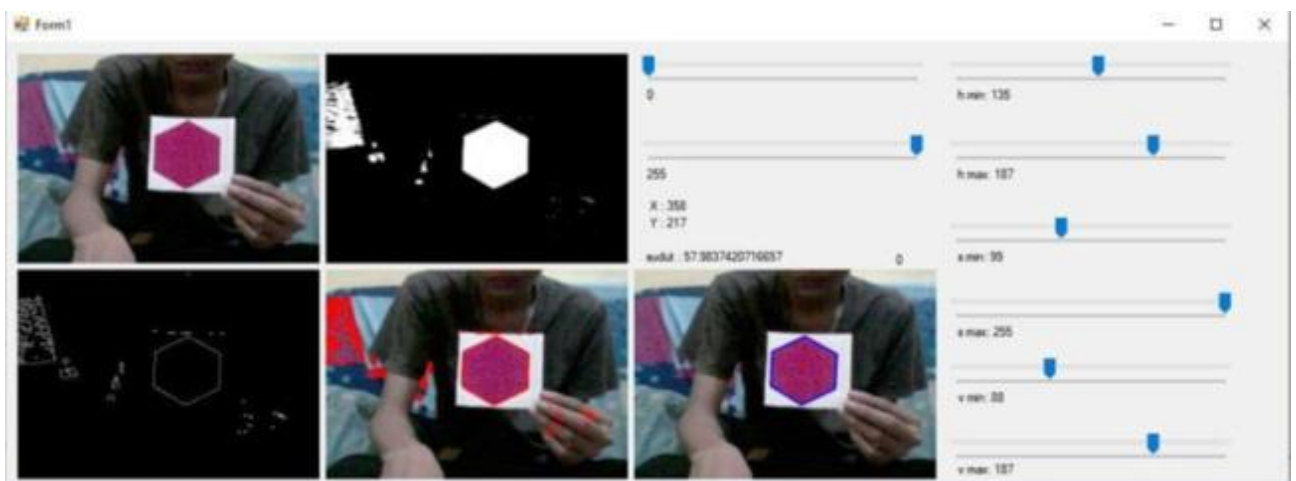


Рисунок 1.3 – Процес визначення положення предмету в дослідженні за допомогою методів нечіткої логіки

Використання нечіткої логіки забезпечує адаптивність і точність руху, це дослідження демонструє успішну інтеграцію обробки зображень і нечіткої логіки для розпізнавання та відстеження об'єктів. Система підходить для застосувань, де потрібен точний горизонтальний рух робота, наприклад, у транспортних стрічках чи автоматизованих лініях виробництва, хоча для більш складних задач з виявлення у складних умовах, може потребуватись складне налаштування правил та механізмів.

Для покращення результатів, часто використовуються більш складні в розробці – гібридні системи на основі нейронних мереж [17].

Дослідження зосереджено на вирішенні проблеми розпізнавання об'єктів на зображеннях дистанційного зондування, що є важливою задачею як для військових, так і для цивільних потреб. У таких задачах значну роль відіграють складнощі, пов'язані зі змінами кліматичних умов, наявністю перешкод і низькою якістю зображень. Особливі труднощі виникають під час роботи з дрібними, нечіткими об'єктами або коли потрібно виявити кілька цілей одночасно.

У дослідженні запропоновано новий метод розпізнавання об'єктів, який використовує гібридну архітектуру, що поєднує LSTM (довготривалу короткочасну пам'ять) і DenseNet (глибоку згорткову мережу). Цей підхід спрямований на подолання недоліків традиційних моделей, забезпечуючи високу точність і стійкість до зовнішніх факторів.

Використання LSTM дозволяє враховувати часові залежності, які можуть бути корисними при аналізі послідовних даних дистанційного зондування. Це дає змогу отримати більш репрезентативне відображення характеристик із серій зображень, що особливо важливо в динамічних умовах.

DenseNet, зі свого боку, забезпечує ефективне вилучення просторових ознак завдяки своїй глибокій згортковій архітектурі, яка сприяє побудові багатих та узгоджених характеристик зображень.

Гібридний підхід з LSTM і DenseNet підвищує точність розпізнавання об'єктів навіть у складних умовах, таких як низька якість зображення чи наявність

дрібних об'єктів.

Результати демонструють, що запропонований метод перевершує сучасні моделі, такі як CNN і YOLOV4, за ключовими метриками: точністю, повнотою, F1-мірою та втратою. Це підтверджує, що інтеграція LSTM і DenseNet дозволяє ефективніше вирішувати проблеми розпізнавання об'єктів у дистанційному зондуванні порівняно з традиційними підходами.

Дослідження [18] зосереджено на покращенні архітектури Faster R-CNN в задачі виявлення військових об'єктів, шляхом додавання декількох етапів обробки.

Faster R-CNN використовується, як основна архітектура нейронної мережі для розпізнавання цілей. Після чого, алгоритм Kmeans++ класифікує та автоматично маркує рамки для цілей в наборі даних. Цей метод дозволяє оптимізувати процес вибору початкових точок для кластеризації, що покращує результати на реальних даних.

ОНЕМ (Online Hard Example Mining), використовується, як додатковий етап до мережі, що покращує точність розпізнавання для складних або важко розпізнаваних зразків. Цей метод дозволяє нейронній мережі зосереджуватися на важких для розпізнавання прикладах, тим самим підвищуючи точність роботи на складних випадках.

Структура покращеної архітектури Faster R-CNN представлена на рис. 1.4.

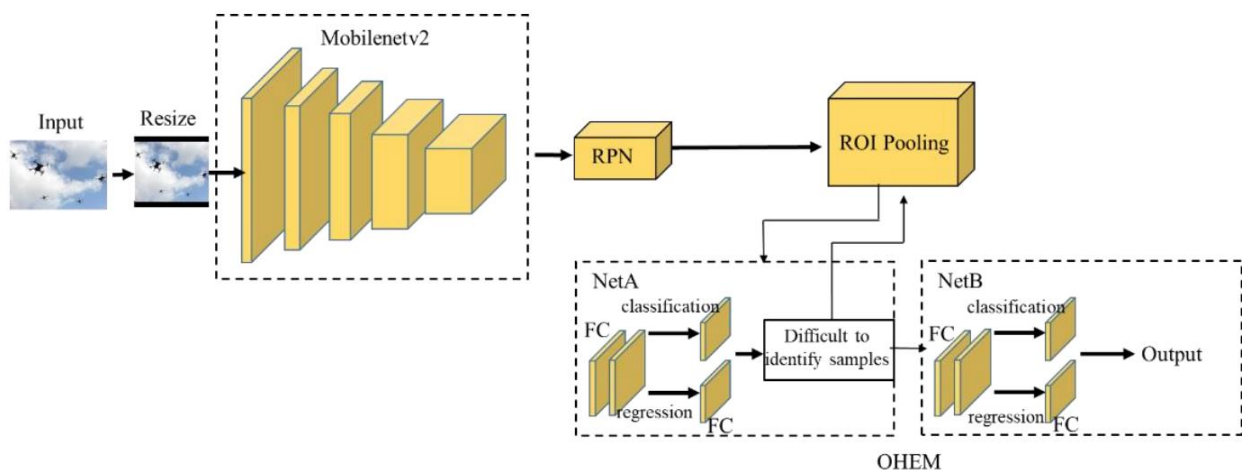


Рисунок 1.4 – Структура покращеної архітектури моделі Faster R-CNN

Запропонований метод досягнув точності 93.8%, що на 2.8% вищий за результат, отриманий за допомогою лише архітектури YOLOv5. Це демонструє, що інтеграція Kmeans++ та ОНЕМ в Faster R-CNN дозволяє значно покращити точність виявлення цілей у військових умовах, де важливо точно розпізнавати навіть найменші або масковані об'єкти.

1.2 Постановка задачі

Можна побачити, що більшість досліджень використовують архітектури нейронних мереж (YOLO, Faster R-CNN, EfficientDET), так як вони є одним з найкращих підходів для вирішення цієї задачі. Хоча, є різні варіанти вирішення завдання виявлення об'єктів на зображенні, такі як гібридні системи чи системи нечіткої логіки.

Варто відзначити, що відсутні дослідження та розроблені системи з новими версіями архітектури YOLO, а саме YOLOv9, YOLOv10, YOLOv11 в яких було покращена швидкодія та зменшення кількості параметрів. Використання цих архітектур може покращити вже наявні результати [19].

Розглядаючи можливість використання нових версій YOLO, слід враховувати різні характеристики цих архітектур, які в кожній версії поділяються на моделі для більш швидкого чи точного навчання за рахунок параметрів, потреб в обчислювальних ресурсах та іншого [20].

Додатковий акцент у дослідженнях робиться на можливості роботи системи в умовах обмежених обчислювальних ресурсів. Це може бути реалізовано через використання більш «легких» моделей (наприклад, MobileNet або оптимізованих версій YOLO) та алгоритмів компресії моделі.

Архітектури та методи, такі як YOLO, RetinaNet, Faster R-CNN та інші, показали себе як потужні інструменти для виявлення об'єктів на зображеннях, зокрема в умовах реального часу. Проте військові об'єкти мають свої специфічні риси, а використання дронів, як основного джерела даних вимагає адаптації моделей до специфіки отриманих зображень. Хоча системи на основі нечіткої

логіки можуть бути корисними для певних аспектів задачі, нейронні мережі забезпечують найкращий баланс між точністю, швидкістю й універсальністю.

Аналізуючи попередні дослідження для задачі виявлення військових об'єктів на динамічних зображеннях, можна побачити спільні риси та проблеми, важливо проаналізувати ці спостереження для подальшого дослідження і постановки задачі.

Узагальнені риси та проблематика наведені в таблиці 1.1.

Таблиця 1.1 – Проблематика та загальні риси систем виявлення об'єктів

Загальні риси систем виявлення об'єктів	Проблематика виявлення військових об'єктів
Потреба у високій точності виявлення	Варіативність об'єктів та складність розпізнавання
Швидкість обробки та придатність для реального часу	Робота системи в складних та динамічних умовах
Масштабованість та стійкість до перешкод	Малі об'єкти та рідкісні класи
Гнучкість у виявленні різних класів об'єктів	Маленька кількість актуальних загальнодоступних датасетів
Використанні модифікованих архітектур нейронних мереж	Складність анотації даних при створенні набору даних
Можливість визначати порівняно дрібні об'єкти на зображенні	
Об'єднання архітектур в гібридні системи	

Основна проблема, полягає в тому, що розпізнавання військових об'єктів є складною задачею. Система потребує високу обчислювальну потужність, здатність працювати з відео високої роздільної здатності та забезпечувати високу точність в умовах складного оточення, наприклад шум та варіабельність зображень, масштаб

об'єктів, погодні умови, швидкість руху. Крім того, актуальність роботи посилюється відсутністю великих загальнодоступних наборів даних для задач виявлення військових об'єктів, що вимагає створення власного набору даних для навчання моделей. Це дозволить не тільки розв'язати конкретну задачу виявлення, але й відкрити нові перспективи для розвитку подібних систем в інших областях, де аналіз зображень з дронів є критично важливим.

Для розробки системи виявлення військових об'єктів на динамічних зображеннях, потрібно створення надійної моделі, яка:

- швидко і точно виявляє військові об'єкти на зображеннях з дронів в реальному часі, зокрема у складних умовах (низька видимість, камуфляж, рух);
- має високу стійкість до шуму та перешкод завдяки методам обробки зображень та використанню додаткових модифікацій, таких як аргументація даних і функції втрат, які підвищують чутливість до малих та рідкісних об'єктів;
- ефективно масштабується для роботи в різних середовищах та умовах, таких як виявлення техніки на полі бою, розпізнавання прапорів або відстеження рухомих цілей.

Метою роботи є вдосконалення точності виявлення, розпізнавання та класифікації військових об'єктів на зображеннях з дрона, шляхом оптимізації і модифікації параметрів різних архітектур нейронних мереж та розробки системи використовуючи найефективніші з них.

Об'єктом є процеси розпізнавання об'єктів на фото та відео.

Предметом дослідження є моделі та архітектури нейронних мереж для задачі розпізнавання військових об'єктів.

Таким чином, задача полягає у створенні високоефективної системи, що відповідає вимогам військових операцій, забезпечуючи як точність, так і швидкість у динамічних умовах.

Для виконання задачі по розробці такої системи, потрібно самостійно зібрати набір даних, який буде містити різні військові об'єкти (танк, бронетехніка, солдат і т.п.).

Створення такої системи забезпечить підвищення безпеки та ефективності військових операцій, скорочення часу на аналіз даних та прийняття рішень, а також зниження ризику для військовослужбовців під час виконання бойового завдання.

Таким чином, дослідження у цій сфері має велике значення, оскільки може допомогти мінімізувати втрати та забезпечити більш точне проведення операцій.

Висновки до розділу 1

У першому розділі обґрунтовано актуальність задачі автоматичного виявлення військових об'єктів, зокрема на зображеннях і відео з дронів, як важливого інструменту для військових операцій. Проаналізовано сучасні виклики, пов'язані з розпізнаванням об'єктів у складних умовах. Проведено огляд останніх досліджень та їх результатів. У рамках постановки задачі визначено основні вимоги до створення системи виявлення військових об'єктів на динамічних зображеннях з використанням нейронних мереж.

2 МЕТОДИ ТА АРХІТЕКТУРИ НЕЙРОННИЙ МЕРЕЖ ДЛЯ ЗАДАЧІ РОЗПІЗНАВАННЯ ВІЙСЬКОВИХ ОБ'ЄКТІВ

2.1 CNN в задачах розпізнавання об'єктів на зображеннях

У задачах розпізнавання об'єктів на зображеннях, CNN є основними моделями для цього, CNN може бути використана для точного передбачення особливих ознак об'єкту. Мережа може навчитися розпізнавати військові об'єкти, використовуючи різні набори даних. Це зумовлено їх здатністю автоматично виділяти ознаки (features) об'єктів на різних рівнях абстракції, що є критичним для таких завдань, як виявлення та класифікація військових об'єктів [21].

Також, CNN може бути використаний для аналізу різних аспектів зображень, таких як місцевість та кількість об'єктів. CNN може оцінювати співвідношення візуальної інформації, включаючи колір та форму.

Створені мережі можуть допомогти розробляти системи для розв'язку різноманітних задач. Наприклад визначення окремих типів об'єктів, їх кількості або положення.

Отже, CNN можуть бути використані для створення потужних систем розпізнавання військових об'єктів на зображеннях, які допоможуть в різних військових операціях та планування, так як мають ряд особливостей, на відміну від класичних методів машинного навчання, де ознаки для розпізнавання об'єктів вибиралися вручну. CNN автоматично вчаться виділяти релевантні ознаки об'єктів. Наприклад, CNN можуть виділяти контури танків, вертольотів або інших військових об'єктів і відрізнити їх від неважливих деталей фону.

CNN добре масштабуються та мають високу точність для обробки великих обсягів даних, тому їх також можна застосовувати до даних, які надходять з дронів у реальному часі. Мережі можна оптимізувати для паралельної обробки на графічних процесорах (GPU), що робить їх придатними для обробки потокових відеоданих.

Застосування CNN в обробці відео, більш складна задача за обробку статичних

зображень, оскільки потрібно враховувати контекст між кадрами, для цього використовуються деякі методи та архітектури роботи з CNN нейромережами, такі як «Frame by frame», де аналізується кожен кадр відео окремо, як статичне зображення, а потім результати для кожного кадру об'єднуються, щоб відстежувати об'єкти через відеопотік. Для покращення точності використовують методи трекінгу, які допомагають зберігати інформацію про положення об'єкта через кілька кадрів. Це знижує ймовірність втрати об'єкта через рух або перешкоди.

Також, часто використовують вже готові архітектури такі як 3D CNN, які враховують часовий вимір (послідовність кадрів) у CNN.

Модель згорткової нейронної мережі (CNN) виконує два етапи: виділення ознак і класифікацію.

На етапі виділення ознак зображення проходить через різні фільтри та шари, які вилучають з нього інформацію та особливості. Після цього зображення переходить до етапу класифікації, де воно класифікується відповідно до цільової задачі. Стандартна структура CNN вказана на рис. 2.1 [22].

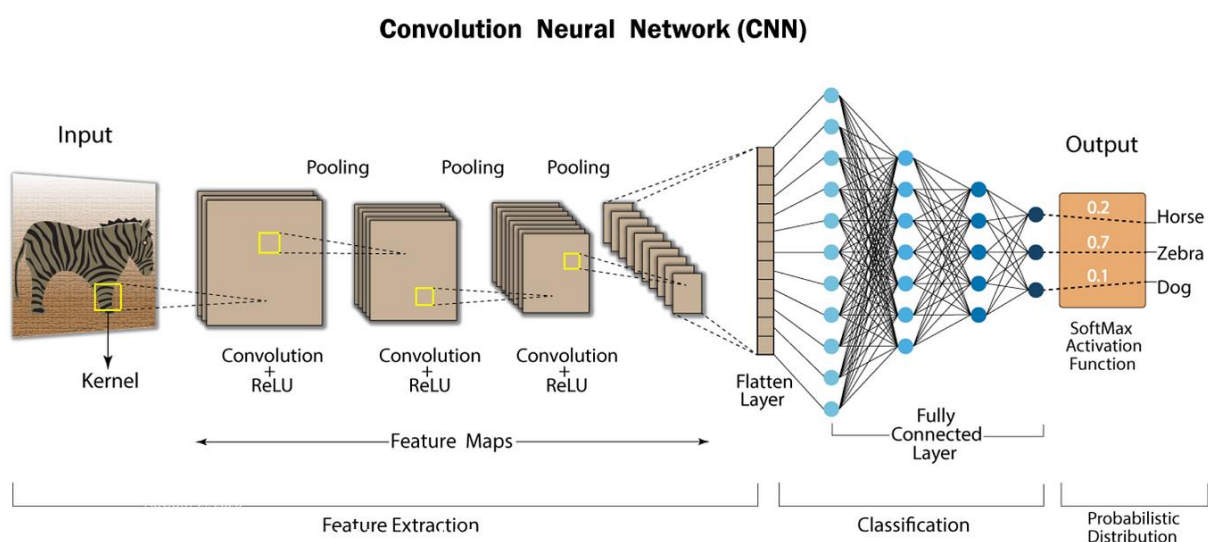


Рисунок 2.1 – Стандартна структура CNN

Модель CNN містить наступні компоненти:

- вхідний шар (Input Layer);

- шар згортки з функцією активації (Convolutional Layer);
- шар об'єднання (Pooling Layer);
- з'єднаний шар (Fully Connected Layer).

Вхідний шар – це вхідне зображення, яке може бути в градаціях сірого чи RGB. Для оптимальної обробки, перед передачею до моделі, пікселі зображення нормалізуються, перетворюючи діапазон від 0 до 255 до діапазону від 0 до 1.

Шар згортки - яка допомагає нам побачити деталі в зображенні. Фільтр, як правило, це квадратний блок, який ковзає по зображенню, вишукуючи певні особливості, такі як краї або контури. Кожне проходження фільтра створює новий шар зображення, який містить інформацію про знайдені особливості. Ці шари зображень, разом взяті, називаються картою ознак (feature maps). Карта ознак містить важливу інформацію про зображення, яку можна використовувати для його класифікації.

Шар об'єднання - це компресор, який зменшує розмір карти ознак, не втрачаючи при цьому важливої інформації. Він робить це, об'єднуючи групи пікселів, які мають схожі значення. Це допомагає знизити кількість параметрів, які потрібно навчити моделі, що може покращити її продуктивність.

Повністю з'єднаний шар - приймає інформацію з шару об'єднання та використовує її для класифікації зображення. Він робить прогнози для кожного можливого класу зображення за допомогою вивченої інформації на попередніх етапах (шари згортки та об'єднання), класифікуючи його у відповідну мітку шари згортки й об'єднання), з вихідним шаром і, зрештою, класифікує вхідне зображення у потрібну мітку

2.2 Використання архітектури YOLO в задачах розпізнавання військових об'єктів

Архітектура YOLO, як і Faster R-CNN та EfficientDet належать до категорії CNN, оскільки основне завдання виявлення об'єктів полягає в обробці зображень, де згорткові шари відіграють ключову роль у вивченні просторових залежностей та

виділенні важливих ознак.

YOLO (You Only Look Once) – це модель нейронної мережі, яка була створена для швидкого і точного розпізнавання об'єктів на зображеннях у реальному часі. Основною метою її розробки було знизити обчислювальні витрати та час обробки, що було досягнуто шляхом застосування підходу "єдиного погляду", де вся картинка аналізується одночасно, а не окремими регіонами, як це робилося у попередніх моделях, наприклад, у CNN.

YOLO розглядає задачу детекції об'єктів як задачу регресії, де для кожного зображення модель намагається передбачити прямокутні області (bounding boxes) для об'єктів та їх класифікацію за одним проходом нейронної мережі. Це забезпечує значно вищу швидкість порівняно з багатьма іншими підходами, такими як R-CNN, які виконують детекцію об'єктів у кілька етапів.

Принцип роботи YOLO полягає в тому, що зображення поділяється на сітку, і кожна клітинка цієї сітки відповідає за передбачення наявності об'єкта в межах цієї клітинки. Кожна клітинка прогнозує ймовірності наявності об'єкта, координати для його bounding box і клас об'єкта. Модель здатна одночасно передбачати декілька об'єктів і визначати їхні межі, що дозволяє використовувати її для складних зображень з багатьма елементами.

На початку вхідне зображення розбивається на рівні частини (сітку). Далі кожна клітинка цієї сітки намагається визначити, чи є об'єкт в межах своєї області. Для цього вона прогнозує межі об'єкта, клас об'єкта (наприклад, людина, автомобіль, військова техніка тощо) і ймовірність, що даний об'єкт присутній в межах її області. Після цього передбачені координати меж об'єкта об'єднуються і створюється остаточний висновок про місцезнаходження об'єкта на зображенні. Це дає змогу одночасно знаходити і класифікувати кілька об'єктів з високою точністю [23].

У задачах розпізнавання військових об'єктів на динамічних зображеннях YOLO використовується через свою здатність швидко і точно аналізувати зображення в реальному часі. Це критично важливо для військових операцій, де

затримки можуть бути фатальними. Завдяки тому, що модель одночасно прогнозує як положення об'єктів, так і їхній клас, це дозволяє ідентифікувати специфічні елементи військової техніки, наприклад, танки, літаки або прапори, на зображеннях з дронів.

Відмінність YOLO від інших моделей, таких як Faster R-CNN або Mask R-CNN, полягає в тому, що інші моделі спочатку пропонують можливі регіони, в яких можуть знаходитися об'єкти, і лише потім виконують класифікацію. Це призводить до збільшення часу обробки, що може бути неприйнятним в умовах реального часу. YOLO ж забезпечує баланс між швидкістю і точністю, що робить його популярним для динамічних задач.

Нові версії YOLO, а саме YOLOv9 та YOLOv10, були розроблені в 2024 році з метою покращення результатів попередніх варіантів, зосереджуючись на трьох ключових аспектах: підвищення точності, зниження обчислювальних витрат і покращення здатності працювати в складних умовах, таких як низька видимість або мала роздільна здатність об'єктів. Версії YOLOv9 та YOLOv10 зосереджені на вдосконаленні механізмів навчання та вдосконаленні архітектури з використанням глибших мереж, ефективніших функцій активації та нових оптимізаційних методів для збільшення точності при менших обчислювальних витратах [24].

YOLOv10 використовує більш просунуті техніки оптимізації, зокрема адаптивні механізми обробки уваги, які покращують здатність моделі зосереджуватись на найбільш значущих частинах зображення. Це дозволяє більш ефективно працювати з великими і складними зображеннями, які включають багато дрібних деталей, таких як військова техніка. Крім того, YOLOv10 інтегрував механізми послідовного подвійного призначення, які дозволяють моделі краще обробляти контексти об'єктів і взаємодію між ними покращуючи прогнозування регіонів, що підвищує її здатність до виявлення складних і багатofункціональних об'єктів на одному зображенні [25].

Це перша архітектура YOLO, яка відійшла від минулого методу розпізнавання зображення та має спрощену структуру, що складається з меншої

кількості шарів та параметрів, не жертвуючи при цьому точністю. Це дозволяє досягти високої швидкості роботи на різних пристроях

Різниця між новими версіями YOLO і попередніми полягає не тільки у збільшенні точності і швидкості, але й у покращеній здатності працювати в більш складних середовищах. Наприклад, для умов бойових дій, де багато шуму, дрон може знімати в умовах низької видимості або туману, і нові версії YOLO здатні краще справлятися з цими умовами завдяки глибшим мережам та новим функціям оптимізації.

2.3 Архітектури Faster R-CNN та EfficientDet в задачах розпізнавання військових об'єктів

Faster R-CNN (Region-based Convolutional Neural Networks, покращує попередні варіанти R-CNN і Fast R-CNN, інтегруючи процес виявлення регіонів з можливими об'єктами в єдиний етап, що робить його швидшим. По своїй суті та меті створення, ця архітектура схожа на YOLO але має інший принцип роботи.

Спочатку йде конволюційна обробка зображення (Extracting Features). На початковому етапі вхідне зображення пропускається через глибоку конволюційну мережу (наприклад, ResNet). Мета цього етапу – витягти ознаки (features) з зображення, які зможуть в подальшому використовуватися для ідентифікації об'єктів. Конволюційні шари генерують карту ознак (feature map), яка містить інформацію про текстури, форми і краї об'єктів. Приклад роботи структури R-CNN та Fast R-CNN наведений на рисунку 2.2 [26].

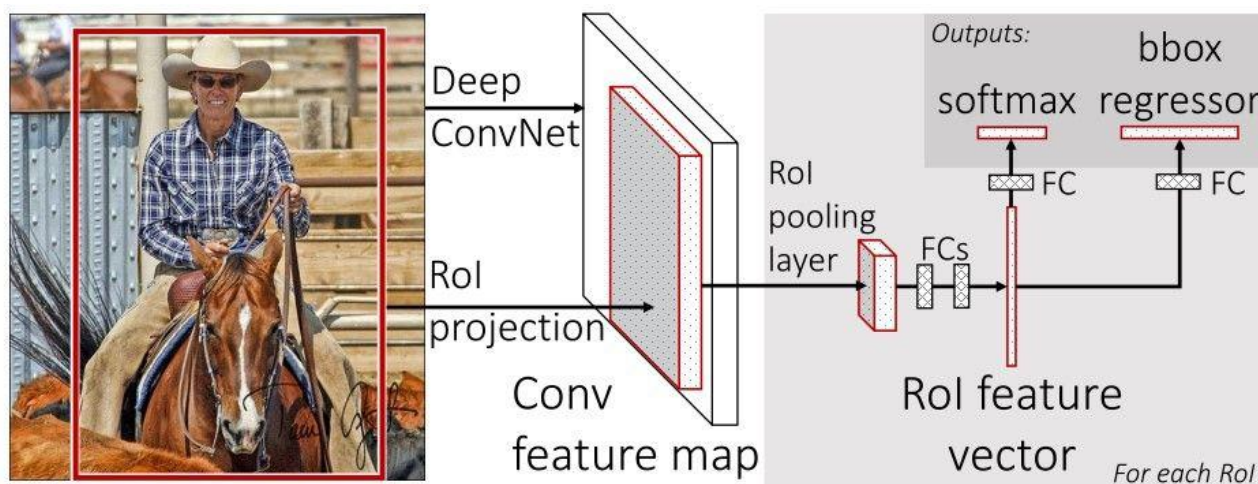


Рисунок 2.2 – Структура R-CNN та Fast R-CNN

Основна відмінність Faster R-CNN від попередніх архітектур полягає у використанні спеціальної мережі для пропозиції регіонів – Region Proposal Network (RPN). На цьому етапі, замість того, щоб спочатку пропонувати можливі регіони, RPN безпосередньо генерує потенційні області, де можуть знаходитися об'єкти. Це робиться шляхом сканування карти ознак, згенерованої на попередньому етапі, за допомогою «ковзного вікна» і визначення «якорів» (anchors) – областей різного розміру та співвідношення сторін, які можуть містити об'єкти.

Далі RPN на кожній карті ознак генерує багато "пропозицій" щодо регіонів, і для кожного такого регіону пропонує ймовірність наявності об'єкта та координати bounding box. Ці пропозиції RoI (Regions of Interest) передаються до наступного етапу для їх класифікації. Після чого, використовується спеціальний шар – RoI Pooling. Він дозволяє нормалізувати розміри всіх запропонованих регіонів до фіксованого розміру, щоб усі регіони могли бути передані на подальшу обробку. Це допомагає об'єднати різнорозмірні пропозиції в єдиний формат для подальшої класифікації та уточнення меж.

На заключному етапі для кожного нормалізованого регіону виконується класифікація об'єкта і уточнення координат оточуючого прямокутника. Мережа визначає, до якого класу належить об'єкт (наприклад, танк, літак, людина) та точно обчислює його місце розташування на зображенні. Якщо об'єкта в регіоні немає, то

регіон відноситься до категорії "фон". Приклад роботи Faster R-CNN наведений на рисунку 2.3 [27].

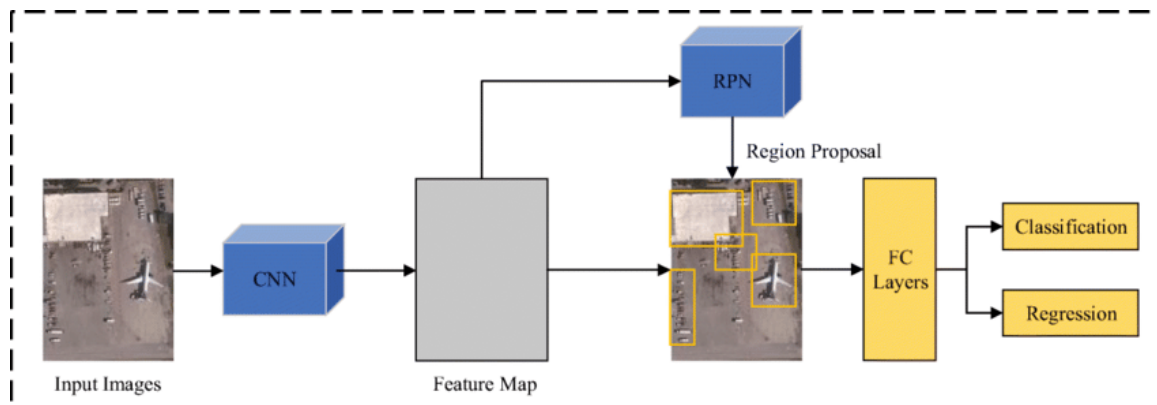


Рисунок 2.3 – Структура Faster R-CNN

Faster R-CNN відрізняється від моделей типу YOLO тим, що використовує двоетапний підхід: спочатку пропонуються можливі регіони, а потім ці регіони класифікуються. Це забезпечує високу точність, але впливає на швидкість. YOLO, навпаки, виконує обидва етапи одночасно, що дозволяє працювати швидше, але іноді з меншою точністю. Але через це, Faster R-CNN забезпечує кращу продуктивність для об'єктів різних масштабів. Використання багаторівневої карти ознак дозволяє йому обробляти як великі, так і дрібні об'єкти, що робить його ефективним для військових зображень, де можуть бути як великі транспортні засоби, так і маленькі деталі, наприклад, зброя або прапори.

EfficientDet – головна ідея цієї архітектури полягає в досягненні балансу між продуктивністю (висока точність) та ефективністю (низька потреба в ресурсах), що робить її ідеальною для застосування в реальних умовах, зокрема для аналізу динамічних зображень, таких як відео.

Модель працює на принципі масштабування глибини (depth), ширини (width) і роздільної здатності (resolution) мережі у вигляді піраміди, що дозволяє збалансувати споживання ресурсів та точність.

На цьому етапі вхідне зображення обробляється через кілька шарів

EfficientNet, які витягують ключові ознаки, як-от краї, текстури, форми і кольори, і створюють карту ознак (feature map).

Особливість EfficientDet – це BiFPN (Bi-directional Feature Pyramid Network). BiFPN об'єднує багаторівневі ознаки, взяті з різних рівнів піраміди ознак. Піраміди ознак потрібні для обробки об'єктів різних розмірів на зображенні, оскільки об'єкти можуть бути як великими, так і малими, і їх потрібно виявляти однаково ефективно. На відміну від традиційних FPN (Feature Pyramid Network), BiFPN використовує бінапрямлений потік даних, що дозволяє більш ефективно зважувати важливі ознаки на різних рівнях мережі [28].

Після того, як ознаки були оброблені BiFPN, EfficientDet виконує прогнозування меж для кожного об'єкта на зображенні, а також його класифікацію. У рамках цього етапу, модель вирішує дві задачі: класифікація об'єкта (який об'єкт знаходиться на зображенні) і регресія прямокутника (точне визначення координат меж об'єкта). Структура архітектури EfficientDet наведена на рисунку 2.4 [29].

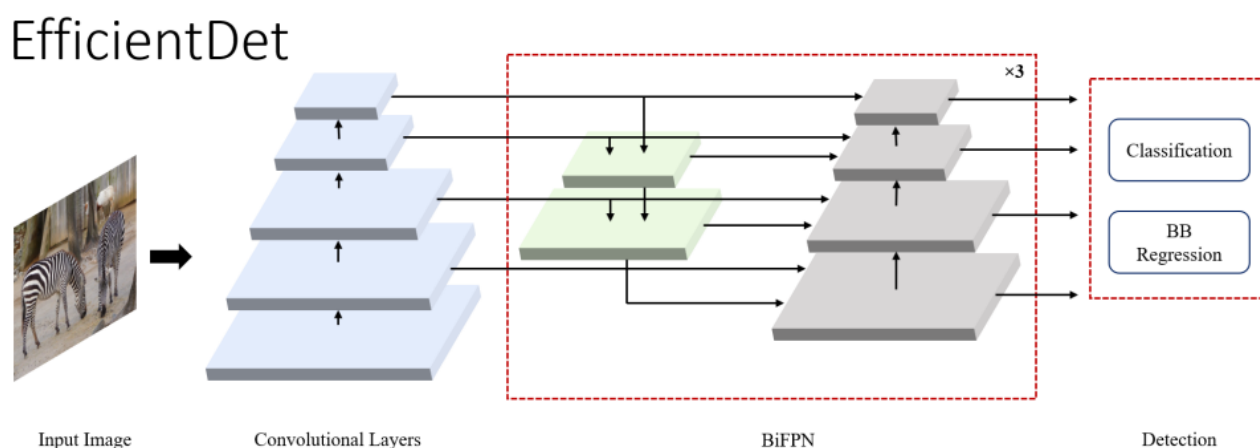


Рисунок 2.4 – Структура архітектури EfficientDet

Одна з головних переваг EfficientDet полягає в її продуктивності при використанні обмежених ресурсів. У військових умовах, де потрібно аналізувати зображення або відео в реальному часі, EfficientDet може працювати на мобільних або вбудованих пристроях (навіть на дронах чи інших військових платформах),

забезпечуючи високу точність при мінімальних затратах часу та ресурсів. Завдяки механізму ViFPN, модель добре розпізнає об'єкти різних розмірів. У військових сценаріях це означає, що EfficientDet може виявляти як великі об'єкти (наприклад, танки, літаки), так і дрібні об'єкти, такі як зброя або деталі спорядження.

Розроблена з акцентом на ефективне використання ресурсів. Вона може працювати на пристроях з обмеженими обчислювальними можливостями, що робить її ідеальною для використання на безпілотах, мобільних пристроях та інших системах, де обмежені ресурси є важливим фактором. Водночас, інші моделі, такі як Faster R-CNN, можуть бути точнішими, але потребують більше ресурсів і не так добре підходять для пристроїв з обмеженими потужностями.

2.4 Використання нечіткої логіки в задачах розпізнавання об'єктів на зображенні

Нечітка логіка є одним з інструментом у задачах розпізнавання об'єктів на зображеннях, оскільки дозволяє працювати зі складними та неповними даними. Ця концепція заснована на моделюванні людського мислення для прийняття рішень в умовах невизначеності.

У розпізнаванні об'єктів нечітка логіка застосовується для визначення меж і характеристик об'єктів, що важливо в умовах шуму, змін освітлення або часткової видимості.

Основною перевагою нечіткої логіки є її здатність інтегрувати нечіткі вхідні дані, такі як розмиті контури або перекриття об'єктів. Наприклад, у військових умовах зображення можуть бути затемненими або забрудненими через погодні умови, дим або пил, фальшиві цілі та інші задачі з якими звичайні алгоритми часто не справляються. Використання нечітких правил дозволяє класифікувати об'єкти навіть за наявності цих перешкод. Графічний приклад такої системи наведений на рис. 2.5 [30].

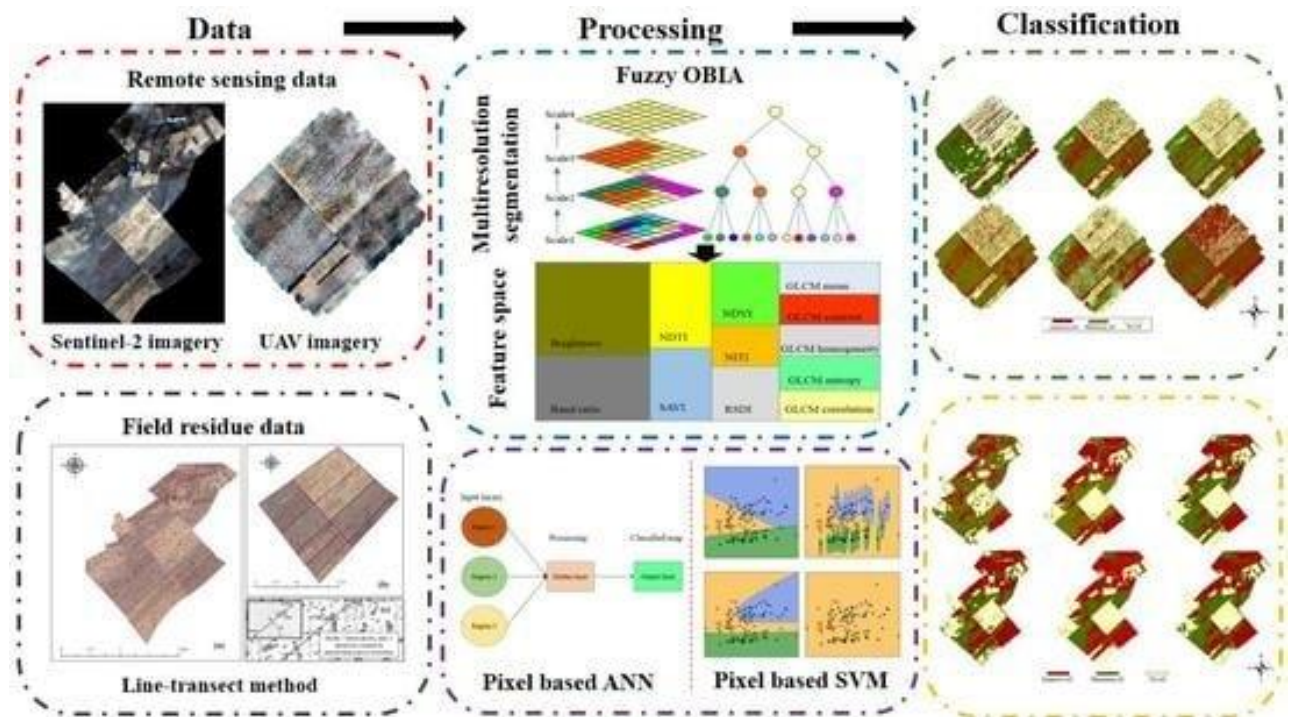


Рисунок 2.5 – Приклад системи визначення об'єкта на зображенні за допомогою нечіткої логіки

Зокрема, нечіткі системи можуть працювати з правилами, що описують взаємозв'язки між розмірами, формами або положенням об'єктів враховуючи експертну оцінку. Наприклад, якщо розпізнається танк, нечітка система може враховувати приблизні пропорції гусениць до корпусу, що дозволяє виключити можливі помилки через схожі за формою об'єкти, такі як вантажівки.

2.5 Гібридні нейронні мережі в задача розпізнавання об'єктів на зображеннях

Гібридні нейронні мережі (ГНМ) є одним із найперспективніших підходів у задачах розпізнавання об'єктів на зображеннях, включаючи військові об'єкти. Вони поєднують різні архітектури або методи для досягнення високої точності, стійкості та швидкості роботи навіть у складних умовах.

Суть таких мереж полягає у тому, що ГНМ об'єднують кілька моделей або підходів для ефективного виявлення та класифікації об'єктів. Наприклад,

комбінація згорткових нейронних мереж, рекурентних нейронних мереж, а також методів нечіткої логіки або машинного навчання дозволяє враховувати як просторові (структура об'єкта), так і часові (послідовність кадрів) аспекти.

За допомогою такого підходу, можна підвищити точність та можливість системи у виявленні малих або перекритих об'єктів або на розмитому зображення [31].

Є декілька популярних прикладів таких систем, такі як поєднання CNN та RNN, де CNN забезпечує аналіз просторових ознак об'єктів, форми або контури, а RNN (особливо LSTM або GRU) аналізує часову послідовність кадрів. Наприклад, на відео CNN виділяє характерні ознаки об'єкта, а RNN аналізує їхню зміну в послідовних кадрах для покращення точності. Для задачі розпізнавання військових об'єктів, таку систему можна створити для розпізнавання військових транспортних засобів на зображеннях дронів. CNN виділяє контури танка, а LSTM уточнює його положення, враховуючи рух у відеопотоку.

Також можна поєднати YOLO та RNN, де YOLO відповідає за швидке визначення положення об'єктів у кадрі, а RNN дозволяє уточнити їхню класифікацію або врахувати залежність між об'єктами. Приклад структури гібридної нейронної мережі наведений на рисунку 2.6 [32].

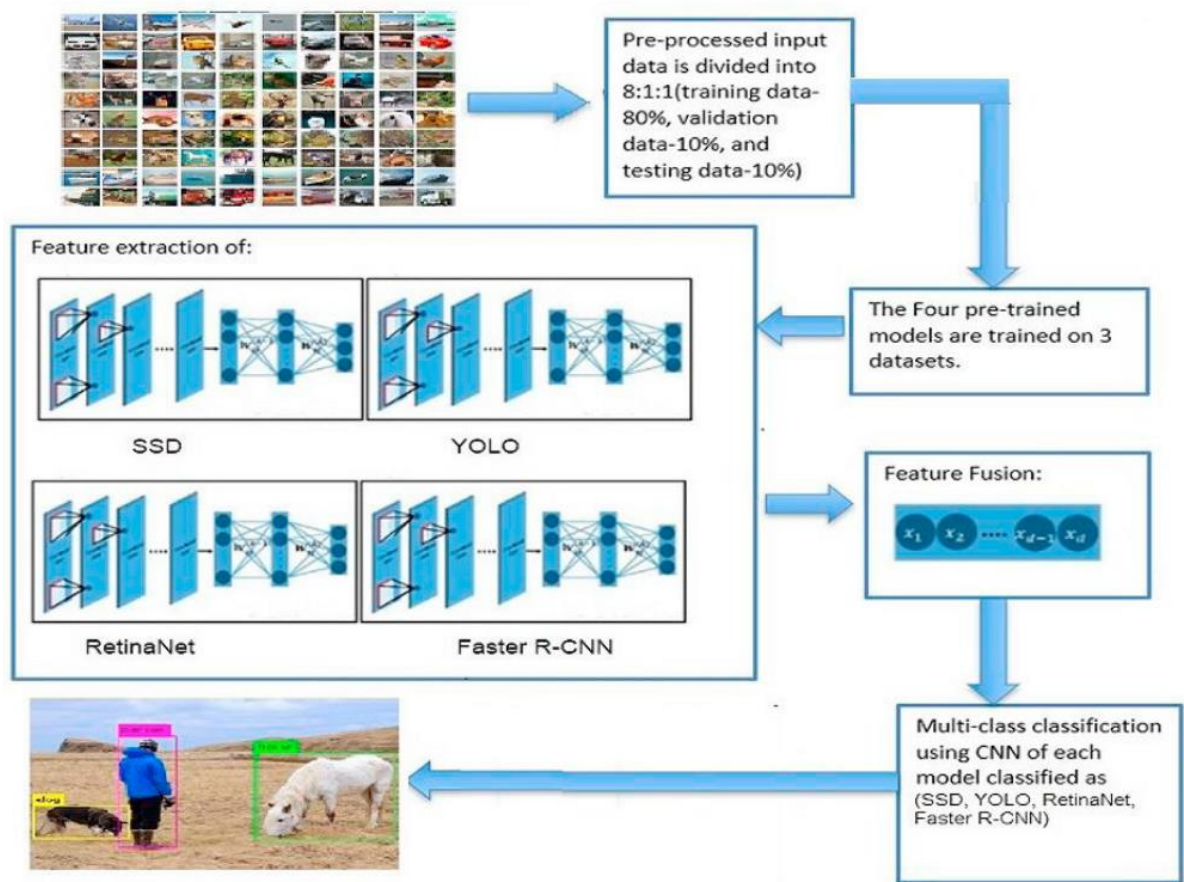


Рисунок 2.6 – Приклад системи на основі гібридної нейронної мережі

Такий підхід допомагає в розпізнаванні військових колон. YOLO визначає місце розташування транспортних засобів, а RNN визначає їхню взаємодію (наприклад, дистанцію між ними).

2.6 Порівняння архітектур та методів, які використовуються для задачі розпізнавання об'єктів на динамічному зображенні

Важливим аспектом для вирішення задачі виявлення військових об'єктів на динамічному зображенні є вибір найбільш ефективної архітектури нейронної мережі або методології для конкретних умов застосування. Серед таких підходів було розглянуто гібридні нейронні мережі, системи нечіткої логіки, а також популярні архітектури YOLO, Faster R-CNN, CNN та EfficientDet. Кожен із цих підходів має свої переваги і недоліки, що визначаються точністю, швидкістю

обробки, стійкістю до шумів і здатністю працювати в реальному часі.

Наприклад архітектура YOLO виділяється своєю швидкістю і здатністю працювати в реальному часі. Faster R-CNN демонструє високу точність при розпізнаванні об'єктів, однак потребує більших обчислювальних ресурсів. EfficientDet поєднує точність із оптимізованим використанням ресурсів. CNN залишається основою багатьох сучасних моделей завдяки своїй здатності обробляти складні візуальні ознаки. Гібридні нейронні мережі поєднують сильні сторони різних архітектур і дозволяють досягти високих результатів навіть у складних умовах. Системи нечіткої логіки, хоча і поступаються нейронним мережам за універсальністю, відрізняються гнучкістю і адаптивністю в умовах невизначеності.

Вибір найкращого підходу залежить від конкретних вимог завдання, таких як точність, швидкість обробки, обчислювальні ресурси та особливості вхідних даних. Для порівняння основних методів і архітектур, їх плюсів та мінусів створена табл. 2.1. Порівняння будуть здійснюватися між популярними архітектурами та підходами на основі швидкості обробки, точності, стійкості до шуму, вимоги до обчислювальних ресурсів, гнучкості та адаптивності, складності реалізації системи. Оцінки по критеріям будуть виставлені з результатів аналізу попередніх досліджень та відкритих джерел інформації, від «низько» до «дуже висока».

Варто зазначити, що ця таблиця дозволяє визначити найкращу архітектуру залежно від потреб конкретного завдання. Також необхідно зауважити, що система розпізнавання військових об'єктів може включати не один підхід чи архітектуру, а можливість вибору між архітектурами (наприклад YOLO або EfficientDet) або ж створену гібридну модель.

Порівняльний аналіз архітектур і підходів необхідний, для того, щоб оцінити ефективність, точність різних архітектур та мати змогу обрати найбільш ефективну за різними показникам для створення фінальної версії.

Оцінка за критеріями в даному випадку є суб'єктивною та спирається на теоретичні знання, спостереження та знайдену інформацію в мережі інтернет.

Спираючись на вимоги для задачі виявлення військових об'єктів на динамічному зображенні за допомогою архітектур нейронних мереж, проведено наступний порівняльний аналіз.

Таблиця 2.1 – Порівняння основних методів та архітектур для задачі виявлення військових об'єктів на динамічних зображеннях

Критерій	YOLO	Faster R-CNN	EfficientDet	CNN	Гібридні мережі	Системи Нечіткої логіки
Швидкість обробки	Дуже висока	Низька	Висока	Середня	Середня	Низька
Точність	Висока	Висока	Висока	Середня	Дуже висока	Середня
Стійкість до шуму	Висока	Висока	Висока	Середня	Висока	Дуже висока
Вимоги до ресурсів	Середні	Високі	Середні	Середні	Високі	Низькі
Гнучкість і адаптивність	Низька	Низька	Середня	Висока	Висока	Дуже висока
Складність реалізації	Низька	Висока	Середня	Низька	Висока	Середня

В умовах реального часу, наприклад для розпізнавання військових об'єктів на динамічних зображеннях, YOLO може бути оптимальним вибором через свою швидкість. У задачах з високими вимогами до точності та стійкості варто розглянути Faster R-CNN чи гібридні нейронні мережі. EfficientDet є збалансованим рішенням для пристроїв із обмеженими обчислювальними можливостями. У нечіткої логіки відносно низька швидкість порівняно з

сучасними архітектурами нейронних мереж, такими як YOLO чи EfficientDet, через те що вона потребує створення бази правил і функцій належності, які визначають, як обробляються входи та генеруються результати. У складних задачах, таких як розпізнавання об'єктів на зображеннях, кількість правил може значно зрости, що збільшує обчислювальні витрати на оцінку кожного з них.

Також у систем на основі нечіткої логіки немає апаратного прискорення, такого як використання GPU.

Для задачі виявлення військових об'єктів на зображеннях найдоцільніше використовувати архітектури YOLO, Faster R-CNN та EfficientDet, оскільки вони забезпечують оптимальний баланс між швидкістю, точністю та адаптивністю до різних умов. YOLO є чудовим вибором для подібних задач завдяки своїй здатності працювати в реальному часі, що особливо важливо у військових операціях, де оперативність прийняття рішень має вирішальне значення. Faster R-CNN, хоч і менш швидка, надає високу точність розпізнавання, що робить її ефективною для складних сцен із великою кількістю деталей або об'єктів, що перекриваються. EfficientDet забезпечує баланс між продуктивністю та ефективністю використання ресурсів, дозволяючи працювати на пристроях із обмеженою обчислювальною потужністю, таких як дрони.

Можливість створення гібридної мережі, яка поєднує сильні сторони цих архітектур, додатково підвищує точність і стійкість системи до складних умов, таких як погана видимість, шум або часткове перекриття об'єктів. Гібридний підхід дозволяє адаптувати систему до специфічних вимог завдання, інтегруючи швидкість YOLO, точність Faster R-CNN та ефективність EfficientDet у єдину структуру. Такий вибір є виправданим для військових застосувань, де потрібна не лише висока точність, а й здатність працювати в режимі реального часу з мінімальними ресурсами.

2.7 Загальна структура інформаційної системи для задачі виявлення військових об'єктів з динамічного зображення

Після дослідження минулих публікацій та збору теоретичної інформації про архітектури, можна спроектувати загальну структуру майбутньої системи.

Структура буде складатись зі збору даних (зображення з камери), дрони збирають відео або зображення з полів бойових дій, ці дані можуть бути в реальному часі або збереженими для подальшого аналізу. Попередня обробка даних може включати підвищення якості зображення, зменшення шуму, нормалізацію розмірів та форматів. Використання моделі обробки зображення, тобто архітектури YOLO, Faster R-CNN або EfficientDet.

Модель визначає точне місце розташування військових об'єктів на зображенні та паралельно, визначається клас об'єкта: танки, літаки, військова техніка, люди або інші військові цілі.

Модель повинна працювати в умовах бойових дій з низькою видимістю, тому важливою є здатність виявляти об'єкти в складних умовах (туман, дим, ніч).

Результати передаються до оператора в реальному часі або зберігаються для подальшого аналізу.

Завдяки проведеному аналізу та визначенню технологій та архітектур, можна побудувати загальну структуру інформаційної системи наведену на рисунку 2.7.

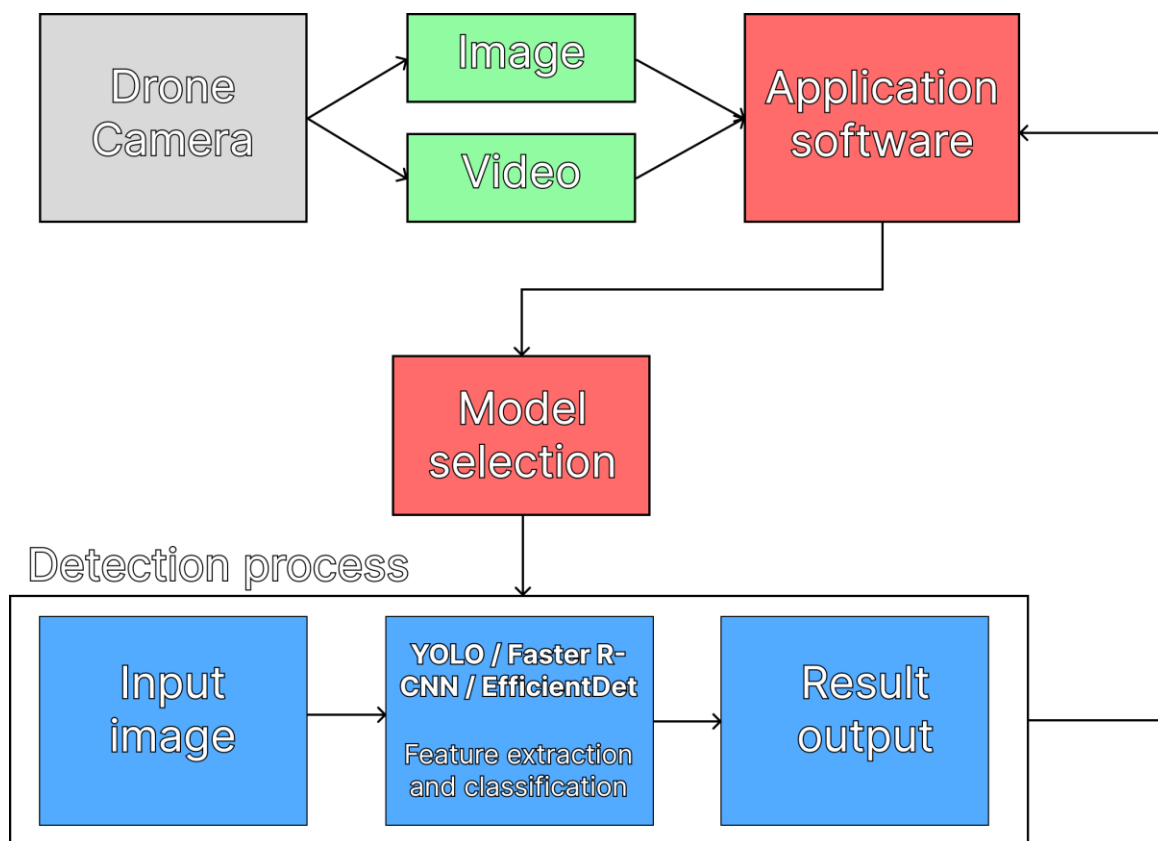


Рисунок 2.7 – Загальна структура інформаційної системи для виявлення військових об'єктів з дрону

Програма призначена для виявлення об'єктів на зображеннях або відео, отриманих з дрона.

Дрон оснащений камерою, яка збирає зображення або відеоматеріал, далі подає вхідні данні в застосунок, де обирається одна з навчених моделей, дані подаються в модель де проходять обробку (прибирається шум, змінюється розмір зображення, колір і т.п). Обрана модель аналізує зображення, виділяючи характерні ознаки об'єктів, які потрібно виявити. Після цього модель класифікує ці ознаки, відносячи їх до відповідних класів об'єктів.

Результатом роботи програми є інформація про виявлені об'єкти, яка може включати координати об'єктів на зображенні, їх класи (наприклад, «танк», «артилерія», «людина») та рівень впевненості в результаті класифікації.

Після аналізу результатів цих моделей, можна дійти висновку, щодо їх ефективності в конкретній ситуації, що дозволить спроектувати та розробити

систему на їх основі.

Приклад структури системи є попереднім і буде змінюватись спираючись на результати подальшого дослідження, при цьому залишиться основна ідея в збирання зображення, його обробці та процесу детекції.

Висновки до розділу 2

У цьому розділі проведено детальний аналіз сучасних архітектур нейронних мереж, таких як CNN, YOLO, Faster R-CNN, EfficientDet, гібридних нейронних мереж та систем на основі нечіткої логіки, з метою визначення їх придатності для задачі розпізнавання військових об'єктів. Охарактеризовано переваги та недоліки кожної з архітектур, що дозволило виявити найбільш перспективні з них для вирішення задач у режимі реального часу. Було визначено загальну структуру для системи для виявлення військових об'єктів з дрону.

Отримані результати аналізу будуть використані для подальшої розробки системи автоматизованого виявлення військових об'єктів, зокрема для вибору оптимальної архітектури та налаштування параметрів моделі.

3 СТВОРЕННЯ НАБОРУ ДАНИХ. ПОРІВНЯЛЬНИЙ АНАЛІЗ МОДЕЛЕЙ НЕЙРОННИХ МЕРЕЖ

Процес навчання моделей нейронних мереж для задач виявлення військових об'єктів базується на використанні сучасних архітектур, таких як YOLO (версії 9, 10, 11), Faster R-CNN та EfficientDet які забезпечують високу продуктивність та точність. Ці моделі здатні ідентифікувати військові об'єкти на зображеннях за допомогою комплексного аналізу просторових і контекстуальних характеристик, дозволяючи їх використання в складних умовах, наприклад, під час військових дій.

Порівняльний аналіз цих моделей, дозволить отримати необхідну інформацію про їх ефективність та застосувати у подальшій роботі.

Тому, у цьому розділі буде описано процес збору зображення та створення датасет. Також, проведена обробка датасету для того, щоб досягти рівномірного розподілу і детально розглянеться процес навчання моделей на основі сучасних архітектур нейронних мереж, зокрема сімейства YOLO версій v9, v10 та v11. Основною метою є визначення ефективних архітектур нейронних мереж для подальшого проектування та розробки системи виявлення військових об'єктів на зображеннях, що є важливим у контексті сучасних технологій спостереження та безпеки.

3.1 Збір та створення набору даних

Для навчання моделей необхідний великий та якісний датасет, що містить зображення з військовими об'єктами різних. Датасет повинен бути збалансованим, містити різноманітні сценарії, умови освітлення та кути огляду, щоб модель могла навчитися розпізнавати об'єкти в складних умовах.

Для формування базового набору зображень використовувалися відкриті бази даних, такі як, Roboflow або відкриті репозиторії. Після завантаження наборів зображень з відкритих джерел, обов'язково перевіряється остання дата оновлення цих зображень, щоб надалі виконувати пошук новішого матеріалу, який не буде

повторюватись.

Більша частина даних зібрана вручну шляхом пошуку відео та зображень в мережі інтернет, соціальних мережах та відкритих джерелах.

Обробка відеофайлів відбувалась розбиттям їх на фрейми (кадри), шляхом рендеру в 10 кадрів на секунду. Зібрані зображення завантажуються в сервіс Roboflow для подальшої обробки та анотації.

Сервіс Roboflow, дозволяє анотувати зображення використовуючи вбудовані веб інструменти, які дозволяють виділити об'єкти на зображенні і позначити їх відповідним лейблом. Також, сервіс має автоматичне анотування, яке обмежено певною кількістю зображень в день [33].

Анотація здійснюється за допомогою міток (класів), що визначають положення об'єктів на зображеннях у форматі, сумісному з моделями YOLO. Процес анотування зображень наведений на рис. 3.1.



Рисунок 3.1 – Анотація зображення за допомогою Roboflow

Після анотації всіх зображень, відбувається процес попередньої обробки. Використовуючи інструменти сервісу Roboflow видаляються зображенні без класів (NULL), а також змінюється розмір усіх зображень, щоб вони відповідали одне

одному.

Некоректний ресайзинг може призвести до спотворення співвідношень об'єктів у зображенні. Наприклад, якщо пропорції зображення не зберігаються, об'єкти можуть бути розтягнутими чи стиснутими, що ускладнює їх розпізнавання. В результаті, модель може навчитися не суттєвим характеристикам, а випадковим шумам чи неправильним ознакам. Це особливо важливо для виявлення військових об'єктів, таких як танки, солдати чи артилерія, де навіть невеликі особливості, такі як форма чи розташування, можуть бути критичними для ідентифікації [34].

Набір даних складається з 10106 зображень, розміром 640 на 640 пікселів та поділених на 9 класів, а саме: «Tank», «AAW», «APC», «Artillery», «IFV», «MTLB», «Soldier», «Truck», «Vehicle».

Ці класи охоплюють найчастіші типи військових об'єктів на полі бою, такі як, танки, броньовані машини піхоти, артилерію, звичайну техніку і т.п.

Особливістю в даному випадку є те, що на одному зображенні може бути декілька різних класів. Через це виникає нерівність між об'єктами, наприклад мітка «Tank» трапляється на більше ніж половині зображень.

Графік розподілу класу наведений на рис. 3.2.

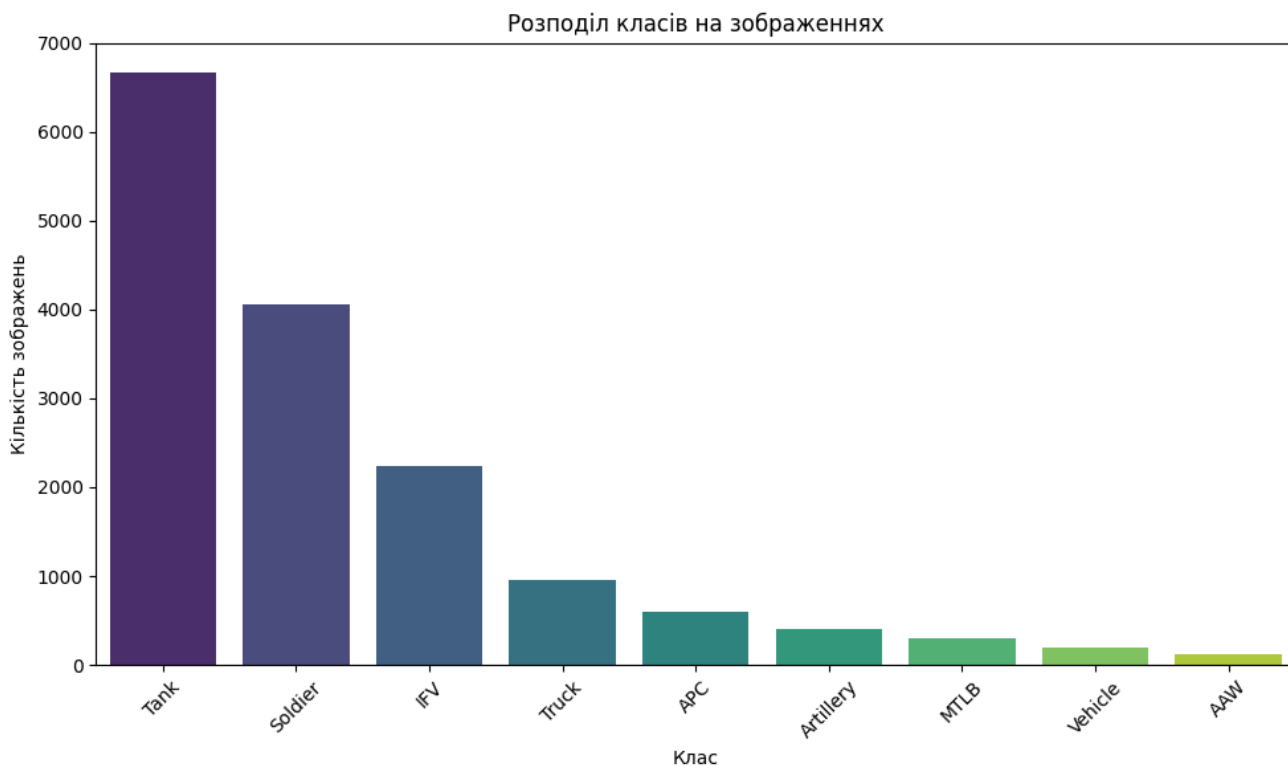


Рисунок 3.2 – Розподіл класів по зображенням в наборі даних

Як ми бачимо з графіку, клас «Tank» має найбільшу кількість зображень (понад 6000), що свідчить про суттєву перевагу цього класу у вибірці. Другим йде клас «Soldier», приблизно 4000 зображень. Клас «IFV» (бойова машина піхоти) займає третє місце із значно меншою кількістю (близько 2500). Інші класи рідше зустрічаються, тому вони представлені в ще меншій кількості, найменший клас займає системи ППО.

Така перевага в кількості класів з піхотою танками та БМП показують, що це є основними військовими одиницями наступу на полі бою.

3.2 Балансування набору даних

Для навчання нейронної мережі, важливим етапом є балансування набору даних. Це дозволить забезпечити рівномірний розподіл класів, представлених у даних, які використовуються для навчання моделей машинного навчання. У випадку виявлення військових об'єктів на зображеннях, балансування є критично

важливим, оскільки якість моделі безпосередньо залежить від структури даних.

Якщо набір даних нерівномірно розподілений, модель буде «зміщена» на користь класів, які мають більше представників. Наприклад, якщо 80% зображень належать до танків, а лише 20% до артилерії, модель може навчитися розпізнавати лише танки, ігноруючи менш представлені класи.

Для того, щоб досягти рівномірного розподілу класів в датасеті, було вирішено провести більш глибоку обробку та налаштування даних.

Методами які використовувались для балансування, були:

- об'єднання схожих класів: наприклад, класи, які включали небойову техніку, були об'єднані в один «Vehicle»;
- метод Undersampling: зменшення домінуючого класу, шляхом відбору підмножини зображень із домінуючого класу, щоб вирівняти їх кількість з менш представленими класами;
- аугментація: генерація нових зображень для менш представлених класів за допомогою різних методів.

Аугментація даних – це техніка штучного розширення набору даних шляхом створення нових варіантів існуючих зображень за допомогою різноманітних трансформацій. Це потрібно для того, щоб збільшити розмір і різноманітність набору даних, покращити узагальнюючу здатність моделей і зменшити ризик перенавчання.

Для даного набору даних, було застосовано декілька видів трансформації, особливості яких описані нижче.

Поворот на $\pm 10^\circ$. Ротація дозволяє моделі навчитися впізнавати об'єкти, які розташовані під різними кутами, наприклад, танк або інша техніка може бути представлення під різними ракурсами, і ця трансформація забезпечує стійкість до таких варіацій. А також збільшує кількість зображень, адже замість одного зображення, з'являються додаткові з іншим нахилом (див. рис. 3.3).



Рисунок 3.3 – Приклад анотованого зображення ППО

Додавання насиченості зображення на 15-20%. Використовується для урахування різних змін освітлення об'єкта чи оточення, наприклад камуфляж або ефекти погоди, такі як яскраве сонце чи тьмяне небо. Така обробка, допомагає моделі навчитися ігнорувати неважливі кольорові зміни і зосереджуватись на формах і структурах об'єктів (див. рис. 3.4).



Рисунок 3.4 – Зображення з підвищеним рівнем насиченості

Аугментація була проведена при 3x множителі, тобто загальна кількість зображень, до яких застосовували різні методи трансформації, збільшилась в 3 рази. Аугментація проводилась на стандартному рівні, тобто шляхом маніпуляцій з усім існуючим зображенням.

Однак, варто зазначити, що можливо покращення виявлення деяких класів за рахунок маніпуляції з їх обмежувальними рамками. Це відносно новий підхід, де метод змінює лише вміст всередині конкретних обмежувальних рамок. Наприклад, можна змінювати яскравість або розмиття об'єкта всередині рамки, залишаючи фон незмінним. Це надає більший контроль над створенням навчальних даних, що відповідають специфічним умовам завдань [35].

Після проведеного балансування набору даних, було отримано фінальну версію датасету.

Кількість класів, становить 5, а саме: «Tank», «Artillery», «APC», «Soldier», «Vehicle».

Кількість зображень в наборі даних становить 18620 зображень, розподіл класів на зображеннях можна побачити на рис. 3.5.

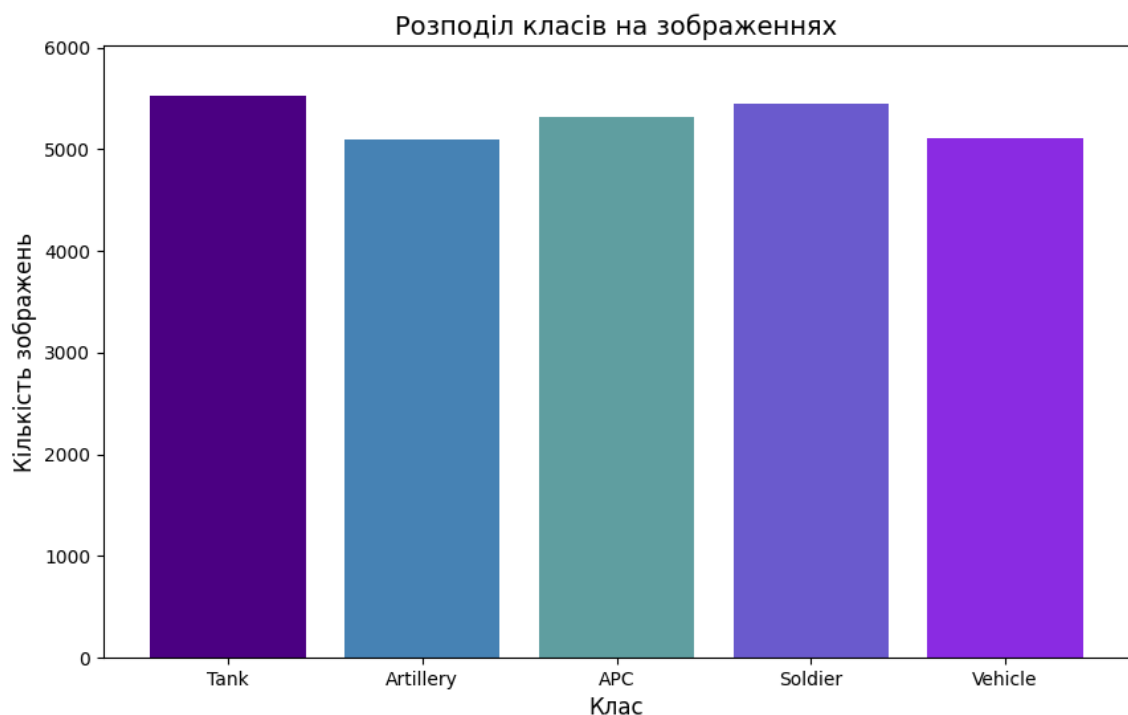


Рисунок 3.5 – Розподіл класів по зображенням в наборі даних після балансування

Можна побачити, що досягнутий розподіл вже більше схожий на рівномірний, хоча і має зміщення до класів «Tank» та «Soldier».

Датасет був поділений на 3 вибірки, тренувальну, валідаційну та тестову, які розподілені у співвідношенні 70%, 15%, 5%, відповідно.

Можна побачити, що обробка даних для їх балансування є ефективним та необхідним кроком в створенні датасету. Рівномірний розподіл, дозволить

нейронній мережі ефективно навчатися на всіх класах не віддаючи перевагу одному чи двом з них, що підвищить загальну точність моделі та ефективність застосування в виявленні військових об'єктів в реальному часі.

3.3 Метрики, які використовуються для перевірки ефективності та точності навчених моделей

Під час та після навчання модель перевіряється на валідаційному та тестовому наборах даних. Використовуються метрики, які дозволяють кількісно оцінити її ефективність та здатність до узагальнення, наприклад такі метрики як Precision (Точність), Recall (Повнота), mAP50 та mAP50-95.

Коли ми оцінюємо якість виявлення моделей, ми зазвичай порівнюємо їх з наземною правдою і ділимо на чотири групи. Випадок, коли модель правильно визначає об'єкт, називається True Positive [TP]. Коли об'єкт, якого насправді немає на зображенні, знайдений, ми говоримо, що він помилково спрацьовує [FP]. З іншого боку, коли об'єкт в основі істини не виявляється, це помилково негативний [FN]. Останню групу утворюють істинні негативи [TN]. Однак у разі виявлення об'єкта він не береться до уваги. Ми можемо інтерпретувати його як всі правильно невиявлені об'єкти – фон [36].

Ці чотири групи утворюють так звану матрицю невідповідності, спираючись на яку, можна порахувати різні показники точності та ефективності навченої моделі нейронної мережі.

Матриця невідповідностей забезпечує всебічну оцінку продуктивності моделі класифікації, пропонуючи уявлення про справжні позитивні, справжні негативні, помилкові спрацьовування та помилкові негативи, допомагаючи тонкому аналізу, що виходить за рамки базової точності. Можна сказати, що матриця невідповідностей є важливим інструментом для оцінки ефективності моделей класифікації.

Матриця невідповідностей продемонстрована в табл. 3.1.

Таблиця 3.1 – Матриця невідповідностей

		Фактично	
		0	1
Предбачення	0	Істинно- позитивний (TP)	Хибно- негативний (FN)
	1	Хибно- позитивний (FP)	Істинно- негативний (TN)

Точність (Precision або Positive Predictive Value, PPV) є суттєвою метрикою в оцінці ефективності класифікаційних моделей, особливо в тих випадках, коли критично важливо зменшити кількість хибних позитивних прогнозів. Вона показує, яку частину позитивних прогнозів моделі можна вважати правильними, тобто відсоток реальних позитивних випадків серед усіх передбачених позитивних результатів.

Точність обчислюється за наступною формулою 3.1:

$$Precision = \frac{TP}{TP+FP}, \quad (3.1)$$

де TP (True Positives) – є кількістю об'єктів, які модель правильно виявила та класифікувала;

FP (False Positives) – кількістю об'єктів, які модель виявила, але вони є помилковими (наприклад, якщо модель ідентифікувала фон як об'єкт).

Узагальнюючи, Precision показує, наскільки модель впевнена в своїх передбаченнях. Висока точність означає, що більшість об'єктів, які були виявлені, є правильними. Використовується в задачах, де помилкові позитивні спрацювання небажані.

Повнота (Recall) показує, наскільки добре модель знаходить всі об'єкти. Висока повнота означає, що модель знаходить більшість об'єктів. Важлива в задачах, де пропуски об'єктів є критичними.

Повнота обчислюється за наступною формулою 3.2:

$$Recall = \frac{TP}{TP+FN}, \quad (3.2)$$

де FN (False Negatives) – це кількість об'єктів, які модель пропустила (не виявила).

mAP метрика використовується для порівняння продуктивності моделей комп'ютерного зору. mAP надає розробникам комп'ютерного зору єдиний основний показник для порівняння, який охоплює як точність, так і запам'ятовування моделі. Пізніше можуть бути проведені порівняння за іншими показниками для кращої оцінки моделі.

Високий mAP означає, що модель має як низький рівень помилкових негативів, так і низький рівень помилкових спрацьовувань. Чим вище mAP, тим точніше і вище відкликання для вашої моделі. Інженери комп'ютерного зору прагнуть вдосконалювати mAP під час створення моделей [37].

mAP50 (Mean Average Precision) – mAP50 оцінює загальну точність моделі виявлення об'єктів при фіксованому рівні збігу 50%.

mAP50-95 – це більш суворі метрика, де середня точність для різних значень IoU від 0.5 до 0.95 з кроком 0.05.

IoU вимірює, наскільки добре передбачена рамка збігається зі справжньою рамкою. Визначається як відношення площі перетину двох рамок до площі їх [38].

Визначення ключової метрики залежить від пріоритетності завдання, яке стоїть перед системою. Наприклад, якщо кщо головна мета – знайти всі військові об'єкти, навіть якщо локалізація не ідеально точна, то mAP50 є більш відповідною метрикою. Високе значення дозволяє зрозуміти чи є на відео чи зображенні танки, системи ППО, військові автівки тощо.

У багатьох системах для моніторингу та безпеки важливо забезпечити швидку роботу моделі. Менш суворі вимоги до точності у метрики mAP50 можуть дозволити використовувати швидші моделі, які працюють в реальному часі.

У той же час, висока точність локалізації, на яку вказує метрика mAP50-95 потрібна, щоб точно визначати положення та розміри об'єктів, наприклад, для нанесення ударів. У військових завданнях дуже важливо мінімізувати кількість хибно позитивних і хибно негативних детекцій. Висока mAP50-95 свідчить про те, що модель добре справляється із суворими умовами.

Узагальнюючи, якщо виявлення військових об'єктів є частиною загальної стратегії моніторингу чи аналізу, mAP50 є ключовою метрикою. Якщо потрібна точна локалізація, то важливішим буде mAP50-95.

В задачі розробки системи виявлення військових об'єктів на динамічному зображенні з дрона, основною метрикою точності буде mAP50 але з урахуванням результатів mAP50-95.

3.4 Навчання моделей YOLO для задачі виявлення військових об'єктів на зображенні

Версії YOLOv9, YOLOv10 та YOLOv11 впроваджуючи більш глибокі та складні архітектури з покращеною здатністю до узагальнення та точнішої детекції дрібних об'єктів. Ці моделі використовують передові методи, такі як сконцентровані механізми уваги, глибокі з'єднання та покращені функції втрат, що дозволяють їм досягати високих показників ефективності.

Деякі моделі YOLO адаптовані до конкретних обчислювальних можливостей пристрою, незалежно від того, чи це центральний або графічний процесор. Більшість моделей YOLO розроблені для обслуговування різних масштабів, наприклад малі, середні та великі, які можна легко серіалізувати. Це дає користувачам свободу вибирати, що найкраще підходить для їхнього застосування [39].

Модель YOLO складається з конволюційних шарів, що витягують ознаки

зображення, та повнозв'язних шарів, що виконують класифікацію та регресію для визначення класів об'єктів та їх координат (див. рис. 3.6) [40].

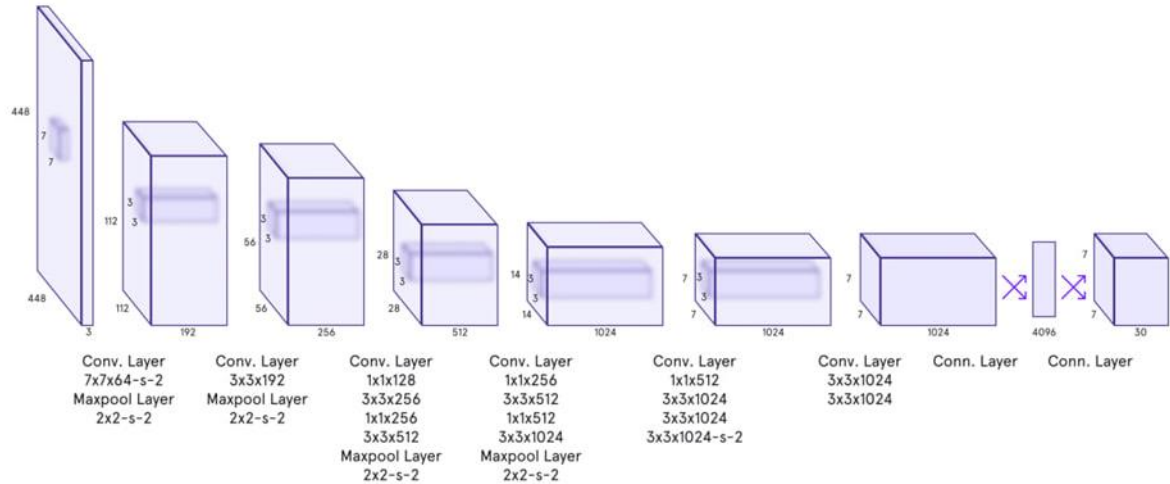


Рисунок 3.6 – Приклад архітектура моделі YOLO

Важливим елементом є функція втрат, яка вимірює розбіжність між передбаченнями моделі та реальними мітками. Вона складається з компонентів, що відповідають за втрати при локалізації (box loss), класифікації (class loss) та додаткових метрик для покращення точності рамок.

В кожній новій версії впроваджуються додаткові блоки в архітектурі або повністю змінюють її частину.

Наприклад, в YOLOv9 впроваджуються новаторські методи, такі як програмована градієнтна інформація (PGI) та узагальнена ефективна мережа агрегації шарів (GELAN), що забезпечує суттєві покращення в ефективності, точності та адаптивності.

Ці технології допомагають зберігати важливу інформацію через глибокі шари мережі, що сприяє надійному генеруванню градієнтів і покращенню збіжності та продуктивності моделі. У той же час, технологія GELAN оптимізує використання параметрів і обчислювальну ефективність, дозволяючи гнучко інтегрувати різні обчислювальні блоки [41].

YOLOv10 пропонує вже інші покращення в порівнянні з попередником, це досягається за рахунок зміни підходу для обробки результатів та оптимізації архітектури. Найбільшою зміною YOLOv10 є відмова від механізму Non-Maximum Suppression (NMS), який довгий час був стандартом у виявленні об'єктів. NMS допомагав усувати зайві прогнози, однак це вимагало додаткових обчислень, що створювало затримку. У YOLOv10 цей механізм замінено на підхід із використанням послідовних подвійних призначень під час навчання. Це не лише спрощує процес передбачень, але й значно знижує затримку, роблячи модель більш ефективною для застосувань у реальному часі [42].

Першою версією моделі YOLO для аналізу була обрана YOLOm11, так як по характеристикам, це є найпотужніша версія на даний момент.

Архітектура YOLOm11 розроблена для оптимізації як швидкості, так і точності, спираючись на досягнення, представлені в попередніх версіях YOLO, таких як YOLOv8, YOLOv9 і YOLOv10. Одним із головних нововведень YOLOv11 є впровадження модуля Cross-Stage Partial with Spatial Attention (C2PSA). Цей модуль поєднує переваги перехресного етапу з просторовою увагою, що дозволяє моделі краще фокусуватися на релевантних частинах зображення. Це особливо корисно при виявленні дрібних об'єктів або об'єктів, що частково перекриваються, підвищуючи загальну точність детекції [43].

Крім того, YOLOv11 оптимізує використання графічних процесорів під час навчання та роботи. Завдяки цієї оптимізації, модель демонструє зниження затримки до 25% порівняно з YOLOv10, що робить її більш придатною для застосувань, де критично важлива швидкість обробки.

Ще одним важливим покращенням є зменшення кількості параметрів моделі без втрати точності. Це досягається завдяки вдосконаленій архітектурі, яка ефективніше використовує обчислювальні ресурси, дозволяючи моделі працювати швидше та споживати менше пам'яті.

Порівняно з YOLOv9 та YOLOv10, YOLOv11 пропонує покращену точність, швидкість та ефективність, зберігаючи при цьому високу продуктивність.

Структура архітектури YOLOv11 наведена на рис. 3.7 [44].

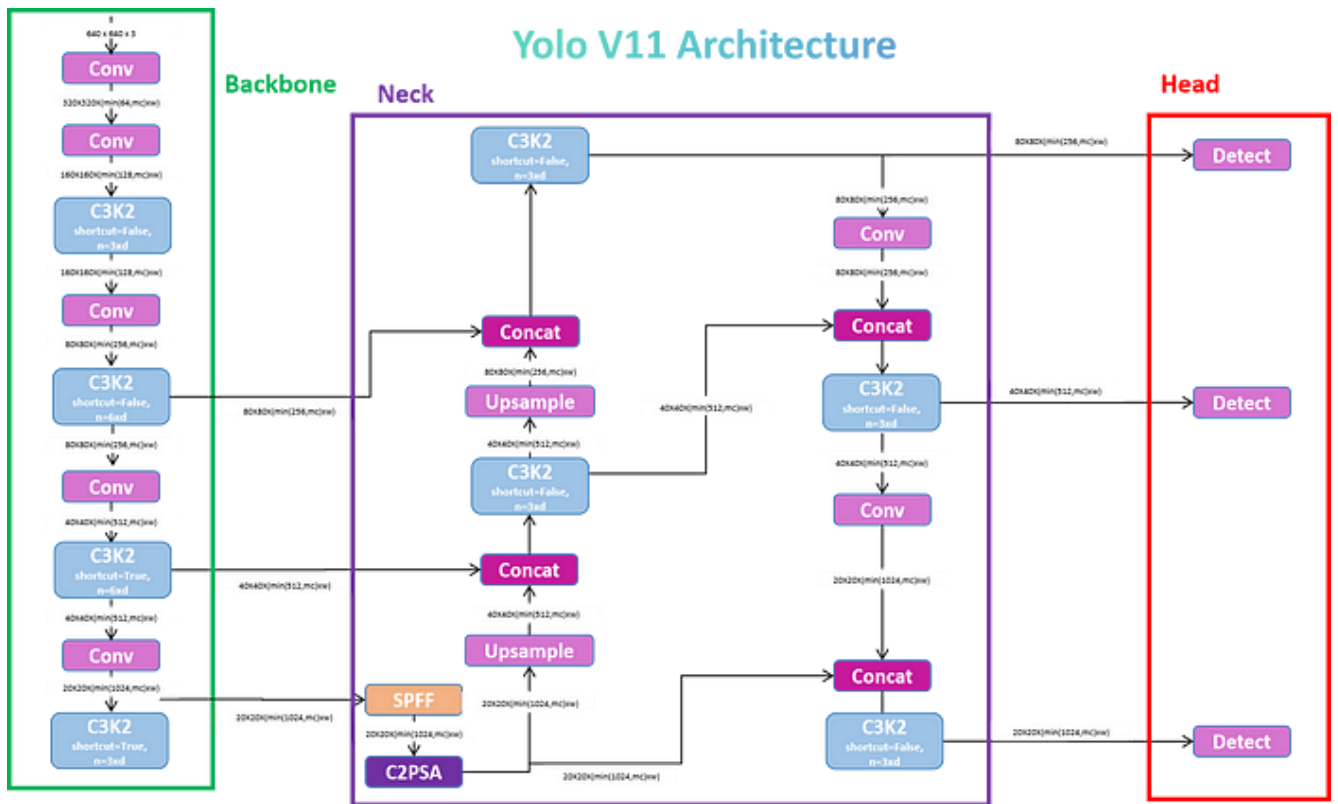


Рисунок 3.7 – Структура архітектури YOLOv11

Перший варіант (YOLO11_test), навчався на 70 епохах з використанням найшвидшої версії yolo11n, результат можна побачити на рис 3.8.

```

all      296      969      0.919      0.812      0.888      0.584

Epoch  GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
70/70   2.45G     0.9709   0.4845   0.9696   15         640: 100% 123/123 [00:36<00:00, 3.36it/s]
Class   Images  Instances  Box(P    R          mAP50    mAP50-95): 100% 10/10 [00:02<00:00, 3.87it/s]
all     296     969       0.913    0.813    0.889    0.586

70 epochs completed in 0.819 hours.
Optimizer stripped from runs/detect/train/weights/last.pt, 5.5MB
Optimizer stripped from runs/detect/train/weights/best.pt, 5.5MB

Validating runs/detect/train/weights/best.pt...
Ultralytics 8.3.36 Python-3.10.12 torch-2.5.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
YOLO11n summary (fused): 238 layers, 2,582,347 parameters, 0 gradients, 6.3 GFLOPs
Class   Images  Instances  Box(P    R          mAP50    mAP50-95): 100% 10/10 [00:04<00:00, 2.45it/s]
all     296     969       0.913    0.813    0.889    0.586
Speed: 0.3ms preprocess, 2.5ms inference, 0.0ms loss, 3.4ms postprocess per image
Results saved to runs/detect/train

```

Рисунок 3.8 – Результат навчання моделі YOLO11_test

Отримані результати базових метрик досить непогані, Точність (Precision)

дорівнює 0.913, Повнота (Recall) = 0.813. Це свідчить про те що модель добре індефікує та знаходить об'єкти на зображенні. Це підтверджує і mAP50, який дорівнює 88.9%.

Але mAP50-95 значно нижче, що означає зниження продуктивності при вищих порогах IoU.

Було вирішено покращити модель і для цього виконано наступні кроки:

- перехід від yolo11n до yolo11s збільшує ємність моделі, дозволяючи їй навчатися складнішим ознакам, що важливо для виявлення військових об'єктів;
- збільшення до 150 епох дає моделі більше можливостей для навчання, покращуючи її продуктивність;
- зменшення початкової швидкості навчання на перших епохах, щоб уникнути пропуску мінімумів функції втрат;
- додати функцію раннього зупинення, якщо протягом 10 епох результат не покращується, це дозволяє уникнути перенавчання.

Для порівняння ефективності різних версій YOLO, надалі будуть використовуватись саме такі налаштування для всіх моделей.

Навчання всіх моделей відбувалось за допомогою сервісу Google Colab, спочатку ми імпортуємо датасет за допомогою бібліотеки roboflow.

Після цього налаштовуємо модель використовуючи методи з пакету ultralytics та запускаємо навчання.

Процес навчання YOLOv11_a на 150 епохах можна побачити на рис. 3.9.

Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size		
15/150	4.38G	1.37	0.9037	1.168	24	640: 100% 123/123 [00:43<00:00, 2.84it/s]		
	Class	Images	Instances	Box (P	R	mAP50	mAP50-95): 100% 10/10 [00:04<00:00, 2.45it/s]	
	all	296	969	0.874	0.695	0.794	0.481	
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size		
16/150	4.35G	1.359	0.9045	1.166	30	640: 100% 123/123 [00:41<00:00, 2.95it/s]		
	Class	Images	Instances	Box (P	R	mAP50	mAP50-95): 100% 10/10 [00:03<00:00, 2.69it/s]	
	all	296	969	0.813	0.678	0.75	0.45	
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size		
17/150	4.35G	1.334	0.8715	1.144	22	640: 100% 123/123 [00:42<00:00, 2.91it/s]		
	Class	Images	Instances	Box (P	R	mAP50	mAP50-95): 100% 10/10 [00:02<00:00, 3.55it/s]	
	all	296	969	0.818	0.697	0.788	0.462	
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size		
18/150	4.33G	1.337	0.8679	1.156	21	640: 100% 123/123 [00:45<00:00, 2.69it/s]		
	Class	Images	Instances	Box (P	R	mAP50	mAP50-95): 100% 10/10 [00:02<00:00, 3.51it/s]	
	all	296	969	0.857	0.692	0.797	0.482	
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size		
19/150	4.37G	1.31	0.851	1.133	40	640: 100% 123/123 [00:43<00:00, 2.81it/s]		
	Class	Images	Instances	Box (P	R	mAP50	mAP50-95): 100% 10/10 [00:03<00:00, 3.33it/s]	
	all	296	969	0.868	0.694	0.802	0.476	
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size		
20/150	4.33G	1.293	0.838	1.128	27	640: 100% 123/123 [00:43<00:00, 2.85it/s]		
	Class	Images	Instances	Box (P	R	mAP50	mAP50-95): 100% 10/10 [00:03<00:00, 2.55it/s]	
	all	296	969	0.872	0.715	0.811	0.501	
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size		
21/150	4.38G	1.281	0.8021	1.127	24	640: 100% 123/123 [00:41<00:00, 2.99it/s]		
	Class	Images	Instances	Box (P	R	mAP50	mAP50-95): 100% 10/10 [00:04<00:00, 2.42it/s]	
	all	296	969	0.879	0.708	0.82	0.493	
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size		
22/150	4.37G	1.303	0.8348	1.142	53	640: 57% 70/123 [00:23<00:16, 3.30it/s]		

Рисунок 3.9 – Процес навчання YOLOv11_a на 150 епохах

Можна побачити, що система в реальному часі показує номер поточної епохи і загальна кількість епох тренування, обсяг пам'яті GPU, який використовує модель під час тренування, втрата для локалізації обмежувальних рамок (bounding boxes). Відображає, наскільки передбачені рамки відрізняються від істинних. На початку тренування значення зазвичай високе, але зменшується в міру навчання, втрата для класифікації об'єктів (cls_loss), метрика відображає, наскільки модель правильно класифікує знайдені об'єкти.

А також результати епохи, Precision (Точність), Recall (Повнота), mAP50 та mAP50-95. Ця інформація дозволяє слідкувати за процесом навчання, щоб у випадку перенавчання чи інших помилок розуміти в який момент це сталося.

Всі результати записані в порівняльну таблицю для визначення найефективнішої версії моделі на архітектурі YOLO.

Ці результати знадобляться нам для визначення найефективнішої версії YOLO та порівнянню з іншими архітектурами для пошуку оптимального рішення та його застосуванню при створенні системи.

Порівняння результатів навчених моделей різних версій та кількості епох сімейства YOLO наведені в табл. 3.2.

Таблиця 3.2 – Порівняння результатів навчених моделей YOLO

Модель	precision	recall	mAP50	mAP50-95	Час (хв)
YOLOv9_a	0.939	0.791	0.891	0.586	59
YOLOv10_a	0.945	0.855	0.904	0.611	132
YOLOv11_a	0.962	0.947	0.933	0,631	106

Модель «YOLOv9_a», навчалась протягом 59 хвилин, використовуючи 7167475 параметрів. Точність та повнота складають 93.9% та 79.1% відповідно, загальна середня точність mAP50 досягає 89.1%, хоча mAP50-95 склала усього 58.6%. При цьому, навчання тривало 60 епох так як було автоматично завершено через перенавчання.

Це означає, що модель ефективно виявляє об'єкти але загальна здатність виявлення та якість локалізації при високих порогах IoU потребує вдосконалення.

Модель «YOLOv10_a», навчалась протягом 132 хвилини, використовуючи 8035734 параметрів. Точність та повнота склали 94.5% та 85.5% відповідно, загальна середня точність mAP50 дорівнює 90.4%, а mAP50-95 досягла 61.1%.

Навчання завершилось достроково через перенавчання за 146 епох.

Остання модель «YOLOv11_a», пройшла всі 150 епох, навчання тривало 106 хвилин. Результати точності та повноти 96.2% та 85.5% відповідно. Середня точність моделі склала 93.3%, а локалізація об'єкта 63.1%.

Результати дослідження свідчать про те, що кожна нова версія моделі сімейства YOLO демонструє поступове покращення в якості та ефективності, що підтверджується зростанням основних метрик: precision, recall, середньої точності mAP50 та покращенням локалізації об'єктів за метрикою mAP50-95. Найгірші результати за всіма параметрами продемонструвала модель «YOLOv9_a», яка, хоча

й має прийнятну загальну точність 89.1% за mAP50, значно відстає від інших моделей у здатності якісно локалізувати об'єкти на високих порогах IoU, лише 58.6%. Час навчання цієї моделі був найменшим – всього 59 хвилин, але це не компенсує її продуктивні недоліки.

Модель «YOLOv10_a» демонструє значний прогрес у порівнянні з попередніми версіями. Її показники точності та повноти досягли 94.5% та 85.5% відповідно, а загальна середня точність зросла до 90.4%. Поліпшення також спостерігається в локалізації об'єктів, де mAP50-95 склала 61.1%. Проте, час навчання суттєво збільшився і становив 132 хвилини, що свідчить про велику кількість параметрів через вдосконалення архітектури моделі, хоча результати все ще залишаються нижчими за показники третьої версії.

«YOLOv11_a», продемонструвала найвищі результати в даному дослідженні. Її точність досягла 96.2%, а повнота – 94.7%, що є найкращими показниками серед усіх представлених моделей. Загальна середня точність mAP50 становить 93.3%, а локалізація об'єктів досягла 63.1%, що свідчить про її високу здатність виявляти та точно локалізувати об'єкти навіть при високих порогах IoU. Час навчання цієї моделі був помірним – 106 хвилин, що вказує на ефективність її архітектури.

Отже, результати дослідження підтверджують, що YOLOv11_a є найбільш оптимальною моделлю для завдань, що вимагають високої точності та якості локалізації. Хоча YOLOv9_a відрізняється найшвидшим часом навчання, її продуктивність не досягає рівня новіших моделей для задачі виявлення військових об'єктів.

Приклад виявлення об'єкта на відео за допомогою моделі «YOLOv11_a» наведений на рис. 3.10



Рисунок 3.10 – Виявлення військових об'єктів на відео за допомогою моделі «YOLOv11_a»

Для підтвердження ефективності, модель «YOLOv11_a» була протестована на відео. Було підтверджено, що модель добре виявляє один або декілька рухомих об'єктів на відео в реальному часі. Також є зауваження стосовно трекінгу об'єкта та його виявленню при перекритті іншим об'єктом, наприклад деревом чи парканом.

3.5 Навчання моделі Faster R-CNN для задачі виявлення військових об'єктів на зображенні

Faster R-CNN (Region-based Convolutional Neural Network), як зазначалось раніше – це потужна модель для об'єктного виявлення, яка одночасно визначає місцезнаходження об'єктів на зображенні (bounding boxes) і класифікує їх. Модель є вдосконаленням R-CNN і Fast R-CNN, але з використанням Region Proposal Network (RPN) для швидкої генерації регіонів інтересу (ROI) [45].

Для навчання моделі використовується датасет з анотованими зображеннями

в форматі coco.json.

Спочатку зображення проходить через базову мережу ResNet50, яка виконує згорткові операції, зменшує розмір зображення і витягує важливі ознаки та отримуються карти ознак. Далі використовується RPN, який працює над картою ознак і генерує регіони-кандидати, які, ймовірно, містять об'єкти. [46]

Структура архітектури моделі Faster R-CNN наведена на рис. 3.11 [47].

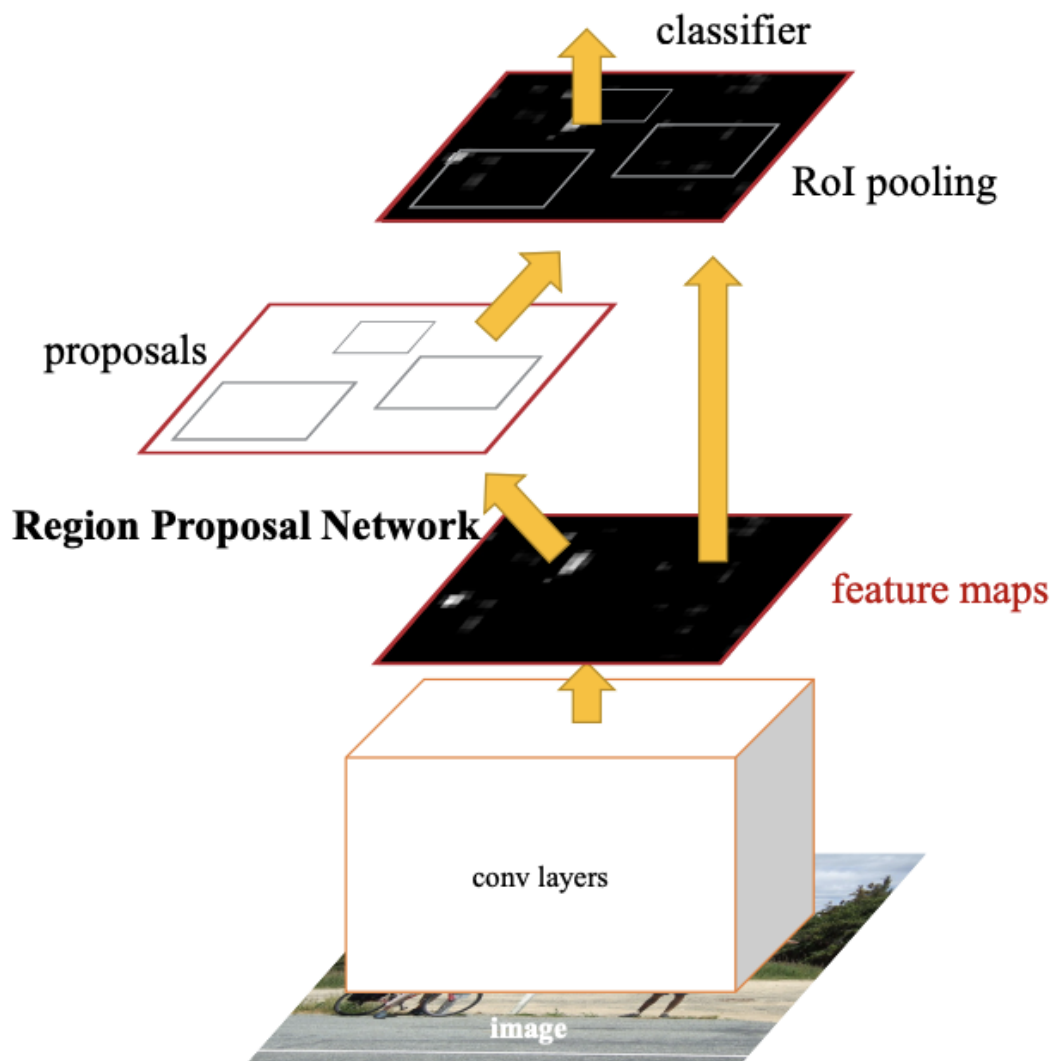


Рисунок 3.11 – Структура архітектури моделі Faster R-CNN

Також використовується регуляція через **weight decay**, щоб уникнути перенавчання. Процес навчання наведений на рис. 3.12.

Епоха 7, Ітерація 960, Втрата: 0.14320360124111176
Епоха 7, Ітерація 970, Втрата: 0.0873129814863205
Епоха 8, Ітерація 0, Втрата: 0.1714094877243042
Епоха 8, Ітерація 10, Втрата: 0.07734304666519165
Епоха 8, Ітерація 20, Втрата: 0.11398838460445404
Епоха 8, Ітерація 30, Втрата: 0.04746747389435768
Епоха 8, Ітерація 40, Втрата: 0.12891553342342377
Епоха 8, Ітерація 50, Втрата: 0.034896209836006165
Епоха 8, Ітерація 60, Втрата: 0.030464958399534225
Епоха 8, Ітерація 70, Втрата: 0.04032544419169426
Епоха 8, Ітерація 80, Втрата: 0.12450076639652252
Епоха 8, Ітерація 90, Втрата: 0.0879000872373581
Епоха 8, Ітерація 100, Втрата: 0.14078770577907562
Епоха 8, Ітерація 110, Втрата: 0.10222741961479187

Рисунок 3.12 – Процес навчання базової моделі Faster R-CNN

Модель Faster R-CNN навчалась протягом 10 епох по 800 ітерацій в кожній. Результатом є «base_faster_rcnn_model», mAP50 дорівнює 83.7% в той час як mAP50-95 дорівнює 59.4%, що є непоганими результатом.

Модель добре визначає регіон військового об'єкта на зображенні. Порівняння передбаченого регіону з реальним наведені на рис. 3.13.



Рисунок 3.13 – Порівняння реального та передбаченого регіону об'єкта в моделі

При перевірці моделі «faster_rcnn_model_a» на відео з дрону, для цього потрібно завантажити відео, розбити його на кадри та написати програму обробки кожного кадру для виявлення об'єктів. Також треба додати нанесення рамок та міток на кадр. Результат обробки відео наведений на рис 3.14.



Рисунок 3.14 – Виявлення військових об'єктів на відео за допомогою моделі «faster_rcnn_model_a»

На відео в реальному часі добре видно, що модель вміє виявляти декілька об'єктів навіть під час руху. Також модель здатна виявляти складні об'єкти, як автівка у будівлі.

3.6 Навчання моделі архітектури EfficientDet для задачі виявлення військових об'єктів на зображенні

На відміну від Faster R-CNN, яка використовує архітектуру FPN, EfficientDet працює за допомогою BiFPN (Bidirectional Feature Pyramid Network), що є покращеною версією FPN, яка дозволяє ефективніше агрегувати інформацію з різних рівнів мережі, забезпечуючи кращу передачу контекстної інформації. BiFPN обробляє ознаки, витягнуті з різних шарів, для покращення якісного об'єднання

інформації з різних масштабів [48].

Якщо більш детально розглядати архітектуру EfficientDet, то вона складається з трьох основних компонентів, таких як базової мережі (backbone), ViFPN та прогнозуючого блоку.

Базовою мережею виступає EfficientNet, яка є високоефективною конволюційною нейронною мережею. Її ключова особливість полягає в збалансованому масштабуванні параметрів глибини, ширини та роздільної здатності, що дозволяє досягти високої продуктивності навіть на обмежених обчислювальних ресурсах. Ця частина моделі ідентифікує ключові візуальні характеристики, такі як форма, текстура та колір, що дозволяє розрізняти, наприклад, танки, людину чи військові транспортні засоби.

Наступним етапом є обробка ознак за допомогою ViFPN, це двонаправлена мережа піраміди ознак. Вона використовує механізм зваженого об'єднання ознак, що дозволяє виділити найважливіші рівні для точного виявлення об'єктів, варто зазначити, що такий підхід ефективно обробляє об'єкти різного розміру, що покращує точність та виявлення.

Завершальним компонентом є прогнозуючий блок, який відповідає за ідентифікацію об'єктів та їх локалізацію на зображенні. Спочатку блок прогнозує координати обмежувальних рамок для об'єктів, а потім визначає їх категорії. Наприклад, для військових задач модель може класифікувати об'єкти як танки, солдат або артилерію, одночасно вказуючи їх точне положення на зображенні.

Завдяки високоточному поєднанню рівнів ознак з ViFPN, цей компонент забезпечує точне визначення навіть складних об'єктів у несприятливих умовах, таких як камуфляж чи низьке освітлення.

Унікальною особливістю EfficientDet є її здатність до масштабування. Модель може бути адаптована для різних вимог продуктивності за допомогою змін у її конфігурації, що включає зміну розміру базової мережі, кількості каналів у ViFPN та роздільної здатності вхідного зображення. Це дозволяє створювати моделі різної складності: від легких версій для швидкого виявлення до потужних

для обробки високоякісних зображень.

Це схоже на версії в YOLO та різний розмір моделей, як то YOLO11n або YOLO11s. Наприклад, для виявлення військових об'єктів за допомогою зображень з дронів, можна використовувати більші моделі, такі як EfficientDet-D4 чи D5.

Структура архітектури моделі EfficientDet наведена на рис. 3.15 [49].

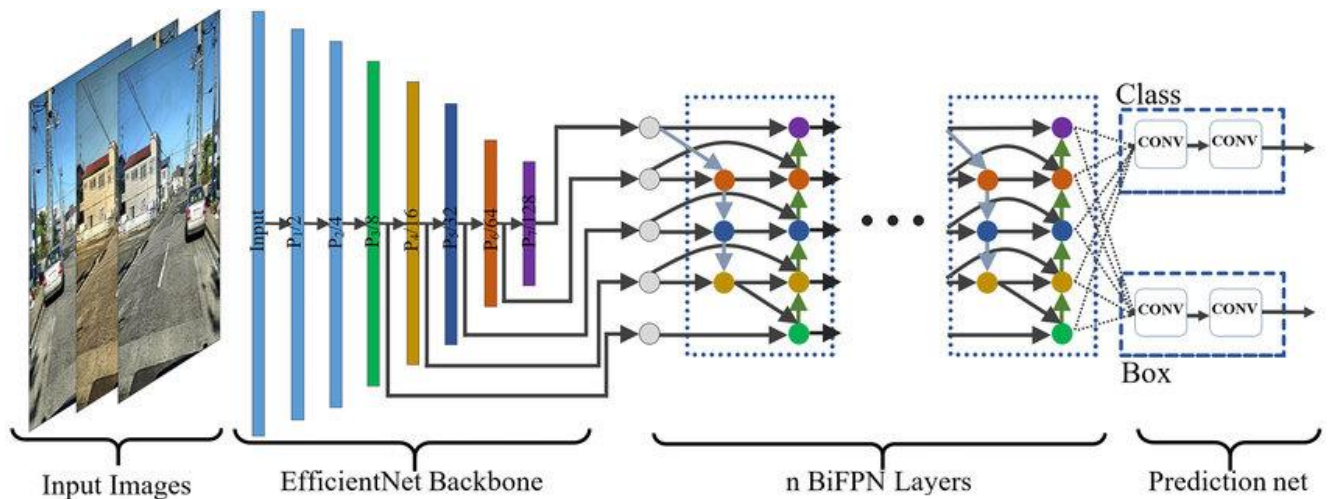


Рисунок 3.15 – Структура архітектури моделі EfficientDet

Для навчання використовується датасет в форматі coco.json та бібліотека tensorflow з keras.

Процес навчання моделі «EfficientDet_a» наведений на рис. 3.16.

```

Your `PyDataset` class should call `super().__init__(**kwargs)` in its constructor. `**kwargs` can include `workers`, `use
61/61 ----- 78s 339ms/step - accuracy: 0.2303 - loss: 1.4346 - val_accuracy: 0.0868 - val_loss: 0.1554
Epoch 2/80
61/61 ----- 44s 271ms/step - accuracy: 0.2481 - loss: 0.0716 - val_accuracy: 0.0799 - val_loss: 0.0902
Epoch 3/80
61/61 ----- 20s 263ms/step - accuracy: 0.2495 - loss: 0.0323 - val_accuracy: 0.1007 - val_loss: 0.0669
Epoch 4/80
61/61 ----- 18s 272ms/step - accuracy: 0.2449 - loss: 0.0214 - val_accuracy: 0.1667 - val_loss: 0.0495
Epoch 5/80
61/61 ----- 18s 270ms/step - accuracy: 0.2281 - loss: 0.0152 - val_accuracy: 0.2014 - val_loss: 0.0386
Epoch 6/80
61/61 ----- 21s 285ms/step - accuracy: 0.2607 - loss: 0.0118 - val_accuracy: 0.2396 - val_loss: 0.0297
Epoch 7/80
61/61 ----- 18s 272ms/step - accuracy: 0.2443 - loss: 0.0097 - val_accuracy: 0.2500 - val_loss: 0.0241
Epoch 8/80
61/61 ----- 18s 274ms/step - accuracy: 0.2495 - loss: 0.0075 - val_accuracy: 0.2847 - val_loss: 0.0193
Epoch 9/80

```

Рисунок 3.16 – Процес навчання базової моделі EfficientDet

Завдяки бібліотеці keras, ми можемо виводити під час навчання різні метрики, такі як mAP, MSE, recall та precision.

Мережа навчалась протягом 100 епох з додаванням 4х конволюаційних шарів. Навчання тривало 74 хвилини та результатом mAP 76.2%, рівень локалізації складає 44.9%.

Проаналізувавши результати та архітектуру EfficientDet, можна зазначити, що архітектура потребує точних налаштувань гіперпараметрів та шарів, це збільшує складність та час розробки але може значно підвищити точність, особливо якщо підібрати параметри для конкретної задачі чи датасету. Також варто зазначити, що рівень локалізації моделі набагато нижчий ніж у попередніх моделях, що говорить про те, що модель погано локалізує об'єкт.

3.7 Порівняльний аналіз ефективності моделей та архітектур нейронних мереж

У межах аналізу будуть враховуватись ключові характеристики моделей, такі як швидкість, точність, гнучкість та складність реалізації. Порівняння навчених моделей наведено в табл. 3.3.

Таблиця 3.3 – Порівняння результатів навчених моделей

Модель	mAP50	mAP50-95	Швидкість (хв)	Складність
EfficientDet_a	0.762	0.449	Низька	Висока
faster_rcnn_model_a	0.904	0.611	Низька	Середня
YOLOv11_a	0.933	0.631	Висока	Низька

Аналізуючи навчені моделі, можна побачити, що найшвидша та найточніша моделлю є модель сімейства YOLO.

«EfficientDet_a» демонструє найнижчу точність серед розглянутих моделей, отримуючи 0.762 за mAP50 та 0.449 за mAP50-95. Це свідчить про обмежену

ефективність у задачах із високими вимогами до розпізнавання дрібних деталей або об'єктів у складних умовах. Її швидкість оцінюється як низька, що робить її менш придатною для використання в реальному часі. Водночас модель має високу складність реалізації, через складне налаштування гіперпараметрів, що ускладнює її інтеграцію в практичні системи з обмеженими ресурсами.

Модель «faster_rcnn_model_a» показує значно вищу точність – 0.904 за mAP50 та 0.611 за mAP50-95. Це робить її придатною для задач, де точність є критичним параметром, наприклад, для розпізнавання військових об'єктів із високим ступенем деталізації. Також варто зауважити, що модель значно краще за інші впорюється з виявленням дрібних деталей, наприклад як частина автівки у будівлі. Проте швидкість обробки також низька, що обмежує застосування моделі в умовах реального часу. Складність моделі оцінюється як середня, що може бути прийнятним компромісом між продуктивністю та зусиллями, необхідними для реалізації.

«YOLOv11_a» демонструє найвищу продуктивність із середньою точністю 0.933 за mAP50 та рівнем локалізації 0.631 за mAP50-95. Крім того, її швидкість оцінюється як висока, що робить цю модель найкращим вибором для задач реального часу. Низька складність моделі сприяє її впровадженню у системи з обмеженими обчислювальними ресурсами.

Таким чином, YOLOv11_a є найоптимальнішим вибором для задач виявлення військових об'єктів, що потребують швидкої обробки та високої точності, особливо в умовах реального часу. Faster R-CNN Model_a підходить для задач, де пріоритетом є висока точність та здатність виявлення дрібних об'єктів, навіть за рахунок швидкості. EfficientDet_a є найменш ефективною та через складність налаштування не є оптимальною для розробки системи.

Майбутня структура системи виявлення військових об'єктів на динамічному зображенні буде будуватись на основі цього аналізу.

Гарним варіантом, є побудова гібридної моделі з YOLOv11 та Faster R-CNN, де YOLOv11 буде відповідати за загальне, швидке виявлення об'єкту та

локалізацію, а Faster R-CNN за виявлення дрібних об'єктів. Об'єднання YOLO та Faster R-CNN дозволяє використовувати переваги обох моделей, швидкість YOLO та точність Faster R-CNN. Оптимальний вибір методу залежить від обчислювальних ресурсів та потреб задачі.

Висновки до розділу 3

У третьому розділі роботи було розглянуто ключові аспекти навчання моделей на основі сучасних архітектур нейронних мереж для задач виявлення військових об'єктів. Головна мета дослідження полягала у визначення ефективної архітектури для розробки системи, здатної швидко працювати зі складними даними на динамічному зображення. Архітектури, такі як YOLO, Faster R-CNN та EfficientDet, обрано за їхню високу продуктивність та здатність працювати в умовах складного оточення, включаючи низьку видимість, шуми та погодні умови. Процес навчання моделей включав підготовку та анотацію набору даних, що є критично важливим для досягнення високої точності системи. Датасет складається зі зображень, які охоплюють різні класи об'єктів, такі як танки, бронетехніка, солдати та інше, з урахуванням варіативності умов освітлення та кутів огляду. Анотація забезпечила точну ідентифікацію об'єктів, що значно підвищило якість навчання моделей.

Важливим етапом роботи стало порівняння ефективності моделей у контексті точності та швидкості роботи. Архітектура YOLO продемонструвала найкращу продуктивність у реальному часі, тоді як Faster R-CNN показала високу точність для виявлення дрібних об'єктів, але потребувала більше обчислювальних ресурсів. Для підвищення ефективності запропоновано гібридний підхід, де YOLO відповідає за швидке виявлення великих об'єктів, а Faster R-CNN уточнює виявлення складних та дрібних деталей.

Метрики, використані для оцінки ефективності моделей, включали середню точність та швидкість обробки. Це дозволило об'єктивно оцінити продуктивність розроблених рішень і визначити оптимальні налаштування для конкретних умов

застосування, таких як військові операції, що вимагають швидкого аналізу та надійних результатів.

Результати аналізу потрібні для подальшої розробки системи автоматизованого виявлення військових об'єктів, що включає створення гібридної архітектури нейромережі, інтеграцію додаткових методів обробки даних та оптимізацію параметрів для використання у складних умовах.

4 РЕАЛІЗАЦІЯ СИСТЕМА ВІЯВЛЕННЯ ВІЙСЬКОВИХ ОБ'ЄКТІВ НА ОСНОВІ ГІБРИДНОГО ПІДХОДУ

4.1 Навчання моделей YOLO_D та Faster_RCNN_D

Базуючись на результатах попереднього розділу, вирішено створити систему використовуючи дві найефективніші для цієї задачі архітектури нейронних мереж YOLO та Faster R-CNN. Навчені моделі будуть названі з позначенням D (detection).

Процес навчання моделей «YOLO_D» та «Faster_RCNN_D» базується на ретельному підборі гіперпараметрів, які суттєво впливають на точність і швидкість роботи алгоритмів. Основними задачами цього етапу є оптимізація конфігурації моделей для забезпечення ефективного виявлення військових об'єктів з мінімальними похибками. Тобто, потрібно покращити результат базових моделей, щоб на основі покращених створити гібридну систему.

За основу моделі «YOLO_D» взята найефективніша модель з порівняльного аналізу, а саме YOLOv11.

Модель «YOLO_D» налаштовувалася з урахуванням особливостей навчальної вибірки, розміру об'єктів і обчислювальних ресурсів. Було визначено такі ключові гіперпараметри: learning rate, яка встановлена на рівні 0.001 для забезпечення поступової конвергенції моделі, а також використана стратегія warm-up, де спочатку починаємо з маленької швидкості навчання поступово повертаємось до стандартної 0.01.

Розмір батчу (batch size), встановлено на рівні 32, що дозволяє зберігати баланс між використанням пам'яті і стабільністю навчання, так як навчання моделі відбувалось на платформі Roboflow з обмеженою кількістю ресурсів.

Кількість епох навчання, при підборі оптимальної варіювалася в межах від 75 до 200 в залежності від результатів проміжної валідації.

Важливим аспектом також є використання функції втрат, адаптованої для задачі виявлення об'єктів, яка враховує похибки в локалізації і класифікації. Для цієї задачі було обрано оптимізатор функції втрат SGD (Stochastic Gradient Descent

Loss), яка мінімізує функцію втрат, оновлюючи параметри моделі на основі обчислення градієнтів на невеликих випадкових батчів. Цей підхід забезпечує ефективне використання обчислювальних ресурсів та дозволяє уникнути локальних мінімумів у функції втрат. Процес навчання моделі наведений на рис. 4.1.

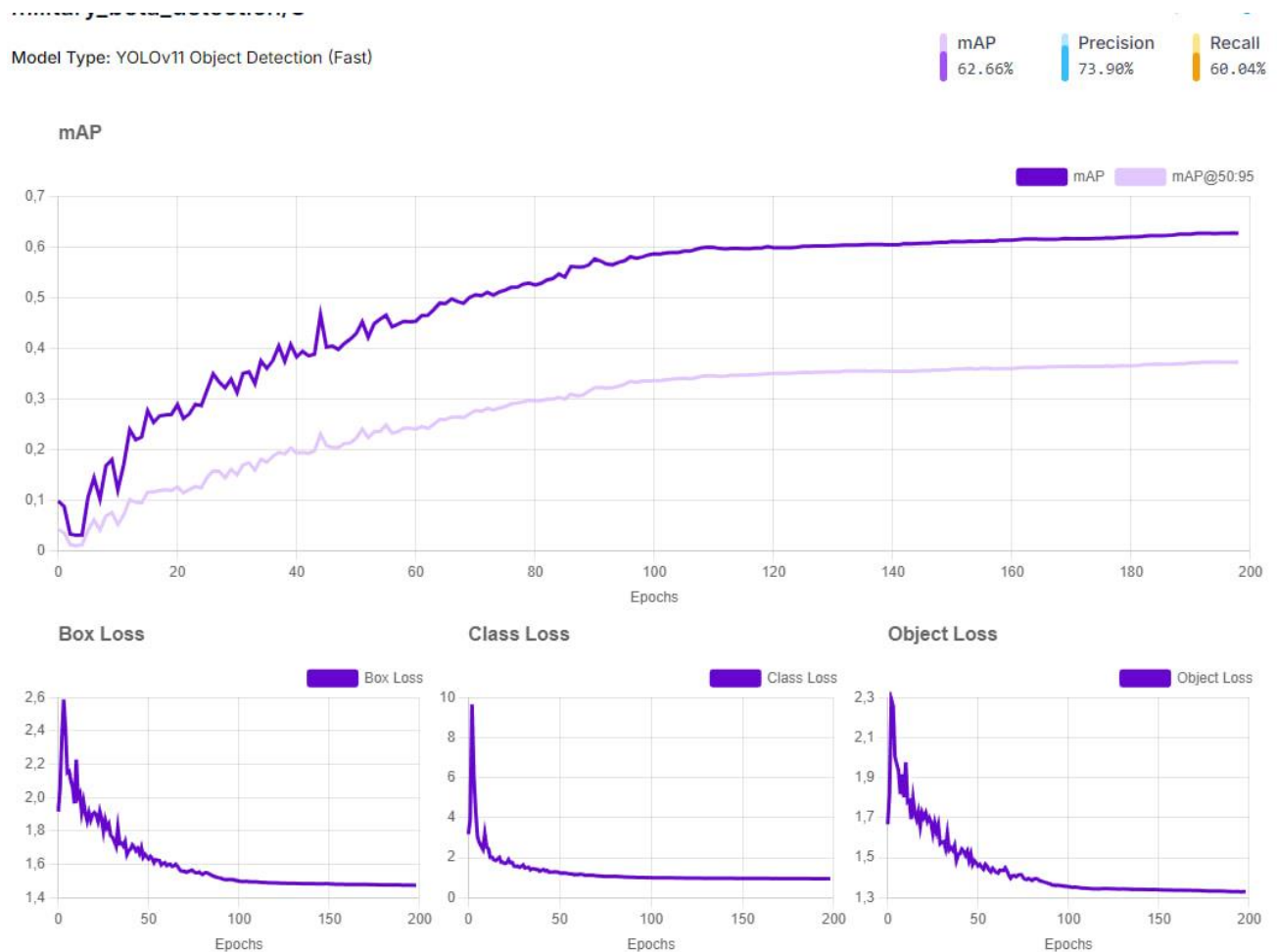


Рисунок 4.1 – Процес навчання моделі YOLO_D

Порівнюючи алгоритми оптимізації Adam та SGD, варто зазначити, що SGD є базовим алгоритмом оптимізації, який оновлює ваги моделі на основі градієнтів, обчислених для випадкових батчів. Це робить його простим у реалізації та ефективним для навчання великих моделей. Однак його поведінка залежить від правильного вибору швидкості навчання, яка часто потребує ручного

налаштування, що і було зроблено.

Для завдань виявлення військових об'єктів SGD добре працює за умови використання методів, таких як затухання чи зростання швидкості навчання. Цей підхід забезпечує більш стабільне навчання.

Adam є більш складним алгоритмом, який використовує адаптивну швидкість навчання для кожного параметра, враховуючи моменти першого та другого порядків (градієнти та їхні квадрати). Це дозволяє швидше сходиться до мінімуму навіть у складних задачах, таких як виявлення об'єктів у динамічних умовах, коли якість зображень може бути неоднорідною через рух дронів, зміну освітлення або погані погодні умови.

Вибір SGD зумовлений тим, що кількість обчислювальних ресурсів обмежено, а розмір моделі не є критичним, SGD може забезпечити кращу узагальнюючу здатність.

Матриця невідповідностей для навченої моделі «YOLO_D» представлена на рис. 4.2.

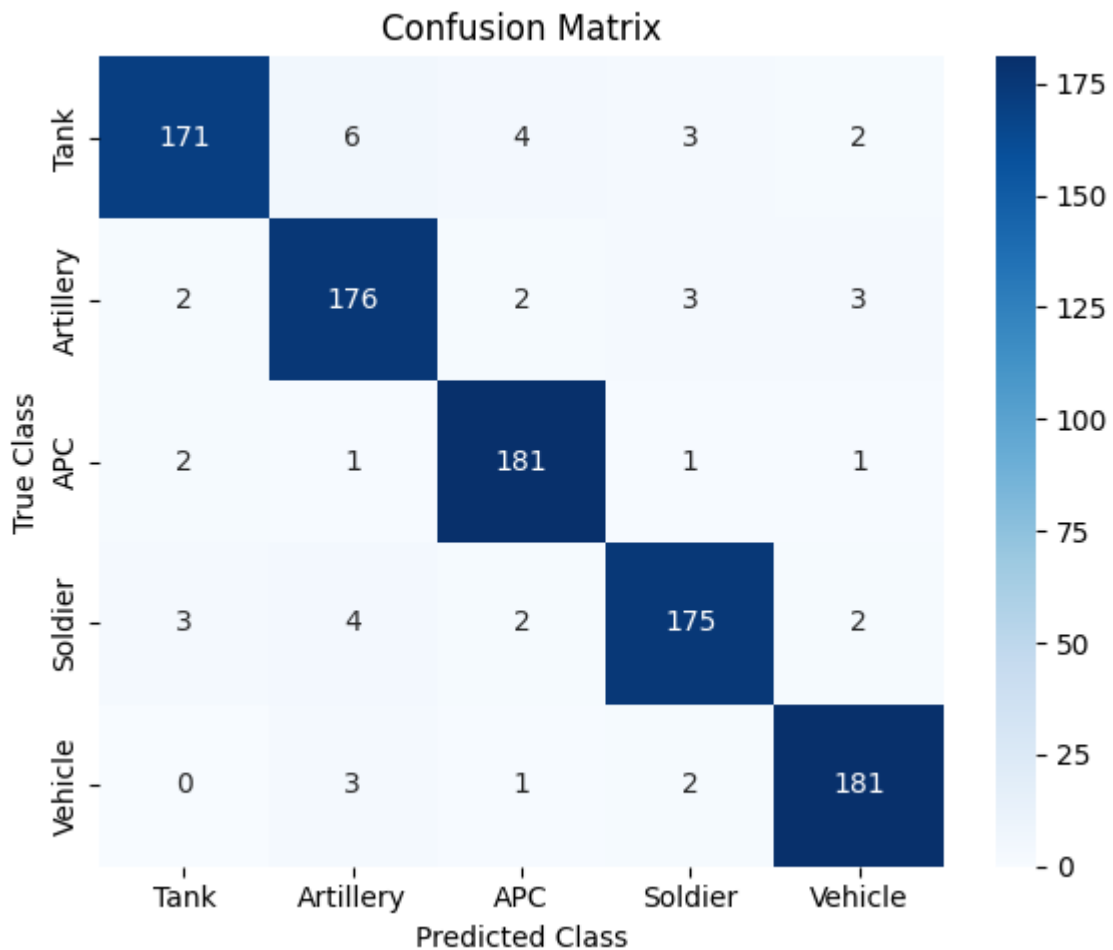


Рисунок 4.2 – Матриця невідповідностей для моделі «YOLO_D»

Модель «Faster_RCNN_D» використовує інший підхід, що базується на двоетапному виявленні об'єктів. Перший етап відповідає за генерацію регіонів-пропозицій, а другий – за класифікацію та уточнення меж об'єктів. У процесі навчання цієї моделі ключовими гіперпараметрами були: learning rate, який початково встановлено на рівні 0.001 з поступовим зниженням після кожних 10 епох. Налаштовано кількість регіонів-пропозицій, яка обмежувалася 300 для підвищення швидкодії моделі. В якості оптимізатора використовувався Adam для забезпечення стабільної мінімізації функції втрат

Навчання моделі проводилося на основі попередньо навчених ваг з минулої моделі, результат якої був ~90%, що дозволило суттєво скоротити час тренування та покращити точність за рахунок використання знань, отриманих на великих

наборах даних. Навчання відбувалось впродовж 35 епох, однак починаючи з 25 епохи було помітно перенавчання. Матриця невідповідностей моделі «Faster_RCNN_D» наведена на рис. 4.3.

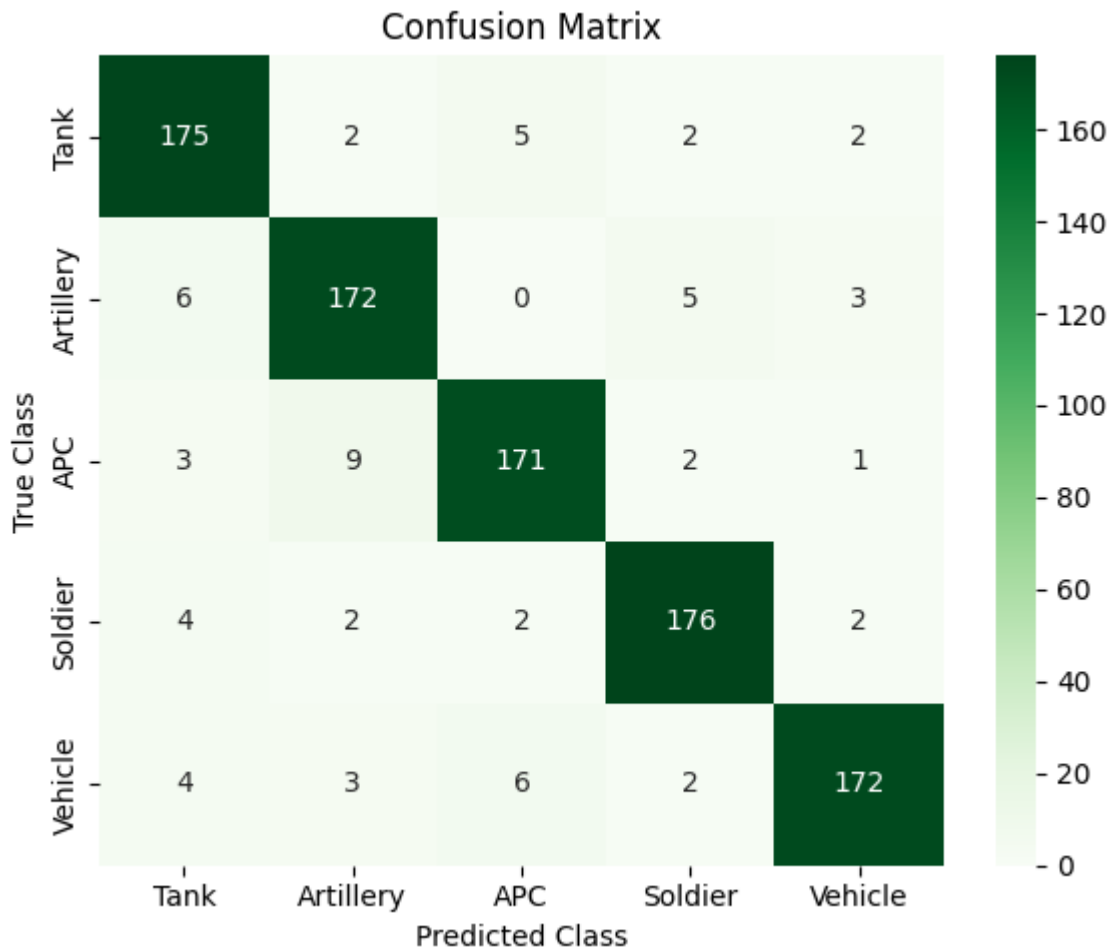


Рисунок 4.3 – Матриця невідповідностей для моделі «Faster_RCNN_D»

Результатом покращення є покращення загальної точності моделей (mAP50), який становить 94.47% для YOLO_D та 92.12% для Faster R-CNN.

4.2 Створення системи на основі моделей YOLO та Faster R-CNN

Принцип побудови гібридної системи для виявлення військових об'єктів з динамічних зображень є двоетапна обробка зображення, де на першому етапі система використовує модель YOLO_D, яка виконує швидке виявлення об'єктів на

зображеннях. На вхід подається зображення або кадр відео з дрону. Модель розділяє зображення на сітку й для кожної клітинки прогнозує ймовірність присутності об'єкта, його клас та координати меж об'єкта (bounding boxes).

Завдяки одноетапній архітектурі, модель «YOLO_D» обробляє зображення цілісно, що забезпечує високу швидкість роботи навіть на обмежених обчислювальних ресурсах. Цей етап спрямований на ідентифікацію всіх потенційних об'єктів, які можуть бути військовими цілями. Проте, через швидкість роботи, результати YOLO_D можуть бути менш точними у виявленні дрібних або складних об'єктів, таких як невеликі військові установки або деталі техніки.

На другому етапі застосовується модель Faster R-CNN для уточнення та детальнішої класифікації об'єктів, визначених на першому етапі. Faster R-CNN отримує виділені регіони з результатів YOLO_D і працює з ними для створення високоточних прогнозів. Ця модель використовує механізм RPN, який уточнює потенційні регіони, а також виконує детальну класифікацію кожного об'єкта.

Система працює послідовно на основі виконанні цих двох етапів.

Важливо, що результати обох етапів об'єднуються, формуючи фінальний висновок системи. У кінцевому результаті система формує точний список виявлених об'єктів із координатами, класами та ймовірностями, а також візуалізує результати на вихідному зображенні.

Структура гібридної системи виявлення військових об'єктів на кадрах з дрону наведена на рис. 4.4.

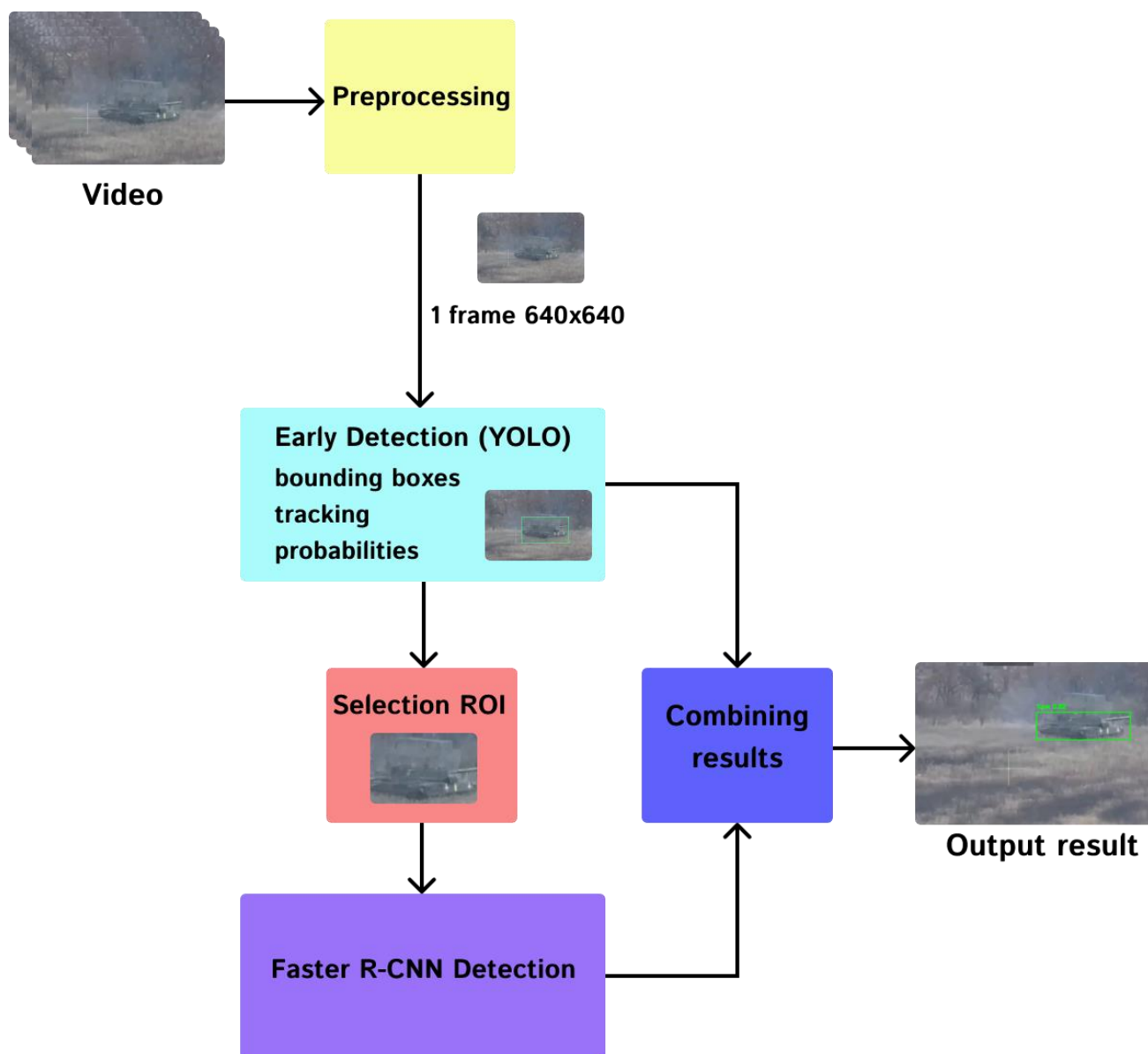


Рисунок 4.4 – Структура гібридної системи виявлення військових об'єктів на основі архітектур нейронних мереж

Синергетичне використання моделей YOLO_D та Faster_RCNN_D у рамках гібридної архітектури забезпечило високий рівень ефективності виявлення військових об'єктів, що було підтверджено під час тестування на реальних зображеннях. Модель YOLO_D показала високу швидкість і придатність для виявлення великої кількості об'єктів, тоді як Faster_RCNN_D забезпечила підвищену точність для складних сцен.

Це дозволило правильно передбачити більше об'єктів на тестовій вибірці,

Кафедра інтелектуальних інформаційних систем
Система виявлення військових об'єктів на динамічних зображеннях на основі архітектур нейронних мереж

mAP50 для системи дорівнює ~96%, що вище за показники моделей окремо.

Система добре показує себе в складних сценах, наприклад в виявленні об'єктів при шумі чи на швидкому зображенні (як FPV), на рис. 4.5 можна побачити вдале виявлення об'єкта танк, який до того є нетипової форми.



Рисунок 4.5 – Виявлення танка за допомогою зображення з FPV

Система також здатна виявляти декілька об'єктів в реальному часі і при задимленому кадрі (рис. 4.6).



Рисунок 4.6 – Виявлення декільком об'єктів

Також варто відзначити, що завдяки обробці деталей за допомогою моделі Faster_RCNN_D, можливо виявлення дрібних цілей, таких як піхота. Для тестування було застосовано камеру ноутбуку та ввімкнена відео з інтернету. Результат можна побачити на рис. 4.7.



Рисунок 4.7 – Виявлення піхотинця у реальному часі з камери

Зображення не є чітким так як відбувається рух рук та самого кадру, але все ж таки система добро впоралась з трекінгом та виявленням об'єкта, що вказує на її ефективність.

4.3 Аналіз результатів системи

Аналізуючи результати системи, створеної на основі моделей YOLO та Faster R-CNN, можна свідчити про високий рівень її ефективності у виявленні військових об'єктів у динамічних умовах. Система функціонує за принципом гібридного підходу, де кожна модель виконує специфічну роль для досягнення балансу між швидкістю і точністю.

У сукупності, ці дві моделі створюють систему, яка досягає високої середньої точності виявлення (mAP50), що перевищує 96%. Така точність зумовлена тим, що система в якій 2 моделі, здатна доповнювати пропуски об'єктів однієї моделі

компенсує їх правильними результатами іншої. Тобто там, де YOLO не зміг точно виявити об'єкт, Faster R-CNN доповнив його результат виявивши його.

Така точність особливо помітна в умовах складних сцен, де військові об'єкти мають варіативні форми, можуть бути частково закриті або перебувати на зашумлених зображеннях. Крім того, система здатна обробляти дані в реальному часі, одночасно виявляючи декілька об'єктів, що критично важливо для військових операцій. У тестових сценаріях вона демонструє стабільну продуктивність навіть у динамічних умовах, таких як обмежена видимість через задимленість або рух.

Система окрім переваг, також має недоліки, а саме:

- погане виявлення дрібних об'єктів на задимлених кадрах або біля більш великих цілей (наприклад піхота біля техніки);
- невелика кількість класі, через що, унікальні об'єкти, такі як ППО виявляються по найбільш схожому класу;
- відсутність інтеграції з платформою Discord, адже саме туди здійснюється трансляція з дронів на полі бою.

Система потребує доопрацювання та налаштування, потрібно додати більше збалансованих зображень до набору даних, це дозволить системі рівномірно навчатися на кожному класі і більш точно виконувати свої задачі.

Вдосконалення виявлення дрібних об'єктів, можна досягти додаванням в систему спеціалізованих модулів, таких як механізми уваги, що може дозволити моделі більше зосереджуватися на дрібних деталях. Одним з можливих методів також є сегментація областей. Після отримання координат об'єктів, виявлених YOLO, можна провести сегментацію цих областей для виділення дрібних деталей, таких як піхота або додаткові елементи військової техніки. Це дозволить зосередити обчислювальні ресурси лише на важливих регіонах зображення.

Додавання додаткових систем комбінування результатів, наприклад комбінування ймовірностей або використання мета моделі.

Інтеграція системи з платформою Dicord, дозволить використовувати її в реальному часі аналізуючи трансляції з дронів.

Загалом, результати підтверджують, що гібридна система, побудована на основі YOLO та Faster R-CNN, є потужним і ефективним інструментом для виявлення військових об'єктів на динамічному зображенні.

Висновки до розділу 4

У цьому розділі, було навчено моделі «YOLO_D» та «Faster_R_CNN_D» та підкреслено ефективність застосування гібридної системи на основі моделей нейронних мереж для виявлення військових об'єктів на динамічних зображеннях.

Навчання моделей «YOLO_D» та «Faster_R_CNN_D» включало ретельний підбір гіперпараметрів, що дозволило досягти високої точності при збереженні швидкодії. Основна увага приділялась оптимізації функцій втрат і стратегії навчання, це забезпечило адаптацію моделей до специфіки задачі. Комбінування цих моделей забезпечило баланс між швидкістю та точністю в виявленні військових об'єктів та їх класифікації. YOLO_D швидко ідентифікує потенційні об'єкти, тоді як Faster R-CNN уточнює їхні характеристики, покращуючи результат.

Загальна середня точність (mAP50) гібридної системи ~96%, що свідчить про її спроможність точно виявляти об'єкти навіть у складних умовах, таких як рух, зашумлення або слабе освітлення. Система демонструє стабільну продуктивність у реальному часі, здатна виявляти об'єкти на динамічних кадрах, включаючи дрібні об'єкти, хоча для цього потрібне подальше вдосконалення.

Окрім переваг, було виявлено проблеми з розпізнаванням дрібних об'єктів, інтеграцією з платформами, такими як Discord, а також недостатня кількість навчальних даних для специфічних класів. Додавання механізмів уваги або сегментації може суттєво покращити результати. Пропонується подальша оптимізація, включаючи розширення наборів даних та вдосконалення алгоритмів, дозволить підвищити адаптивність і універсальність системи, що особливо важливо для її використання в польових умовах.

ВИСНОВКИ

Результатом виконання кваліфікаційної роботи стало створення ефективної системи виявлення військових об'єктів на динамічних зображеннях, яка базується на гібридному підході з використанням сучасних архітектур нейронних мереж, таких як YOLO та Faster R-CNN. Система демонструє високу точність 96% у складних реальних умовах та є придатною для роботи в реальному часі.

Під час проведення дослідження та виконання роботи, було виконано наступні дії:

- проаналізовано попередні результати досліджень та зібрані теоретичні матеріали;
- розроблено новий, унікальний набір даних із зображеннями військових об'єктів, отриманих із дронів, з чіткою анотацією для покращення навчання моделей;
- проведено порівняльний аналіз сучасних архітектур нейронних мереж із визначенням їхніх переваг та недоліків у контексті розпізнавання військових об'єктів;
- обрано найефективніші архітектури нейромереж для поставленої задачі;
- оптимізовано гіперпараметри моделей Faster R-CNN та YOLO, що дозволило підвищити їхню ефективність у задачах з виявленням складних об'єктів при низькій видимості та високим рівнем шуму;
- спроектовано гібридну систему, яка поєднує найкращі властивості архітектур Faster R-CNN та YOLO для досягнення балансу між точністю і швидкістю обробки;
- проаналізовано результати роботи системи, виявлено її позитивні та негативні риси, надано рекомендації, щодо покращення та використання системи.

Розроблена система повністю відповідає цілям і завданням, поставленим у вступі до роботи. Вона продемонструвала високий рівень автоматизації процесу

розпізнавання військових об'єктів, що значно покращує ефективність збору інформації для військових операцій.

Систему доцільно інтегрувати у вже існуючі платформи комунікації між операторами дронів та штабом прийняття рішень для підвищення їхньої функціональності.

Під час експлуатації системи слід враховувати її адаптацію до специфічних військових умов, зокрема через регулярне оновлення набору даних та впровадження нових методів комбінування результатів.

Подальші дослідження можуть бути спрямовані на покращення результатів окремих моделей за рахунок зміни датасету та методів навчання, наприклад використовуючи під час навчання ваги однієї моделі для навчання іншої.

Таким чином, під час дослідження та виконання кваліфікаційної роботи, було виконано не лише всі поставлені завдання, але й продемонструвала перспективність використання штучного інтелекту у вирішенні складних задач обробки динамічних зображень у реальному часі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Survey of review spam detection using machine learning techniques / M. Crawford et al. *Journal of Big Data*. 2015. Vol. 2, no. 1. DOI: <https://doi.org/10.1186/s40537-015-0029-9> (date of access: 10.10.2024).
2. Content-Based Video Recommendation System Based on Stylistic Visual Features / Y. Deldjoo et al. *Journal on Data Semantics*. 2016. Vol. 5, no. 2. P. 99–113. DOI: <https://doi.org/10.1007/s13740-016-0060-9> (date of access: 10.10.2024).
3. Li S., Deng W. Deep Facial Expression Recognition: A Survey. *IEEE Transactions on Affective Computing*. 2020. P. 1. URL: <https://doi.org/10.1109/taffc.2020.2981446> (date of access: 14.12.2023).
4. Vyas A. S., Prajapati H. B., Dabhi V. K. Survey on Face Expression Recognition using CNN. 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), Coimbatore, India, 15–16 March 2019. 2019. URL: <https://doi.org/10.1109/icaccs.2019.8728330> (date of access: 10.10.2024).
5. Military Target Detection Method Based on Improved YOLOv5 / X. Du et al. 2022 International Conference on Cyber-Physical Social Intelligence (ICCSI), Nanjing, China, 18–21 November 2022. 2022. URL: <https://doi.org/10.1109/iccsi55536.2022.9970675> (date of access: 10.10.2024).
6. Parcell K., Choi B., Starling J. Optimizing Performance of Real-time Detection and Classification for Military Personnel and Weapons. *The ITEA Journal of Test and Evaluation*. 2024. Vol. 45, no. 3. URL: <https://doi.org/10.61278/itea.45.3.1005> (date of access: 10.10.2024).
7. MobileNetV2: Inverted Residuals and Linear Bottlenecks / M. Sandler et al. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, 18–23 June 2018. 2018. URL: <https://doi.org/10.1109/cvpr.2018.00474> (date of access: 10.10.2024).

8. Ding P., Qian H., Chu S. SlimYOLOv4: lightweight object detector based on YOLOv4. *Journal of Real-Time Image Processing*. 2022. URL: <https://doi.org/10.1007/s11554-022-01201-7> (date of access: 10.10.2024).

9. Computer Vision-Based Military Tank Recognition Using Object Detection Technique: An application of the YOLO Framework / S. Ali et al. 2023 1st International Conference on Advanced Innovations in Smart Cities (ICAISC), Jeddah, Saudi Arabia, 23–25 January 2023. 2023. URL: <https://doi.org/10.1109/icaisc56366.2023.10085552> (date of access: 10.10.2024).

10. Computer Vision Based Drone Detection Using Mask R-CNN / A. S. Mubarak et al. 2022 International Conference on Artificial Intelligence in Everything (AIE), Lefkosa, Cyprus, 2–4 August 2022. 2022. URL: <https://doi.org/10.1109/aie57029.2022.00108> (date of access: 10.10.2024).

11. Ba Alawi A. E., Mohammed H. M. A. The Role of YOLOv8 in Enhancing Strategic Military Equipment Detection. 2024 4th International Conference on Emerging Smart Technologies and Applications (eSmarTA), Sana'a, Yemen, 6–7 August 2024. 2024. P. 1–5. URL: <https://doi.org/10.1109/esmarta62850.2024.10638856> (date of access: 10.10.2024).

12. A Military Object Detection Model of UAV Reconnaissance Image and Feature Visualization / H. Liu et al. *Applied Sciences*. 2022. Vol. 12, no. 23. P. 12236. URL: <https://doi.org/10.3390/app122312236> (date of access: 15.10.2024).

13. Anusha N., Swapna T. Deep Learning-Based Aerial Object Detection for Unmanned Aerial Vehicles. 2023 International Conference on Sustainable Communication Networks and Application (ICSCNA), Theni, India, 15–17 November 2023. 2023. URL: <https://doi.org/10.1109/icscna58489.2023.10370618> (date of access: 15.10.2024).

14. Papers with Code - UAVDT Dataset. The latest in Machine Learning | Papers With Code. URL: <https://paperswithcode.com/dataset/uavdt> (date of access: 15.10.2024).

15. Dousai N. M. K., Lonearic S. Detecting Humans in Search and Rescue Operations Based on Ensemble Learning. IEEE Access. 2022. Vol. 10. P. 26481–26492. URL: <https://doi.org/10.1109/access.2022.3156903> (date of access: 15.10.2024).

16. Object Detection Robot Using Fuzzy Logic Controller Through Image Processing / M. Khairudin et al. Journal of Physics: Conference Series. 2021. Vol. 1737, no. 1. P. 012045. URL: <https://doi.org/10.1088/1742-6596/1737/1/012045> (date of access: 15.10.2024).

17. Enhancing Remote Sensing Object Detection with a Hybrid Densenet-LSTM Model / V. L. Narayana et al. 2024 IEEE International Conference on Computing, Power and Communication Technologies (IC2PCT), Greater Noida, India, 9–10 February 2024. 2024. URL: <https://doi.org/10.1109/ic2pct60090.2024.10486394> (date of access: 15.10.2024).

18. Military Unmanned Equipment Image Target Recognition Method based on Improved Deep Learning / L. Gao et al. Journal of Physics: Conference Series. 2024. Vol. 2732, no. 1. P. 012004. URL: <https://doi.org/10.1088/1742-6596/2732/1/012004> (date of access: 15.10.2024).

19. Buhl N. YOLO Object Detection Explained: Evolution, Algorithm, and Applications. Encord. URL: <https://encord.com/blog/yolo-object-detection-guide/> (date of access: 15.10.2024).

20. Kowalczyk A. Comparing YOLOv10, YOLOv9, and YOLOv8: A Performance Study. Medium. URL: <https://medium.com/@az.tayyebi/comparing-yolov10-yolov9-and-yolov8-a-performance-study-41e3072fa0e8> (date of access: 19.10.2024).

21. Dharmaraj. Convolutional Neural Networks (CNN)–Architectures Explained. Medium. URL: <https://medium.com/@draj0718/convolutional-neural-networks-cnn-architectures-explained-716fb197b243> (date of access: 19.10.2024).

22. Haque K. N. What is Convolutional Neural Network – CNN (Deep Learning). LinkedIn: Log In or Sign Up. URL: <https://www.linkedin.com/pulse/what->

convolutional-neural-network-cnn-deep-learning-nafiz-shahriar (date of access: 19.10.2024).

23. Nelson J. What is YOLO? The Ultimate Guide [2024]. Roboflow Blog. URL: <https://blog.roboflow.com/guide-to-yolo-models/> (date of access: 19.10.2024).

24. Wang C.-Y., Liao H.-Y. M. YOLOv1 to YOLOv10: The Fastest and Most Accurate Real-time Object Detection Systems. APSIPA Transactions on Signal and Information Processing. 2024. Vol. 13, no. 1. URL: <https://doi.org/10.1561/116.20240058> (date of access: 19.10.2024).

25. Potrimba P. What is YOLOv10? An Architecture Deep Dive. Roboflow Blog. URL: <https://blog.roboflow.com/what-is-yolov10/> (date of access: 19.10.2024).

26. Skelton J. Faster R-CNN Explained for Object Detection Tasks | DigitalOcean. DigitalOcean | Cloud Infrastructure for Developers. URL: <https://www.digitalocean.com/community/tutorials/faster-r-cnn-explained-object-detection> (date of access: 19.10.2024).

27. Object detection in remote sensing images based on deep transfer learning / J. Chen et al. Multimedia Tools and Applications. 2021. URL: <https://doi.org/10.1007/s11042-021-10833-z> (date of access: 20.10.2024).

28. Tan M., Pang R., Le Q. V. EfficientDet: Scalable and Efficient Object Detection. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020. 2020. URL: <https://doi.org/10.1109/cvpr42600.2020.01079> (date of access: 20.10.2024).

29. GdPicture.NET Table Extraction Series - Part 2: Deep Learning Approaches. GdPicture.NET Imaging SDKs. URL: <https://www.gdpicture.com/blog/table-extraction-deep-learning/#EfficientDet> (date of access: 20.10.2024).

30. Najafi P., Feizizadeh B., Navid H. A Comparative Approach of Fuzzy Object Based Image Analysis and Machine Learning Techniques Which Are Applied to Crop Residue Cover Mapping by Using Sentinel-2 Satellite and UAV Imagery. Remote Sensing. 2021. Vol. 13, no. 5. P. 937. URL: <https://doi.org/10.3390/rs13050937> (date of access: 20.10.2024).

31. Wahi A., Mohamed Athiq F., Palanisamy C. A hybrid feature extraction method-based object recognition by neural network. 2008 International Conference on Computing, Communication and Networking (ICCCN), Karur, Tamil Nadu, India, 18–20 December 2008. 2008. URL: <https://doi.org/10.1109/icccnet.2008.4787728> (date of access: 20.10.2024).
32. Hybrid Neural Network Architecture for Multi-Label Object Recognition using Feature Fusion / S. D. Meena et al. *Procedia Computer Science*. 2022. Vol. 215. P. 78–90. URL: <https://doi.org/10.1016/j.procs.2022.12.009> (date of access: 20.10.2024).
33. Skalski P. How to Train YOLOv8 Object Detection on a Custom Dataset. *Roboflow Blog*. URL: <https://blog.roboflow.com/how-to-train-yolov8-on-a-custom-dataset/> (date of access: 01.11.2024).
34. Nelson J. You Might Be Resizing Your Images Incorrectly. *Roboflow Blog*. URL: <https://blog.roboflow.com/you-might-be-resizing-your-images-incorrectly/> (date of access: 01.11.2024).
35. Nelson J. Introducing Bounding Box Level Augmentations. *Roboflow Blog*. URL: <https://blog.roboflow.com/introducing-bounding-box-level-augmentations/> (date of access: 01.11.2024).
36. Understanding the Confusion Matrix in Machine Learning - GeeksforGeeks. *GeeksforGeeks*. URL: <https://www.geeksforgeeks.org/confusion-matrix-machine-learning/> (date of access: 05.11.2024).
37. Solawetz J. What is Mean Average Precision (mAP) in Object Detection?. *Roboflow Blog*. URL: <https://blog.roboflow.com/mean-average-precision/> (date of access: 06.11.2024).
38. Hui J. mAP (mean Average Precision) for Object Detection. *Medium*. URL: <https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173> (date of access: 06.11.2024).
39. Mastering All YOLO Models from YOLOv1 to YOLOv9: Papers Explained (2024). *LearnOpenCV – Learn OpenCV, PyTorch, Keras, Tensorflow with code, &*

tutorials. URL: <https://learnopencv.com/mastering-all-yolo-models/#aioseo-yolov9> (date of access: 06.11.2024).

40. Parti A. Introduction to YOLO Object Detection: Understanding the Basics. Pareto: Premium AI & LLM Training Data Labeled by Elite Teams. URL: <https://pareto.ai/blog/yolo-object-detection> (date of access: 06.11.2024).

41. Ultralytics. YOLOv9. Home - Ultralytics YOLO Docs. URL: <https://docs.ultralytics.com/models/yolov9/> (date of access: 15.11.2024).

42. Ultralytics. YOLOv10. Home - Ultralytics YOLO Docs. URL: <https://docs.ultralytics.com/models/yolov10/#overview> (date of access: 15.11.2024).

43. Ultralytics. YOLO11 🚀 NEW. Home - Ultralytics YOLO Docs. URL: <https://docs.ultralytics.com/models/yolo11/#how-do-i-train-a-yolo11-model-for-object-detection> (date of access: 15.11.2024).

44. Rao S. N. YOLOv11 Explained: Next-Level Object Detection with Enhanced Speed and Accuracy. Medium. URL: <https://medium.com/@nikhil-rao-20/yolov11-explained-next-level-object-detection-with-enhanced-speed-and-accuracy-2dbe2d376f71> (date of access: 15.11.2024).

45. Boesch G. The Fundamental Guide to Faster R-CNN [2025] - viso.ai. viso.ai. URL: <https://viso.ai/deep-learning/faster-r-cnn-2/> (date of access: 01.12.2024).

46. GeeksforGeeks. Faster R-CNN | ML - GeeksforGeeks. GeeksforGeeks. URL: <https://www.geeksforgeeks.org/faster-r-cnn-ml/> (date of access: 01.12.2024).

47. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks / S. Ren et al. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2017. Vol. 39, no. 6. P. 1137–1149. URL: <https://doi.org/10.1109/tpami.2016.2577031> (date of access: 01.12.2024).

48. Sarkar A. Understanding EfficientNet—The most powerful CNN architecture. Medium. URL: <https://arjun-sarkar786.medium.com/understanding-efficientnet-the-most-powerful-cnn-architecture-eaeb40386fad> (date of access: 05.12.2024).

49. An Efficient and Scalable Deep Learning Approach for Road Damage Detection / S. Naddaf-Sh et al. 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 10–13 December 2020. 2020. URL: <https://doi.org/10.1109/bigdata50022.2020.9377751> (date of access: 05.12.2024).

ДОДАТОК А

Апробація роботи

Робота пройшла апробацію під час Всеукраїнської науково-практичної конференції молодих вчених, аспірантів та студентів, 2 – 4 грудня 2024 р. у м. Миколаєві.

Тема конференції: «МАШИННЕ НАВЧАННЯ ТА ШТУЧНИЙ ІНТЕЛЕКТ»

Посилання: <https://meet.google.com/dqk-qgte-bqp>.

Голова секції: д.т.н., проф. Гожий О.П.

Секретар секції: Сомряков Б.

Міністерство освіти і науки України
Чорноморський національний
університет ім. Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних
систем

УДК 004.8

Архіпов Д. В., Сіденко Є. В.
Чорноморський національний університет
ім. Петра Могили,
Миколаїв, Україна



Інформаційний лист

*Всеукраїнська науково-
практична конференція
молодих вчених, аспірантів і
студентів*

Інтелектуальні інформаційні системи

2 – 4 грудня 2024 року

Миколаїв

СИСТЕМА ВИЯВЛЕННЯ ВІЙСЬКОВИХ ОБ'ЄКТІВ НА ДИНАМІЧНИХ ЗОБРАЖЕННЯХ НА ОСНОВІ АРХІТЕКТУР НЕЙРОННИХ МЕРЕЖ

Сучасні військові конфлікти вимагають ефективних рішень для автоматичного аналізу зображень, отриманих з дронів, для оперативного виявлення та класифікації військових об'єктів. Дана робота пропонує систему для виявлення військових об'єктів на зображеннях з дронів на основі різних архітектур нейронних мереж, таких як YOLO (You Only Look Once) та Faster R-CNN. [1]

У ході роботи було виконано порівняння моделей нейронних мереж для задачі розпізнавання військових об'єктів з зображення дрону, включаючи точність та швидкість. Основною метрикою визначення точності є mAP50 з урахуванням результатів mAP50-95. [2]

Для побудови моделі було створено власний набір даних на основі анотованих зображень з дронів, що містять різні класи військових об'єктів (танки, артилерійські установки, ДОТи тощо). Зображення були взяті з соціальних мереж, відкритих джерел в інтернеті та відео різних підрозділів ЗСУ.

Обробка даних відбувалась за допомогою сервісу Roboflow та інструментів анотування. Вихідні, анотовані, кольорові зображення були зменшені до розміру 640 на 640 пікселів. Навчання моделей відбувалось на сервісі Google colab з графічним ядром «T4 GPU», обсягом відеопам'яті 15360 МБ.

Модель була реалізована на мові програмування Python із використанням бібліотек NumPy, Pandas, scikit-learn, Keras, PyTorch та ultralytics.

Модель Faster R-CNN, є потужною архітектурою для об'єктного виявлення, яка одночасно визначає місцезнаходження об'єктів на зображенні і класифікує їх.