

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»  
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ  
ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ  
КАФЕДРА ШТУЧНОГО ІНТЕЛЕКТУ

На правах рукопису  
УДК 004.852

До захисту допущено  
Завідувач кафедри ШІ

Олена ЧУМАЧЕНКО

«\_\_\_» \_\_\_\_\_ 2022 р.

## Магістерська дисертація

на здобуття ступеня магістра за спеціальністю 122 «Комп'ютерні науки»  
на тему: «Розпізнавання стратегічних технічних об'єктів за допомогою  
згорткових нейронних мереж»

Виконав:

студент 2 курсу, групи КІ-11мп  
Іванов Сергій Ігорович \_\_\_\_\_

Керівник: доцент кафедри ММСА,  
к.ф.-м.н., доц. Шубенкова І. А. \_\_\_\_\_

Рецензент: професор кафедри ММСА,  
д.т.н., проф. Зайченко О. Ю. \_\_\_\_\_

Засвідчую, що у цій магістерській дисертації  
немає запозичень з праць інших авторів  
без відповідних посилань

Студент: \_\_\_\_\_

Київ  
2022

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»  
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ  
ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ  
КАФЕДРА ШТУЧНОГО ІНТЕЛЕКТУ

Рівень вищої освіти — другий (магістерський)  
Спеціальність (ОПП) — 122 «Комп'ютерні науки» («Системи і методи штучного інтелекту»)

ЗАТВЕРДЖУЮ  
Завідувач кафедри ШІ  
Олена ЧУМАЧЕНКО  
«\_\_» \_\_\_\_\_ 2022 р.

**ЗАВДАННЯ**

на магістерську дисертацію студенту Іванову Сергію Ігоровичу

**1. Тема дисертації:** «Розпізнавання стратегічних технічних об'єктів за допомогою згорткових нейронних мереж», науковий керівник дисертації Шубенкова Ірина Анатоліївна, к.ф.-м.н., доцент, затверджені наказом по університету від «03» листопада 2022 р. № 4046-с.

**2. Термін подання студентом дисертації:** 12.12.2022 р.

**3. Об'єкт дослідження:** Методи та моделі розпізнавання об'єктів.

**4. Предмет дослідження:** Системи розпізнавання стратегічних технічних об'єктів на основі глибоких згорткових нейронних мереж.

**5. Перелік завдань, які потрібно зробити:**

- 1) здійснити огляд літератури за темою роботи;
- 2) дослідити актуальність обраної теми;
- 3) ознайомитись із існуючими методами та моделями розпізнавання об'єктів;
- 4) здійснити порівняльний аналіз існуючих методів;

- 5) розробити та реалізувати систему, що вирішує задачу розпізнавання стратегічних технічних об'єктів з використанням апарату згорткових нейронних мереж;
- 6) провести експеримент, що засвідчує працеспроможність запропонованої моделі, виконати аналіз результатів;
- 7) провести аналіз ринкових можливостей запуску стартап проекту;
- 8) зробити висновки;
- 9) підготувати ілюстративний матеріал;
- 10) оформити пояснювальну записку.

## 6. Перелік ілюстративного матеріалу.

7. Дата видачі завдання: 1 вересня 2022 року.

### Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1.	Вивчення літератури за темою роботи.	02.09.2022 – 14.09.2022	Виконано
2.	Підготовка першого розділу.	15.09.2022 – 29.09.2022	Виконано
3.	Підготовка другого розділу.	30.09.2022 – 18.10.2022	Виконано
4.	Розробка програмного продукту.	20.10.2022 – 05.11.2022	Виконано
5.	Підготовка третього розділу	06.11.2022 – 18.11.2022	Виконано
6.	Підготовка частини стартап-проекту	19.11.2022 – 22.11.2022	Виконано
7.	Концептуальні висновки. Перспективи розвитку отриманих рішень	23.11.2022 – 25.11.2022	Виконано
8.	Оформлення пояснювальної записки	26.11.2022 – 03.12.2022	Виконано

Студент

Сергій Іванов

Науковий керівник дисертації

Ірина ШУБЕНКОВА

## РЕФЕРАТ

Дипломна робота: 119 с., 42 рис., 1 додаток, 43 джерел

НЕЙРОННІ МЕРЕЖІ, ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ,  
КОМП'ЮТЕРНИЙ ЗІР, РОЗПІЗНАВАННЯ ОБРАЗІВ, ВИЯВЛЕННЯ  
ОБ'ЄКТІВ, МАШИННЕ НАВЧАННЯ, ГЛИБОКЕ НАВЧАННЯ,  
КЛАСИФІКАЦІЯ

Мета роботи – використати апарат згорткових штучних нейронних мереж для розпізнавання стратегічних технічних об'єктів.

Об'єктом дослідження є методи та моделі розпізнавання об'єктів.

Предметом досліджень є системи розпізнавання стратегічних технічних об'єктів на основі глибоких згорткових нейронних мереж.

Актуальність даного методу впливає із стратегічної важливості швидкого та точного розпізнавання, обробки та аналізу візуальної інформації із камер дронів, наземної техніки та об'єктів та інших джерел. Така інформація може бути життєво важливою як у військовій, так і у цивільній сферах, наприклад для журналістики, своєчасного попередження мирних громадян про небезпеку та їх захисту.

У даній роботі запропонована модель для розпізнавання стратегічних технічних об'єктів на основі глибоких згорткових нейронних мереж, орієнтована на використання на БПЛА різних класів та за умов значно обмежених обчислювальних ресурсів.

## ABSTRACT

Thesis: 119 p., 42 fig., 1 appendice, 43 sources

NEURAL NETWORKS, CONVOLUTIONAL NEURAL NETWORKS, COMPUTER VISION, PATTERN RECOGNITION, OBJECT DETECTION, MACHINE LEARNING, DEEP LEARNING, CLASSIFICATION

The purpose of this work is to use the apparatus of convolutional artificial neural networks for recognition of strategic technical objects.

The object of research is methods and models of object recognition.

The subject of research is recognition systems of strategic technical objects based on deep convolutional neural networks.

The relevance of this method stems from the strategic importance of rapid and accurate recognition, processing and analysis of visual information from drone cameras, ground vehicles and objects and other sources. Such information can be vital in both the military and civilian areas, such as journalism, the timely warning of civilians of danger and their protection.

In this thesis, a model for recognizing strategic technical objects based on deep convolutional neural networks is proposed, aimed at use on UAVs of various classes and under conditions of significantly limited computing resources.

## ЗМІСТ

ВСТУП .....	9
1 ЗАГАЛЬНІ ВІДОМОСТІ ПРО ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ ТА ПОПУЛЯРНІ АРХІТЕКТУРИ .....	10
1.1 Загальні відомості про штучні нейронні мережі .....	10
1.2 Процес навчання .....	11
1.3 Метод зворотнього розповсюдження помилки .....	12
1.4 Проблема перенавчання .....	13
1.5 Архітектури нейронних мереж .....	14
1.5.1 Нейронні мережі прямого поширення .....	14
1.5.2 Рекурентні нейронні мережі .....	16
1.5.3 Згорткові нейронні мережі .....	17
1.5.4 Автокодувальники .....	19
1.6 Архітектури глибоких згорткових нейронних мереж .....	20
1.6.1 Агрегувальні шари, ваги та повнозв'язні шари .....	20
1.6.2 LeNet та AlexNet .....	22
1.6.3 VGG-16 та VGG-19 .....	23
1.6.4 ResNet .....	24
1.6.5 R-CNN .....	27
1.6.6 You Only Look Once .....	30
1.6.7 Single Shot Detector .....	32
1.6.8 Feature Pyramid Network .....	33
1.7 Постановка задачі .....	35
1.8 Висновки до розділу 1 .....	35
2 ЗАСТОСУВАННЯ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ ДО ЗАДАЧ РОЗПІЗНАВАННЯ СТРАТЕГІЧНИХ ТЕХНІЧНИХ ОБ'ЄКТІВ .....	36
2.1 Етичні питання розпізнавання стратегічних технічних об'єктів та можливі застосування .....	36
2.2 Основні напрями застосування апарату розпізнавання образів .....	37

2.3 Особливості застосування систем розпізнавання образів на дронах та наземних об'єктах.....	37
2.4 Ускладнюючі фактори та специфіка застосування апарату комп'ютерного зору для розпізнавання стратегічних технічних об'єктів ..	41
2.4.1 Вхідні дані для навчання .....	41
2.4.2 Первинна обробка даних та складності розпізнавання .....	45
2.4.3 Маскування .....	47
2.4.4 Технічні обмеження .....	51
2.4.5 Довіра та безпека .....	52
2.5 Огляд деяких відомих цивільних та спеціалізованих рішень .....	53
2.6 Висновки до розділу 2 .....	55
<b>3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ ТА АНАЛІЗ РЕЗУЛЬТАТІВ</b>	<b>56</b>
3.1 Обрані вхідні дані для навчання .....	56
3.2 Вибір платформ та програмних засобів для реалізації моделі .....	58
3.3 Метрики порівняння моделей .....	60
3.4 Порівняльний аналіз моделей .....	62
3.5 Побудована модель та результати експериментів .....	63
3.5.1 Зниження шуму .....	63
3.5.2 Квантування .....	65
3.5.3 Архітектура моделі .....	66
3.5.4 Результати експериментів на власному наборі даних .....	72
3.6 Аналіз результатів експериментів .....	74
3.7 Висновки до розділу 3 .....	78
<b>4 РОЗРОБКА СТАРТАП ПРОЄКТУ</b> .....	<b>79</b>
4.1 План розробки стартапу .....	80
4.2 Опис ідеї стартап-проєкту.....	81
4.3 Технологічний аудит ідеї проєкту .....	83
4.4 Аналіз ринкових можливостей запуску стартап-проєкту .....	85
4.5 Розроблення ринкової стратегії проєкту .....	93
4.6 Розроблення маркетингової програми стартап-проєкту.....	96

	8
4.7 Висновки до розділу 4.....	101
ВИСНОВКИ.....	102
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ .....	103
ДОДАТОК А. ЛІСТИНГ ПРОГРАМИ .....	108



## ВСТУП

Сучасні конфлікти все більше перетворюються на інформаційні. Більша кількість та якість стратегічно-важливої інформації є ваговою складовою перемоги на полі бою. Інформація також корисна й для порятунку людських життів та збереження інфраструктури.

Своєчасне отримання та обробка даних стратегічної значущості є вирішальним фактором при веденні розвідки, прийнятті тактичних та стартагічних рішень, визначенні загроз та цілей.

Одним із головних джерел такої інформації в сучасних конфліктах часто стають безпілотні літальні апарати. Гнучкість та широкі можливості їх застосування дозволяють отримувати достовірні дані безпосередньо з лінії фронту та навіть за нею. Окрім інформації, вони також здатні уражати цілі та виявляти постраждалих.

Значно збільшити ефективність застосування БПЛА, безпеку людей та наземної техніки здатні допоміжні системи обробки візуальних даних та розпізнавання об'єктів. Вони можуть значно збільшити швидкість та точність обробки інформації та прийняття рішень виконавцями або ж повністю їх замінити.

При застосуванні у військовій та супутніх галузях, системи розпізнавання образів стикаються зі значною кількістю додаткових складностей, від питань безпеки та вимог надійності до доступності та якості вхідних даних, вимог до швидкості та надзвичайно складних умов роботи.

Таким чином, існує потреба в системі, що була б придатна та готова до використання за складних реальних бойових умов, мала б достатню точність та могла обробляти дані в реальному часі для допомоги виконавцям у своєчасному прийнятті рішень.

# 1 ЗАГАЛЬНІ ВІДОМОСТІ ПРО ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ ТА ПОПУЛЯРНІ АРХІТЕКТУРИ

## 1.1 Загальні відомості про штучні нейронні мережі

Штучні нейронні мережі являють собою обчислювальні системи, подібні до відповідних систем людського мозку. Вони складаються із штучних нейронів, що приймають вхідні сигнали від певної кількості зв'язків із нейронами попереднього шару або входом мережі. Такі зв'язки є аналогами аксонів в біологічних нейронах та мають на входах ваги, що визначають їх силу. Всередині нейрона інформація обробляється через суматор зв'язків та певну функцію активації і передається до виходу нейрону.

Будова штучного нейрона представлена на рисунку 1.1.

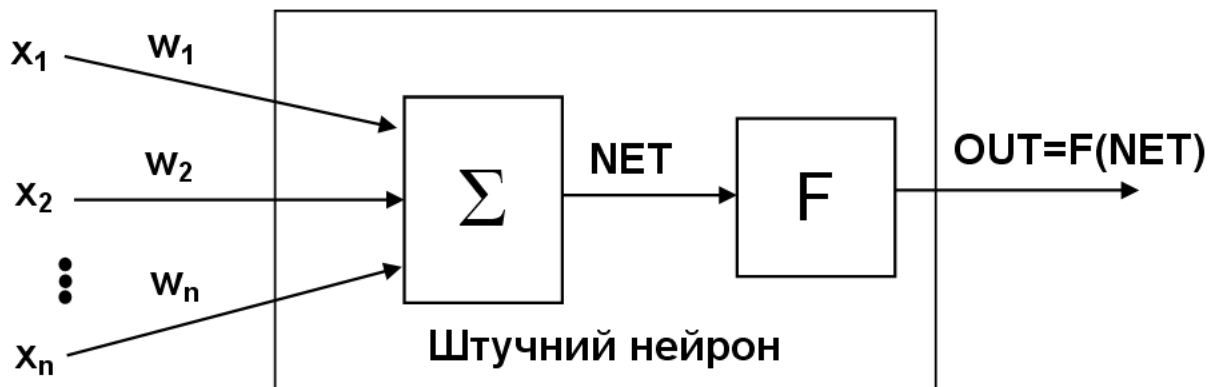


Рисунок 1.1 – Будова штучного нейрона

Такі штучні нейрони в мережі об'єднують у шари за виконуваною функцією. Традиційно, в мережі є вхідний шар, певна кількість внутрішніх шарів, що зветься прихованими шарами, та вихідний шар. Вхідний сигнал від вхідного шару передається через приховані й далі на вихідний шар мережі.

Схему простої штучної нейронної мережі наведено на рисунку 1.2.

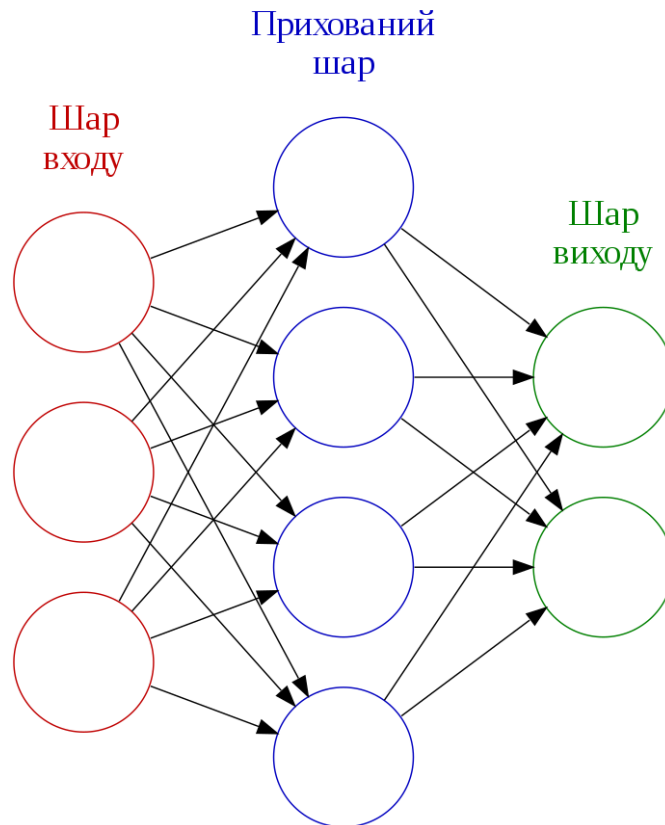


Рисунок 1.2 – Схема простої нейронної мережі із одним прихованим шаром

## 1.2 Процес навчання

Процес навчання нейронної мережі полягає у корегуванні параметрів мережі для отримання кращих результатів. Для більшості задач в нейронних мережах використовують навчання з учителем, коли системі подаються дані із готовими результуючими складовими для поступової підготовки мережі до роботи із реальними даними, де правильна відповідь не буде їй заздалегідь відома. Система вчиться відтворювати зв'язки між вхідними даними та відповідними виходами.

Покращення результатів та точності мережі проводиться шляхом мінімізації певної помилки, різниці між спрогнозованим та реальним

виходами. Позначимо цю помилку  $e$ . Тоді процес навчання мережі описується як

$$f(x) - e \leq Y \leq f(x) + e \quad (1.1)$$

### 1.3 Метод зворотнього розповсюдження помилки

Одним із найпопулярніших методів мінімізації помилки при навчанні мережі є метод зворотнього розповсюдження помилки. Суть методу полягає у поширенні сигналу похибки в зворотньому напрямку по мережі для корегування ваг та покращення точності мережі.

Метод працює ітеративно, для поступового зменшення похибки, а такі ітерації зазвичай називають епохами навчання мережі.

Початкові ваги мережі зазвичай встановлюють на певному невеликому випадковому значенні. Розглянемо покроковий алгоритм методу.

Позначимо  $t_i$  – бажані значення,  $y_i$  – виходи мережі,  $x_i$  – навчальні входи,  $\eta$  – швидкість навчання.

На кожній епосі подаємо вхідний сигнал  $x_i$  та отримуватимемо виходи  $y_i$ .

Похибка обраховується за формулою

$$E = \frac{1}{2} \sum_{i=1}^n (y_i - t_i)^2 \quad (1.2)$$

Після отримання виходу мережі на кожній епосі проводиться корегування ваг. Для цього знаходиться

$$\delta_i = y_i(1 - y_i)(t_i - y_i) \quad (1.3)$$

Для вузлів внутрішнього рівня

$$\delta_i = y_i(1 - y_i) \sum_{j \in I(j)} \delta_j \omega_{i,j} \quad (1.4)$$

Ваги оновлюються за формулою

$$\omega_{i,j} = \omega_{i,j} + \Delta\omega_{i,j}, \quad (1.5)$$

де

$$\Delta\omega_{i,j} = -\eta\delta_j x_i \quad (1.6)$$

Всі ці операції виконуються послідовно на кожній ітерації (епосі) роботи мережі, поки не буде досягнена бажана точність, певний ліміт епох або ж помилка не перестане змінюватись достатньо значуще тобто мережа майже припинить «навчатись».

#### 1.4 Проблема перенавчання

Часта проблема, що виникає під час навчання мережі з учителем полягає у перенавчанні мережі, тобто отриманні задовільних результатів на тренувальних даних, але, водночас, значного падіння точності при роботі на реальних даних.

Ця проблема часто трапляється у складніших та глибших моделях, з більшою кількістю параметрів та шарів відповідно. Для вчасного виявлення цієї проблеми зазвичай вхідні дані для мережі діляться на тренувальні та тестові, конкретне співвідношення для поділу залежить від задачі та даних, після чого тестові дані використовують для перевірки моделі після її навчання на тренувальних даних. Для вирішення ж проблеми або її завчасного уникнення використовують метод перехресного затвердження (cross-validation), ранню зупинку моделі до настання перенавчання, наприклад при

припиненні значного покращення точності на перевірочному наборі даних, регуляризацію мережі та її порівняння із результатами інших, змінюючи в процесі основні параметри моделі або ж її архітектуру, використання часткового відкидання нейронів в шарах.

## 1.5 Архітектури нейронних мереж

Одним з головних факторів ефективності використання нейронної мережі для вирішення задачі є правильно обрана й підібрана архітектура мережі. Розглянемо декілька типів архітектур популярних для обробки двовимірних даних та розпізнавання та ключові напрями їх використання.

### 1.5.1 Нейронні мережі прямого поширення

Найпростішою архітектурою є нейронні мережі прямого поширення або ж повнозв'язні мережі. В цих мережах нейрони кожного шару зв'язані із кожним нейроном наступного шару, водночас не маючи зв'язків всередині одного шару.

Найпростіший приклад повнозв'язної мережі прямого поширення зображено на рисунку 1.3.

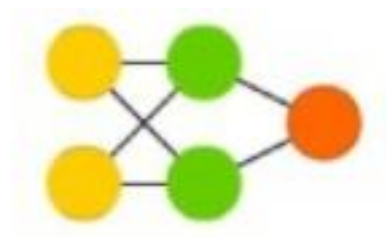


Рисунок 1.3 – Перцептрон з одним прихованим шаром

Якщо ж в мережі наявно більше одного прихованого шару, то вона вважається глибокою (Deep Neural Network). Глибоку нейронну мережу прямого поширення зображено на рисунку 1.4.

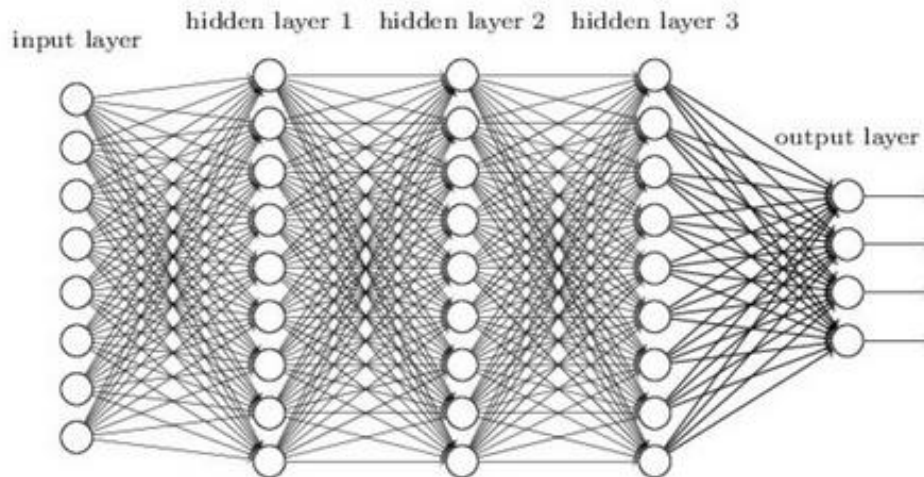


Рисунок 1.4 – Глибока нейронна мережа

Нейронні мережі прямого поширення мають широке застосування в задачах класифікації, проте мають значні проблеми щодо стрімкого зростання кількості параметрів при обробці значних об'ємів даних або даних більших розмірностей, як наприклад зображення. Для досягнення достатньої точності може знадобитись досить багато шарів, що, разом із зростанням кількості параметрів в мережі, стрімко збільшує обчислювальну складність та збільшує ймовірність виникнення перенавчання та проблеми спадних градієнтів при зворотньому поширенні похибки, коли зміна ваг мережі значно знижується або майже не відбувається за значної кількості прихованих шарів. Саме тому для багатьох практичних задач найуспішніші моделі часто використовують повнозв'язні шари лише в поєднанні із блоками інших архітектур.

### 1.5.2 Рекурентні нейронні мережі

Ще однією популярною архітектурою є рекурентні нейронні мережі, що мають будову певною мірою схожу на мережі прямого поширення, але додатково мають циклічні зв'язки в деяких нейронах прихованих шарів, таким чином реалізуючи «пам'ять» мережі й враховуючи часові зміни.

Основні використання таких мереж пов'язані із послідовностями та часовими рядами, проте часто використовуються і їх поєднання із іншими архітектурами, наприклад із згортковими нейронними мережами для забезпечення останніх здатністю до врахування змін часу. Прикладом задачі де таке використання є досить популярним це прогнозування руху об'єктів на зображенні чи відеоряді.

Приклад простої рекурентної нейронної мережі наведено на рисунку 1.5.

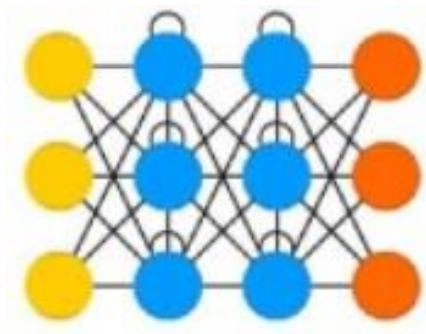


Рисунок 1.5 – Рекурентна нейронна мережа

Певним покращенням відносно класичних рекурентних мереж, що мали часті проблеми із спаданням градієнтів при використанні в навчанні методу зворотнього поширення помилки є мережі довгої короткострокової пам'яті (Long Short-Term Memory, LSTM), що мають можливість очищення й перезапису «пам'яті» нейронів. Схему мережі LSTM наведено на рисунку 1.6.



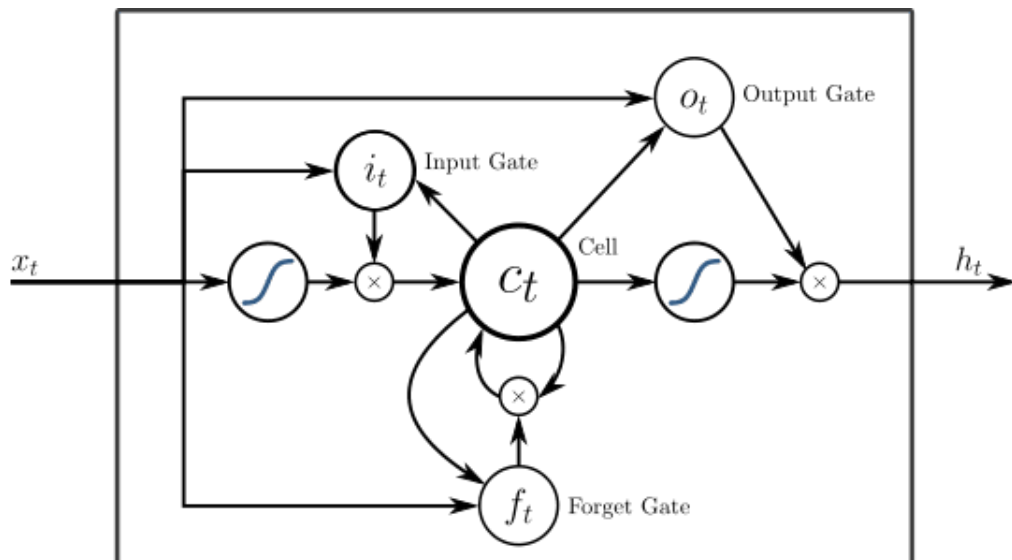


Рисунок 1.6 – Мережа довгої короткострокової пам'яті

### 1.5.3 Згорткові нейронні мережі

Значним покращенням в обробці багатовимірних та великих об'ємів даних в порівнянні із повнозв'язними нейронними мережами стали згорткові нейронні мережі (Convolutional Neural Networks, CNN).

В основі згорткових нейронних мереж лежать згорткові шари та відповідна операція згортки – обчислення скалярного добутку вхідних даних та фільтру певної розмірності, результатом чого є карта збудження фільтру.

Згорткові нейронні мережі прийшли на заміну звичайним повнозв'язним мережам для обробки зображень через необхідність в мережах, що могли б впоратись із неймовірним збільшенням кількості параметрів, що існує в задачах обробки зображень. Так, якщо ми маємо зображення всього 512 на 512 пікселів, то звичайна повнозв'язна нейронна мережа мала б 262,144 ваг (параметрів).

Згорткові нейронні мережі допомагають вирішити цю проблему завдяки застосуванню операції згортки. Як вже зазначалося, операція згортки

передбачає обчислення скалярного добутку даних фільтру та вхідних даних, що формує карту збудження фільтру.

Представлення виконання операції згортки на зображенні наведено на рисунку 1.7.

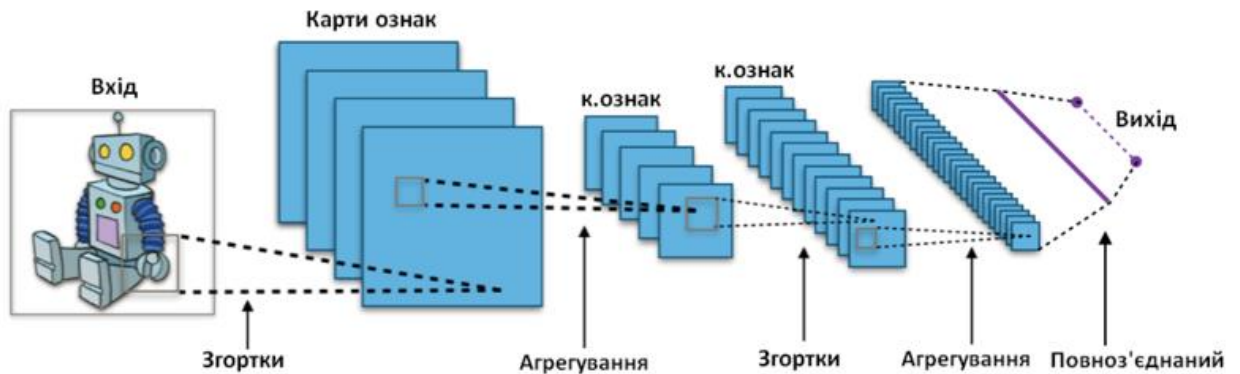


Рисунок 1.7 – Спрощене представлення операції згортки на зображенні в простій згортковій нейронній мережі

Ядро фільтру може мати розмірність, наприклад, 7 на 7, що залишає всього 49 вільних параметрів і кожне таке ядро матиме спільні ваги.

Таке значне зменшення кількості параметрів в мережі дозволяє, перш за все, позбавитись у мережах із зворотним поширенням похибки, проблем зникаючих градієнтів, а також проблеми вибуху градієнтів.

З цього також випливає, що згорткові нейронні мережі можуть бути значно глибшими за відповідні повнозв'язні мережі, за меншої кількості параметрів, таким чином зберігаючи точність та уникаючи перенавчання.

Саме тому зазвичай для задач розпізнавання образів із зображень та відеоряду застосовуються згорткові нейронні мережі, а найвідоміші прикладні застосування використовують надглибокі мережі із більше ніж 15-ма шарами.

Додатково допомагає покращити швидкість роботи згорткових нейронних мереж та зменшити обсяг використовуваної пам'яті використання спільних ваг в згорткових шарах для кожного рецептивного поля.

#### 1.5.4 Автокодувальники

Ще один тип архітектури мережі, що використовують для навчання відтворенню нерозмічених заздалегідь даних це автокодувальники. Найпростіші автокодувальники можуть бути побудовані на базі простої мережі прямого поширення, в той же час відповідно до задачі в їх основі часто застосовуються інші базові архітектури. Сигнал в автокодувальнику так само як і в інших мережах прямого поширення рухається від вхідних шарів через приховані й до виходів мережі, проте задача автокодувальника зводиться не до отримання певної результуючої змінної із входів, а, натомість, до відтворення входів мережі у вихідному її шарі. Типовий автокодувальник складається з двох частин, кодувальника що навчається запам'ятовувати певні властивості входу мережі, та декодувальника, що навчається за ними відтворювати вхідні дані. Найчастіше задача автокодувальника зводиться не до простого відтворення, а, натомість, до додавання до відтвореного сигналу певних корисних властивостей. Серед таких задач варто виділити згладжування сигналу, прибирання шумів та зниження розмірності даних.

Популярним є застосування автокодувальників на основі згорткових нейронних мереж для знешумлювання, згладжування, стискання або навпаки збільшенні розмірності зображень, зміну контрастності й подібні. Ці властивості автокодувальників також допомагають покращити якість даних для подачі їх на вхід інших мереж для проведення, наприклад, класифікації зображень.

## 1.6 Архітектури глибоких згорткових нейронних мереж

### 1.6.1 Агрегувальні шари, ваги та повнозв'язні шари

Згорткові нейронні мережі для об'єднання виходів кількох нейронів (їх кластеру) до входу єдиного нейрону наступного шару застосовує спеціальний тип шарів – агрегувальні шари.

Агрегувальні (пулінгові) шари зазвичай відрізняються основним принципом за яким із вихідних значень попереднього шару отримується значення для подачі на наступний.

Популярним підходом в цьому є застосування максимізації, також відоме як *max pooling*. Відповідно до своєї назви, в такому варіанті шар пулінгу обирає для подачі далі максимальне значення із попереднього шару. На практиці це означає, що максимальне агрегування вибирає яскравіші пікселі із зображення. Це корисно, наприклад, коли фон зображення темний і нас більше цікавлять саме світлі пікселі із зображення.

Усереднювальне агрегування (*average pooling*) вираховує середнє із значень кластеру нейронів попереднього шару. Усереднювальне агрегування згладжує зображення, тож чіткі риси можуть не бути ідентифіковані, коли використовується цей метод об'єднання, однак водночас це може бути корисним як раз для згладжування шумів та інших викидів на зображенні.

Менш популярним є застосування мінімізації (*min pooling*). Це повна протилежність до максимального агрегування і його основне застосування – зображення із світлим фоном, адже в той час як максимізація обирає світліші, мінімізація обирає темніші пікселі на зображенні.

В деяких роботах набирає популярність застосування адаптивного агрегування (*adaptive pooling*) [1]. Існує багато різних підходів для формулювання алгоритму відбору значень із кластерів попереднього шару, проте звернемо увагу на варіант представлений в [1], що має значні переваги в задачах класифікації. Тут оператор агрегування параметризується як матриця

розподілу. Таким чином, модель навчається знаходити відповідні параметри розподілу для агрегування паралельно з основним процесом навчання, наприклад класифікації об'єктів на зображенні, і, в результаті, кожному класу об'єктів відповідає свій тип агрегування.

Принцип знаходження значень для наступного шару у випадках популярних максимального та усереднювального агрегування наведено на рисунку 1.8.

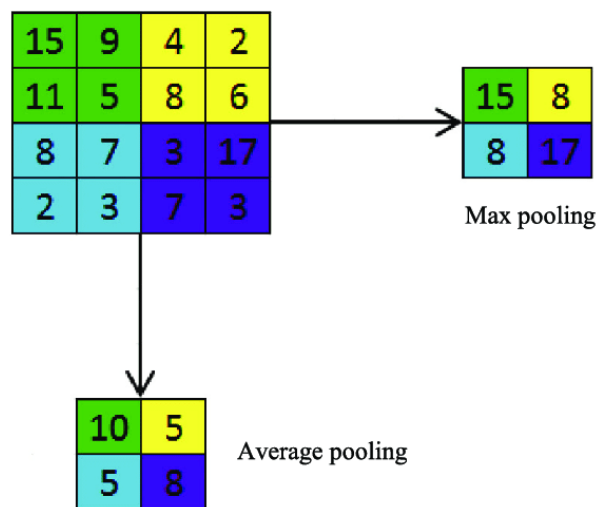


Рисунок 1.8. – Ілюстрація процесу максимального та усереднювального агрегування.

На рисунку 1.9 наведено результати операцій максимального та усереднювального агрегування для зображення.

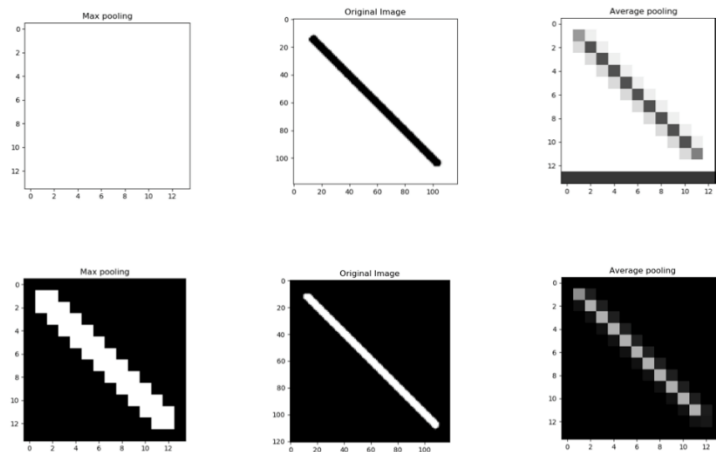


Рисунок 1.9 – Приклад максимального та усереднювального агрегування для простого зображення із чорним або білим фоном.

Окрім шарів згортки та агрегування, типові архітектури згорткових нейронних мереж також включають повнозв'язні шари для отримання із виходу попередніх шарів кінцевого виходу мережі. Такі шари еквівалентні звичайній повнозв'язній мережі і слують, наприклад, для проведення класифікації на значно зменшених за кількістю параметрів даних після згортки та агрегування.

### 1.6.2 LeNet та AlexNet

LeNet є простою архітектурою згорткової нейронної мережі, типовим представником таких мереж є LeNet5. Це одна з найперших відомих згорткових нейронних мереж, розроблена Яном ЛеКаном та ін. як засіб розпізнавання простих написаних від руки символів, цифр та букв.

Традиційна мережа включає усі типові шари сучасних ЗНМ та має два згорткові шари, два шари агрегації за кожним згортковим відповідно, та три повнозв'язні шари для класифікації.

Продовженням ідеї LeNet стала мережа AlexNet, створена та названа на честь Алекса Кричевскі. Модель стала відомою отримавши на 10.8% кращий результат за конкурентів на конкурсі із розпізнавання на датасеті ImageNet. Модель складається із 8 шарів, п'ять згорткових з яких лише за частиною слідував шар агрегування.

Одним із ключових переваг AlexNet можна вважати застосування функції активації ReLU, яка показувала значно кращі результати навчання та вирішувала деякі ключові проблеми традиційних функцій сигмоїди та гіперболічного тангенсу, що застосовувались в LeNet та значній частині архітектур того часу, таких як обчислювальна складність, спадання сили активації, нелінійність на всьому відрізку, що унеможлиблювало побудову ефективних глибших мереж. Саме через це оригінальну публікацію AlexNet [2] вважають однією із найвизначніших в історії розвитку ЗНМ адже вона дала початок багатьом глибшим архітектурам.

### 1.6.3 VGG-16 та VGG-19

Архітектури ЗНМ VGG були створені для задач виділення ознак на зображенні К. Сімонян та А. Зіссерман [3]. Модель продемонструвала один із кращих на той час показник точності в розпізнаванні об'єктів на зображенні на датасеті ImageNet.

Вона представляє собою певне покращення відносно мережі AlexNet та найбільш відома в двох модифікаціях – VGG-16 та VGG-19, де число відповідає кількості шарів, згорткових та повнозв'язних.

Традиційна VGG-16 має 13 шарів згортки, а шари агрегації слідують після 2, а потім після кожних 3 шарів згортки. Для класифікації використовуються 3 повнозв'язні шари. VGG-19 є модифікацією цієї мережі де замість трійок згорткових шарів, після двох пар слідують четвірки шарів

згортки. Таке поглиблення моделі продемонструвало збільшення точності в порівнянні із попередньою. В обох мережах використовуються функції активації ReLU, та функція softmax після повнозв'язних шарів для класифікації.

Архітектури VGG попри збільшення точності мають вади щодо використання пам'яті та дуже низької швидкості, тож в деяких практичних задачах замість них віддається перевага меншим архітектурам.

На рисунку 1.10 наведено архітектуру та процес масштабування вхідного зображення мережею VGG-16.

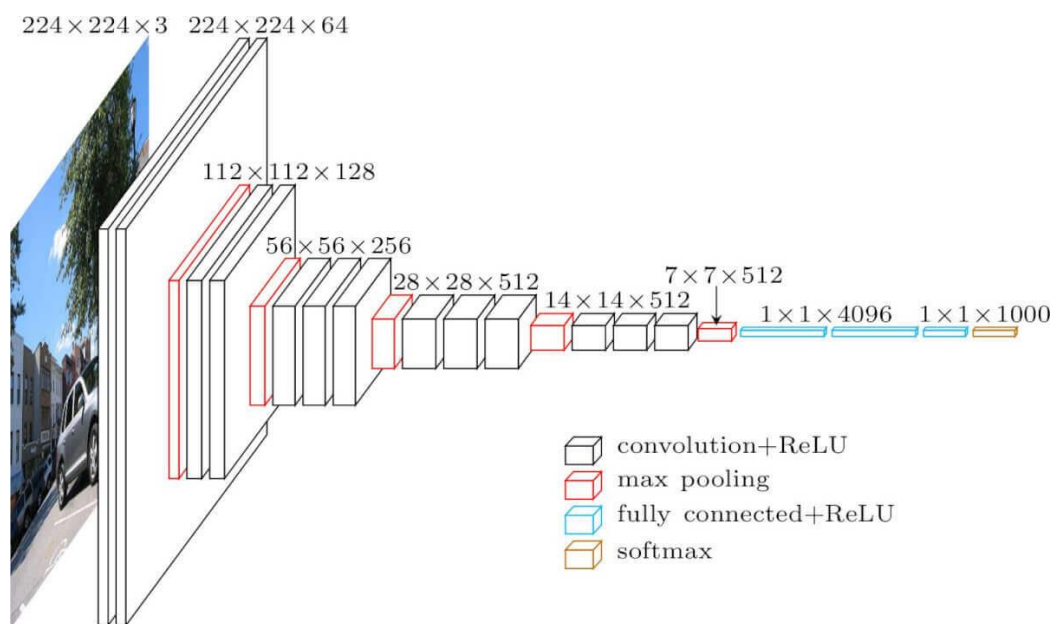


Рисунок 1.10 – Архітектура мережі VGG-16

#### 1.6.4 ResNet

Значним проривом в класифікації зображень була поява «залишкових» ЗНМ (Residual Networks). Основною причиною появи таких мереж стало те, що просте збільшення числа згорткових шарів в мережі швидко призвело до того, що в певний момент точність мережі перестає зростати і навпаки, починає різко падати.



Це було вирішено компанією Microsoft, що розробила мережу із з'єднаннями швидкого доступу (shortcut connections), що пропускають один або кілька згорткових шарів мережі та їх з'єднань, після чого виходи таких шарів, що рівні «залишковому» представленню, додаються до виходів шарів без з'єднань швидкого доступу, як показано на рисунку 1.11.

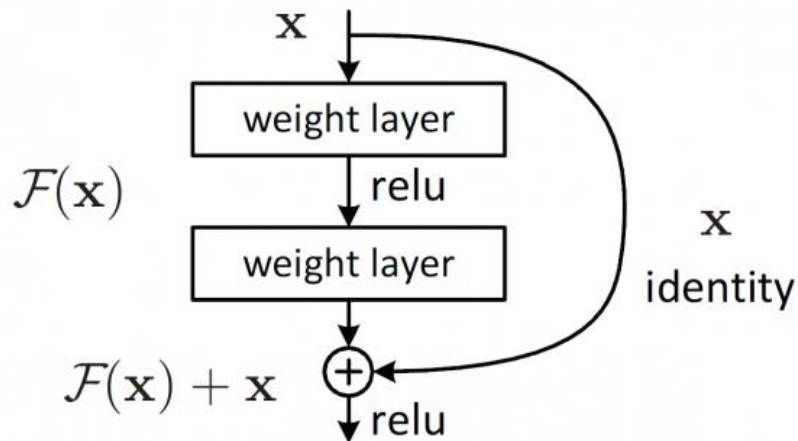


Рисунок 1.11 – Схема з'єднання швидкого доступу та отримання виходу такого з'єднання

Основа архітектури мережі була взята із архітектури VGG, однак модель демонструє значно більшу швидкість у порівнянні із нею, маючи меншу складність обчислень та потребуючи менше ресурсів, що значно краще підходить для практичних задач і є особливо цікавим для задач розпізнавання в реальному часі.

Як і в мережах VGG, у ResNet використовуються фільтри згортки 3 на 3, та кількість фільтрів у мережі збільшується при зменшенні поданої на них вхідної карти зображень. Однак, основна відмінність полягає саме в застосуванні в таких випадках з'єднань швидкого доступу. Якщо розмірність шарів входу та виходу однакова, то таке з'єднання використовується напряму, відповідно до простої формули  $F(x) + x$ . Якщо ж розмірності змінюються, то у випадку ResNet34 ідентифікатори шарів співставляються з нулями, що додаються для збільшення розмірності, а у випадку ResNet50 і більше –

використовується проекція такого з'єднання із застосуванням згорток 1 на 1 з обох боків, що являє собою блок із 3 шарів із фільтрами 1 на 1, 3 на 3 та знов 1 на 1. Шари агрегації використовуються лише до та після усіх згорткових шарів, реалізуючи на початку максимальну, а перед одним повнозв'язним шаром для класифікації – усереднювальну агрегацію.

На рисунку 1.12 представлено порівняння архітектури звичайного VGG-16, 34-шарової основи ResNet34, та фактичної архітектури ResNet34 із з'єднаннями швидкого доступу.

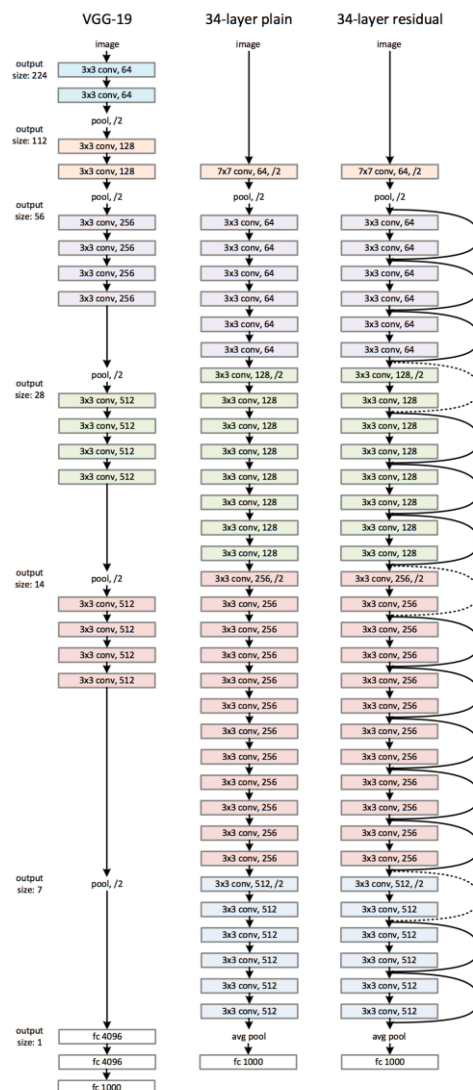


Рисунок 1.12 – VGG-16 (зліва), основні шари ResNet34 та ResNet34 із з'єднаннями швидкого доступу

### 1.6.5 R-CNN

Від традиційних задач класифікації зображень перейдемо до задач детекції або знаходження об'єктів на зображенні. Різниця між ними полягає в тому, що в алгоритмах знаходження об'єктів ми намагаємося намалювати певну візуальну рамку навколо об'єкта, що нас цікавить, щоб знайти його в зображенні. Таких об'єктів на зображенні може бути як один, так і декілька, зокрема це може бути декілька об'єктів різного класу. Окрім цього, часто такі задачі вирішуються для випадків зйомки в реальному часі, тож важлива не тільки точність мереж, що здатні були б знаходити, класифікувати та розмічувати об'єкти на зображенні, але і їх швидкість і, у випадках багатьох практичних застосувань на техніці, що не має великих потужностей для обробки даних, простота та практичність таких мереж.

Одними із прикладів мереж, що покликані зробити задачу детекції швидкою є мережі R-CNN. Ці моделі створені для вирішення проблеми виявлення одразу багатьох регіонів інтересу (region of interest, RoI) на зображенні. Замість пошуку безлічі можливих регіонів, було запропоновано [4] обирати лише певну кількість, наприклад, 2000 регіонів за допомогою вибіркового пошуку. Після цього, жадібний алгоритм пошуку об'єднує маленькі регіони у якнайбільші, що подаються на звичайну згорткову нейронну мережу для виявлення ознак та подальшої класифікації.

Схема роботи цього алгоритму виділення регіонів наведена на рисунку 1.13.

### R-CNN: *Regions with CNN features*

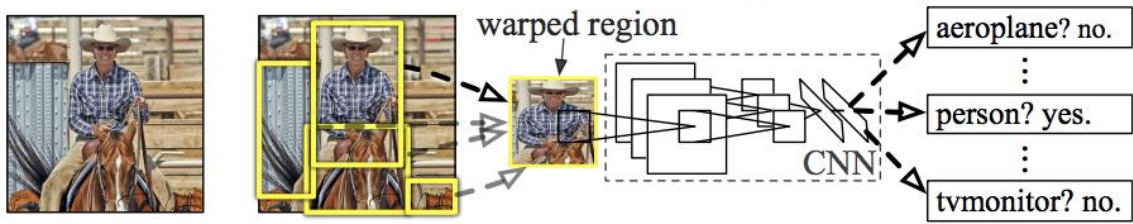


Рисунок 1.13 – Схема виділення та обробки регіонів інтересу в R-CNN

Звісно, такий підхід все ще залишає проблеми для застосування в реальному часі через необхідність виділення та класифікації значної кількості регіонів.

Певною мірою цю проблему допоміг вирішити підхід, реалізований у Fast R-CNN. В новій архітектурі обробка зображення починалась із подачі його до згорткових шарів для виділення ключових ознак, за рахунок чого така операція проводилася всього раз для 1 зображення, замість її проведення для кожного окремого регіону інтересу. Після виділення ознак використовується спеціальний шар агрегації регіонів інтересу (RoI pooling). Такі регіони передаються до повнозв'язних шарів мережі для визначення класів.

Схему обробки регіонів у мережі Fast R-CNN наведено на рисунку 1.14.

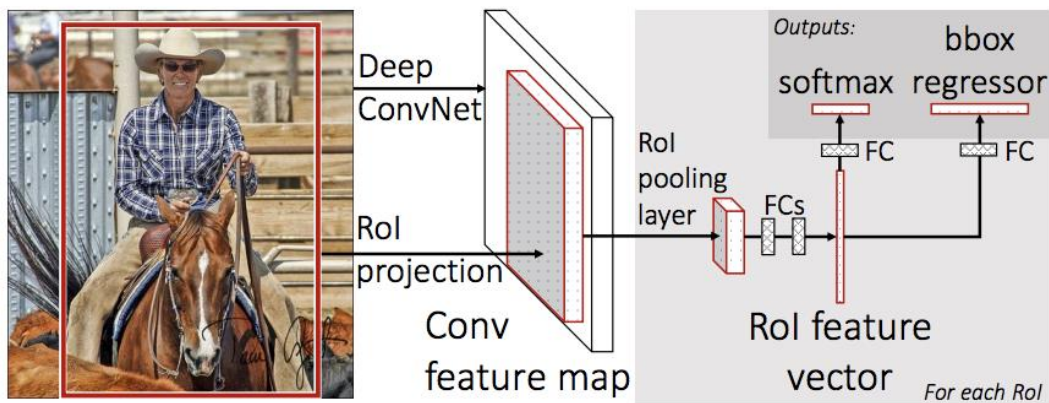


Рисунок 1.14 – Схема виділення та класифікації регіонів інтересу в Fast R-CNN

Попри ці покращення, існувала ще одна проблема зі швидкістю роботи мережі – застосування дуже повільного за своєю природою вибіркового пошуку. Для її вирішення було запропоновано [4] повністю відмовитись від такого пошуку та натомість на кроці одразу після виділення ознак за допомогою згорткового шару використовувати додаткову мережу для вибору регіонів інтересу (Region Proposal Network), після чого отримані регіони так само подавались до шару агрегування та до повнозв'язних шарів класифікації. Така архітектура отримала назву Faster R-CNN та пропонувалася для використання у детекції об'єктів в реальному часі.

Архітектура Faster R-CNN представлена на рисунку 1.15.

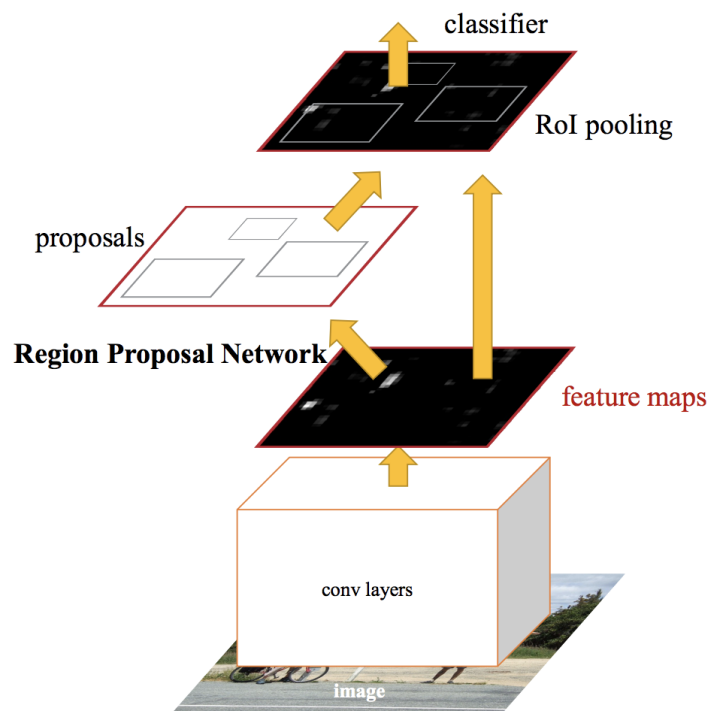


Рисунок 1.15 – Схема виділення та класифікації регіонів інтересу в Faster R-CNN

### 1.6.6 You Only Look Once

Мережі що базувались на виділенні регіонів інтересу на зображенні поступово набирали швидкість та наближались до можливостей розпізнавання та детекції в реальному часі. Однак, вони все ще мали певні вади у швидкості спричинені в тому числі самою необхідністю виділення, розгляду та прогнозування класу для багатьох окремих регіонів інтересу.

Архітектура YOLO (You Only Look Once або поглянути можна лише раз), натомість, розглядає проблему детекції та класифікації як проблему регресії. Зображення ділиться на сітку певних розмірів  $N$  на  $N$ , після чого для кожного сегменту сітки прогноуються можливі рамки, рівень впевненості, та ймовірності приналежності до певного класу.

Ці прогнози мають наступну формулу:

$$P = N * N * (B * 5 + C), \quad (2.1)$$

де  $B$  – кількість рамок;

$C$  – ймовірність приналежності до класу.

YOLO повністю орієнтована на розпізнавання в реальному часі і може робити це навіть трохи швидше, на швидкості близько 49 кадрів на секунду.

На рисунку 1.16 наведено принцип роботи системи обчислення ймовірностей для сегментів зображення в YOLO.

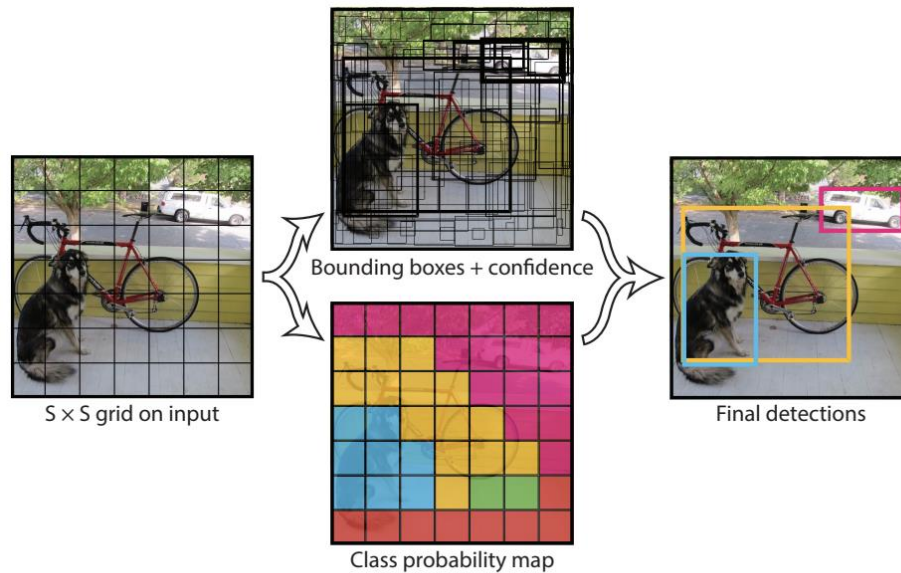


Рисунок 1.16 – Поділ зображення на сітку та прогнозування ймовірностей для кожного сегменту в YOLO

Сама мережа представляє собою так звану FCNN – повнозгорткову мережу і складається із 24 згорткових та 2 повнозв'язних шарів для класифікації. Навколо деяких шарів згортки використовуються залишкові блоки із згортками 1 на 1.

Із появи першої версії мережі, на світ з'явилося багато оновлених версій, спрямованих на покращення таких параметрів як швидкість, на що наприклад були спрямовані YOLOv2 та v5, що мали можливість працювати в реальному часі на великій кількості кадрів на секунду за умови достатньо потужного графічного процесору, а також модифікації такі як YOLOv3, спрямовані більшою мірою на покращення точності мережі.

Існують також екстремальніші модифікації мережі, як FastYOLO, що відрізняється використанням всього 9 згорткових шарів замість 24, що додатково пошвидшує її роботу і може бути особливо корисно у практичному застосуванні в реальному часі на менш потужних вбудованих системах що, зокрема, може бути особливо цікаво у військовій сфері оскільки не потребує наявності надпотужного графічного процесору для використання мережі для детекції в реальному часі.

### 1.6.7 Single Shot Detector

Ще одним прикладом однокрокового підходу для розпізнавання об'єктів на зображеннях є архітектура Single Shot Multibox Detector (SSD) [5]. Базова мережа була побудована на основі архітектури VGG16 через високу точність останньої. Для пошвидшення мережі та переходу до розпізнавання об'єктів в один крок повнозв'язні шари VGG були замінені на додаткові згорткові шари.

В основі процесу розпізнавання об'єктів мережа SSD використовує спеціальні якірні околи. Спершу, подібно до алгоритму You Only Look Once, зображення ділиться на сітку й у кожному сегменті сітки проводиться розпізнавання об'єктів. Для об'єктів що займають декілька таких сегментів використовуються якірні околи. Фактично, в процесі навчання мережа вчиться підбирати якірні околи із найбільшим показником перетину (IoU) відносно реального околу об'єкту. Тоді, такий якірний окіл, визначений співвідношенням сторін та рівнем наближення у зображенні, закріплюється за об'єктом і буде використаний для його виявлення та класифікації.

Принцип використання якірних околів об'єктів на поділеному на сітку зображенні в алгоритмі SSD наведено на рисунку 1.17.



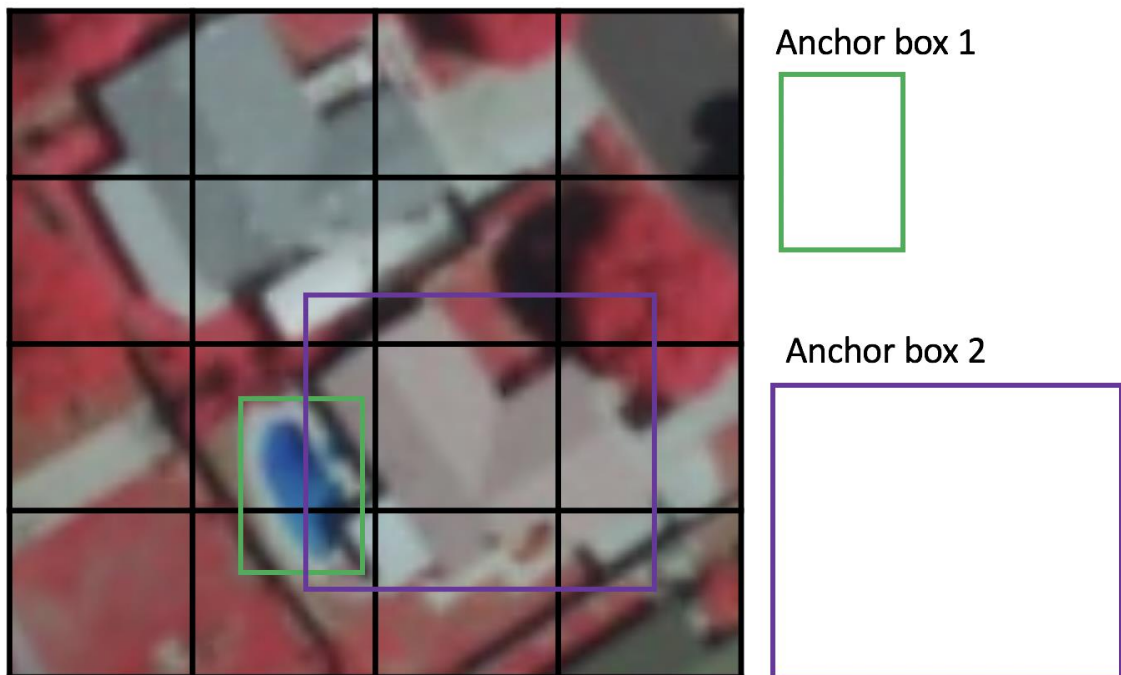


Рисунок 1.17 – Поділ зображення на сітку та застосування якірних околів в алгоритмі SSD

Чимало сучасних мереж використовують SSD для проведення кінцевого розпізнавання об'єктів, зазвичай замінюючи основну частину мережі із VGG16 на іншу мережу для пошуку ознак на зображенні, як наприклад компактна MobileNet-SSD.

### 1.6.8 Feature Pyramid Network

Окрему складність в задачах розпізнавання об'єктів створює різний масштаб шуканих об'єктів на різних зображеннях. Так, шуканий об'єкт може бути зображений дуже близько в тренувальній вибірці, проте реальні задачі потребуватимуть здатності моделі до його виявлення на значній відстані. В залежності від роздільної здатності вхідного зображення це також може означати, що об'єкт буде представлено значно меншою кількістю пікселів для виявлення в ньому відповідних вивчених ознак.

Для вирішення цієї проблеми часто використовують мережі масштабування та об'єднання ознак. Прикладом такої мережі є мережа пірамід ознак (Feature Pyramid Network, FPN).

Принцип роботи масштабування в FPN зображено на рисунку 1.18.

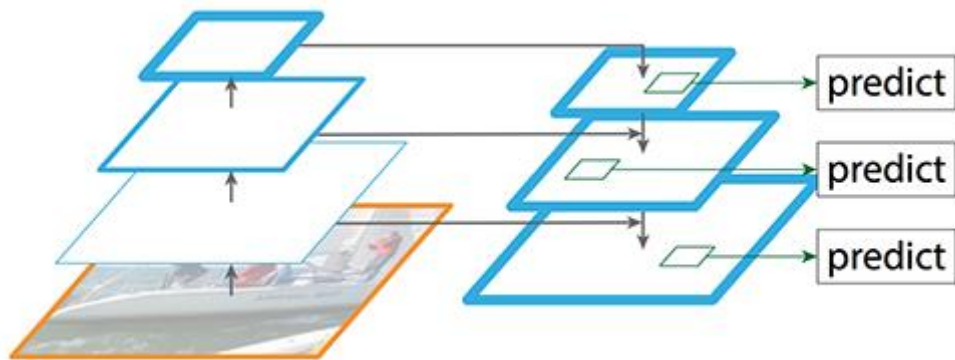


Рисунок 1.18 – Масштабування ознак в FPN

Основна перевага їх використання полягає у тому, що у початковому масштабі зображення може не бути достатньо семантично значимим. Такі алгоритми як SSD часто не використовують для розпізнавання найнижчі шари через невідповідність обчислювальної складності якості отримуваних ознак. Натомість, використовуються семантично наповненіші верхні, масштабовані шари. FPN допомагає збільшити можливості використання ознак за різних масштабів зображення вводячи додатковий шлях масштабування згори-вниз починаючи із більш семантично наповненого шару. Таким чином модель може навчатись розпізнаванню ознак об'єктів за різних масштабів без значних втрат в швидкості та збільшення обчислювальної складності алгоритму.

## 1.7 Постановка задачі

Основні задачі роботи наступні:

- розглянути основні архітектури та особливості будови штучних нейронних мереж що використовуються в задачах розпізнавання об'єктів;
- розглянути основні проблеми та особливості застосування апарату згорткових нейронних мереж для розпізнавання стратегічних технічних об'єктів;
- обрати набори вхідних даних для експериментів, провести обробку вхідних даних;
- розробити програмне забезпечення у вигляді глибокої згорткової нейронної мережі для розпізнавання стратегічних технічних об'єктів;
- провести експерименти та проаналізувати отримані результати та якість побудованої моделі.

## 1.8 Висновки до розділу 1

В даному розділі було розглянуто основні особливості будови нейронних мереж, проведено ознайомлення із загальними поняттями.

Було розглянуто основні складові базових архітектур нейронних мереж, популярні архітектури глибоких згорткових нейронних мереж, що є основою сучасних моделей у задачах розпізнавання образів, їх ключові складові та переваги.

## 2 ЗАСТОСУВАННЯ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ ДО ЗАДАЧ РОЗПІЗНАВАННЯ СТРАТЕГІЧНИХ ТЕХНІЧНИХ ОБ'ЄКТІВ

### 2.1 Етичні питання розпізнавання стратегічних технічних об'єктів та можливі застосування

Апарат розпізнавання образів почав застосовуватись до розпізнавання стратегічних технічних об'єктів відносно нещодавно, із розвитком більш точних та одночасно менш витратних за часом та необхідними обчислювальними потужностями мереж. Однак, чимало дослідників ставить перед собою питання етики у розробці такого виду моделей. Основне з цих питань полягає у тому як буде використано їхню розробку.

Звісно основне застосування таких технологій справді полягає у отриманні військових розвідданих, інформації про положення техніки, об'єктів зв'язку супротивника. Подібний апарат також застосовується безпосередньо на техніці та ракетах напрямлених на ураження супротивника, як наприклад камери дронів, що несуть бойовий заряд, або системи наведення деяких ракет. І так само як для оборони, такі технології можуть бути застосовані і для нападу та збройної агресії – для виявлення скупчень стратегічно важливої техніки, живої сили супротивника тощо, наведення на них відповідних видів озброєнь.

Однак, слід зазначити, що, окрім безпосередньо військових застосувань, такі технології можуть дуже допомогти для своєчасного застереження цивільних громадян про можливу небезпеку, а також для отримання інформації та висвітлення подій журналістами. В деяких випадках, як от перед початком російського вторгнення в Україну 2022 року, супутникові знімки на яких було виявлено скупчення техніки біля державного кордону стали для багатьох сигналом та дали підстави та інформацію про те, що потрібно готуватися до можливих провокацій або реального збройного конфлікту. Розповсюдження цих світлин в пресі також допомогло звернути увагу світової

громадськості та лідерів інших держав до проблеми навіть до початку повномасштабного вторгнення.

Окрім цього, такі технології можуть застосовуватись для того, щоб рятувати життя поранених військових або людей у небезпечній ситуації. [6]

## 2.2 Основні напрями застосування апарату розпізнавання образів

Серед основних задач, що вирішуються в розпізнаванні стратегічних технічних об'єктів методами згорткових нейронних мереж зокрема та апарату розпізнавання образів та комп'ютерного зору в цілому, слід виділити детекцію, створення рамок та класифікацію зображень і об'єктів на них як на статичних зображеннях із супутника [7][8], дрона [8] або відкритих джерел як соціальні мережі [9], так і в реальному часі із камери дрона [10].

Додаткову складність та вимогливість до швидкості та точності роботи має навігація ракет на основі комп'ютерного зору [11], задачі пов'язані із прогнозуванням руху і подібні [12].

## 2.3 Особливості застосування систем розпізнавання образів на дронах та наземних об'єктах

Із неупинним розвитком технологій комп'ютерного зору такі системи знаходять все більше різноманітних застосувань у військовій сфері.

Одним із ключових застосувань є системи, що допомагають оператору помічати, визначати та реагувати, або, у деяких випадках [13], самостійно реагують на потенційні загрози в полі зору систем камер на борту. Сам транспортний засіб може бути як повністю безпілотним, так і навпаки, перевозити персонал і обслуговуватись як водієм так і стрільком.

В першому випадку слід відзначити застосування комп'ютерного зору в багатьох випадках одночасно для автоматизованого орієнтування в просторі та переміщення безпілотного транспортного засобу і, в той же час, для миттєвого виявлення, визначення та реагування на потенційні цілі або загрози безпеці. Відповідні системи для наземних транспортних засобів можуть реагувати на зміни ландшафту, орієнтуватись на інші транспортні засоби в задачах охорони конвою, для безпілотних літальних апаратів можуть допомагати із процесами зльоту й посадки, автоматичного повернення на базу, і, водночас, чітко й в реальному часі визначати потенційні загрози такі як ворожі броньовані машини різного класу, військовослужбовців або просто потенційні загрози у вигляді людей зі зброєю, літальні апарати різного класу.

На рисунку 2.1 зображено віддалену збройну станцію Pitbull повністю автономного бойового колісного робота RoBattle, оптична система якого дозволяє за секунди реагувати на потенційні загрози.



Рисунок 2.1 – Віддалена збройна станція General Robotics Pitbull

В другому випадку, такі системи застосовуються в першу чергу як допоміжний інструмент для водія, оператора або стрілка, що покликані

допомогти у вчасному розпізнаванні цілей та загроз у складних умовах та вчасному попередженні про них, зменшуючи вплив людського фактору, однак залишаючи прийняття складних рішень щодо, наприклад, відкриття вогню, на оператора системи.

Приклад зображення із рамкою та класифікацією об'єктів для допомоги оператору наведено на рисунку 2.2.



Рисунок 2.2 – Зображення із відповідними рамками навколо значимих для безпеки об'єктів

Окрім безпосереднього бойового застосування, технічні засоби обладнані системами камер із обробкою зображення методами комп'ютерного зору знаходять широке застосування у розвідці, спостереженні та забезпеченні безпеки військових та стратегічних об'єктів [14][15], аналізі та обробці значних об'ємів візуальних даних [9][16]. Відомі й застосування систем комп'ютерного зору для виявлення наземних мін [17].

Варто зазначити, що в більшості такі системи потребують досить значних обчислювальних ресурсів, тому не можуть бути вбудовані в будь-яку систему на борту будь-якого військового або цивільного дрону або наземного транспортного засобу. У випадку важкої техніки, в більшості країн-лідерів із виробництва сучасної військової техніки вже почали оснащувати танки, подекуди БМП та БТР системами камер та бортовими системами обробки

зображень з камер, що можуть включати апарат комп'ютерного зору для розпізнавання об'єктів військової та безпекової значимості на зображенні.

Схематичний приклад такої системи з камер та апаратури для обробки інформації з них наведено на рисунку 2.3.

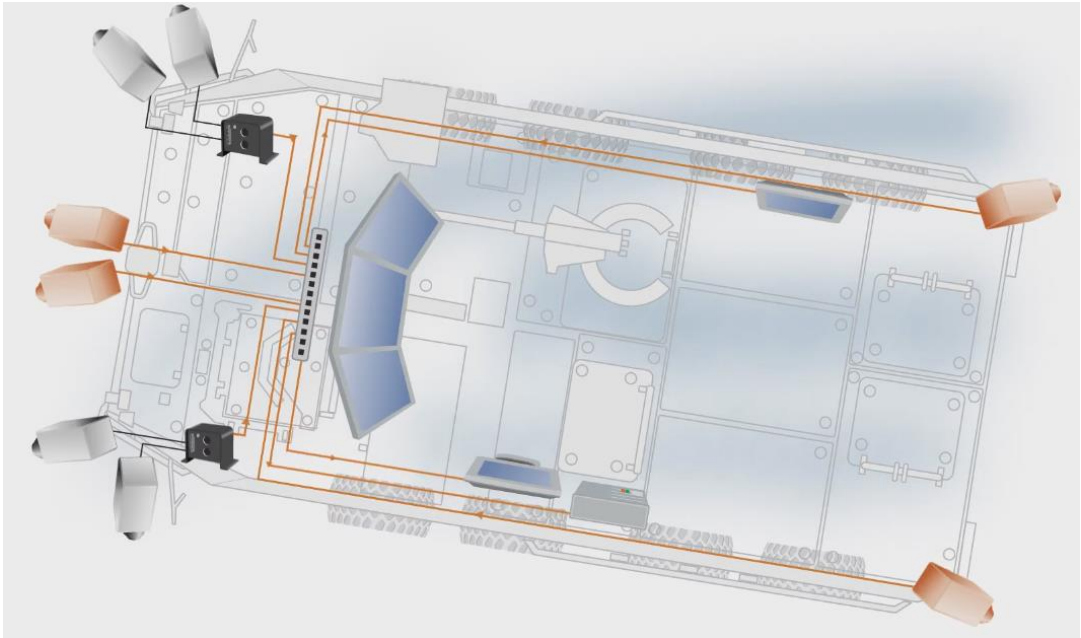


Рисунок 2.3 – Схема системи отримання та обробки зображення для подачі операторам з камер важкого військового транспортного засобу

Ще одне рішення, що є популярнішим у випадках безпілотних літальних апаратів (дронів) полягає у обробці зображення системою на стороні оператора, наприклад, в центрі керування, як у випадках із великими військовими ударними БПЛА, або на відповідній мобільній системі що може представляти собою переносну робочу станцію, ноутбук або навіть смартфон. Другий варіант популярніший у випадках використання баражуючих боєприпасів, також відомих як дрони-камікадзе, деяких легких військових дронів або ж застосування цивільних дронів для військових задач таких як розвідка або скидання боєприпасів. Такі переносні системи також можуть мати серйозні обмеження в обчислювальних ресурсах і, відповідно, потребують рішень швидких в роботі, для застосування в реальному часі, та ефективних з точки зору балансу точності та низьких вимог до обладнання.



## 2.4 Ускладнюючі фактори та специфіка застосування апарату комп'ютерного зору для розпізнавання стратегічних технічних об'єктів

### 2.4.1 Вхідні дані для навчання

Однією з головних проблем для навчання мережі на даних, що мають військову важливість, є доступність інформації. Це ж стосується й візуальних даних. У відкритому доступі із очевидних причин та з міркувань безпеки знаходиться дуже обмежена кількість зображень військової техніки та інших стратегічних технічних об'єктів, особливо таких, що були б хоч частково наближені до ситуації їх виявлення на полі бою.

Один із небагатьох відомих частково відкритих наборів даних, Moving and Stationary Target Acquisition and Recognition (MSTAR), має досить обмежений набір класів, хоч і покриває певне різноманіття наземної військової техніки.

На рисунку 2.4 наведено приклади об'єктів та їх кількість для кожного класу в публічній версії набору даних MSTAR.

Targets	BMP2	BTR70	T72	BTR60	2S1	BRDM2	D7	T62	ZIL131	ZSU234
17°	233	233	232	256	299	298	299	299	299	299
15°	587	196	582	195	274	274	274	273	274	274



Рисунок 2.4 – Візуальні приклади об'єктів та класи публічного датасету MSTAR

Через обмеженість варіативності кутів, дальності кадру в наведеному наборі даних та специфіки використання даних із радару з синтезованою апертурою в багатьох дослідженнях використовують дані із вільного доступу [18] [19], в тому числі з соціальних мереж [9] або їх поєднання із невеликими публічними наборами даних що є у відкритому доступі. Варто зазначити, що названі проблеми із даними це повністю не вирішує, тому, задля отримання зображень ближчих до реальних умов зазвичай використовуються аугментація [18] [19] або різні підходи до створення синтетичних даних [20].

Основними методами аугментації є розтягнення зображень, обрізка, ротації і подібні. Однак, слід зазначити що подібні процеси необхідно проводити із урахуванням необхідності наближення даних до реальних умов, наприклад, корисними для збагачення вхідних даних були б симуляції віддалення або зменшення на зображенні об'єкта або регіону інтересу і, в той же час, перевертання зображення деяких об'єктів призвело б до появи у вхідних даних нереалістичних зображень. До інших наближених до реалій способів аугментації даних слід віднести гаусівський фільтр та інші способи створення розмитостей на зображеннях, зміна тону, контрастності, затінення та інші способи вирівнювання образу об'єкту та заднього фону, синтетична симуляція туману та інших погодних умов що можуть ускладнювати розпізнавання.

Типові приклади аугментації вхідних зображень із відкритих джерел наведені на рисунку 2.5 [18].

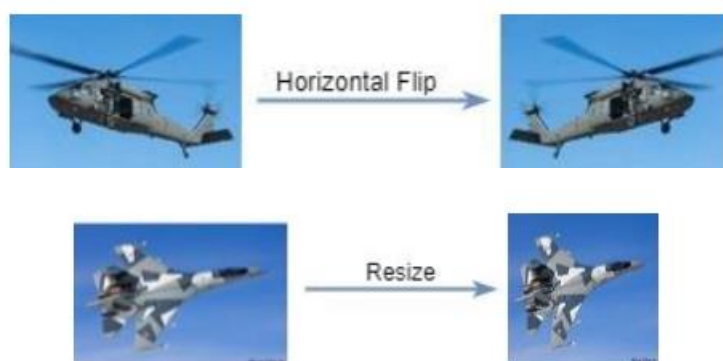


Рисунок 2.5 – Прості приклади аугментації даних із вільного доступу

Окрім вже названих проблем із отриманням та коректним збагаченням даних у звичайних типах спектру, варто зазначити майже повну відсутність у вільному доступі достатньої кількості даних для навчання мереж на зображеннях або відеоряді інфрачервоного або термального інфрачервоного спектру, що застосовується у багатьох реальних системах, часто одночасно із звичайним зображенням.

Досить часто для тестування ефективності та швидкості розроблених моделей, їх порівняння із іншими на однакових даних, а інколи й донавчання моделі, використовуються цивільні датасети, орієнтовані на тестування основних параметрів моделей комп'ютерного зору для відповідних задач як розпізнавання й класифікація об'єктів на зображеннях, відео, прогнозування їх руху [21]. Цікавими є також підходи до збагачення навчальних даних для навчання мережі шляхом передавального навчання [22].

На рисунку 2.6 зображено приклад передавального донавчання мережі для розпізнавання військових об'єктів на основі моделі, навченої на певних класах об'єктів популярного відкритого набору даних ImageNet [22].

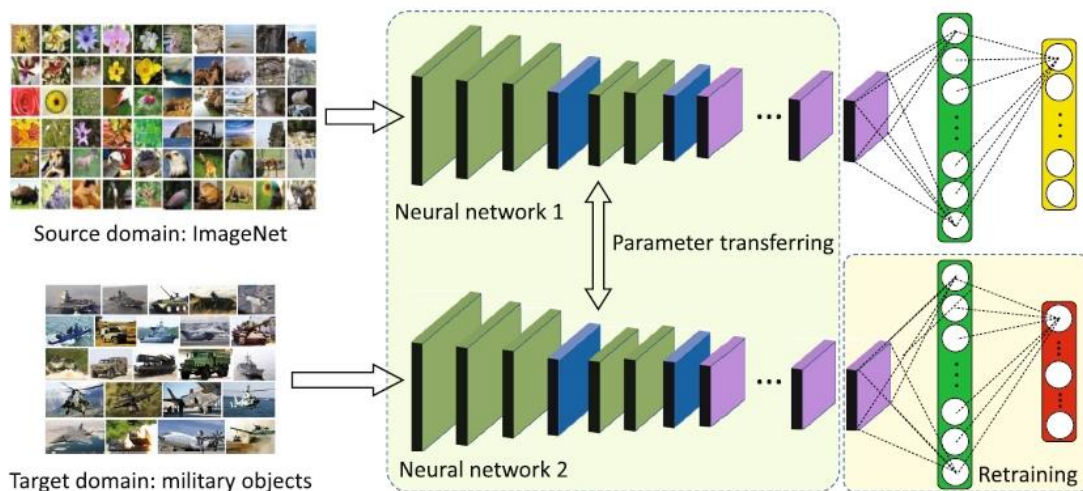


Рисунок 2.6 – Схема процесу донавчання мережі на невеликому об'ємі даних

Серед широкого вибору популярних сучасних наборів даних що підходять для тестування моделей призначених для різних типів задач комп'ютерного зору можна виділити великий набір даних для задач виявлення

об'єктів Microsoft Common Objects in Context (MS COCO) [23], який часто застосовується як еталонний для порівняння моделей в узагальненій задачі виявлення об'єктів. Для більш специфічних задач, пов'язаних із розпізнаванням в русі, варто відзначити й набір даних KITTI [24], що включає дуже велику кількість кадрів із камери закріпленої на авто і може імітувати задачі розпізнавання для систем, вмонтованих на наземні військові транспортні засоби, в тому числі й системи на автопілоті. Для цих тестів набір даних KITTI має деякі ключові властивості через наявність кадрів із перекриттям об'єктів інтересу, різними погодними умовами, складними сценами, різними кутами та віддаленням об'єктів. Приклади кадрів із набору даних KITTI наведені на рисунку 2.7.

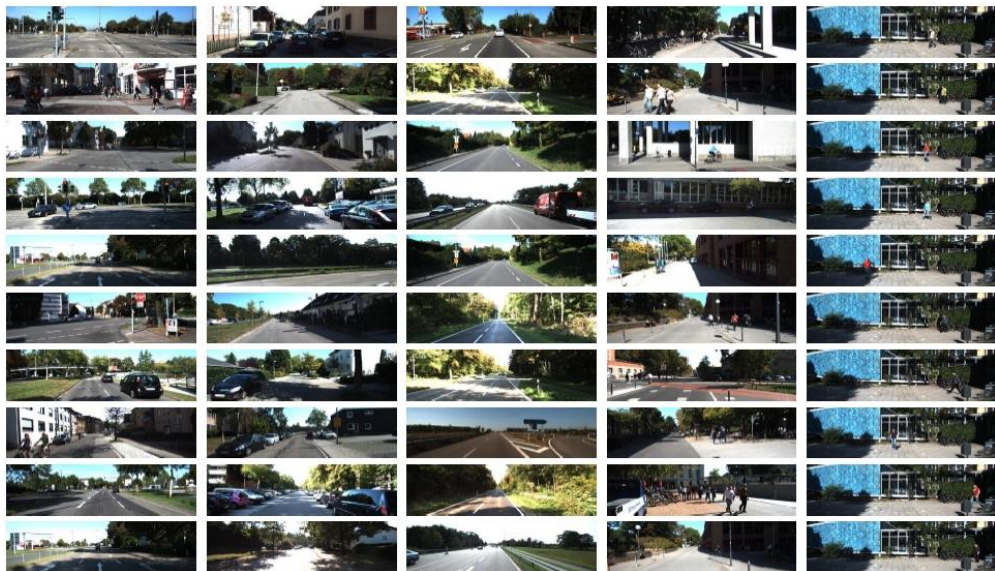


Рисунок 2.7 – Приклади кадрів із набору даних KITTI із складними сценами із об'єктами, що належать до різних класів

Ще один чудовий приклад набору даних, що має схожі до KITTI показники по різноманіттю та складності кадрів, однак більш придатний для використання для систем орієнтованих для застосування на дронах – VisDrone [25]. Цей набір даних має різні модифікації, відповідно до різних задач розпізнавання. Приклад кадрів із набору перевірочних даних VisDrone-VID для задач розпізнавання та класифікації об'єктів наведено на рисунку 2.8.



Рисунок 2.8 – Приклад кадрів із набору даних VisDrone-VID із значною відстанню до об’єктів, складними сценами та освітленням

Слід також зауважити, що хоч сучасні війни, особливо російсько-українська війна після ескалації 2022 року й додали до відкритих джерел неймовірну кількість зображень та відеозаписів, в тому числі і реальних кадрів із бойових дронів та іншої техніки, проте їх використання в наукових чи учбових цілях несе в собі як етичні, так і безпекові проблеми й можливе лише за ретельного відбору даних із офіційних джерел, що не несли б будь-якої стратегічної важливості та використання яких було б можливе із норм етики та моралі.

#### 2.4.2 Первинна обробка даних та складності розпізнавання

При використанні даних, наявних у відкритому доступі або у відкритих наборах даних не обійтись без необхідності не лише їх збагачення, але й фільтрації та первинної обробки. Ці дії включають в себе як прості зміни розмірності, так і перевірку на відповідність задачі, класу.

Що ж стосується реальних даних, отримуваних із систем камер вмонтованих на важкій техніці чи дронах, не обійтись і без інших проблем та викривлень, які необхідно усунути або мінімізувати для того щоб модель могла коректно працювати та отримувати достатню точність. До проблем

розпізнавання на реальних даних можна віднести часто низьку роздільну здатність камер. Наприклад, значна кількість систем моніторингу оточення для танків багатьох європейських країн та США досі використовує зображення із роздільною здатністю від 640 на 480 до 1200 на 900 пікселів. Це означає що достатньо віддалений об'єкт на такому зображенні може складатись із вкрай низької кількості пікселів і майже не піддаватись детекції або коректній класифікації. Більшість дронів, однак, дозволяють певною мірою варіювати використовувані камери або системи камер, хоча залишається обмеження по вазі, особливо якщо відносно легкий дрон також має нести бойовий заряд.

Приклад типової якості зображення при його наближенні для камери дрона наведено на рисунку 2.9.



Рисунок 2.9 – Кадри скупчення військової техніки зроблені із дрона

Деякі інші типові проблеми та викривлення зображення включають використання лінз із більшим кутом огляду, особливо у випадках розвідувальних дронів або оптичних систем для огляду оточення важкої техніки, розриви у зображеннях, розмитість і подібні. Проблемним фактором також може бути складність наведення камери відповідного дрону або

наземного транспортного засобу так, щоб усі об'єкти інтересу були повністю в кадрі. Окрім цього, у великій кількості випадків апарат із встановленою камерою сам знаходиться в русі, одночасно знімаючи рухомий об'єкт що створює надскладні умови для виявлення та класифікації такого об'єкту. Сукупність цих факторів показує складність самої задачі розпізнавання стратегічних технічних об'єктів у реальних умовах та необхідність у створенні систем, добре підготовлених до роботи із складними, та часто малоінформативними з точки зору ознак, даними.

Деякі рішення проблеми нечіткості зображення, складного вирізнення об'єкту на однотонному фоні включають накладання певних фільтрів, збільшення контрастності зображення, інтенсивності кольорів [26].

#### 2.4.3 Маскування

Додаткові проблеми для систем розпізнавання образів в реальних умовах створює пряме та непряме маскування. Приклад найпростішого маскування із яким стикаються дрони зображено на рисунку 2.10.



Рисунок 2.10 – Просте маскування стратегічних технічних засобів із використанням тіні та природних перешкод

Якщо у випадках простих перешкод мережа, що мала достатню навчальну вибірку для вивчення основних ознак могла б визначити об'єкт на зображенні, то використання професійного маскування значно ускладнює цей процес і потребує значно більшої натренованості мережі або використання спеціальних методів для нівелювання ефекту маскування на розпізнавання обрисів та ознак об'єкта.

На рисунку 2.11 показано приклад маскування зенітно-ракетного комплексу за допомогою військової маскувальної сітки що добре співпадає гамою кольорів із оточенням.





Рисунок 2.11 – Маскування ЗРК за допомогою спеціальної маскувальної сітки

Рішення проблеми маскування часто потребують серйозного ускладнення мережі та використання покращених технологій вибору регіонів інтересу або виявлення ключових ознак на зображенні. Так, наприклад, автори [19] застосували аналог апарату виявлення найпомітніших ознак та звертання уваги аналогічний до людського мозку, заснований на принципах гештальт-візуальної психології. Іншим прикладом спеціалізованого рішення проблеми маскування є використання спеціального оператора виявлення згладжених об'єктів на карті інтенсивності зображення [27]. Використання цього оператора продемонстроване на рисунку 2.12.

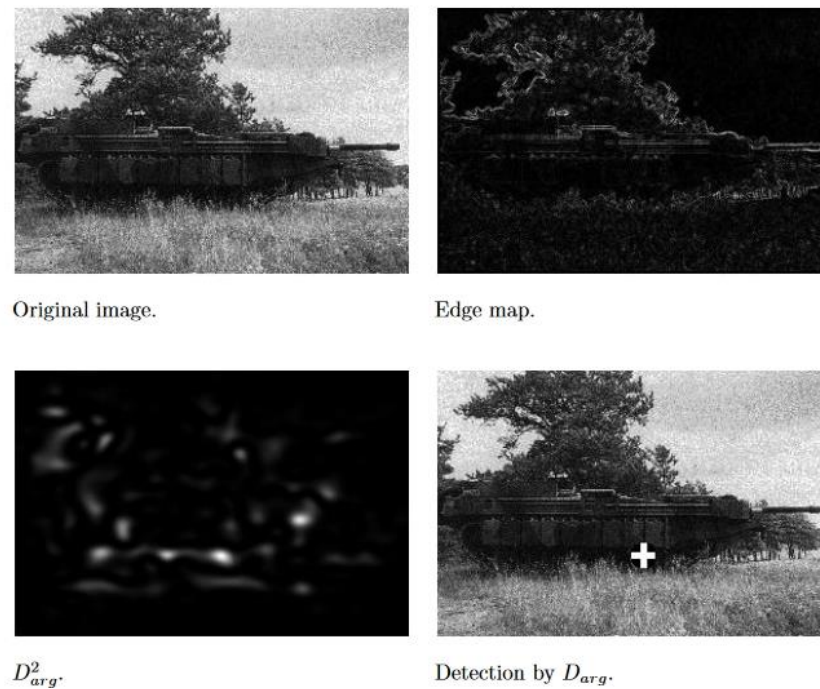


Рисунок 2.12 – Застосування оператора  $D^{2arg}$  запропонованого в [27] для виявлення об'єкту що використовує маскування

Ще одним рішенням проблеми маскування, що найчастіше використовується у військовій сфері є застосування додаткових каналів зображення у інфрачервоному або термальному інфрачервоному і подібних спектрах. Через важкість отримання даних для виявлення стратегічних технічних об'єктів у відкритому доступі, більшість рішень із використанням цих додаткових каналів для одночасного навчання моделі на кількох типах даних або виключного навчання лише на відповідному їх типі залишаються лише у поодиноких застосуваннях в цивільній сфері [21].

Варто зазначити, що розвиток систем комп'ютерного зору та їх військового застосування викликав увагу і до зворотнього процесу в сфері маскування: створення спеціального маскування, яке не буде виявлене системами комп'ютерного зору через особливу текстуру [28].

#### 2.4.4 Технічні обмеження

В більшості своїй у випадку використання систем комп'ютерного зору на наземній техніці вони використовують вбудовані бортові системи та обчислювальні потужності для обробки даних. Додаткові обмеження тут створюють певні вимоги до компактності та необхідність створення систем готових або пристосованих до складних умов, вологи, вібрацій та інших зовнішніх факторів впливу. Досить часто графічні процесори або повні системи розташовані в спеціальних кейсах для захисту від факторів середовища. Що стосується БПЛА, в переважній більшості випадків складні обчислення пов'язані із виявленням та класифікацією стратегічних технічних об'єктів що потрапили в об'єктив камери дрону виконуються не на борту дрону. Існують різні рішення, від командних центрів до мобільних додатків. В цивільній сфері популярним рішенням є перенесення обчислень в хмару [29], однак із міркувань безпеки такий підхід зазвичай не розглядається для застосування у військовій справі. Ключовим фактором в оснащенні як БПЛА так і відповідної робочої станції зазвичай є маса самого літального апарату, його вантажопідйомність без втрати в швидкості та маневреності, дальність польоту та бойова задача що впливають на вибір бажаної робочої станції для керування дроном.

У всіх із перерахованих вище випадків, окрім хіба що командних пунктів, існують строгі обмеження того, наскільки обчислювально потужною може бути робоча станція для обробки даних, виявлення та класифікації об'єктів. Сучасні моделі згорткових нейронних мереж зазвичай орієнтовані на проведення більшої частини обчислень на графічному процесорі і саме потужність цього процесору часто є вирішальною у можливості застосування системи комп'ютерного зору та її потужності, а разом і точності.

Важливим фактором у створенні рішення для розпізнавання стратегічних технічних об'єктів таким чином є його масштабованість та

можливість практичного застосування як у добре оснащених системах, так і у компактніших системах меншої потужності, ближчих до умов поля бою та пристосованих більшою мірою до вимог надійності ніж обчислювальної здатності.

#### 2.4.5 Довіра та безпека

Проблема довіри до систем машинного навчання, штучного інтелекту та комп'ютерного зору існує в багатьох сферах, але вона більш сильна саме в тих сферах, що пов'язані із безпекою напряду. Чимало операторів військової техніки, командувачів та іншого персоналу можуть проявляти недовіру до систем комп'ютерного зору, встановлених на техніці.

Для вирішення цієї проблеми [14] рекомендується достатнє інформування персоналу про функціонал та принцип роботи системи, її типові слабкості й переваги, обмеження у застосуванні. Ця проблема додатково зростає при використанні повністю автономної або безпілотної техніки поряд із персоналом, наприклад для захисту конвоїв, моніторингу території. Додатковий зв'язок між операторами або працівниками підрозділів що відповідають за використання відповідного автономного транспортного засобу або дрону та персоналом на землі може допомогти у вирішенні цих проблем та побудові ланцюгу довіри людини до роботи іншої людини. Що стосується самих операторів, то системи комп'ютерного зору часто використовуються разом із подачею відповідного вхідного зображення тож оператори мають змогу порівняти та перевірити результати визначення цілей або потенційних загроз.

Додаткову недовіру не лише персоналу а й командування та держав відносно фінансування програм із розвитку штучного інтелекту у військовій сфері є міркування щодо кібербезпеки та захищеності таких систем від

хакерських атак [14]. Такі міркування особливо стосуються використання цивільних моделей та моделей із відритим кодом, оскільки вони можуть мати вбудовані «бекдори» що нададуть небажаний доступ хакерам або відому супротивнику слабкість. Теж саме стосується й використання відкритих даних. Саме тому багато країн почали також інвестиції в збір великих масивів даних та закриту розробку власних систем, задля забезпечення власних моделей власними зібраними даними, яким повністю можна довіряти та відслідкувати доступ до них.

## 2.5 Огляд деяких відомих цивільних та спеціалізованих рішень

За всіх наявних складнощів, у сфері розпізнавання стратегічних технічних об'єктів було зроблено чимало цікавих досліджень та побудовано специфічну методологію для вирішення проблем, що не є типовими для інших сфер застосувань систем комп'ютерного зору якщо врахувати, що у військовій сфері вони часто наявні одразу у сукупності.

Серед деяких цікавих рішень можна виділити застосування регіонів цікавості для початкової сегментації та обробки вхідних зображень відповідно до принципів гештальт-візуальної психології для подальшого виявлення стратегічних технічних об'єктів в складних умовах: за умов маскування, завіс, низької роздільної здатності зображення та всіх факторів цих одночасно [19]. Тестування моделі проводилось на власних даних зібраних із відкритих джерел, а також на невійськових, загальнодоступних наборах даних. Результати роботи моделі на зображеннях із застосуванням маскування наведені на рисунку 2.13.



Рисунок 2.13 – Результати виявлення та класифікації стратегічних технічних об’єктів та персоналу на відносно складних зображеннях низької роздільної здатності

Що стосується прискорення моделі та можливості інтегрування її в системи із меншими обчислювальними потужностями, популярним рішенням є заміна або відмова від використання повнозв’язних шарів як таких, що мають велику кількість параметрів та потребують додаткових витрат часу, ресурсів та пам’яті. Серед потенційних заміників в інтересах прискорення класифікації розглядається зокрема екстремальна машина для навчання (Extreme Learning Machine, ELM) [18], що, однак, має значні проблеми із падінням точності класифікації.

Серед інших рішень цікавими є розширення можливостей навчання мережі та наборів вхідних даних шляхом використання зображень із соціальних мереж, що також може становити додатковий інтерес для так званої розвідки на основі відкритих джерел [9], та застосування техніки передавального навчання [22].

Попри це, слід зазначити що відкриті дослідження в цій сфері залишаються проблематичними як із етичних міркувань, так і з міркуванням наявності та якості даних. Тож, значним чином сучасний етап розвитку систем розпізнавання образів для виявлення та класифікації стратегічних технічних об’єктів завдячує роботам та моделям із цивільних сфер, як наприклад моделі

для розпізнавання об'єктів в ускладнених умовах із дронів. Чимало цивільних моделей, орієнтовані на вирішення таких проблем, для виявлення та класифікації об'єктів на зображенні чи відеоряді будуються для вирішення певних проблем подібних інтегрованих систем, як рух камери, перекриття об'єкта, погодні умови, низька якість зображення.

Корисними для військової сфери є й рішення спрямовані на інтеграцію систем на компактніші платформи і загальне зменшення потреб моделей у пам'яті, обчислювальних ресурсах, за збереження достатньої точності. Так деякі моделі виявлення об'єктів із відеоряду здатні бути використані на смартфонах [30] [31].

Збільшення точності та поліпшення процесу слідкування за об'єктом в реальному часі було досягнуто [32] [33] шляхом використання в згорткових нейронних мережах шарів з пам'яттю LSTM.

## 2.6 Висновки до розділу 2

В даному розділі було визначено основні проблеми, етичні питання та побоювання, пов'язані із застосуванням апарату розпізнавання образів до задач розпізнавання стратегічних технічних об'єктів, розглянуто шляхи їх вирішення, різні застосування апарату у військовій та цивільній сфері.

Було розглянуто типові рішення проблем знаходження, доповнення та обробки вхідних даних, вибору архітектури та технічних вимог до мережі.

Було визначено потребу у масштабованому рішенні, здатному поєднати швидкість обробки даних в реальному часі, здатність робити достатньо точні прогнози та можливість застосування за відносно низьких обчислювальних потужностей.

### 3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

#### 3.1 Обрані вхідні дані для навчання

Для забезпечення підготовленості мережі до реальних умов в якості вхідних даних було обрано набір даних, зібраний студентами та викладачами Факультету прикладних наук Українського Католицького Університету у співпраці із громадською організацією «Тех Інституції» [34].

Набір даних складається з більш ніж 100 відеозаписів, зібраних із відкритих та офіційних джерел за перші пів року повномасштабної війни Росії проти України 2022 року.

Ці відеозаписи представляють собою реальні кадри з дронів різних класів та напрямів застосування, зібрані безпосередньо під час виконання ними бойових завдань. Об'єкти інтересу в кадрах із набору даних поділено на 3 класи: військові транспортні засоби, військовослужбовці та невизначені рухомі об'єкти.

Головна перевага використання цього набору даних полягає у його відповідності та репрезентативності відносно реальних бойових умов, з якими стикаються камери та оператори дронів при виконанні завдань. Серед ключових прикладів типових ускладнюючих факторів, що наявні в наборі даних в першу чергу варто виділити досить низьку якість зображення. Так, у більшості відеозаписів роздільна здатність камери не перевищує 1280 на 720 пікселів, а в деяких випадках складає близько 270 на 270 пікселів. Окрім цього, у відеозаписах подекуди присутні шуми та розмитість. У сценах на відеозаписах часто велика кількість різноманітних об'єктів як от будівлі, деякі відеозаписи також включають складні погодні умови як падаючий сніг та техногенні ускладнюючі фактори як задимлення та горіння в процесі бою або в результаті удару по об'єкту. Багато кадрів демонструють перекриття об'єктів інтересу іншими або оточенням, багато кадрів зроблені під час знаходження дрона в русі, або ж демонструють рухомі об'єкти, в окремих випадках



присутні певні спотворення кутів та форм. Об'єкти інтересу, від військової техніки до персоналу, використовують як наявні природні чи техногенні укриття, так і рукотворне маскування, як маскувальні сітки.

Об'єкти інтересу в наборі даних поділені на три класи: військові технічні засоби, військовослужбовці та невизначені транспортні засоби, що часто відповідають випадкам, коли конкретний тип транспортного засобу та його форми з тих чи інших причин важко відрізнити при розмічуванні, однак чітко можна зрозуміти приналежність до транспортних засобів.

На рисунку 3.1 наведено приклади кадрів із набору даних, що демонструють деякі типові фактори, ускладнюючі розпізнавання об'єктів.



Рисунок 3.1 – Кадри із набору даних, що демонструють деякі типові викривлення та ускладнюючі розпізнавання об'єктів фактори

В результаті виділення кадрів та їх фільтрації було отримано вибірку із 11776 зображень. Набір даних має певну незбалансованість класів представлених об'єктів із переважанням об'єктів класу військової техніки що, однак, є оптимальним для побудови системи відповідно до завдання роботи. Набір було поділено на набори для навчання мережі, валідації при навчанні та тестовий набори у співвідношенні 80, 10 та 10 відсотків відповідно.

### 3.2 Вибір платформ та програмних засобів для реалізації моделі

Основною орієнтацією розроблюваної моделі є системи, що базуватимуться на робочих станціях операторів дронів, а в окремих випадках можуть знаходитись і безпосередньо на дроні. За таких умов виникають значні обмеження за обчислювальними потужностями, що будуть доступні для використання системи та, відповідно, обмеження обчислювальної складності самої системи. Окрім архітектури моделі, на швидкість роботи за обмежених обчислювальних ресурсів та можливість застосування на різних робочих станціях значний вплив має вибір платформи для розробки та використання побудованої моделі.

Таким чином, визначимо головні вимоги до платформи для розробки моделі:

- можливість використання на різних пристроях, таких як ноутбуки, планшети та мобільні пристрої, а також інтегровані пристрої;
- висока швидкість роботи моделей, мінімальний час завантаження;
- мінімальні потреби у використовуваній пам'яті та ресурсах процесорів;
- можливість використання різних типів процесорів, від центральних до графічних та тензорних процесорів, наявних на

пристрої для максимально ефективного використання наявних ресурсів.

У таблиці 3.1 наведені результати еспериментів із проведення розпізнавання об'єктів на еталонному наборі даних за допомогою повнорозмірної моделі YOLOv3 на мобільному пристрої Xiaomi Blackshark [35].

Таблиця 3.1 – Порівняння платформ для роботи моделі для мобільних пристроїв.

Фактор	OpenCV	TFLite	Snapdragon NPE
Попередня обробка	9.3 мс	56.3 мс	56.3 мс
Розмір моделі	237.08 Мб	236.66 Мб	212.41 Мб
Час завантаження	456 мс	271 мс	2618 мс
Час розпізнавання	5203 мс	4379 мс	595 мс
Середнє використання ОЗУ	633 Мб	263 Мб	707 Мб
Максимальне використання ОЗУ	1.5 Гб	1 Гб	1.1 Гб

Варто зазначити, що, із наведених моделей, OpenCV представляє собою платформу, в першу чергу сфокусовану на швидкості попередньої обробки зображення та може використовувати лише центральний процесор пристрою. Платформа Snapdragon Neural Processing Engine має можливості для використання всіх ресурсів пристрою, однак використовує дуже низькорівневий спосіб розробки моделей та обмежена пристроями на базі процесорів Snapdragon. В той же час, платформа Tensorflow Lite за можливості використання всіх потужностей смартфона також має широкий вибір

підтримуваних моделей та зручний процес розробки. Платформа також дозволяє створювати моделі, придатні для використання на багатьох пристроях, в тому числі дозволяє підлаштування побудованої моделі для її використання на інтегрованих системах, що використовують тензорні процесори, оптимізовані для машинного навчання, EdgeTPU.

Для розробки моделі було обрано мову програмування Python та середовище програмування Google Colab, що дозволяє використовувати віддалені обчислювальні потужності, в тому числі графічні та тензорні процесори. В якості платформи для розробки та потртування на мобільні пристрої було обрано TensorFlow Lite.

### 3.3 Метрики порівняння моделей

Для попереднього порівняння моделей традиційно використовуються еталонні набори даних та відповідні рекомендовані метрики. Для розпізнавання об'єктів одним із найпопулярніших еталонних наборів даних є Microsoft Common Objects in Context (MS COCO) [23].

В наборі даних MS COCO використовується спеціальна метрика усередненої по класам середньої точності (mAP). Показник середньої точності береться за різних значень перетину над об'єднанням (Intersection over Union, IoU), від 50 до 95% з кроком у 5%, тобто mAP[0.5;0.95]. Для конкретного значення перетину над об'єднанням він розраховується як

$$mAP = \frac{1}{|classes|} \sum_{c \in classes} \frac{|TP_c|}{|FP_c| + |TP_c|}, \quad (3.1)$$

де classes – множина класів набору даних COCO, TP<sub>c</sub> – правильно визначені об'єкти, що належать класу, FP<sub>c</sub> – неправильно визначені об'єкти, що належать класу.

Візуалізація поняття перетину над об'єднанням, що використовується для вимірювання якості побудованих моделлю околів об'єктів відносно реальних околів, наведена на рисунку 3.2.

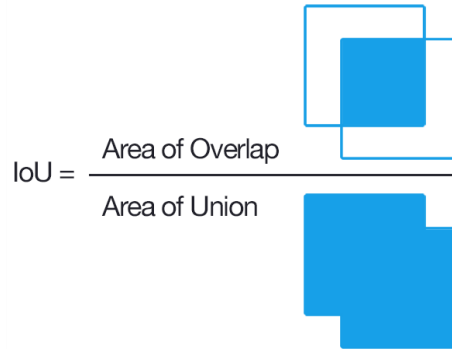


Рисунок 3.2 – Перетин над об'єднанням для вимірювання якості околів об'єктів.

Відповідно до специфіки задачі та реальних потреб для допомоги оператору дрона, враховуватимемо також значення середньої точності (AP) за перетину за об'єднанням від 50%, тобто AP[0.5], та середню точність AP[0.5;0.95] для класу військової техніки, враховуючи певну незбалансованість вхідних даних, а також основну задачу роботи.

Для порівняння швидкості та компактності моделі, для її використання в реальному часі у системах із обмеженими обчислювальними ресурсами, будемо використовувати показники часу, витраченого на обробку одного кадру (latency) у мілісекундах та відповідного йому показнику пропускної здатності (Frames per Second, FPS) у кадрах на секунду, а також показники кількості параметрів та ваги моделі.

### 3.4 Порівняльний аналіз моделей

Для порівняння широкого набору сучасних моделей були обрані легкі моделі, націлені на досягнення високої швидкості роботи на пристроях із обмеженими ресурсами. Для порівняння використовувались результати моделей на наборі даних MS COCO, піднабір test-dev, на мобільному пристрої Google Pixel. Результати порівняльних тестів наведено в таблиці 3.2.

Таблиця 3.2 – Порівняння моделей на наборі даних MS COCO на смартфоні Google Pixel

Модель	mAP, %	latency, мс	FPS	Нпараметрів	Розмір, Мб
YOLOv3	55.3	~1000	~1	65.2M	234
MobileNetV2-1.0-224	28.4	73.8	13.5	3.4M	11.4
Tiny-YOLOv3 [35]	33.1	57.8	17.7	8.9M	32.8
LightDenseYOLO [36]	31.8	50	20	5.7M	20.1
EfficientDet-Lite1	31.5	49	20.4	4.2M	5.8
EfficientDet-Lite0 [37]	26.4	36	27.8	3.2M	4.4

Порівняння включає в себе повнорозмірну модель YOLOv3, що демонструє чудові результати в точності, але також потребує значних обчислювальних витрат і зовсім не придатна для розпізнавання в реальному

часі із пристроїв із обмеженими ресурсами. В той же час, представлені модифікації на базі архітектури YOLO, орієнтовані на швидкість та легкість, показують значно кращі результати за швидкістю та значне зменшення кількості параметрів, за певних втрат у точності, що однак не є критичними й залишають точність на цілком прийнятному для практичного використання рівні.

Варті уваги й надлегкі моделі, що використовують форм-фактори для зменшення розмірів та пришвидшення роботи при адаптації їх до різних практичних задач та доступних ресурсів. За менш ніж 5 мільйонів параметрів, такі моделі як MobileNetV2 та EfficientDet-Lite демонструють mAP більше 25%.

Задача забезпечення розпізнавання об'єктів глибокою згортковою мережею в реальному часі на мобільних пристроях є дуже складною й більшість сучасних моделей може надати лише відчуття реального часу на швидкості до 20 кадрів на секунду, проте пройти поріг реального часу вдається EfficientDet-Lite0 із показником в 27.8 кадрів на секунду.

### 3.5 Побудована модель та результати експериментів

#### 3.5.1 Зниження шуму

В реальних умовах рівень зашумленості вхідного зображення може значно перевищувати той, що представлений в наборі даних. На якість сигналу та зображення може впливати як якість з'єднання, так і застосування в зоні використання дрона засобів радіо-електронної боротьби (РЕБ), що можуть створювати значний білий шум та впливати на якість розпізнавання з камери БПЛА або іншої техніки за умови проведення розпізнавання на робочій станції оператора.

Традиційним рішенням для усунення шумів із вхідного зображення є використання знешумлюючого автокодувальника. Такий згортковий автокодувальник складається із двох сегментів – кодувальника та декодувальника та навчається виділяти основні ознаки вхідного зображення та реконструювати його у знешумленому вигляді.

Традиційно, архітектура згорткових автокодувальників для цієї задачі характеризується відносно невеликою глибиною, що спричинено необхідністю відбудови вхідного зображення із зниженої розмірності після операції згортки. Застосування ж залишкових з'єднань за більшої глибини мережі [38] надає високу точність за можливості використання зображень більшої роздільної здатності, однак в умовах розпізнавання з камери дронів за роздільної здатності подекуди нижче за 270 на 270 пікселів застосування заглибокої мережі може призводити до втрати кольорів та значної частини значимих пікселів та, відповідно, ознак для розпізнавання, що подекуди у випадках віддалених об'єктів може означати повну втрату об'єкту з поля зору камери. Окрім цього, чимало досліджень [39] показали ефективність застосування мереж із меншою кількістю шарів та більшою кількістю карт ознак і, відповідно, згорткових фільтрів в межах одного шару, а також ефективність використання пакетної нормалізації після початкових згорткових шарів. Деякі дослідження також вказують на ефективність використання при навчанні додатково зашумлених даних, що також буде ефективним для підготовки мережі до роботи у складніших умовах ніж у наведених в наборі даних випадках, здебільшого включаючих лише шуми від природної низької якості сигналу або камери.

Послугуючись цим, а також в результаті експериментів на власних даних із додаванням нормально розподіленого шуму було отримано оптимальну архітектуру згорткового автокодувальника із 12 шарами.

Архітектуру побудованої мережі наведено на рисунку 3.3.



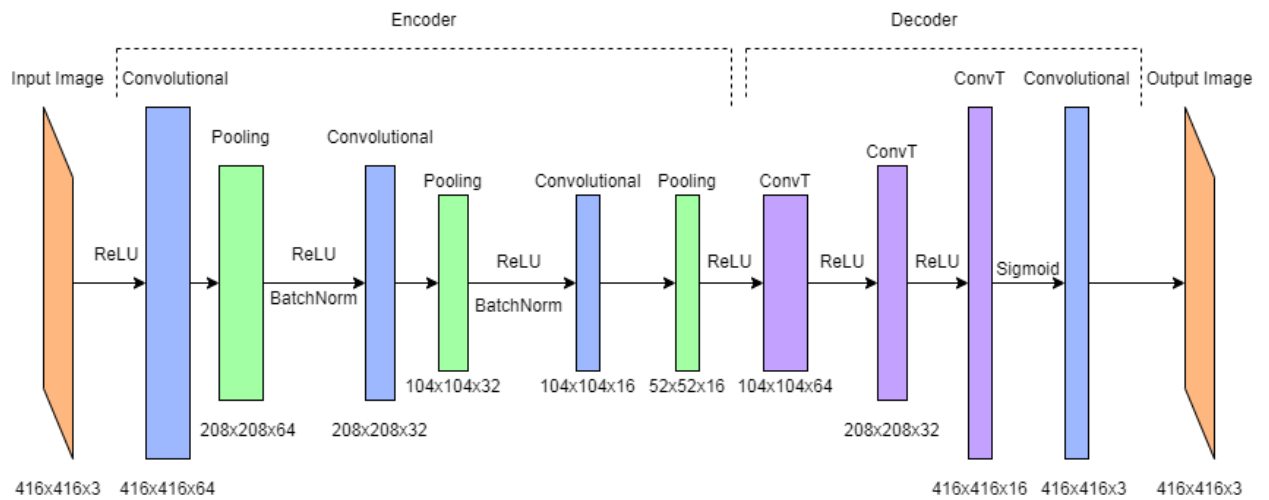


Рисунок 3.3 – Архітектура автокодувальника для зниження шуму на зображенні

Кодування здійснюється трьома згортковими шарами із трьома шарами максимального агрегування, поміж шарами також використовується пакетна нормалізація для пошвидшення та підвищення стабільності мережі. Декодувальна частина мережі включає в себе 3 транспоновані згорткові шари, що навчаються правильному збільшенню розмірності зображення до розмірів вхідного та одного згорткового шару для отримання вихідного 3-канального кольорового зображення.

### 3.5.2 Квантування

Для додаткового зменшення розмірів моделі та прискорення її роботи на пристроях із обмеженими обчислювальними ресурсами, як на центральних, так і на графічних та тензорних процесорах було використано технологію квантування моделі.

В роботі використано повне integer-квантування із 32-бітних чисел із плаваючою точкою (float32) у цілі 8-бітні числа (int8), що передбачає калібрування значень ваг, зміщень, вхідних змінних та входів прихованих

шарів і дозволяє зменшити модель до 4 разів, збільшивши швидкість роботи до 3 разів на центральних та тензорних процесорах та мікроконтролерах, трохи менше пошвидшуючи модель на графічному процесорі.

Рисунок 3.4 демонструє процес перенесення значень із простору чисел із плаваючою точкою у простір цілих чисел.

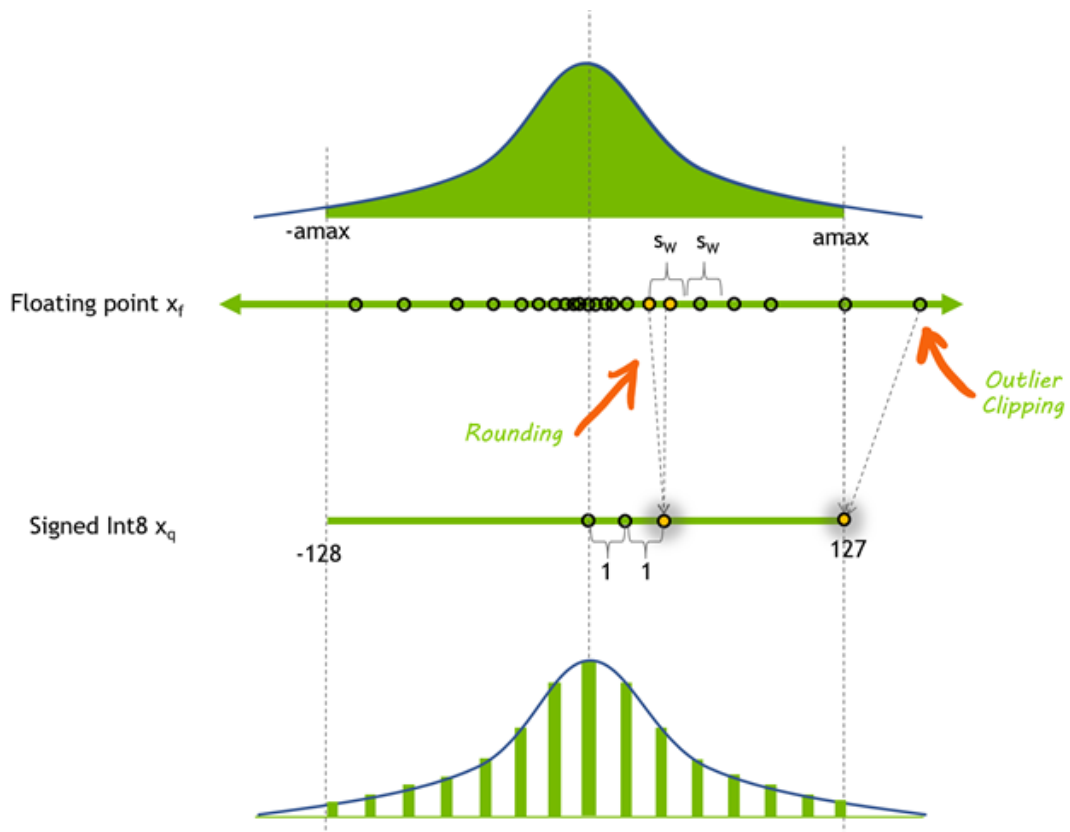


Рисунок 3.4 – Квантування значень в моделі з чисел із плаваючою точкою у цілі числа

### 3.5.3 Архітектура моделі

Для проведення розпізнавання було використано модель, що складається із 3 основних частин та послуговується принципом проведення розпізнавання в один крок.

Оснoву моделі (backbone) складає мережа виділення ознак із вхідного зображення EfficientNet-Lite, що використовує «вузькі» блоки (bottlenecks), із згортковими шарами 1 на 1 та шарами глибокої згортки 3 на 3 та 5 на 5. Глибока згортка представляє собою поділ вхідного сигналу на канали й проведення згортки відносно кожного каналу із їх об'єднанням для отримання виходу шару, замість традиційного проведення згортки відносно усіх каналів одночасно. Ці «вузькі» блоки використовуються в поєднанні із спеціальними оберненими залишковими блоками, за аналогією із MobileNetV2 [40], що дозволяють пропускати та з'єднують повноцінні «вузькі» блоки, замість окремих залишкових шарів. Їх використання дозволяє значно збільшити швидкість та адаптовність мережі. Приклад такого інвертованого залишкового блоку наведено на рисунку 3.5.

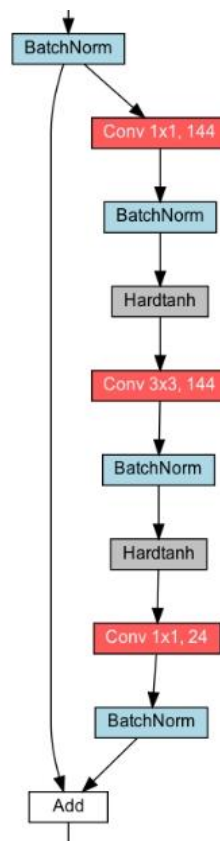


Рисунок 3.5 – Інвертований залишковий блок

Процес виділення ознак в моделі та масштабування вхідного зображення схематично наведено на рисунку 3.6.

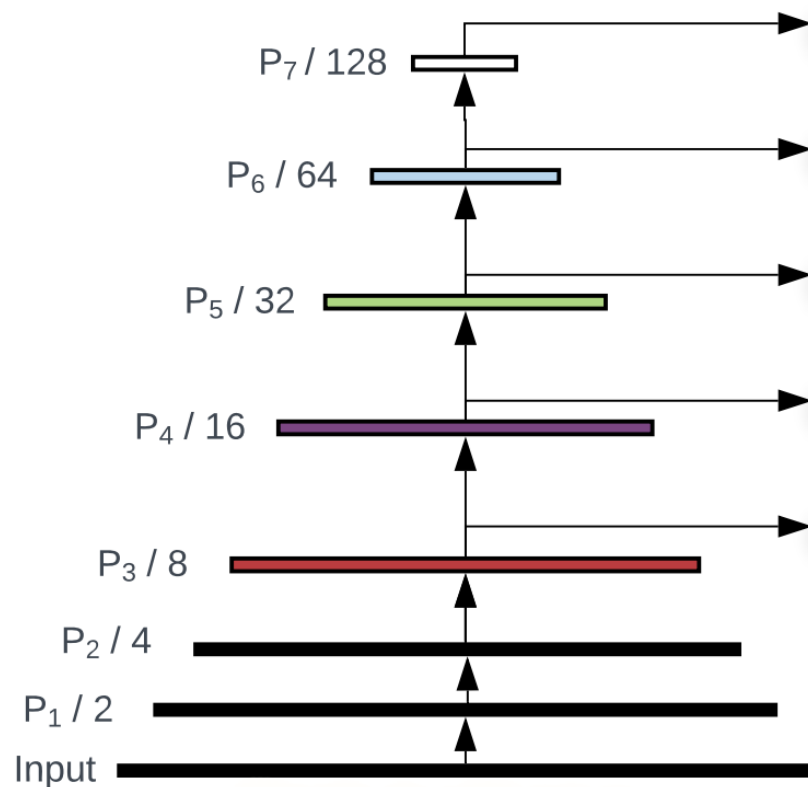


Рисунок 3.6 – Мережа виділення ознак EfficientNet

Виділені ознаки подаються до мережі масштабування та об'єднання ознак. Для цього використовується поліпшена версія класичної мережі пірамід ознак (Feature Pyramid Network, FPN), що називається двонаправленою FPN (Bidirectional FPN, Bi-FPN). Вона приймає на вхід виділені ознаки з 3 до 7 рівня масштабування із мережі виділення ознак та проводить їх масштабування між рівнями в двох напрямках. В порівнянні із класичною FPN, що збирає та масштабує ознаки згори-вниз, Bi-FPN не лише проводить процес в дві сторони, але також використовує систему пропускних між-масштабних блоків для прискорення проходження ознак до другого етапу (зменшення масштабу). Один набір таких двонаправлених з'єднань між рівнями з 3 по 7 зазвичай називається окремим шаром Bi-FPN.

В моделі використовується повноцінна секція із трьох шарів (блоків) Bi-FPN наведена на рисунку 3.7.

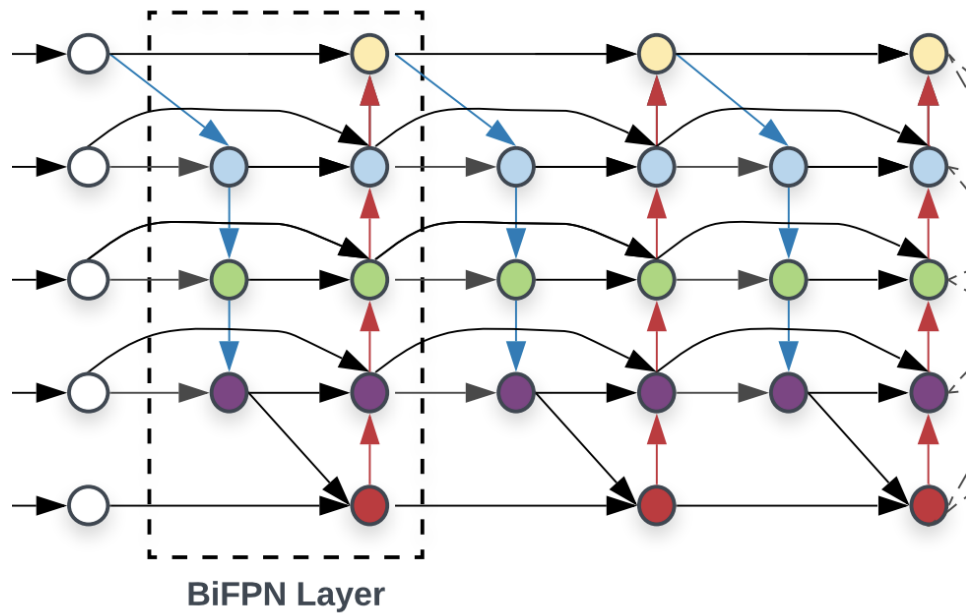


Рисунок 3.7 – Блоки Vi-FPN

Окрім цього, оскільки різні вхідні ознаки мають різні масштаби від початку, вони мають різний вплив на вихідні масштабовані та об'єднані ознаки. Тому мережа також навчається встановлювати ваги для кожної вхідної ознаки, що збільшує точність та допомагає їй краще адаптуватися до змін масштабів об'єктів. Мережа Vi-FPN використовує швидке нормалізоване об'єднання зважених ознак замість класичного softmax, через нестабільність в мережі та додаткову обчислювальну складність останнього [37]. Натомість, використовується нормалізація ваг та ReLU, а вихідні об'єднання ознак обраховуються за формулою

$$O = \sum_i \frac{w_i}{\epsilon + \sum_j w_j} \cdot I_i, \quad (4.1)$$

де  $w_i, w_j$  – відповідні ваги ознак,  $I_i$  – вхідна ознака,  $\epsilon = 0.0001$  – мала величина, що дозволяє уникнути чисельної нестабільності.

Такий підхід дозволяє прискорити виконання операції об'єднання до 30% на графічному процесорі за мінімальних втрат у точності (близько 0.1-0.01% AP).

Для додаткового підвищення ефективності тут також використовується глибока згортка, на відміну від звичайної, при об'єднанні ознак у традиційній FPN.

Таким чином, для проміжного, шостого, рівня ознак

$$P_6^{out} = DConv \left( \frac{w_1' \cdot P_6^{in} + w_2' \cdot P_6^{td} + w_3' \cdot Resize(P_5^{out})}{w_1' + w_2' + w_3' + \epsilon} \right), \quad (4.2)$$

де  $P_6^{out}$  – вихідна ознака для 6 рівня на шляху знизу-вгору,  $P_6^{td}$  – проміжна ознака для 6 рівня, що обраховується як

$$P_6^{td} = DConv \left( \frac{w_1 \cdot P_6^{in} + w_2 \cdot Resize(P_7^{in})}{w_1 + w_2 + \epsilon} \right) \quad (4.3)$$

Після отримання масштабованих та об'єднаних ознак виконується отримання прогнозів щодо околів об'єктів на зображенні та їх класів. Для цього використовуються блоки так званої Class/Vox мережі, яка виконує задачі класифікації та регресії за допомогою згорткових шарів. Ваги в мережі спільні для всіх рівнів ознак, за аналогією з RetinaNet [41]. Кількість фільтрів та блоків відповідає їх кількості в Vi-FPN.

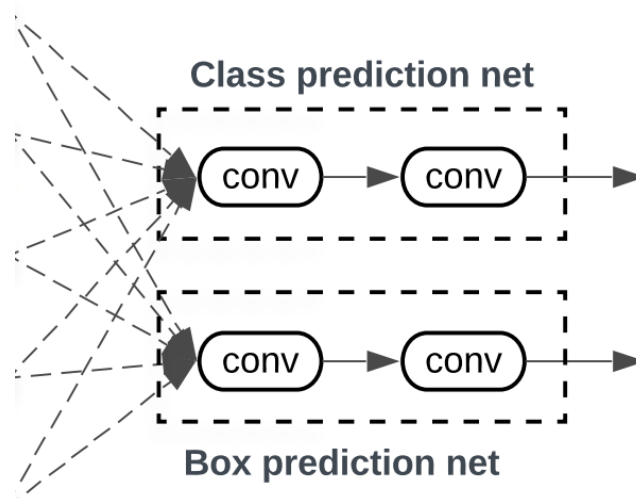


Рисунок 3.8 – Блоки класифікації та регресійного прогнозування околів

Розмірність зображення на вході змінюється до 320 на 320 пікселів для оптимального проходження мережі масштабування ознак Vi-FPN. Загалом, мережа налічує 276 шарів та має 3.25 мільйонів тренуваних параметрів. Вага моделі після квантування складає всього 5.4 мегабайти.

Побудована модель для розпізнавання, за рішенням оператора у випадку виявлення шумів як наприклад при застосуванні супротивником засобів РЕБ, може працювати у поєднанні із автокодувальником для зниження шуму на вхідному зображенні.

Узагальнену схему роботи системи наведено на рисунку 3.9.

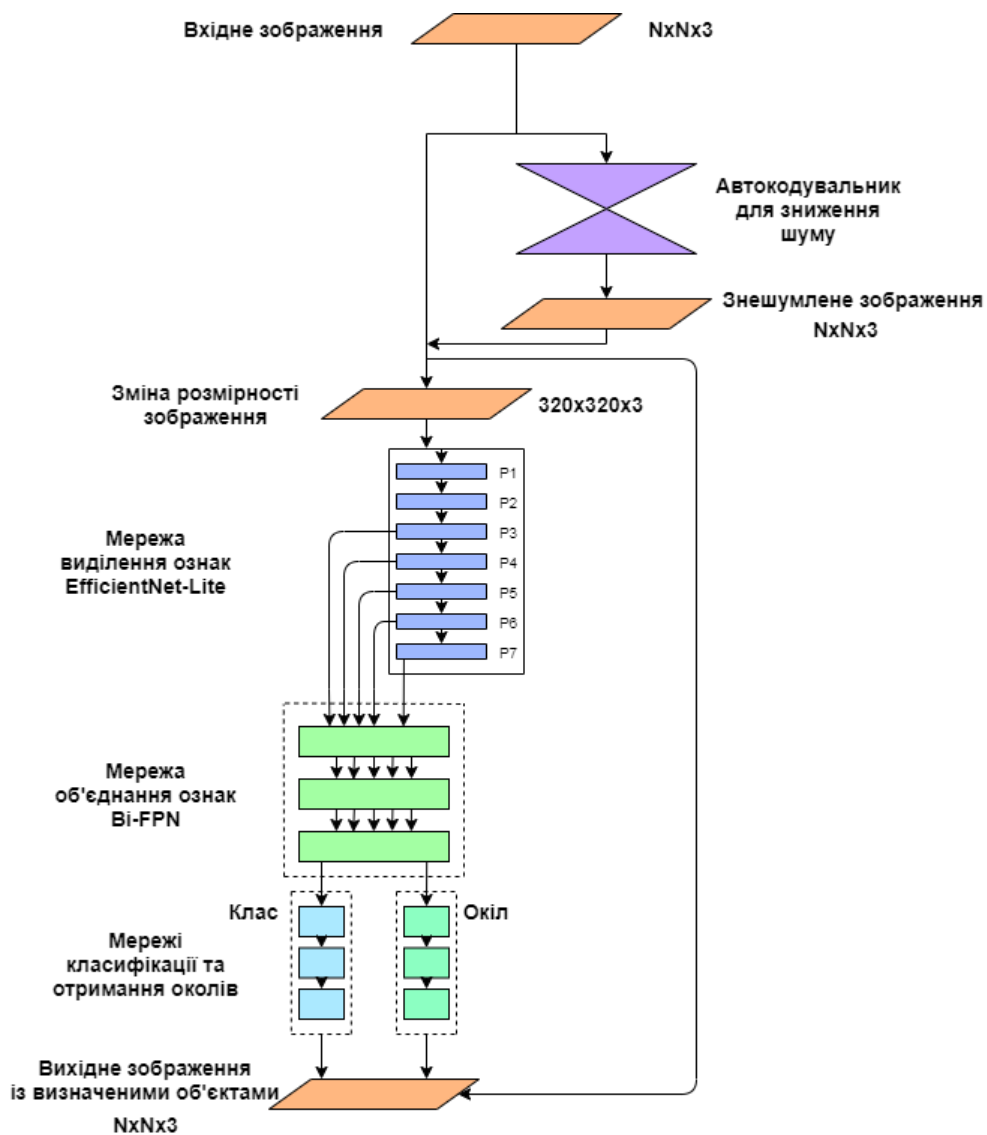


Рисунок 3.9 – Схема побудованої системи

### 3.5.4 Результати експериментів на власному наборі даних

Для початку наведемо результати роботи автокодувальника для зниження шуму на додатково замшумлених зображеннях із власного набору даних. Основними метриками порівняння виступатимуть точність (MSE) та швидкість обробки кадру. Враховано було також відповідність результуючого зображення вхідному, такі характеристики як збереження кольорів та якості, основних ознак для проведення подальшого розпізнавання. Основною відмінністю наведених архітектур є кількість згорткових та відповідних транспонованих згорткових шарів для кодувальника та декодувальника. Результати роботи для кількох різних варіантів архітектури наведені в таблиці 3.3.

Таблиця 3.3 – Результати роботи автокодувальника для зниження шуму

Архітектура (Conv/ConvT)	MSE, %	Час обробки кадру, мс
2x2	1.11	8.78
3x3	0.67	10.76
4x4	0.61	20.6

Варто відзначити, що за збільшення глибини мережі вище за 3 кроки масштабування в обидва боки спостерігається серйозна втрата в збереженні кольорів та якості зображення, що може негативно впливати на якість розпізнавання.

За результатами експериментів було обрано архітектуру із 3 шарами згортки та транспонованої згортки в кодувальнику та декодувальнику відповідно як оптимальну у поєднанні точності та швидкості, а також виходячи із потреби в збереженні ключових ознак об'єктів.



При виборі базової моделі та прийнятті рішення про застосування квантування для використання моделі на центральних та графічних процесорах таких робочих станцій як ноутбук середньої потужності було проведено експерименти для порівняння точності та швидкості моделей EfficientDet із низькими форм-факторами. Для відповідності типовим обмеженням у ресурсах було враховано також обсяг використовуваної пам'яті графічного процесора. Результати експериментів наведені у таблиці 3.4.

Таблиця 3.4 – Результати роботи моделей на CPU та GPU із використанням квантування

Модель	mAP, %	AP50, %	AP <sub>v.t.</sub> , %	latency <sub>(GPU)</sub> , мс	latency <sub>(CPU)</sub> , мс	MemUsage, ГБ
EfficientDet-Lite1	26.6	45.0	42.3	30.0	50.5	3.88
EfficientDet-Lite1 (quantized)	26.4	44.7	41.9	28.75	36.95	3.58
EfficientDet-Lite0	22.4	44.0	42.0	23.125	41.85	3.56
EfficientDet-Lite0 (quantized)	22.3	42.8	41.7	22.8	30.2	3.09

За результатами експериментів було встановлено доречність використання квантування, що не призводить до значних змін в точності мережі, однак пришвидшує її роботу, що особливо добре проявляється на центральних та тензорних процесорах.

В таблиці 3.5 наведено результати випробувань побудованої системи в повному варіанті, враховуючи умови наявності шуму у вхідному сигналі.

Таблиця 3.5 – Результати роботи моделі за наявності шуму

Модель	mAP	AP50	APв.т.	latency <sub>(GPU)</sub> , мс	FPS <sub>(GPU)</sub>	latency <sub>(CPU)</sub> , мс	FPS <sub>(CPU)</sub>
без шуму	22.3	42.8	41.7	22.8	43.86	30.2	33.11
за наявності шуму	14.7	28.3	31.5	22.8	43.86	30.2	33.11
з використанням знешумлення	20.8	39.8	40.8	33.7	29.67	43.6	22.9

### 3.6 Аналіз результатів експериментів

Результати проведених експериментів встановили ефективність та доречність використання цілочисельного квантування для компактизації та прискорення моделі. Квантування дозволило відчутно прискорити модель на центральному процесорі за відносно незначних втрат у точності в усього кілька десятих відсотка усередненої по класам AP[0.5;0.95]. Варто відзначити, що воно також дозволяє прискорити модель для використання на тензорних процесорах та мікроконтролерах і хоча й має значно менший ефект для пристроїв із графічним процесором все ж може бути на них доречним через зменшення розмірів і вкарй низькі втрати в точності розпізнавання.

Основна складова системи – модель для розпізнавання об'єктів мала дуже високі для легкої та компактної моделі показники точності, в тому числі з середньою точністю AP[0.5;0.95] вище 40% для пріоритетного класу військової техніки, а також вище 40% за показником AP[0.5], що є прийнятним показником для моделей орієнтованих для використання у складних умовах в якості допоміжних для оператора. За достатньої точності модель також мала дуже високу швидкість, із показником у 44 кадри на секунду для графічного процесора, використовуючи менше 4 гігабайт пам'яті процесору та 33 кадри

на секунду для центрального процесора. Попередній порівняльний аналіз також вказує на можливість використання моделі на мобільних пристроях за пропускної здатності у 27 кадрів на секунду, що робить її придатною для використання в реальному часі навіть на таких обмежених в обчислювальних ресурсах пристроях.

На рисунку 3.10 наведено результати застосування моделі на деяких тестових зображеннях із вхідного набору даних, що демонструють типові фактори ускладнюючі розпізнавання.



Рисунок 3.10 – Візуальні результати роботи моделі на тестових зображеннях

Модель показала готовність до використання в реальних умовах та здатність до розпізнавання за складних погодних умов, перекриття об'єктів,

низької якості вхідного зображення, складних сцен із великими скупченнями об'єктів та використання простого маскуванню як природні перекриття частини об'єкту та маскувальні сітки.

Результати роботи розробленого автокодувальника для зниження шуму на вхідному зображенні показали досить високу точність збереження по-піксельних значень в усіх каналах зображення по метриці середньоквадратичної похибки (MSE). Окрім цього, було отримано модель збалансовану в швидкості, точності та таку, що мала б мінімальний вплив на якість та кольори вхідного зображення, що значно поліпшує можливість проведення подальшого розпізнавання на основі знешумленого зображення. За використання автокодувальника у повній версії системи у поєднанні із моделлю для розпізнавання об'єктів було досягнуто розпізнавання в реальному часі на графічному процесорі та дуже наближене відчуття реального часу за майже 23 кадрів на секунду на центральному процесорі. При цьому, точність розпізнавання в моделі зазнала досить незначних втрат у кілька відсотків по метриці усередненої по класам AP[0.5;0.95], що також доводить доречність його використання для реальних умов у порівнянні із відсутністю додаткових методів зниження шуму, що значно погіршує розпізнавання.

Результати роботи моделі для розпізнавання об'єктів на додатково зашумлених вхідних зображеннях, знешумлених за допомогою автокодувальника наведено на рисунку 3.11.



Рисунок 3.11 – Візуальні результати роботи моделі у поєднанні із автокодувальником на зашумлених вхідних даних

Таким чином, система придатна для використання на ноутбуках, планшетах та мобільних пристроях в реальному часі за низького шуму й за рішенням оператора про відсутність необхідності у використанні додаткового модуля для його зниження. Система також придатна для використання на інтегрованих системах та із застосуванням портативних тензорних процесорів EdgeTPU, ефективно використовуючи їх потужності для моделей машинного

навчання. За необхідності використання модулю зниження шуму за рішенням оператора дрону система здатна зберігати високу точність та наближену до реального часу швидкість у 23 кадри на секунду.

### 3.7 Висновки до розділу 3

У даному розділі було проведено порівняльний аналіз платформ для розробки моделі та варіантів базової моделі для проведення розпізнавання, варіантів архітектурних рішень.

Було обрано оптимальну архітектуру моделі згорткового автокодувальника для зниження шуму на вхідному зображенні та наведено результати відповідних експериментів.

Було встановлено ефективність застосування квантування для пришвидшення моделі та здатності моделі до проведення розпізнавання на реальних даних із достатньою точністю в реальному часі та за наближення до реального часу при застосуванні у поєднанні із знешумлюючим автокодувальником. Модель також показала готовність до роботи за умов дії факторів, ускладнюючих розпізнавання.

#### 4 РОЗРОБКА СТАРТАП ПРОЄКТУ

У наш час у світі існує високий попит на системи розпізнавання об'єктів у багатьох сферах людського життя й одією із найбільш фінансованих сфер була й залишається військова. Велика кількість компаній займається розробкою систем комп'ютерного зору та обробки інформації для військових компаній приватного та державного сектору.

Попри це, в сфері систем розпізнавання об'єктів для дронів найбільше рішень орієнтовані на великі дрони, оперативно-тактичного рівня, що нескладно оснастити необхідною апаратурою або ж використовувати ці системи із командних пунктів. В той же час, із значним розширенням використання легких дронів для ведення розвідки та виконання бойових задач існує потреба у системі, що була б здатна до проведення розпізнавання об'єктів для допомоги оператору на мобільних робочих станціях у польових умовах, за умов значних обмежень у обчислювальних ресурсах [42][43]. З цього випливає актуальність розробленого стартап-проєкту.

#### 4.1 План розробки стартапу

План розробки стартапу включає в себе проведення маркетингового аналізу, зокрема:

- конкурентний аналіз;
- формування ідеї самого проєкту та виділення цільової аудиторії;
- розробка стратегії виведення товару на ринок, базуючись на аналізі ринкового середовища.

Після проведення маркетингового аналізу, проводиться організація самого стартап-проєкту, зокрема:

- складання плану розробки та запуску продукту;
- планування обсягу виробництва та оцінка потенційного обсягу ресурсу, необхідного для виконання плану;
- розрахунок витрат, необхідних для реалізації проєкту, та витрат на запуск.

Далі проводиться фінансово-економічний аналіз стартап-проєкту та оцінка ризиків. В межах цього етапу:

- визначається обсяг інвестиційних витрат;
- розраховуються основні фінансово-економічні показники проєкту та показники інвестиційної привабливості проєкту;
- визначаються основні ризики та способи їх запобігання.

Останнім кроком є планування заходів із комерціалізації та масштабування, що включає в себе:

- проведення дослідження на предмет інтересів потенційних інвесторів та бізнесів;
- складання інвестиційної пропозиції, яка включає в себе як опис самого продукту, так і можливі шляхи розширення та розвитку;
- обрати канали комунікації із потенційно зацікавленими сторонами.



Далі в розділі наведено результати виконання кожного з цих кроків.

#### 4.2 Опис ідеї стартап-проєкту

В межах пункту було проаналізовано і подано у вигляді таблиць:

- зміст ідеї;
- можливі напрямки застосування;
- можливі напрямки застосування;
- основні вигоди, що може отримати користувач товару;
- відмінності від існуючих аналогів та замінників.

У таблиці 4.1 наведений опис ідеї стартапу.

Таблиця 4.1 – Опис ідеї стартапу

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Розроблена система здатна проводити розпізнавання військових об'єктів із кадрів відеоряду з камери дронів у реальному часі за обмежених обчислювальних ресурсів	Допомога оператору у веденні розвідки або патрулюванні	Значне збільшення швидкості реагування та кількості виявлених об'єктів інтересу, протидія шуму та іншим поскладнюючим виявлення факторам
	Допомога у наведенні на цілі	Протидія шуму та іншим поскладнюючим виявлення факторам, висока точність та допомога у швидкому реагуванні та виявленні цілей
	Допомога у пошуку поранених	Протидія шуму та іншим поскладнюючим виявлення факторам, допомога у виявленні на великих відстанях

Проведемо порівняльний аналіз сильних (S), слабких (W) та нейтральних (N) сторін проєкту та конкурентів і наведемо результати у таблиці 4.2.

Таблиця 4.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Технікоекономічні характеристики ідеї	(потенційні) товари/концепції конкуrentів			W	N	S
		Мій проект	Matrixspace	TheiaTech			
1.	Точність	Середня	Висока	Висока		+	
2.	Собівартість	Низька	Висока	Висока			+
3.	Портованість	Висока	Низька	Низька			+

#### 4.3 Технологічний аудит ідеї проекту

Проаналізуємо технологічну здійсненність проекту та наведемо результати в таблиці 4.3.

Таблиця 4.3 – Технологічна здійсненність ідеї проєкту

№ п/п	Ідея проєкту	Технології і реалізації	Наявність технологій	Доступність технологій
1	Створення системи розпізнавання об'єктів із кадрів відеоряду з камери БПЛА	Використання мови програмування Python	Наявні	Доступні
2		Використання мови програмування C#	Не наявні, необхідні допрацювання	Доступні
3		Використання мови програмування Java	Наявні, необхідні допрацювання	Доступні
Обрана технологія реалізації ідеї проєкту: Python				

За результатами аналізу було зроблено висновок, що проєкт є технологічно здійсненним. Технологічним засобом реалізації проєкту було обрано мову програмування Python через доступність, багатоплатформність та широкий функціонал для розробки систем машинного навчання.

#### 4.4 Аналіз ринкових можливостей запуску стартап-проєкту

Далі проведемо попередній аналіз ринку для запуску стартап-проєкту (таблиця 4.4).

Таблиця 4.4 – Попередня характеристика потенційного ринку стартап-проєкту

№ п/п	Показники ринку (найменування)	Характеристика
1	Кількість головних гравців, од	30
2	Загальний обсяг продаж, грн/ум.од	1000000
3	Динаміка ринку (якісна оцінка)	Стрімко зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Відсутні
5	Специфічні вимоги до стандартизації та сертифікації	Відсутні
6	Середня норма рентабельності в галузі (або по ринку), %	30%

Проведемо характеристику потенційних клієнтів, які можуть бути зацікавлені в проєкті (таблиця 4.5).

Таблиця 4.5 – Характеристика потенційних клієнтів стартап-проєкту

№ п/п	Потреби, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Програмне забезпечення робочих станцій операторів	Держава та державні компанії оборонного сектору	Цікавить надійність та простота впровадження у існуючі системи	Портованість та низька собівартість впровадження
2	Покращення систем розпізнавання об'єктів для БПЛА та програмне забезпечення робочих станцій	Приватні компанії	Цікавить простота адаптування під різні задачі та умови	Висока точність та швидкість роботи моделі, портованість
3	Допомога у виконанні бойових задач	Кінцевий користувач, оператор БПЛА	Потреба у зручності та швидкості роботи системи	Робота в реальному часі, надійність

Проведемо аналіз ринкового середовища. Для цього наведемо фактори, що перешкоджають ринковому впровадженню проєкту (таблиця 4.6), та фактори, що йому сприяють (таблиця 4.7).

Таблиця 4.6 – Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Конкуренція	Попри відкритість ринку до нових проєктів, багато найбільших гравців мають певний спектр постійних клієнтів	Знайти точки додаткової цінності та свою клієнтську нішу
2	Собівартість й доцільність впровадження	Конкуренти можуть коштувати менше або ж надавати альтернативні сервіси	Сфокусуватися на точності моделей та швидкодії, відповідним чином побудувати маркетингову стратегію
3	Адаптованість до різних задач	При потребі у використанні моделі у інших сферах застосування може виникнути необхідність збору даних, перенавчання та/або зміни архітектури моделі у відповідності до задач	Мати достатній штаб і ресурси, для побудови різних моделей для різних предметних областей, налагодити збір даних

Таблиця 4.7 – Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Універсальність	Можливість використання продукту на різних платформах та робочих станціях	Зробити акцент при маркетингу
2	Простота у впровадженні	Використання системи потребує лише завантаження платформи та може бути додане до інтегрованої системи у вигляді модуля із тензорним процесором	Зробити акценти при маркетингу, розвивати співпрацю із розробниками БПЛА, робочих станцій та інших супутніх систем для розробки готових рішень
3	Надійність та гарантії	Забезпечувати надійність системи та сервісну підтримку	Пропонувати моделі з найкращими результатами, проводити польові випробування, а також надавати усю необхідну технічну підтримку

Результати аналізу пропозиції, визначення загальних рис конкуренції на ринку наведені у таблиці 4.8.



Таблиця 4.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	У чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною )
1. Тип конкуренції: чиста конкуренція	Представлена велика кількість компаній та продуктів	Знайти точки додаткової цінності та свою клієнтську нішу
2. За рівнем конкурентної боротьби: міжнародний	Розроблені рішення можуть бути доступні у всьому світі	Розширити цільову аудиторію, розробити інтерфейс на різних мовах
3. За галузевою ознакою: внутрішньогалузева	Продукт здебільшого використовуватиметься в межах галузі	Вдосконалення загальних якостей продукту
4. Конкуренція за видами товарів: товарно-видова	Конкуренція з іншими рішеннями та продуктами	Підтримувати та покращувати якість та функціонал
5. За характером конкурентних переваг: нецінова	Різні компанії пропонують різну якість	Розробляти якісніші алгоритми і моделі
6. За інтенсивністю: немарочна	Якість системи часто важливіша за бренд	Робити акцент маркетингу на якості продукції

Детальніший аналіз умов конкуренції в галузі за моделлю Майкла Портера наведено в таблиці 4.9.

Таблиця 4.9 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти у галузі	Потенційні конкуренти	Постачальники	Клієнти	Товарозамінники
	Matrixpace, TheiaTech, Google AI, AlwaysAI, Brodmann17	Якість, капіталовкладення	–	Контроль якості	–
Висновки	Конкуренція з невеликою інтенсивністю, ринок відкритий для нових проєктів	Існують можливості входу на ринок для нових потенційних конкурентів протягом пів року	Постачальники майже відсутні	Клієнти мають вплив на умови роботи на ринку визначаючи основні напрями застосування систем	Товари заміники відсутні

З огляду на конкурентну ситуацію, робота на ринку є цілком можливою. Ринок відкритий для нових проєктів, а сильними сторонами у конкурентній боротьбі можуть стати точність та надійність моделей, їх швидкість а також можливість застосування на різних платформах.

За наведеними результатами, проведемо обґрунтування факторів конкурентоспроможності проєкту (таблиця 4.10).

Таблиця 4.10 – Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування
1	Універсальність	Можливість використання продукту на різних платформах
2	Надійність	Доведена готовність до використання в польових умовах, підтримка та покращення
3	Якість моделей	Включає такі фактори як точність та швидкість роботи, додатковий функціонал, що є запоруками конкурентоспроможності
4	Собівартість впровадження	Можливість швидкого введення в експлуатацію як в нових так і із існуючими системами

Проведемо аналіз сильних та слабких сторін стартап-проєкту (таблиця 4.11).

Таблиця 4.11 – Порівняльний аналіз сильних та слабких сторін системи

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів							
			-3	-2	-1	0	1	2	3	
1	Універсальність	20	+							
2	Надійність	12				+				
3	Якість моделей	17				+				
4	Собівартість впровадження	16		+						

Далі проведемо SWOT-аналіз проєкту (таблиця 4.12).

Таблиця 4.12 – SWOT-аналіз стартап-проєкту

Сильні сторони	Слабкі сторони
Універсальність Швидкість роботи Простота та доступність впровадження у використання	Не сформована база клієнтів Невисокі капіталовкладення Відсутність польових випробувань
Можливості	Загрози
Покращення моделей Адаптація під конкретні способи застосування Співпраця із виробниками БПЛА та іншого військового ПЗ	Якісна конкуренція, стрімкий розвиток ринку Неготовність до масштабування виробничого процесу

На основі SWOT-аналізу було розроблено та проаналізовано альтернативи ринкової поведінки для виведення стартап-проекту на ринок, наведені у таблиці 4.13.

Таблиця 4.13 – Альтернативи ринкового впровадження стартап проекту

№ п/п	Альтернатива ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Початок розповсюдження ПЗ на волонтерських засадах	50%	4 місяці
2	Проведення маркетингу та пошук клієнтів, участь в тендерах	70%	6 місяців
3	Співпраця із розробниками іншого військового ПЗ та БПЛА	80%	8 місяці

За результатами аналізу було обрано проводити маркетингову кампанію та шукати клієнтів для підписання контрактів на розробку та впровадження систем.

#### 4.5 Розроблення ринкової стратегії проекту

Першим кроком для розробки ринкової стратегії є опис цільових груп споживачів (таблиця 4.14).

Таблиця 4.14 – Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит у межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Державний оборонний сектор	Середня	25%	Середня	Середня
2	Приватні компанії	Висока	30%	Висока	Середня
3	Кінцеві користувачі	Середня	20%	Середня	Висока
Обрані цільові групи: 1, 2, 3					

В таблиці 4.15 сформовано базову стратегію розвитку проєкту.

Таблиця 4.15 – Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проєкту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	2	Масового маркетингу	Якість та універсальність моделей	Стратегія диференціації

Наступним кроком було обрано стратегію конкурентної поведінки (таблиця 4.16).

Таблиця 4.16 – Визначення базової стратегії конкурентної поведінки

Чи є проєкт «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
Ні	Компанія буде шукати нових споживачів та забирати існуючих у конкурентів	Буде створювати власні унікальні пропозиції, зважаючи на особливості конкурентів	Стратегія наслідування лідеру

За наведеними результатами, сформуємо стратегію позиціонування проєкту (таблиця 4.17).

Таблиця 4.17 – Визначення стратегії позиціонування

Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проєкту	Вибір асоціацій, які мають сформувати комплексну позицію власного проєкту
Універсальність Надійність Точність та швидкість в роботі Простота та низька собівартість впровадження	Стратегія диференціації	Універсальність Точність Швидкість Портованість Простота впровадження	Проста у впровадженні система Точна робота в реальному часі Універсальна система для різних платформ

#### 4.6 Розроблення маркетингової програми стартап-проєкту

Підсумуємо результати проведеного комплексного аналізу та визначимо ключові переваги концепції потенційного товару (таблиця 4.18).



Таблиця 4.18 – Ключові переваги концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами
1	Універсальність	Портованість та широкий спектр застосувань	Можливість використання на різних платформах та для різних задач
2	Швидкість роботи	Система дозволяє проводити операції в реальному часі	Значно вища швидкість роботи за обмежених обчислювальних ресурсів
3	Точність	Система має досить високу точність	Постійне покращення моделей, збір та використання нових даних

Розроблену трирівневу маркетингову модель товару наведено у таблиці 4.19.

Таблиця 4.19 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Допомога оператору БПЛА у швидкому реагуванні та виявленні об'єктів на кадрах із камери дрону		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Універсальність.	1.Нм	1.Технологічна
	2. Точність та швидкість.	2.Нм	2.Технологічна
	3. Низька собівартість впровадження	3.Нм	3.Економічна
	Якість: перевірка відповідності продуктів державним стандартм безпеки в країні клієнта		
Пакування: відсутнє			
Марка: HiveMind.AI, UAV-ODS			
III. Товар із підкріпленням	До продажу: допомога в освоєнні та користуванні системою		
	Після продажу: безперервна підтримка		
За рахунок чого потенційний товар буде захищено від копіювання: захист поєднання властивостей і характеристик товару			

Експертним методом було проведено аналіз меж встановлення ціни (таблиця 4.20).

Таблиця 4.20 – Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	2500\$	8000\$	Високий рівень доходів у перших двох груп, середній рівень у третьої цільової групи	2000-3000\$

Наступним кроком є визначення оптимальної системи збуту (таблиця 4.21).

Таблиця 4.21 – Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Канал нульового рівня	Продаж	0 (напрямую)	Власна
2	Збут через партнерські продукти	Продаж, підтримка споживача	1 (в поєднанні із товарами партнерів)	Горизонтальна

Останньою складовою маркетингової програми є розроблення концепції маркетингових комунікацій (таблиця 4.22).

Таблиця 4.22 – Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Пошук спеціалізованих систем для інтеграції в існуючі та власні нові системи	Публікація в спеціалізованих виданнях, журналах Спеціалізовані канали зв'язку Оголошення тендерів	Низька собівартість та Універсальність та портованість Висока точність та швидкість	Показати переваги рішення над конкурентами, виділити ключові особливості	Створення сайту продукту, розповсюдження інформації про продукт на спеціалізованих ресурсах
2	Пошук готових інтегрованих рішень	Спеціалізовані канали зв'язку Оголошення тендерів	Надійність Висока точність та швидкість Зручність у використанні операторами	Показати зручність та синергованість рішення із супутнім продуктом, ключові переваги	Використання спільних каналів комунікацій для висвітлення комплексного продукту із партнерами

Концепція товару передбачає систему для опомоги оператору БПЛА у швидкому реагуванні та виявленні об'єктів на кадрах із камери дрону.

Буде використана стратегія масового маркетингу та диференціації із адаптуванням під різні групи споживачів.

Основними способами збуту продукту буде розповсюдження інформації на спеціалізованих ресурсах, власному сайті в мережі Інтернет та через спільні канали із виробникам супутніх продуктів, а також шляхом участі в тендерах у державному секторі. Збут відбуватиметься як напрямую, так і через партнерські продукти.

#### 4.7 Висновки до розділу 4

В даному розділі проведено підготовчий аналіз для впровадження розробленої системи в якості стартап-проєкту.

У рамках розділу було досліджено конкурентні продукти, визначено сильні та слабкі сторони системи у порівнянні з ними, проведено аналіз ринку, що було визначено як сприятливий та відкритий для нових проєктів. Було визначено високий попит та стрімкий розвиток ринку. Також було проведено SWOT-аналіз та визначено технологічну та економічну доцільність впровадження стартап-проєкту.

Було проведено аналіз цільової аудиторії продукту та визначено ключові шляхи збуту та маркетингову стратегію продукту, що включає як прямий збут, так і програми партнерства із виробниками супутніх продуктів.

Проєкт було визначено як конкурентоспроможний, як з цінової так і з якісної точки зору із перевагами у швидкості роботи системи та універсальності її використання.

В результаті було встановлено доцільність подальшої імплементації проєкту.

## ВИСНОВКИ

У даній роботі було запропоновано нову модель для розпізнавання стратегічних технічних об'єктів на основі згорткових нейронних мереж.

Розпізнавання об'єктів у військовій сфері пов'язане із додатковими ускладнюючими факторами, такими як доступ до інформації, дані, низька якість зображення та інші перепони, протидія розпізнаванню, рух.

Традиційні моделі, що ґрунтуються на використанні одноетапних згорткових нейронних мереж та використовуються для вирішення цих задач, орієнтовані на використання на потужних робочих станціях та зазвичай не пристосовані для використання в польових умовах.

У даній дисертації була запропонована легка одноетапна модель, що орієнтована на використання у реальних бойових умовах на пристроях із обмеженими обчислювальними ресурсами.

Для прискорення моделі та зменшення розмірів було використано цілочисельне квантування та було розроблено згортковий автокодувальник для зниження шуму у вхідних даних.

Розроблена модель продемонструвала достатню точність та придатність до використання на пристроях із обмеженими обчислювальними ресурсами за значно більшої швидкості за аналогічні існуючі моделі, забезпечуючи розпізнавання в реальному часі на мобільних пристроях та наближення до реального часу за використання автокодувальника для зниження шуму. Модель також показала готовність до застосування в реальних умовах на реальних даних, включаючи такі ускладнюючі розпізнавання фактори як низька якість зображення, велика відстань до об'єктів, часткове перекриття, маскування.

Розроблена модель може використовуватись для допомоги оператору у веденні розвідки та патрулюванні для легких розвідувальних БПЛА, пошуку цілей та наведенні на них для БПЛА, що несуть бойовий заряд. Система також може бути використана для пошуку поранених на полі бою.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Krizhevsky A., Sutskever I., Hinton G. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*. 20XX. Vol. 60. P.
2. Liu Y., Zhang Y.-M., Zhang X.-Y., Liu C.-L. Adaptive spatial pooling for image classification. *Pattern Recognition*. 2016. Vol. 6. P. 58–67.
3. Simonyan K., Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *ICLR 2015: Proceedings of the 3rd International Conference on Learning Representations*. San Diego, California, USA, 7 May – 9 May, 2015.
4. Ren S., et al. Faster R-CNN: Towards real-time object detection with region proposal networks. *NIPS 2015: Proceedings of Advances in neural information processing systems*. Montreal, Quebec, Canada. 7 December – 12 December, 2015. P.91–99.
5. Wei Liu et. al. SSD: Single shot multibox detector. *ECCV 2016: In proceeding of European conference on computer vision*. Amsterdam, The Netherlands, 11 October – 14 October, 2016. P. 21–37.
6. Ismail R., Muthukumaraswamy S. Military Reconnaissance and Rescue Robot with Real-Time Object Detection. *Intelligent Manufacturing and Energy Sustainability. Smart Innovation, Systems and Technologies*. 2021. Vol. 213. P. 637–648.
7. Mohammed H.A.A. Multi-Scale Aircraft Detection from Satellite Images. *Erzincan University Journal of Science and Technology*. 2021. Vol. 14. P. 322–330.
8. DiRenzo N. Identifying Military Vehicles in Satellite Imagery. *Python in Plain English*. 2022.
9. Hiipala T. Recognizing military vehicles in social media images using deep learning. *ISI 2017: Proceedings of the 2017 IEEE International Conference on Intelligence and Security Informatic*, Beijing, China. July 22–24, 2017.

10. Hua X., et al. Military Object Real-Time Detection Technology Combined with Visual Saliency and Psychology. *Electronics*. 2018. Vol. 7. P. 216–241.
11. Diwani D., Chougule A., Mukhopadhyay D. Artificial Intelligence based Missile Guidance System. *SPIN 2020: Proceedings of 7th International Conference on Signal Processing and Integrated Networks*. Noida, India. 27 February – 28 February, 2020. P. 873–878.
12. J. Wang, S. Simeonova, and M. Shahbazi, “Orientation- and scaleinvariant multi-vehicle detection and tracking from unmanned aerial videos. *Remote Sens*. 2019. Vol. 11, No. 18.
13. Egozi A. Robotics: Israel Aerospace Industries (IAI) ground robotic systems. *Asia-Pacific Defence Reporter*. 2016. Vol. 42, No. 7. P. 46–47.
14. Svenmarck P., Luotsinen L., Nilsson M., Schubert J. Possibilities and challenges for artificial intelligence in military applications. In proceedings of the NATO Big Data and Artificial Intelligence for Military Decision Making Specialists’ Meeting. Bordeaux, France. 30 May –1 June, 2018. P. 1–16.
15. Gupta A., Gupta U. (2018, December). Military Surveillance with Deep Convolutional Neural Network. *IEEE 2018: In proceedings of 2018 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques*. USA. December, 14–15, 2018. P. 1147–1152.
16. Wilson N.A. Understanding the Battle for AI in Warfare through the Practices of Assemblage: A Case Study of Project Maven. *Utrecht University*. August 3, 2020.
17. Achkar R., Owayjan M. Implementation of a vision system for a landmine detecting robot using artificial neural network. *International Journal of Artificial Intelligence & Applications (IJAIA)*. 2012. Vol. 3, No. 5. P.73–92.
18. Surrisyad H., Wahyono. A Fast Military Object Recognition using Extreme Learning Approach on CNN. *International Journal of Advanced Computer Science and Applications(IJACSA)*. 2020. Vol. 11, No. 12. P.210–220.



19. Xia H., Xinqing W., Dong W., Jie H., Hu X. Military Object Real-Time Detection Technology Combined with Visual Saliency and Psychology. *Electronics*. 2018. Vol. 7, No. 216.
20. Wisniewski M, Rana Z. A, Petrunin I. Drone Model Classification Using Convolutional Neural Network Trained on Synthetic Data. *J Imaging*. 2022. Vol. 12, No 8.
21. Wu X., Li W., Hong D., Tao R., Du Q. Deep learning for unmanned aerial vehicle-based object detection and tracking: a survey. *IEEE Geoscience and Remote Sensing Magazine*. 2018. Vol. 10, No. 1. P.91–124.
22. Lin Tsun-Yiet. al. Microsoft COCO: Common objects in context. *ECCV 2014: Proceeding of the 13th European Conference on Computer Vision*. Zurich, Switzerland, September 6-12, 2014. P. 740–755.
23. Yang Z., Yu W., Liang P., Guo H., Xia L., Zhang F., Ma J. Deep transfer learning for military object recognition under small training set condition. *Neural Computing and Applications*. 2019. Vol. 31, No. 10. P.6469–6478.
24. Geiger A., Lenz P., Stiller C., Urtasun R. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*. 2013. Vol. 32, No. 11. P.1231–1237.
25. Zhu P., Wen L., Du D., Bian X., Ling H., Hu Q., Ma Z. VisDrone-VDT2018: The vision meets drone video detection and tracking challenge results. *ECCV Workshops 2018: In Proceedings of the European Conference on Computer Vision*.
26. Pajares G.. Overview and current status of remote sensing applications based on unmanned aerial vehicles (UAVs). *Photogramm. Eng. Remote Sens.* 2015. Vol. 81, No. 4. P.281–330.
27. Tankus A., Yeshurun Y. Convexity-based visual camouflage breaking. *Computer Vision and Image Understanding*. 2001. Vol. 82, No. 3. P.208–237.
28. Wise C., Plested J. Developing Imperceptible Adversarial Patches to Camouflage Military Assets From Computer Vision Enabled Technologies. (Preprint / arXiv:2202.08892). 2022.

29. Lee J., Wang J., Crandall D., Šabanović S., Fox G. Real-time, cloud-based object detection for unmanned aerial vehicles. *IRC 2017: Proceedings of the 1st IEEE International Conference on Robotic Computing*. P.36–43.

30. M. Liu and M. Zhu, “Mobile video object detection with temporally aware feature maps. In proceedings of CVPR 2018. P.5686–5695.

31. Zhu X., Dai J., Zhu X., Wei Y., Yuan L. Towards high performance video object detection for mobiles. (Preprint / arXiv:1804.05830). 2018.

32. Li Q., Zhao X., Huang K. Learning temporally correlated representations using LSTMs for visual tracking. *IEEE/ICIP: Proceedings of International Conference on Image Processing (ICIP)*. Phoenix, Arizona, USA, Sep 25–28, 2016. P.1614–1618.

33. Liang Y., Zhou Y. LSTM multiple object tracker combining multiple cues. *IEEE/ICIP: Proceedings of International Conference on Image Processing (ICIP)*. Athens, Greece, Oct 7–10, 2018. P.2351–2355.

34. Факультет прикладних наук УКУ. На передовій безпілотники відіграють надзвичайно важливу роль, тому дуже важливо вміти швидко і безперешкодно аналізувати отримані дані. URL: <https://www.facebook.com/100063512295462/posts/pfbid02uEWNEKcAVT4na6aSgR1DJCfXnrKX1jcCMzQHnoivoNUPbD5iGv93ZTTKYREFeRbBl/> (дата звернення: 02.11.2022).

35. Martinez-Alpiste I. et. al. Smartphone-based real-time object recognition architecture for portable and constrained systems. *Journal of Real-Time Image Processing*. 2022. Vol. 19. P. 103–115.

36. Nguyen P.H. et. al. LightdenseYOLO: a fast and accurate marker tracker for autonomous UAV landing by visible light camera sensor on drone. *Sensors*. 2018. Vol. 18. P. 1–30.

37. Mingxing T., Ruoming P., Quoc V. Le. EfficientDet: Scalable and Efficient Object Detection. *IEEE/CVF: Proceedings of the Conference on*

Computer Vision and Pattern Recognition (CVPR). Seattle, Washington, USA, June 13–19, 2020. P. 10781–10790.

38. Bajaj K., Singh D. K., Ansari M. A. Autoencoders based deep learner for image denoising. *Procedia Computer Science*. 2016. Vol. 171. P. 1535–1541.

39. Mizuho et al. Convolutional auto-encoder for image denoising of ultra-low-dose CT. *Heliyon*. 2017. Vol. 3. №e00393.

40. Sandler M. et. al. MobilenetV2: Inverted residuals and linear bottlenecks. Proceedings of the IEEE conference on computer vision and pattern recognition. Salt Lake City, Utah, USA, June 18–23, 2018. P. 4510–4520.

41. Lin T. Y. et. al. Focal loss for dense object detection. In proceedings of the IEEE international conference on computer vision. Venice, Italy, Oct 22–29, 2017. P. 2980–2988.

42. Nex F. et. al. UAV in the advent of the twenties: Where we stand and what is next. *ISPRS Journal of Photogrammetry and Remote Sensing*. 2022. Vol. 184. P. 215–242.

43. Pong B. The Art of Drone Warfare. *Journal of War & Culture Studies*. 2022. Vol. 15. P. 377–387.

## ДОДАТОК А. ЛІСТИНГ ПРОГРАМИ

```
import os
import glob
import time
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from PIL import Image

import tensorflow as tf
from tensorflow import keras
from keras import layers
def preprocess(array):

    array = array.astype("float32") / 255.0
    array = np.reshape(array, (len(array), 416, 416, 3))
    return array

def noise(array):

    noise_factor = 0.4
    noisy_array = array + noise_factor * np.random.normal(
        loc=0.0, scale=1.0, size=array.shape
    )

    return np.clip(noisy_array, 0.0, 1.0)
```

```
def display(array1, array2):

    n = 5

    indices = np.random.randint(len(array1), size=n)
    images1 = array1[indices, :]
    images2 = array2[indices, :]

    plt.figure(figsize=(12, 8))
    for i, (image1, image2) in enumerate(zip(images1, images2)):
        ax = plt.subplot(2, n, i + 1)
        plt.imshow(image1.reshape(416, 416, 3))
        plt.gray()
        ax.get_xaxis().set_visible(False)
        ax.get_yaxis().set_visible(False)

        ax = plt.subplot(2, n, i + 1 + n)
        plt.imshow(image2.reshape(416, 416, 3))
        plt.gray()
        ax.get_xaxis().set_visible(False)
        ax.get_yaxis().set_visible(False)

    plt.show()

def display_one(array):
    data = array.reshape(416, 416, 3)
    fig, ax = plt.subplots(figsize=(8, 8))
    fig.set_size_inches(8, 8)
    ax.imshow(data, interpolation='none')
```

```

plt.axis('off')
plt.tight_layout()
plt.savefig("current.png", bbox_inches='tight', dpi=53)

def save_results(array, folder_name):
    if not os.path.exists(folder_name):
        os.mkdir(folder_name)
    for i in range(len(array)):
        name = folder_name + '/' + str(i)+'.png'
        image = array[i:i+1]
        data = image.reshape(416, 416, 3)
        fig, ax = plt.subplots(figsize=(8, 8))
        fig.set_size_inches(8, 8)
        ax.imshow(data, interpolation='none')
        plt.axis('off')
        plt.tight_layout()
        plt.savefig(name, bbox_inches='tight', dpi=53)

from google.colab import drive
drive.mount('/content/gdrive')
!ln -s /content/gdrive/MyDrive /mydrive
filelist_train = glob.glob('/mydrive/STOR/data/denoise_train/*.jpg')
train_data = np.array([np.array(Image.open(fname)) for fname in filelist_train])
train_data = preprocess(train_data)
noisy_train_data = noise(train_data)
display(train_data, noisy_train_data)
filelist_test = glob.glob('/mydrive/STOR/data/denoise_test/*.jpg')
test_data = np.array([np.array(Image.open(fname)) for fname in filelist_test])
test_data = preprocess(test_data)
noisy_test_data = noise(test_data)
autoencoder = keras.Sequential(name="Denoise_CNN")

```

```
# Encoder
```

```
autoencoder.add(layers.Conv2D(64, kernel_size=(3, 3), padding="same",  
activation="relu", input_shape=(416, 416, 3)))  
autoencoder.add(layers.MaxPooling2D((2, 2), padding="same"))  
autoencoder.add(layers.BatchNormalization())  
autoencoder.add(layers.Conv2D(32, kernel_size=(3, 3), padding="same",  
activation="relu"))  
autoencoder.add(layers.MaxPooling2D((2, 2), padding="same"))  
autoencoder.add(layers.BatchNormalization())  
autoencoder.add(layers.Conv2D(16, kernel_size=(3, 3), padding="same",  
activation="relu"))  
autoencoder.add(layers.MaxPooling2D((2, 2), padding="same"))
```

```
# Decoder
```

```
autoencoder.add(layers.Conv2DTranspose(64, kernel_size=(3, 3), strides=2,  
padding="same", activation="relu"))  
autoencoder.add(layers.Conv2DTranspose(32, kernel_size=(3, 3), strides=2,  
padding="same", activation="relu"))  
autoencoder.add(layers.Conv2DTranspose(16, kernel_size=(3, 3), strides=2,  
padding="same", activation="relu"))  
autoencoder.add(layers.Conv2D(3, kernel_size=(3, 3), padding="same",  
activation="sigmoid"))
```

```
early_stop = keras.callbacks.EarlyStopping(monitor='loss', patience=2)  
autoencoder.compile(optimizer="adam", loss="mse", metrics=["accuracy"])  
autoencoder.summary()  
autoencoder.fit(  
    x=noisy_train_data,  
    y=train_data,
```

```
epochs=20,
batch_size=4,
shuffle=True,
validation_data=(noisy_test_data, test_data),
)

start_time = time.time()

predictions = autoencoder.predict(noisy_test_data)

end_time = time.time()

print("Denoising inference time: ")
print("--- %s seconds ---" % (end_time - start_time))
display(noisy_test_data, predictions)
results = autoencoder.evaluate(noisy_test_data, test_data, batch_size=4)
print("test loss, test accuracy:", results)
from google.colab import files
save_results(noisy_test_data, 'noisy_test_data')
!zip -r /content/noisy_test_data.zip /content/noisy_test_data
files.download('/content/noisy_test_data.zip')
save_results(predictions, 'pred')
!zip -r /content/pred_s20.zip /content/pred
files.download('/content/pred.zip')

from google.colab import drive
drive.mount('/content/gdrive')
!ln -s /content/gdrive/MyDrive /mydrive
!sudo apt -y install libportaudio2
!pip install -q --use-deprecated=legacy-resolver tf-lite-model-maker
```



```
!pip install -q pycocotools
!pip install -q opencv-python-headless==4.1.2.30
!pip uninstall -y tensorflow && pip install -q tensorflow==2.8.0
import numpy as np
import os
import glob
import xml.etree.ElementTree as ET
import pandas as pd
import tensorflow as tf

from tflite_model_maker.config import QuantizationConfig
from tflite_model_maker.config import ExportFormat
from tflite_model_maker import model_spec
from tflite_model_maker import object_detector

assert tf.__version__.startswith('2')

tf.get_logger().setLevel('ERROR')
from absl import logging
logging.set_verbosity(logging.ERROR)
validation_data =
object_detector.DataLoader.from_pascal_voc("/mydrive/STOR/data/validation/im
ages", "/mydrive/STOR/data/validation/annotations", label_map={1:
"UnidentifiedMovingObject", 2: "Servicemen", 3: "Vehicle"})
validation_data = object_detector.DataLoader.from_cache("/mydrive/STOR/data/c
ached/validation/fd2c112752054ea1fcebcbf49d7779205")
print(validation_data.size)
test_data =
object_detector.DataLoader.from_pascal_voc("/mydrive/STOR/data/test/images",
```

```

"/mydrive/STOR/data/test/annotations", label_map={ 1:
"UnidentifiedMovingObject", 2: "Servicemen", 3: "Vehicle"})
test_data =
object_detector.DataLoader.from_cache("/mydrive/STOR/data/cached/test/testdata
")
print(test_data.size)
train_data =
object_detector.DataLoader.from_pascal_voc("/mydrive/STOR/data/train/images",
"/mydrive/STOR/data/train/annotations", label_map={ 1:
"UnidentifiedMovingObject", 2: "Servicemen", 3: "Vehicle"})
train_data =
object_detector.DataLoader.from_cache("/mydrive/STOR/data/cached/train/traindata")
print(train_data.size)
spec = model_spec.get('efficientdet_lite0')
model = object_detector.create(train_data, model_spec=spec, epochs=30,
batch_size=16, train_whole_model=True, validation_data=validation_data)
model.evaluate(test_data)
model.export(export_dir='/mydrive/STOR/training')
model.evaluate_tflite('/mydrive/STOR/training/model.tflite', test_data)
#@title Load the trained TFLite model and define some visualization functions

import cv2
import time

from PIL import Image

model_path = '/mydrive/STOR/training/model.tflite'

# Load the labels into a list

```

```

classes = ['???'] * model.model_spec.config.num_classes
label_map = model.model_spec.config.label_map
for label_id, label_name in label_map.as_dict().items():
    classes[label_id-1] = label_name

# Define a list of colors for visualization
COLORS = np.random.randint(0, 255, size=(len(classes), 3), dtype=np.uint8)

def preprocess_image(image_path, input_size):
    """Preprocess the input image to feed to the TFLite model"""
    img = tf.io.read_file(image_path)
    img = tf.io.decode_image(img, channels=3)
    img = tf.image.convert_image_dtype(img, tf.uint8)
    original_image = img
    resized_img = tf.image.resize(img, input_size)
    resized_img = resized_img[tf.newaxis, :]
    resized_img = tf.cast(resized_img, dtype=tf.uint8)
    return resized_img, original_image

def detect_objects(interpreter, image, threshold):
    """Returns a list of detection results, each a dictionary of object info."""

    signature_fn = interpreter.get_signature_runner()

    # Feed the input image to the model
    output = signature_fn(images=image)

    # Get all outputs from the model
    count = int(np.squeeze(output['output_0']))

```

```
scores = np.squeeze(output['output_1'])
classes = np.squeeze(output['output_2'])
boxes = np.squeeze(output['output_3'])
```

```
results = []
for i in range(count):
    if scores[i] >= threshold:
        result = {
            'bounding_box': boxes[i],
            'class_id': classes[i],
            'score': scores[i]
        }
        results.append(result)
return results
```

```
def run_odt_and_draw_results(image_path, interpreter, threshold=0.5):
    """Run object detection on the input image and draw the detection results"""
    # Load the input shape required by the model
    _, input_height, input_width, _ = interpreter.get_input_details()[0]['shape']

    # Load the input image and preprocess it
    preprocessed_image, original_image = preprocess_image(
        image_path,
        (input_height, input_width)
    )

    start_time = time.time()
    # Run object detection on the input image
    results = detect_objects(interpreter, preprocessed_image, threshold=threshold)
```

```

end_time = time.time()
print("OD inference time: ")
print("--- %s seconds ---" % (end_time - start_time))
# Plot the detection results on the input image
original_image_np = original_image.numpy().astype(np.uint8)
for obj in results:
    # Convert the object bounding box from relative coordinates to absolute
    # coordinates based on the original image resolution
    ymin, xmin, ymax, xmax = obj['bounding_box']
    xmin = int(xmin * original_image_np.shape[1])
    xmax = int(xmax * original_image_np.shape[1])
    ymin = int(ymin * original_image_np.shape[0])
    ymax = int(ymax * original_image_np.shape[0])

    # Find the class index of the current object
    class_id = int(obj['class_id'])

    # Draw the bounding box and label on the image
    color = [int(c) for c in COLORS[class_id]]
    cv2.rectangle(original_image_np, (xmin, ymin), (xmax, ymax), color, 2)
    # Make adjustments to make the label visible for all objects
    y = ymin - 15 if ymin - 15 > 15 else ymin + 15
    label = "{:}: {:.0f}%".format(classes[class_id], obj['score'] * 100)
    cv2.putText(original_image_np, label, (xmin, y),
                cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 2)

# Return the final image
original_uint8 = original_image_np.astype(np.uint8)
return original_uint8

```

```
#@title Run object detection and show the detection results
```

```
INPUT_IMAGE_URL =
```

```
"https://www.dropbox.com/s/46j1mnyivcbahcm/aisvqqymmsjppnt.jpg?dl=0"
```

```
#@param {type:"string"}
```

```
DETECTION_THRESHOLD = 0.12 #@param {type:"number"}
```

```
TEMP_FILE = "/mydrive/STOR/tmp/tempor1.png"
```

```
!wget -q -O $TEMP_FILE $INPUT_IMAGE_URL
```

```
im = Image.open(TEMP_FILE)
```

```
im.thumbnail((512, 512), Image.ANTIALIAS)
```

```
im.save(TEMP_FILE, 'PNG')
```

```
# Load the TFLite model
```

```
interpreter = tf.lite.Interpreter(model_path=model_path)
```

```
interpreter.allocate_tensors()
```

```
# Run inference and draw detection result on the local copy of the original file
```

```
detection_result_image = run_odt_and_draw_results(
```

```
    TEMP_FILE,
```

```
    interpreter,
```

```
    threshold=DETECTION_THRESHOLD
```

```
)
```

```
# Show the detection result
```

```
Image.fromarray(detection_result_image)
```

```
!curl https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -
```

```
!echo "deb https://packages.cloud.google.com/apt coral-edgetpu-stable main" |  
sudo tee /etc/apt/sources.list.d/coral-edgetpu.list
```

```
!sudo apt-get update
```

```
!sudo apt-get install edgetpu-compiler
```

```
NUMBER_OF_TPUS = 1
```

```
!edgetpu_compiler '/mydrive/STOR/training/model.tflite' --  
num_segments=$NUMBER_OF_TPUS
```