

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерної інженерії і управління
(повна назва)

Кафедра _____ електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти _____ другий (магістерський)

Модель та метод ідентифікації об'єктів
по зображенням повітряного оптичного
моніторингу
(тема)

Виконав:

студент _____ II курсу, групи _____ СПМ-22-2
Собода А.П.
(прізвище, ініціали)

Спеціальність _____
123 «Комп'ютерна інженерія»
(код і повна назва спеціальності)

Тип програми _____ освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма _____
Системне програмування
(повна назва освітньої програми)

Керівник: _____ проф. Рубан І.В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ

_____ Коваленко А.А.
(підпис) (прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерної інженерії та управління _____

Кафедра _____ електронних обчислювальних машин _____

Рівень вищої освіти _____ другий (магістерський) _____

Спеціальність _____ 123 «Комп'ютерна інженерія» _____
(код і повна назва)

Тип програми _____ освітньо-професійна _____
(освітньо-професійна або освітньо-наукова)

Освітня програма _____ Системне програмування _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

“ _____ ” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту _____ Сободе Андрію Павловичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи Модель та метод ідентифікації об'єктів по зображенням повітряного оптичного моніторингу

затверджена наказом по університету від “ 06 ” листопада 2023 р. № 1299 Ст

2. Термін подання студентом роботи до екзаменаційної комісії _____ 15 січня 2024 р.

3. Вхідні дані до роботи _____

1. Зображення повітряного оптичного моніторингу.

2. Програмне середовище Google Colab Pro.

4. Перелік питань, що потрібно опрацювати у роботі _____

1. Аналіз проблеми.

2. Огляд базових методів.

3. Застосування методів ідентифікації об'єктів на зображеннях.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) 10 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Огляд методів ідентифікації об'єктів	07.11.23-13.11.23	
2	Вибір інструментальних засобів	14.11.23-23.11.23	
3	Проведення експериментів	24.11.23-23.12.23	
4	Оформлення матеріалів кваліфікаційної роботи	26.12.23-02.01.24	
5	Подання кваліфікаційної роботи керівникові та її попередній захист	03.01.24-06.01.24	
6	Подання кваліфікаційної роботи на рецензування	09.01.24-12.01.24	

Дата видачі завдання 06 листопада 2023 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис)

проф. Рубан.І.В.
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 60 с., 29 рис., 4 табл., 1 дод., 15 джерел.

ІДЕНТИФІКАЦІЯ, НЕЙРОМЕРЕЖА, МЕТОД, МАСКА, ЗОБРАЖЕННЯ, ЗГОРТКА.

Метою кваліфікаційної роботи є аналіз методів ідентифікації об'єктів на зображеннях і їх застосування для зображень повітряного оптичного моніторингу.

У ході виконання кваліфікаційної роботи були проаналізовані методи ідентифікації об'єктів на зображеннях. Показана їх архітектура та можливі області застосування. Розглянуто та застосовано набір даних з зображеннями повітряного оптичного моніторингу для задачі ідентифікації об'єктів.

У результаті виконання роботи були натреновані та порівняні дві моделі нейромереж здатних ідентифікувати до 15 різновидів об'єктів на зображеннях повітряного оптичного моніторингу.

ABSTRACT

Master's thesis: 60 pages, 29 figures, 4 tables, 1 appendices, 15 sources.

IDENTIFICATION, NEURAL NETWORK, METHOD, MASK, IMAGE, CONVULSION.

The major goal of this thesis is to analyze the methods of identifying objects in the images and their application to aerial optical monitoring images.

During the execution of the qualification work, methods of object identification in images were analyzed. Their architecture and possible areas of application were shown. A dataset with aerial optical monitoring images was considered and applied to the object identification task.

As a result of the work, two neural network models capable of identifying up to 15 different types of objects in aerial optical monitoring images were trained and compared.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	7
ВСТУП	8
1 АНАЛІЗ ПРОБЛЕМИ	9
2 ТЕОРЕТИЧНІ ОСНОВИ	12
2.1 Згорткові нейронні мережі	13
2.2 Операція згортки	14
2.3 Властивості згорки	16
2.4 Функції згорткових нейронних мереж	19
2.5 Розрахунок рецептивного поля	22
2.6 Ідентифікація об'єктів	23
3 ВИБІР АРХІТЕКТУРИ НЕЙРОННОЇ МЕРЕЖІ	26
3.1 Single Shot MultiBox Detector	26
3.2 Архітектура YOLOv5	28
3.3 Архітектура YOLOv8	30
4 НАБІР ДАНИХ ДЛЯ НАВЧАННЯ НЕЙРОМЕРЕЖІ	33
4.1 Розмір зображення	33
4.2 Різноманітність орієнтацій екземплярів	33
4.3 Інформація просторового розрішення	34
5 ПРАКТИЧНІ РЕЗУЛЬТАТИ	36
5.1 Результати тренування YOLOv8 nano.	36
5.2 Результати тренування YOLOv8 small	42
5.3 Результати ідентифікації об'єктів на зображеннях повітряного оптичного моніторингу.	49
ВИСНОВКИ	52
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	53
ДОДАТОК А ГРАФІЧНИЙ МАТЕРІАЛ КВАЛІФІКАЦІЙНОЇ РОБОТИ	55

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

NN – нейронна мережа (англ., Neural network)

CNN – згортова нейронна мережа (англ., Convolutional neural network)

YOLO – ти дивишся лише один раз (англ., You Only Look Once)

CV – Комп'ютерний зір (англ., Computer Vision)

ВСТУП

Питання ідентифікації об'єктів на зображеннях виникає вже протягом тривалого часу, але надзвичайна актуальність цієї проблеми стає очевидною у зв'язку з ростом обсягів зображень та даних, які потребують обробки та аналізу. Спочатку, коли були відсутні автоматизовані засоби аналізу, люди мусили вручну переглядати фотографії та відеоматеріали для ідентифікації об'єктів. Однак з розвитком технологій та розвитком обчислювальної сфери, такий підхід став неефективним і недоцільним. Аналіз відео та зображень з повітря стає все більше важливим в контексті спостереження над довколишнім середовищем та дистанційним моніторингом. Високе навантаження на обробку даних і потреба в точній ідентифікації об'єктів вимагають розвитку нових методів та алгоритмів. Проблеми, пов'язані з ідентифікацією об'єктів на зображеннях, можуть бути спричинені недостатньою інформацією про сам об'єкт. Об'єкти можуть бути частково видимі на зображенні, мати нетипову форму, бути погано освітлені, чи приховані перешкодами. Для вирішення цих проблем були натреновані моделі, які здатні ідентифікувати об'єкти на зображеннях повітряного оптичного моніторингу.

1 АНАЛІЗ ПРОБЛЕМИ

Виявлення об'єктів є важливим завданням комп'ютерного зору, яке полягає у виявленні екземплярів візуальних об'єктів певного класу (наприклад, людей, тварин або автомобілів) у цифрових зображеннях [1]. Метою виявлення об'єктів є розвиток обчислювальних засобів, моделей і методів, які забезпечують одну з найосновніших частин знань, необхідних для програм комп'ютерного зору: які об'єкти де розташовані? Дві найважливіші метрики для виявлення об'єктів - це точність (включаючи точність класифікації та точність локалізації) та швидкість.

Виявлення об'єктів служить основою для багатьох інших завдань комп'ютерного зору, таких як сегментація екземплярів [2], підписування зображень [3], відстеження об'єктів [4] тощо. В останні роки швидкий розвиток методів глибокого навчання значно сприяв прогресу виявлення об'єктів, що призвело до визначних проривів та перетворило його на дослідницьку гарячу точку з безпрецедентною увагою. Виявлення об'єктів тепер широко використовується в багатьох реальних програмах, таких як автономне водіння, робототехніка, відеоспостереження тощо. Рисунок 1.1 показує зростання кількості публікацій, пов'язаних з "виявленням об'єктів", за останні два десятиліття.

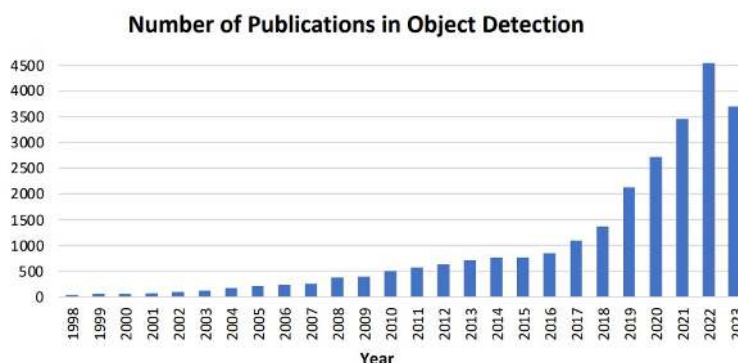


Рисунок 1.1 - Збільшення кількості публікацій з виявлення об'єктів з 1998 по 2023 рік

Задача ідентифікації об'єктів полягає в визначенні місцезнаходження об'єктів на зображенні (локалізації) та визначенні категорії, до якої вони належать (класифікації). Процес традиційного виявлення об'єктів зазвичай складається з трьох ключових фаз: вибору інформативної області, екстракції характеристик та класифікації.

У першому етапі, вибору інформативної області, через велику різноманітність потенційних положень і розмірів об'єктів у зображенні, зазвичай застосовується метод сканування зображення за допомогою вікон різних масштабів. Ця всеохоплююча стратегія дозволяє виявити всі можливі позиції об'єктів, але в той же час є ресурсоємною та часто призводить до великої кількості помилкових детекцій. Обмежена кількість шаблонів вікон може призвести до упущення релевантних областей.

Другий етап, екстракція характеристик, потребує виділення візуальних ознак, які можуть надавати значуще та надійне представлення для відрізнення об'єктів. SIFT, HOG та Хаар-подібні ознаки є добре відомими прикладами. Ці характеристики можуть імітувати складні функції зорових клітин людського мозку, але через різноманітність зовнішнього вигляду, умов освітлення та фону, створення універсального дескриптора вручну є складним завданням.

На завершальному етапі, класифікації, необхідний класифікатор для відокремлення цільових об'єктів від інших категорій та надання представлення більшої ієрархічності, семантики та інформативності. Зазвичай використовуються такі методи, як SVM, AdaBoost та DPM. DPM є гнучкою моделлю, яка інтегрує частини об'єкта з урахуванням деформацій. Застосування графічної моделі, ретельно підібраних низькорівневих ознак та інспірованих кінематикою декомпозицій дозволяє створювати точні моделі для різноманітних класів об'єктів.

Завдяки розвитку глибинних нейронних мереж (DNN) [5], значно зросла ефективність завдяки застосуванню методу R-CNN, який базується на характеристиках CNN [6]. Глибинні нейронні мережі, зокрема найбільш

відомі CNN, функціонують зовсім по-іншому, ніж традиційні методи. Вони мають більш складну архітектуру, що дозволяє їм навчатися більш складним характеристикам, аніж поверхневі нейронні мережі. Також завдяки виразності та міцним алгоритмам тренування вони можуть вивчати детальні та інформативні характеристики об'єктів.

Після R-CNN було запропоновано багато вдосконалених моделей, у тому числі Fast R-CNN, яка спільно оптимізує задачі регресії класифікації та обмежувальної рамки, Faster R-CNN, яка використовує додаткову підмережу для створення пропозицій регіону і YOLO, який виконує виявлення об'єктів за допомогою регресії з фіксованою сіткою. Усі вони забезпечують різні ступені покращення продуктивності виявлення порівняно з основним R-CNN і роблять точніше виявлення об'єктів у реальному часі більш досяжним.

Переваги CNN у порівнянні з традиційними методами можна підсумувати наступним чином.

1) Ієрархічне представлення ознак, яке є багаторівневими представленнями від пікселів до високорівневих семантичних характеристик, навчених за допомогою ієрархічної багатоетапної структури може бути автоматично навчене з даних, і приховані фактори вхідних даних можуть бути розкриті через багаторівневі нелінійні відображення.

2) У порівнянні з традиційними поверхневими моделями, більш глибока архітектура забезпечує експоненціально збільшену виразну здатність.

3) Архітектура CNN надає можливість спільно оптимізувати кілька пов'язаних завдань разом (наприклад, Fast R-CNN поєднує класифікацію та регресію обмежувальних рамок у багатозадачному способі навчання).

4) Завдяки великій навчальній потужності глибоких CNN, деякі класичні виклики комп'ютерного зору можуть бути переформульовані як проблеми перетворення високовимірних даних та вирішені з іншої точки зору.

2 ТЕОРЕТИЧНІ ОСНОВИ

Глибинне навчання у строгому сенсі включає використання кількох шарів штучних нейронів. Перші штучні нейронні мережі були розроблені наприкінці 1950-х років із презентацією алгоритмів перцептронів [7]. Однак обмеження, пов'язані з обчислювальними витратами цих алгоритмів у той період, а також часто неправильно цитоване твердження Мінського та Паперта [8], що перцептрони не здатні навчатися нелінійним функціям, таким як XOR, спричинили значне зниження інтересу до подальших досліджень цих алгоритмів і сприяли так званій зимі штучного інтелекту. Зокрема, у своїй книзі [8] Мінський та Паперт обговорювали, що одношарові перцептрони здатні навчатися лише лінійно роздільних патернів. Часто помилково вважалося, що вони також передбачили, що це стосується і багатошарових перцептронних мереж. Дослідження нейронних мереж відновилися більш ніж через 10 років, і в [9] деякі з цих питань були уточнені та додатково обговорені.

Навіть якщо протягом цього періоду не було багато досліджень, пов'язаних із перцептронами, були введені дуже важливі алгоритми, такі як алгоритм зворотного поширення помилки [10] та рекурентні нейронні мережі [11]. Перцептрон спочатку був розроблений для наглядних бінарних задач класифікації та був натхненний роботами неврологів, таких як Дональд Хебб.

Він був побудований на основі нелінійного нейрона, а саме, моделі нейрона Мак-Каллока—Піттса. Більш формально, ми шукаємо функцію $f(x; w, b)$, таку що $f(\cdot; w, b): \mathbf{x} \in R^p \rightarrow \{+1, -1\}$ де w та b є параметрами f , а вектор $\mathbf{x} = [x_1, \dots, x_p]^T$ є вхідними даними. Набір тренування складається з $\{(x^{(i)}, y^{(i)})\}$. Зокрема, перцептрон покладається на лінійну модель для виконання класифікації:

$$f(x; w, b) = \begin{cases} +1 & \text{if } w^T x + b \geq 0 \\ -1 & \text{otherwise} \end{cases} \quad (2.1)$$

Перцептрон на якому кожен з елементів входу описаний як нейрон, і всі елементи поєднуються шляхом зважування з параметрами моделі, а потім передаються до активаційної функції для остаточного рішення (рисунок 2.1).

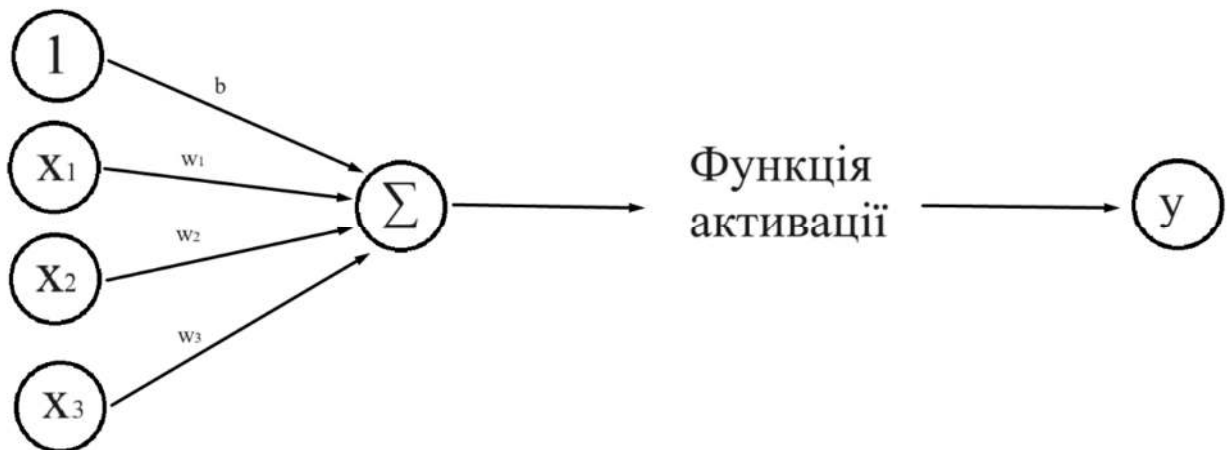


Рисунок 2.1 – Модель простого перцептрону

2.1 Згорткові нейронні мережі

Згорткові нейронні мережі (CNN) є спеціалізованою категорією глибинних нейронних мереж, які використовують операцію згортки для обробки вхідних даних [12].

На відміну від глибоких повністю з'єднаних мереж, згорткові нейронні мережі (CNN) відзначаються в обробці даних з просторовою або сітчастою організацією (наприклад, часові ряди, зображення, відео тощо) і водночас зменшують кількість параметрів, які можна навчити, завдяки їх властивостям спільного використання ваг. Решта цього розділу спочатку представляє операцію згортки та мотивацію до її використання як будівельного блоку/модуля нейронних мереж. Потім представлено ряд різних варіацій разом із прикладами найважливіших архітектур CNN. Нарешті, буде обговорено важливість рецептивного поля – центральної властивості таких

мереж.

2.2 Операція згортки

Операція згортки визначається як інтеграл добутку двох функцій (f, g) після того, як одна з них віддзеркалена та зсунута відносно іншої функції. Формально ми записуємо:

$$h(t) = \int_{-\infty}^{\infty} f(t-\tau)g(\tau)d\tau. \quad (2.2)$$

Таку операцію також можна позначити зірочкою (*), тому вона записується як:

$$h(t) = (f * g)(t) \quad (2.3).$$

Сутністю операції згортки є відображення того, як одна функція впливає на іншу. Ця інтуїція виникає з області обробки сигналів, де зазвичай важливо знати, як сигнал буде змінений фільтром. Наприклад, візьмемо одновимірний безперервний сигнал, як активність мозку пацієнта на деякому електроенцефалографічному електроді, та гаусівський фільтр. Результат операції згортки між цими двома функціями даватиме ефект гаусівського фільтру на цьому сигналі, який насправді буде згладженою версією вхідного сигналу.

Інший спосіб думати про операцію згортки полягає в тому, що вона показує, як пов'язані між собою дві функції. Іншими словами, вона показує, наскільки схожі або несхожі ці дві функції при різних відносних позиціях. Насправді операція згортки дуже схожа на операцію перехресної кореляції, з тонкою відмінністю в тому, що в операції згортки одна з двох функцій є інвертованою. У контексті глибинного навчання конкретні відмінності між цими двома операціями можуть бути другорядними; однак операція згортки

має більше властивостей, ніж кореляція, таких як комутативність. Зверніть також увагу, що коли сигнали симетричні, обидві операції дають однаковий результат[12].

Для роботи з дискретними та кінцевими сигналами, ми можемо розширити визначення операції згортки.

Зокрема, враховуючи два дискретні сигнали $f[k]$ та $g[k]$ де $k \in Z$, операція згортки визначається так:

$$h[k] = \sum_n f[k - n]g[n]. \quad (2.4)$$

Операцію згортки можна також розширити на багатовимірні сигнали аналогічним чином. Наприклад, ми можемо описати операцію згортки між двома дискретними та скінченними двовимірними сигналами (наприклад, $I[i, j], K[i, j]$) так:

$$H[i, j] = \sum_m \sum_n I[i - m, j - n]K[m, n] \quad (2.5)$$

Часто буває так, що перший сигнал є основним об'єктом аналізу, як, наприклад, зображення великого формату, а другий сигнал має набагато менший розмір, такий як матриця 3x3 або 4x4, і призначений для виконання специфічної функції. Другий сигнал називають ядром. На рисунку 2.2 можна побачити візуалізацію процесу згортки для двовимірного дискретного сигналу, такого як зображення, із застосуванням ядра розміром 3 на 3. Для кожної позиції на вхідному сигналі ядро згортки зміщується, проводиться покомпонентне множення ядра на відповідну частину вхідного сигналу та сумування результатів для отримання кінцевого значення згортки.

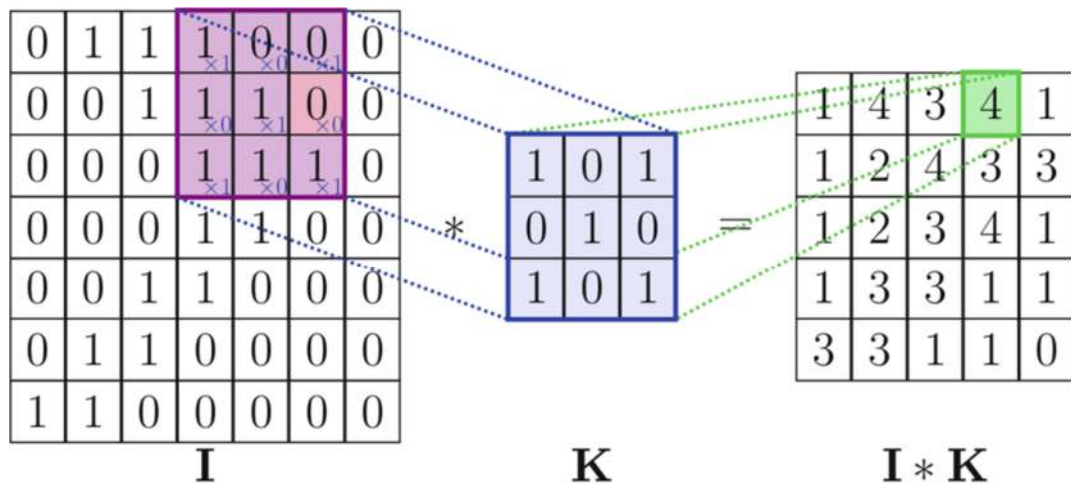


Рисунок 2.2 - Візуалізація дискретної згортки [12]

Приклади того, як згортки можуть бути застосовані до зображень, представлені на рисунку 2.3.

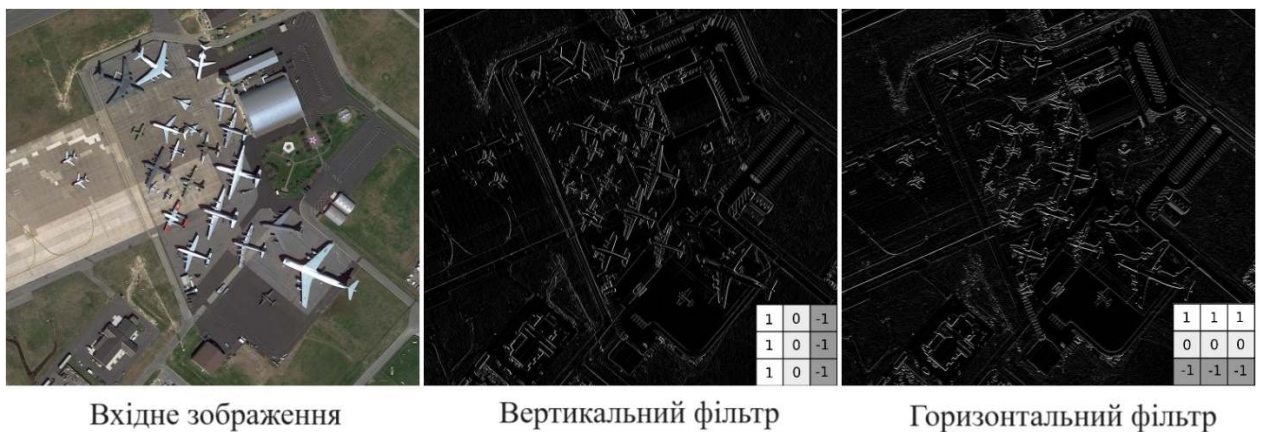


Рисунок 2.3 – Два приклади застосування згорток до зображення. Один з фільтрів діє як детектор вертикальних країв, а інший — як детектор горизонтальних країв.

2.3 Властивості згортки

У випадку дискретної області, операцію згортки можна виконати, використовуючи просте матричне множення, без необхідності зсувати один сигнал над іншим. Це можна по суті досягти, використовуючи перетворення матриці Тепліца. Перетворення Тепліца створює розріджену матрицю з

повторюваними елементами, яка при множенні на вхідний сигнал дає результат згортки. Щоб ілюструвати, як операція згортки може бути реалізована як матричне множення, розгляньмо приклад з ядром 3 на 3 (K) та входом 4 на 4 (I).

$$K = \begin{bmatrix} k_{00} & k_{01} & k_{02} \\ k_{10} & k_{11} & k_{12} \\ k_{20} & k_{21} & k_{22} \end{bmatrix} \text{ and } I = \begin{bmatrix} i_{00} & i_{01} & i_{02} & i_{03} \\ i_{10} & i_{11} & i_{12} & i_{13} \\ i_{20} & i_{21} & i_{22} & i_{23} \\ i_{30} & i_{31} & i_{32} & i_{33} \end{bmatrix} \quad (2.6)$$

Тоді операцію згортки можна обчислити як матричне множення між ядром, перетвореним у вигляді матриці Тепліца (рисунок 2.4).

$$K = \begin{bmatrix} k_{00} & k_{01} & k_{02} & 0 & k_{10} & k_{11} & k_{12} & 0 & k_{20} & k_{21} & k_{22} & 0 & 0 & 0 & 0 & 0 \\ 0 & k_{00} & k_{01} & k_{02} & 0 & k_{10} & k_{11} & k_{12} & 0 & k_{20} & k_{21} & k_{22} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & k_{00} & k_{01} & k_{02} & 0 & k_{10} & k_{11} & k_{12} & 0 & k_{20} & k_{21} & k_{22} & 0 \\ 0 & 0 & 0 & 0 & 0 & k_{00} & k_{01} & k_{02} & 0 & k_{10} & k_{11} & k_{12} & 0 & k_{20} & k_{21} & k_{22} \end{bmatrix}$$

і перетворений вхід:

$$I = [i_{00} \ i_{01} \ i_{02} \ i_{03} \ i_{10} \ i_{11} \ i_{12} \ i_{13} \ i_{20} \ i_{21} \ i_{22} \ i_{23} \ i_{30} \ i_{31} \ i_{32} \ i_{33}]^T.$$

Рисунок 2.4 – Матриця Тепліца для операції згортки

Отриманий результат буде переформовано в матрицю розміром 2 на 2, щоб відновити вихідні дані згортки. Реалізація цієї матричної множення підсвічує декілька важливих властивостей операції згортки, які є ключовими для її використання в глибоких нейронних мережах.

Перетворивши операцію згортки на операцію матричного множення, стає ясно, що вона може бути вписана в формалізацію лінійних функцій. Таким чином, глибокі нейронні мережі можуть бути спроектовані так, щоб використовувати навчальні згорткові ядра. На практиці, в кожному згортковому блоку навчаються кілька ядер, а декілька таких блоків складаються один на одного, формуючи глибокі CNN. Зазвичай результат

операції згортки називають картою ознак або просто ознаками [12].

Однією з важливих особливостей операції згортки є те, що вона потребує набагато менше параметрів, ніж повністю з'єднані глибокі нейронні мережі на основі MLP. Як можна побачити з матриці K (рисунок 2.4), одні й ті ж параметри використовуються на всіх позиціях. Врешті-решт, замість навчання різного набору параметрів для різних місць розташування вхідних даних, вчиться тільки один набір. Це явище називається спільним використанням параметрів або ваг і може значно зменшити кількість пам'яті, яка потрібна для зберігання параметрів мережі. Ілюстрація процесу спільного використання ваги на різних позиціях, разом з тим фактом, що для даного шару обчислюються кілька фільтрів (що призводить до кількох карт ознак), представлена на рисунку 2.5.

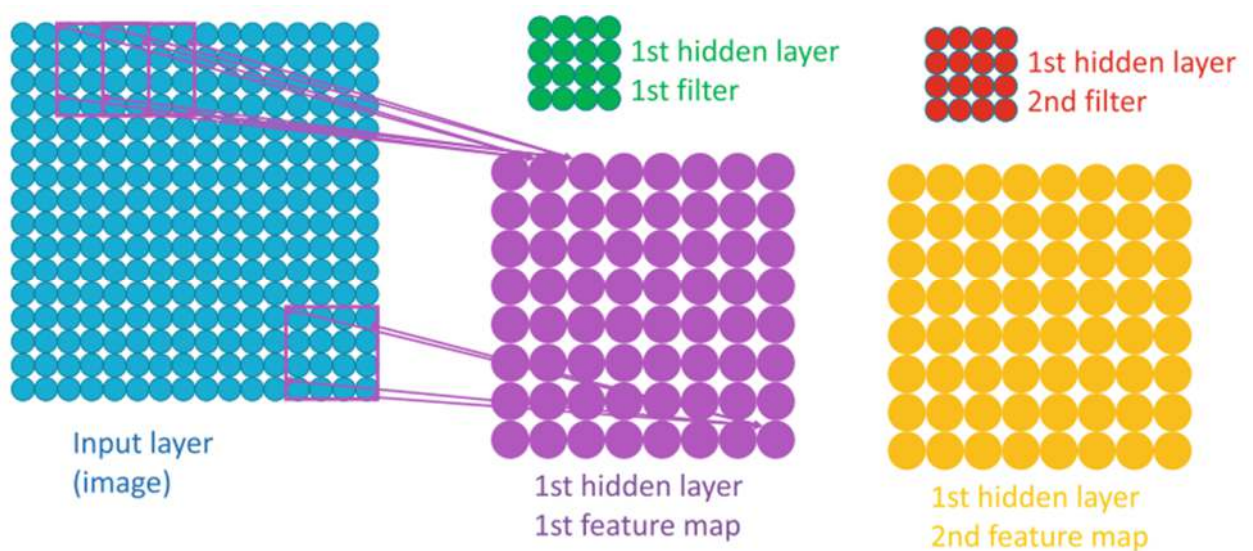


Рисунок 2.5 – На даному шарі вчиться багато фільтрів, кожен з яких виявляє певну характеристику в зображенні, що призводить до створення кількох карт ознак [12]

Згорткові нейронні мережі довели свою високу ефективність у обробці даних з просторовою структурою, як-от зображення, відео тощо. Це ефективність базується на факті локальної зв'язності елементів ядра, при цьому одне й те саме ядро застосовується в різних місцях вхідних даних

[12].Така обробка надає вельми корисну властивість, що називається еквівалентністю перекладу, яка дозволяє мережі виводити подібні відгуки на різних розташуваннях вхідних даних. Як приклад корисності такої властивості можна навести завдання детекції на зображеннях. Зокрема, коли тренують мережу для виявлення пухлин на МРТ-зображенні головного мозку, модель повинна реагувати подібним чином незалежно від місця, де може проявлятися аномалія.

2.4 Функції згорткових нейронних мереж

Нарешті, іншою важливою властивістю операції згортки є те, що вона робить розмір вхідних даних незалежним від параметрів, які можна навчити. Наприклад, у випадку з повністю з'єднаними мережами (MLP), розмір матриці вагів є функцією виміру вхідних даних. Зокрема, щільно з'єднаний шар, що відображає 256 ознак на 10 виходів, матиме розмір $W \in \mathbb{R}^{10 \times 256}$.

Навпаки, у згорткових шарах кількість параметрів, які можна навчити, залежить лише від розміру ядра і кількості ядер, які має шар. Це дозволяє обробляти вхідні дані довільного розміру, наприклад у випадку повністю згорткових мереж. Операція згортки може змінити розмірність вихідних даних. У прикладі (рисунок 2.2), зображення розміром 7×7 , після згортки з ядром розміром 3×3 , дає карту ознак розміром 5×5 . Хоча зміну розмірів можна уникнути за допомогою відповідного додавання крайових даних перед операцією згортки (рисунок 2.6), в деяких випадках насправді бажано зменшити розміри вхідних даних. Таке зменшення можна досягти різними способами, залежно від завдання, яке стоїть перед розробником.

У цьому підрозділі будуть обговорюватися деякі з найбільш типових функцій, які використовуються в CNN.

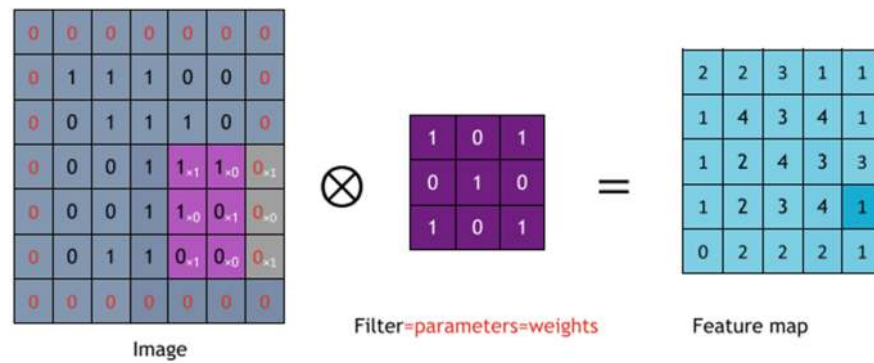


Рисунок 2.6 – Додавання відступу навколо матриці вхідного зображення [12]

2.4.1 Функція об'єднання шарів

У багатьох архітектурах згорткових нейронних мереж (CNN) широко використовуються операції зменшення вибірки, метою яких є стиснення розміру карт ознак та зменшення обчислювального навантаження. Такі операції, також відомі як шари об'єднання, агрегують значення свого входу в залежності від їхньої конструкції. Деякі з найпоширеніших шарів зменшення вибірки — це максимальне об'єднання, середнє об'єднання або глобальне середнє об'єднання. У перших двох випадках використовується або максимальне, або середнє значення як ознака для виходу по неперекриваючим регіонам заздалегідь визначеного розміру об'єднання. У випадку глобального середнього об'єднання всі просторові виміри представлені середнім значенням.

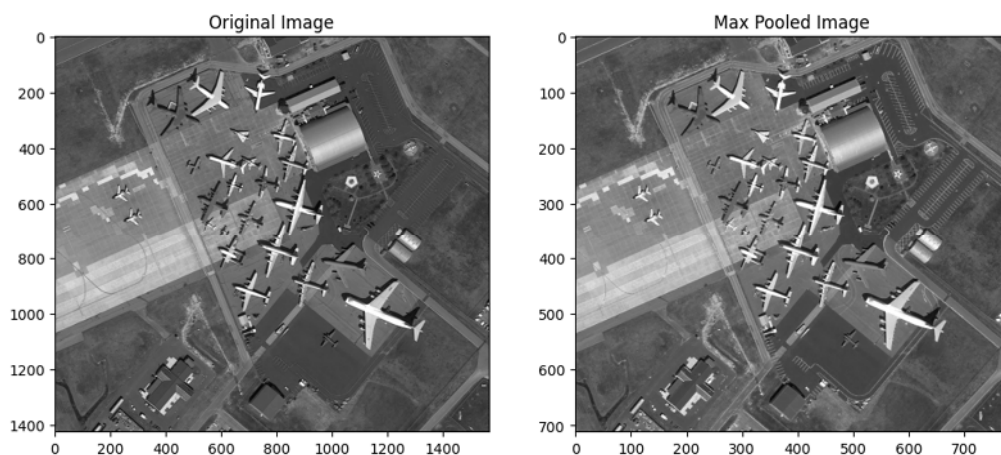


Рисунок 2.7 – Результат операції об'єднання

2.4.2 Згортка зі збільшеним кроком

Згортка з великим кроком - це техніка у згорткових нейронних мережах, де замість того, щоб застосовувати згортку на кожній позиції з мінімальним кроком (розміром 1), використовуються більші інтервали кроків. Це означає, що операція згортки виконується через визначені проміжки, пропускаючи деякі позиції, що призводить до створення вихідної карти ознак з меншою кількістю елементів. Такий підхід дозволяє зменшити розміри ознак у проміжних шарах нейронної мережі, тим самим знижуючи її обчислювальне навантаження.

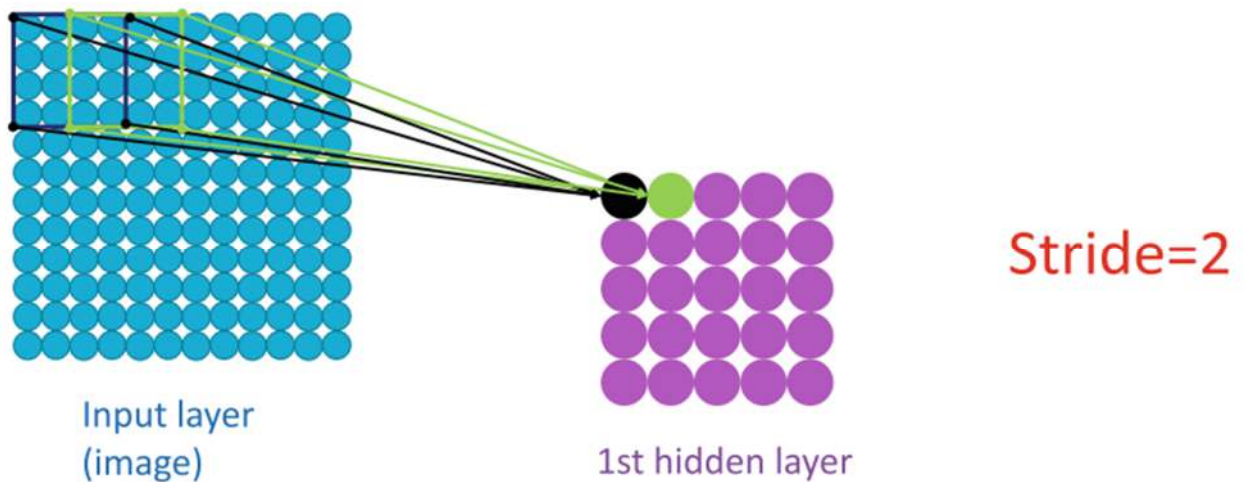


Рисунок 2.8 – Приклад згортки з кроком 2 [12]

2.4.3 Розширена згортка

Розширена згортка, яку також називають атрозною, - це згортка з ядрами, які були розширені шляхом вставляння нульових відступів («à trous» у французькій мові, що означає "з дірками") між ненульовими значеннями ядра. У цьому випадку до операції згортки додається додатковий параметр (d), який змінює відстань між елементами ядра. По суті, це збільшує охоплення ядра, але при цьому кількість параметрів, які можуть навчитися, залишається тією ж. Наприклад, розширена згортка з розміром ядра 3 на 3 та

ступенем розширення $d = 2$ буде розташована на сітці розміром 5 на 5, але з більшими проміжками між елементами.

2.4.4 Транспонована згортка

У деяких випадках необхідно не тільки зменшити просторові розміри вхідних даних, але й, зазвичай на більш пізніх етапах мережі, застосувати операцію збільшення розмірів. Найбільш характерний приклад – завдання сегментації зображень, де очікується класифікація на рівні пікселів, і, отже, вихід нейронної мережі повинен мати той же розмір, що й вхід. У таких випадках зазвичай застосовуються декілька операцій збільшення розмірів. Збільшення може бути досягнуте за допомогою операції транспонованої згортки, яка в кінцевому підсумку збільшить розмір виходу. Детально, транспонована згортка виконується шляхом розширення входу замість ядра перед застосуванням згорткової операції. Таким чином, вхід розміром 5 на 5 досягне розміру 10 на 10 після розширення з $d = 2$. З належним додаванням крайових даних і використанням ядра розміром 3 на 3, вихід в кінцевому підсумку подвоїться за розміром.

2.5 Розрахунок рецептивного поля

У контексті глибоких нейронних мереж, а конкретно у згорткових нейронних мережах (CNN), термін "рецептивне поле" використовується для визначення частини входу, яка генерує конкретну ознаку. Наприклад, CNN, яка приймає зображення як вхідні дані та застосовує лише одну операцію згортки з ядром розміром 3 на 3, матиме рецептивне поле 3 на 3. Це означає, що для кожного пікселя першої карти ознак розглядається 3 на 3 ділянка входу. Тепер, якщо додати ще один шар, знову з розміром 3 на 3, то рецептивне поле нової карти ознак з огляду на вхід CNN буде 5 на 5. Іншими словами, частина входу, яка використовується для розрахунку кожного

елемента карти ознак другого згорткового шару, збільшується. Обчислення рецептивного поля на різних ділянках згорткової нейронної мережі (CNN) є ключовим при спробі зрозуміти внутрішні механізми конкретної архітектури. Наприклад, CNN, яка призначена для обробки зображень розміром 256×256 і якій потрібна інформація з усіх її частин, повинна мати рецептивне поле, близьке до розміру входу. Рецептивне поле може бути змінене всіма різними параметрами згортки та операціями зменшення/збільшення розміру.

2.6 Ідентифікація об'єктів

Задача виявлення об'єктів полягає у визначенні та маркуванні кожного об'єкта в зображенні за допомогою обмежувальних рамок. Час, необхідний для виявлення об'єктів, скорочується за рахунок використання методу ковзаючого вікна, який є основоположним для виявлення об'єктів [13]. У цьому методі використовується вікно певного розміру, наприклад $M \times N$, щоб просканувати цільове зображення. Спочатку на тренувальних зображеннях створюється класифікатор, який вчиться розрізняти шукані об'єкти як одну категорію та неправильні об'єкти як іншу. Об'єкти, які мають відповідати шуканим, класифікуються як позитивні зразки, тоді як помилкові випадки вважаються негативними.

Існує два основних напрями методів виявлення: методи, засновані на пропозиції регіонів, і методи, засновані на класифікації. До першої групи належать такі системи, як R-CNN та Faster R-CNN, а друга категорія включає системи типу "YOLO" (You Only Look Once) та SSD (Single Shot Detector).

2.6.1 Швидший R-CNN

Швидший R-CNN складається з двох мереж: мережі пропозиції регіонів, яка використовується для генерації регіонів інтересу, тобто регіону пропозицій, та мережі, яка використовує ці пропозиції для виявлення об'єкта.

Таким чином, у Швидшому R-CNN генерація пропозиції регіону та виявлення об'єкта відбувається за допомогою подібної мережі. Він запропонував ідею якірних коробок. Ці коробки розміщуються у ковзному вікні та виконуються з трьома масштабами (128x128, 256x256 та 512x512) та трьома співвідношеннями сторін (1:1, 2:1 та 1:2). Таким чином, всього 9 коробок базуються на мережі пропозиції регіону для прогнозування ймовірності того, що вона належить до фону або переднього плану. Отже, Швидший R-CNN вважається однією з найточніших технік виявлення об'єктів.

2.6.2 You Only Look Once

YOLO — це техніка, у якій окремі компоненти виявлених об'єктів об'єднуються в єдину нейронну мережу. Вона обробляє вхідне зображення та використовує його ознаки для прогнозування кожної обмежувальної рамки. Вона ділить кожне зображення на сітку $S \times S$ і кожна сітка передбачає N обмежувальних рамок і підтверджує оцінку для цих рамок. Довірчість виявлення показує точність обмежувальної рамки, чи вона охоплює вірне зображення чи ні. YOLO є надшвидкою і може використовуватися у реальному часі, але її недоліком є те, що вона робить висновок лише про один тип класу в одній сітці.

2.6.3 Детектор одного знімку (SSD)

SSD, як і передбачає назва, призначений для виявлення численних об'єктів на зображенні лише за один "знімок", що означає, що згортована нейронна мережа працює лише один раз і передбачає ознаки на карті ознак. Вона також використовує якірні рамки з різними співвідношеннями аспектів та масштабами. Для передбачення об'єктів різного розміру мережа комбінує піраміди ознак з різними роздільностями. Це значно швидший метод

порівняно з методами, які базуються на двофазній RPN.

2.6.4 RetinaNet

RetinaNet — це техніка, яка створена шляхом об'єднання двох покращень над методами, подібними до YOLO та SSD:

- пірамідальні мережі ознак для виявлення об'єктів;
- вогнищеві втрати для щільного виявлення об'єктів.

Пірамідальна мережа ознак (FPN) використовує характеристику багаторівневої пірамідальної структури глибоких CNN для створення пірамід ознак. Вона поєднує низько-роздільні, семантично міцні ознаки з високо-роздільними, семантично слабкими ознаками через шлях зверху-вниз та з'єднання на краях. Вогнищеві втрати призначені для вирішення проблем одноетапного виявлення об'єктів з неправильностями, де є надзвичайно велика кількість можливих фонових класів і лише незначна кількість класів переднього плану. Focal loss є покращенням для втрат крос-ентропії, що зменшує відповідні втрати для добре організованих моделей та встановлює більш суворі штрафи на помилково класифіковані моделі.

RetinaNet є самостійною, єдиною системою, що складається з основної мережі та двох специфічних підмереж завдань. Основна мережа є основою і відповідає за обробку вхідного зображення. Перша підмережа виконує класифікацію об'єктів, використовуючи виходи основної мережі, тоді як друга підмережа працює над визначенням обмежувальної рамки.

3 ВИБІР АРХІТЕКТУРИ НЕЙРОННОЇ МЕРЕЖІ

3.1 Single Shot MultiBox Detector

Підхід SSD полягає у використанні прямого згорткового мережевого процесу, що створює задану кількість обмежувальних рамок та оцінює імовірність присутності класів об'єктів у цих рамках. Після цього виконується етап немаксимального пригнічення для визначення кінцевих локацій об'єктів. SSD використовує початкові шари звичайної мережевої архітектури, призначеної для класифікації зображень високої якості (але без шарів класифікації), яку ми називаємо базовою мережею. До цієї базової мережі додаються додаткові згорткові шари для створення багатомасштабних карт ознак, які дозволяють виявляти об'єкти на різних рівнях масштабу. Ці додаткові шари поступово зменшуються у розмірах, що дозволяє мережі робити передбачення на декількох рівнях (рисунок 3.1).

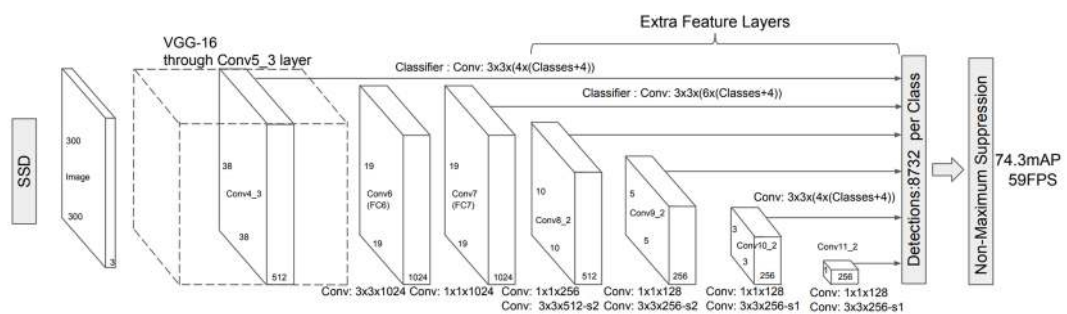


Рисунок 3.1 -Архітектура SSD

Кожен з цих нових або існуючих шарів ознак може використовувати набір згорткових фільтрів для створення фіксованої кількості передбачень обмежувальних рамок. Для шару ознак розміром $m \times n \times r$ з r каналами використовується базовий $3 \times 3 \times r$ фільтр, який передбачає категорію об'єкта або зміщення рамки відносно її стандартного положення. Цей процес

відбувається на кожному з $m \times n$ позицій, де застосовується фільтр, створюючи відповідне вихідне значення. Вихідні значення зміщення обмежувальних рамок визначаються відносно положення стандартної рамки, яке залежить від кожного місця розташування на карті ознак. Це відрізняється від архітектури YOLO, де на цьому етапі використовується проміжний повністю з'єднаний шар замість згорткового фільтра.

У мережі SSD для кожної клітини на верхніх картах ознак задаються стандартні обмежувальні рамки, які розташовані у вигляді сітки, що відповідає конволюційному розташуванню рамок на карті. Кожна рамка має фіксоване положення відносно своєї клітини на карті. Для передбачення об'єктів у цих рамках, мережа визначає зміщення від стандартної форми рамки та оцінки класу, які показують наявність об'єкта певного класу в рамці. Для кожної рамки з заданої кількості k в конкретному місці, мережа обчислює s оцінок класів та чотири зміщення відносно первісної форми рамки. В результаті, на кожному місці карти ознак застосовуються $(s + 4)k$ фільтрів, що генерують $(s + 4)kmp$ вихідних сигналів для карти ознак розміром $m \times n$. Цей підхід до стандартних рамок схожий на метод використання якірних рамок у Faster R-CNN, але з важливою відмінністю: у SSD ці рамки застосовуються до декількох карт ознак з різною роздільною здатністю. Таке застосування рамок на кількох картах ознак дозволяє ефективно покривати різноманітність можливих форм і розмірів об'єктів, які можуть зустрічатися на зображеннях.

Мета тренування SSD визначається за принципами MultiBox, але розширена для обробки декількох категорій об'єктів. Нехай $x_{ij}^p = \{0,1\}$ буде індикатором, що зіставляє i -ту стандартну рамку з j -тою істинною рамкою p -тої категорії об'єктів. У стратегії зіставлення, описаній вище, ми можемо мати суму $\sum_i \sum_j x_{ij}^p \geq 1$. Загальна втрата об'єктивності є зваженою сумою втрат локалізації (loc) та втрат довіри (conf):

$$L(x, c, l, g) = \frac{1}{N} (L_{\text{conf}}(x, c) + \alpha L_{\text{loc}}(x, l, g)) \quad (3.1)$$

де N - це кількість зіставлених стандартних рамок. Якщо $N = 0$, то втрата встановлюється на 0. Втрата локалізації - це гладкий L1 втрат між передбачуваною рамкою (l) та істинною рамкою (g) параметрів. Подібно до Faster R-CNN, ми виконуємо регрес до зміщень для центру (c_x, c_y) стандартної обмежувальної рамки (d) та її ширини (w) та висоти (h).

$$L_{\text{loc}}(x, l, g) = \sum_{i \in \text{Pos}} \sum_{m \in \{cx, cy, w, h\}} x_{ij}^k \text{smooth}_{L1}(l_i^m - \widehat{g}_j^m) \\ \widehat{g}_j^{cx} = \frac{(g_j^{cx} - d_j^{cx})}{d_i^w}, \quad \widehat{g}_j^{cy} = \frac{(g_j^{cy} - d_j^{cy})}{d_i^h}, \quad (3.2) \\ \widehat{g}_j^w = \log\left(\frac{g_j^w}{d_i^w}\right), \quad \widehat{g}_j^h = \log\left(\frac{g_j^h}{d_i^h}\right).$$

Втрата впевненості є втратою softmax за впевненостями декількох класів.

$$L_{\text{conf}}(x, c) = - \sum_{i \in \text{Pos}} x_{ij}^p \log(\widehat{c}_i^p) - \sum_{i \in \text{Neg}} \log(\widehat{c}_i^0) \text{ where } \widehat{c}_i^p = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)} \quad (3.3)$$

Та ваговий коефіцієнт α встановлюється рівним 1 шляхом перехресної перевірки.

3.2 Архітектура YOLOv5

YOLOv5 була випущена через кілька місяців після YOLOv4 у 2020 році Гленом Джочером, засновником та генеральним директором компанії Ultralytics. Він використовує багато покращень YOLOv4, але розроблений на Pytorch замість Darknet. YOLOv5 включає алгоритм від Ultralytics, який називається AutoAnchor. Цей інструмент попереднього тренування перевіряє

та налаштовує якорні рамки, якщо вони погано підходять для набору даних та параметрів тренування, таких як розмір зображення. Спочатку він застосовує функцію k-середніх до міток набору даних для генерації початкових умов для алгоритму генетичної еволюції (GE). Потім алгоритм GE розвиває ці якорі протягом 1000 поколінь за замовчуванням, використовуючи втрату СІоU та Найкращий Можливий Відклик як функцію пристосування. На рисунку 3.2 показана детальна архітектура YOLOv5.

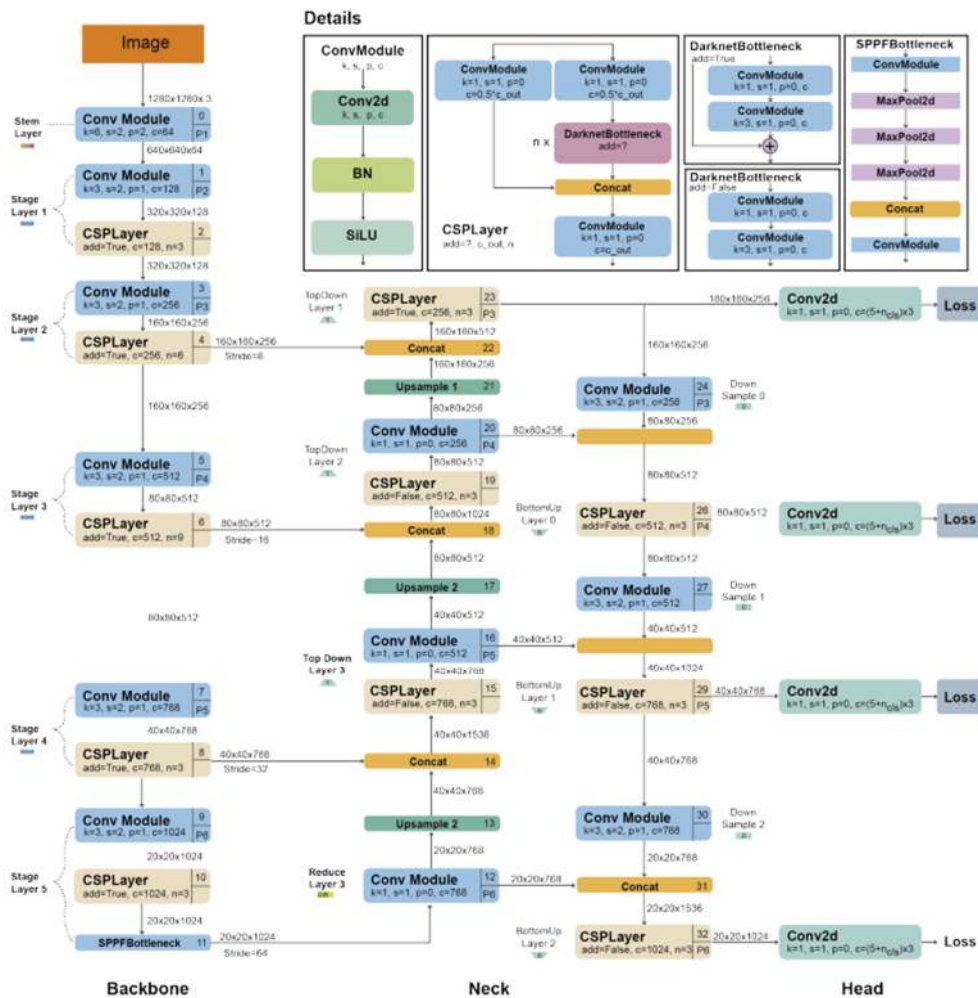


Рисунок 3.2 – Архітектура YOLOv5 [14]

Основою є модифікований CSPDarknet53, який починається зі Stem, шару зі зсувною конволюцією з великим розміром вікна для зменшення використання пам'яті та обчислювальних витрат; за ним слідує конволюційні шари, які витягують відповідні ознаки з вхідного зображення.

Шар SPPF (швидке просторове пірамідальне пулінг) та наступні конволюційні шари обробляють ознаки на різних масштабах, тоді як шари збільшення підвищують роздільну здатність карти ознак. Шар SPPF має на меті прискорити обчислення мережі шляхом пулінгу ознак різних масштабів у карту ознак фіксованого розміру. За кожною конволюцією слідує нормалізація пакету (BN) та активація SiLU. Шия використовує SPPF та модифікований CSP-PAN, тоді як голова схожа на YOLOv3.

YOLOv5 використовує кілька аугментацій, таких як Мозаїка, копіювання вставка, випадкова афінна трансформація, MixUp, аугментація HSV, випадкове горизонтальне перевертання, а також інші аугментації з пакета `albumentations`. Він також покращує чутливість сітки, щоб зробити її більш стабільною до стрибкоподібних градієнтів.

YOLOv5 надає п'ять масштабованих версій: YOLOv5n (nano), YOLOv5s (small), YOLOv5m (medium), YOLOv5l (large) та YOLOv5x (extra large), де ширина та глибина конволюційних модулів варіюються для конкретних застосувань та вимог обладнання. Наприклад, YOLOv5n та YOLOv5s є легкими моделями, призначеними для пристроїв із низькими ресурсами, тоді як YOLOv5x оптимізовано для високої продуктивності, хоча це йде на шкоду швидкості.

YOLOv5 є відкритим джерелом та активно підтримується Ultralytics, з понад 250 учасниками та частими нововведеннями. YOLOv5 легко використовувати, навчати та розгортати. Ultralytics надає мобільну версію для iOS та Android, а також багато інтеграцій для розмітки, навчання та розгортання.

3.3 Архітектура YOLOv8

Зображена на (рисунок 3.3) архітектура YOLOv8, це модель глибокого навчання для завдань виявлення об'єктів. Архітектура YOLO (You Only Look Once) відома своєю швидкістю та ефективністю в реальному часі, що

дозволяє використовувати її для завдань з виявлення об'єктів у реальному часі.

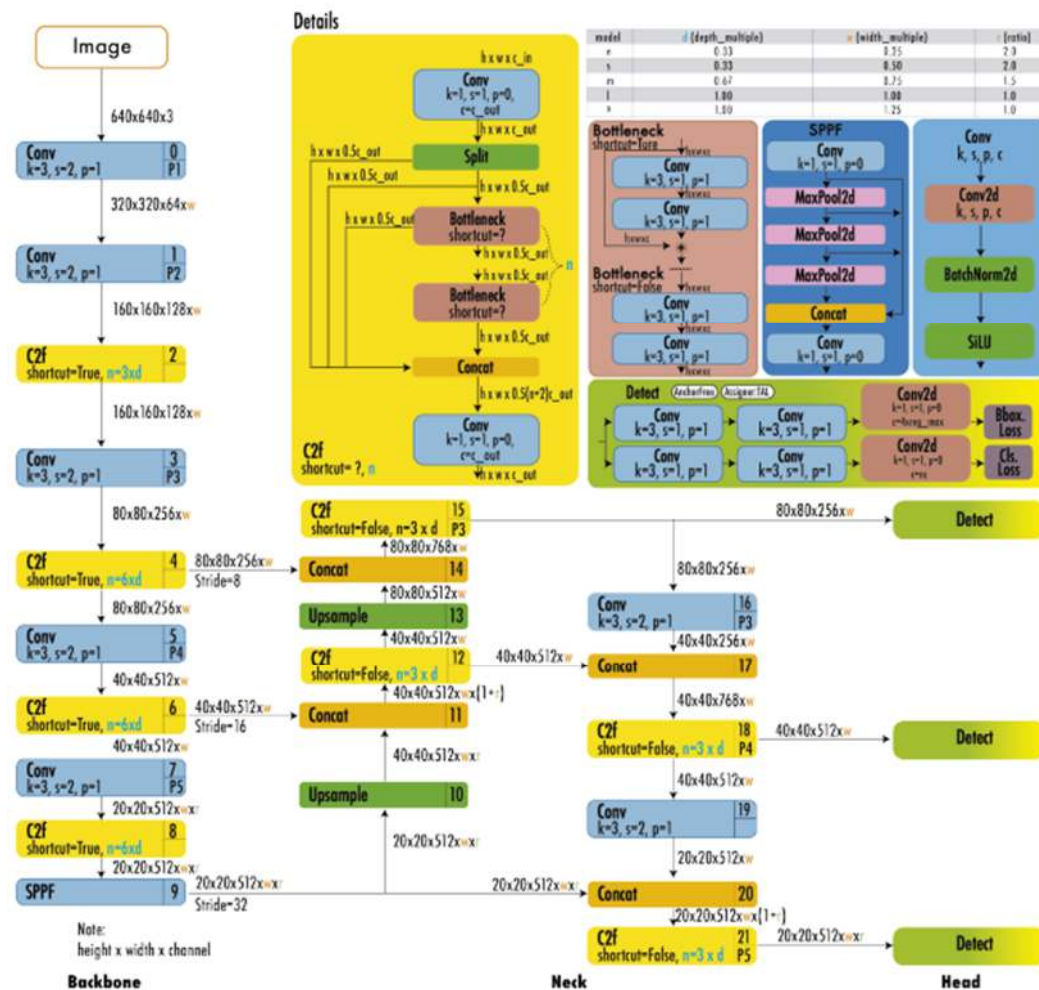


Рисунок 3.3 – Архітектура YOLOv8 [14]

На рисунку 3.3 представлена детальна архітектура YOLOv8. Вона використовує схожий каркас, як і YOLOv5, але з деякими змінами в CSPLayer, який тепер називається модулем C2f. Модуль C2f (частковий перехресний флакон з двома згортками) поєднує високорівневі характеристики з контекстною інформацією для підвищення точності детекції.

YOLOv8 використовує модель без якорів з розділеним головним блоком для незалежної обробки завдань визначення об'єктів, класифікації та регресії. Такий дизайн дозволяє кожній гілці концентруватися на своєму завданні і підвищує загальну точність моделі. На вихідному шарі YOLOv8

вони використовували сигмоїдну функцію як активаційну функцію для оцінки наявності об'єкта, що представляє ймовірність, що обмежувальний прямокутник містить об'єкт. Для ймовірностей класів використовується функція `softmax`, що представляє ймовірності належності об'єктів до кожного можливого класу.

YOLOv8 використовує функції втрат CIOU та DFL для втрат обмежувального прямокутника та двійкову крос-ентропію для втрат класифікації. Ці втрати покращили продуктивність детекції об'єктів, особливо при роботі з меншими об'єктами.

YOLOv8 також надає модель семантичної сегментації під назвою модель YOLOv8-Seg. Основою є екстрактор характеристик CSPDarknet53, за яким слідує модуль C2f замість традиційної архітектури шиї YOLO. За модулем C2f слідує дві голови сегментації, які навчаються передбачати маски семантичної сегментації для вхідного зображення.

Модель має схожі детектувальні голови до YOLOv8, що складаються з п'яти модулів детекції та шару передбачення. Модель YOLOv8-Seg досягла найкращих результатів на різних бенчмарках детекції об'єктів та семантичної сегментації, зберігаючи високу швидкість та ефективність.

YOLOv8 можна запускати з командного рядка (CLI), або також можна встановити як пакет PIP. Крім того, він поставляється з численними інтеграціями для маркування, тренування та впровадження.

4 НАБІР ДАНИХ ДЛЯ НАВЧАННЯ НЕЙРОМЕРЕЖІ

Для тренування нейромережі, яка спеціалізується на ідентифікації об'єктів на зображеннях, отриманих за допомогою повітряного оптичного моніторингу, було обрано датасет Dataset for Object Detection in Aerial images. DOTA - це велика колекція зображень з високою роздільною здатністю. Він містить тисячі зображень, на яких позначені різні об'єкти, такі як транспортні засоби, літаки, кораблі, бейсбольні поля та інші [15]. Цей датасет використовується для тренування та оцінки алгоритмів, що дозволяють виявляти та класифікувати об'єкти на аерофотознімках. Він включає велику кількість анотованих об'єктів і є особливо корисним для розробки та тестування алгоритмів комп'ютерного зору, оскільки представляє виклик через різноманітність та складність зображень.

4.1 Розмір зображення

Аерофотознімки зазвичай значно більші за розміром порівняно з тими, що містяться у наборах даних природних зображень. Початковий розмір зображень у нашому наборі даних варіюється від приблизно 800×800 до близько 4000×4000 , тоді як більшість зображень у звичайних наборах даних (наприклад, PASCAL-VOC і MSCOCO) не перевищують 1000×1000 . Ми робимо анотації на оригінальному повному зображенні без його розділення на частини, щоб уникнути випадків, коли один екземпляр розділяється на різні частини.

4.2 Різноманітність орієнтацій екземплярів

Як показано на рисунку 4.1, набір даних досягає гарного балансу в екземплярах різних напрямків, що значно допомагає у навчанні надійного

детектора. Більше того, набір даних DOTA близький до реальних сцен, оскільки у реальному світі звичайно зустрічається велика варіація об'єктів.

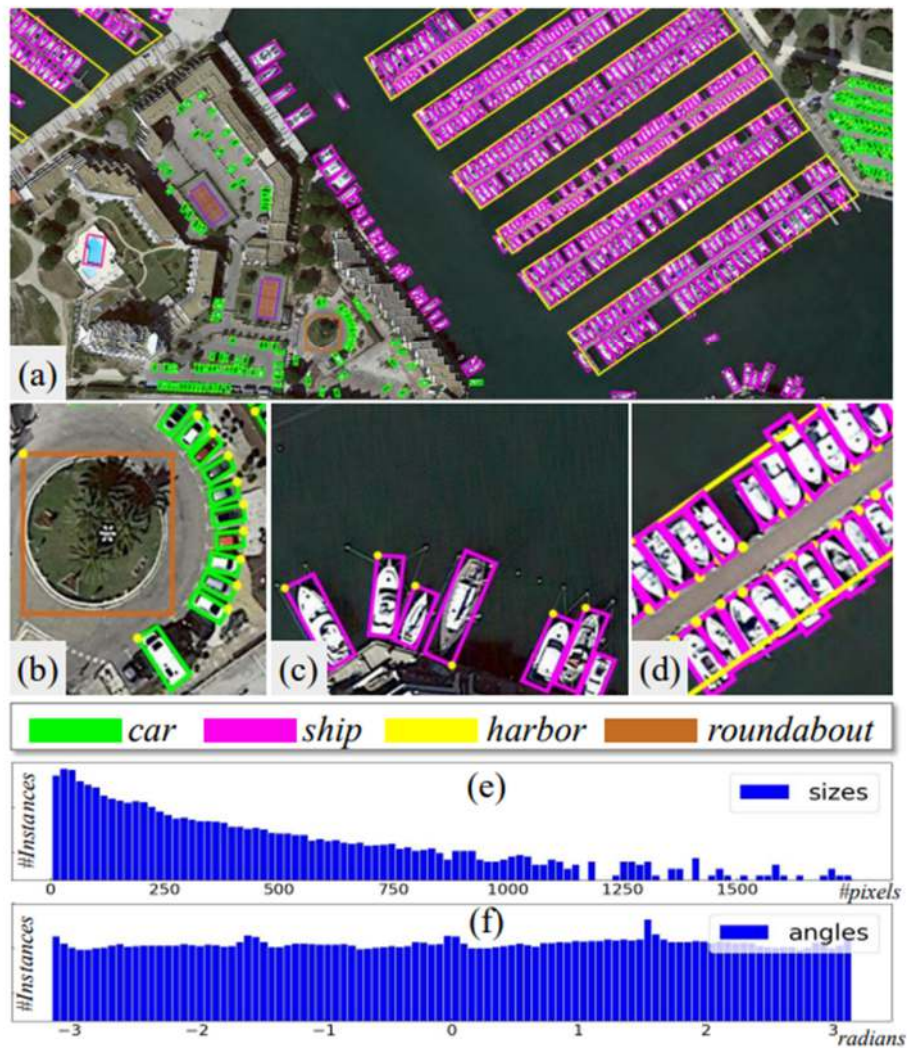


Рисунок 4.1 - Приклад з DOTA. (a) Типове зображення в DOTA, яке містить багато інстанцій у різних категоріях. (b) Ілюстрація різноманіття орієнтацій та розмірів інстанцій. (c),(d) Ілюстрації рідкісних та щільних інстанцій відповідно. Тут ми бачимо чотири з п'ятнадцяти можливих категорій у DOTA. Приклади, показані на (b),(c),(d), вирізані з джерельного зображення (a). Гістограми (f),(g) показують розподіл інстанцій згідно з розміром та орієнтацією в DOTA [16]

4.3 Інформація просторового розрешення

Важливість просторового розрішення для завдання виявлення є подвійною. По-перше, воно дозволяє моделі бути більш адаптивною та надійною для різноманітних об'єктів однієї категорії. Відомо, що об'єкти виглядають меншими, коли їх спостерігати з відстані. Той самий об'єкт різних розмірів може ускладнити роботу моделі та погіршити класифікацію. Однак, модель може звертати більше уваги на форму з наданою інформацією про розрішення замість розміру об'єктів. По-друге, це краще для детальної класифікації. Наприклад, буде просто відрізнити маленький човен від великого військового корабля.

Просторове розрішення також може бути використане для фільтрації неправильно позначених викидів у нашому наборі даних, оскільки внутрішньокласові різноманітності фактичних розмірів для більшості категорій обмежені. Викиди можна виявити, вибравши об'єкти, розмір яких значно відрізняється від розмірів об'єктів тієї ж категорії у невеликому діапазоні просторового розрішення.

5 ПРАКТИЧНІ РЕЗУЛЬТАТИ

Для реалізації методу ідентифікації об'єктів на зображеннях повітряного оптичного моніторингу було обрано середовище Google Colab Pro, яке забезпечує ефективність і гнучкість у виконанні обчислень. Мова програмування, яка лягла в основу реалізації, - Python, а як основний фреймворк було використано PyTorch, що дозволяє ефективно працювати з неймережами.

Тренування неймережі виконувалося на датасеті DOTA v1.5, який є досить репрезентативним і містить широкий спектр об'єктів для ідентифікації. У рамках проекту було треновано дві моделі неймереж: YOLOv8 nano та YOLOv8 small. Модель YOLOv8 nano пройшла процес тренування у 10 епох з використанням високопродуктивного GPU A100, що дозволило забезпечити швидке і ефективне навчання. Тренування моделі YOLOv8 small було проведено у 20 епох, використовуючи GPU V100, який також забезпечує високу продуктивність обчислень.

5.1 Результати тренування YOLOv8 nano

Тренування моделі YOLOv8 nano на датасеті DOTA v1.5 за допомогою GPU NVIDIA A100-SXM4-40GB, 40514MiB зайняло 0.135 годин (рисунок 5.1).

```

Epoch   GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
10/10   10.2G    1.176    0.8664    1.141     6          640: 100%|██████████| 89/89 [00:35<00:00, 2.49it/s]
      Class  Images  Instances  Box(P  R  mAP50  mAP50-95): 100%|██████████| 29/29 [00:00<00:00, 3.48it/s]
      all    458    69565    0.762    0.313    0.359    0.212

10 epochs completed in 0.135 hours.
Optimizer stripped from runs/obb/train/weights/last.pt, 6.5MB
Optimizer stripped from runs/obb/train/weights/best.pt, 6.5MB

```

Рисунок 5.1 – Час витрачений на тренування YOLOv8_n у 10 епох

На рисунку 5.2 можна побачити як змінюються метрики впродовж

тренування YOLOv8 nano, що відбувалося на протязі 10 епох Плавні криві (позначені помаранчевим кольором) відображають усереднені значення метрик для візуальної чистоти і можуть бути корисними для ідентифікації загальних тенденцій, в той час як сині лінії показують актуальні значення на кожній епосі.

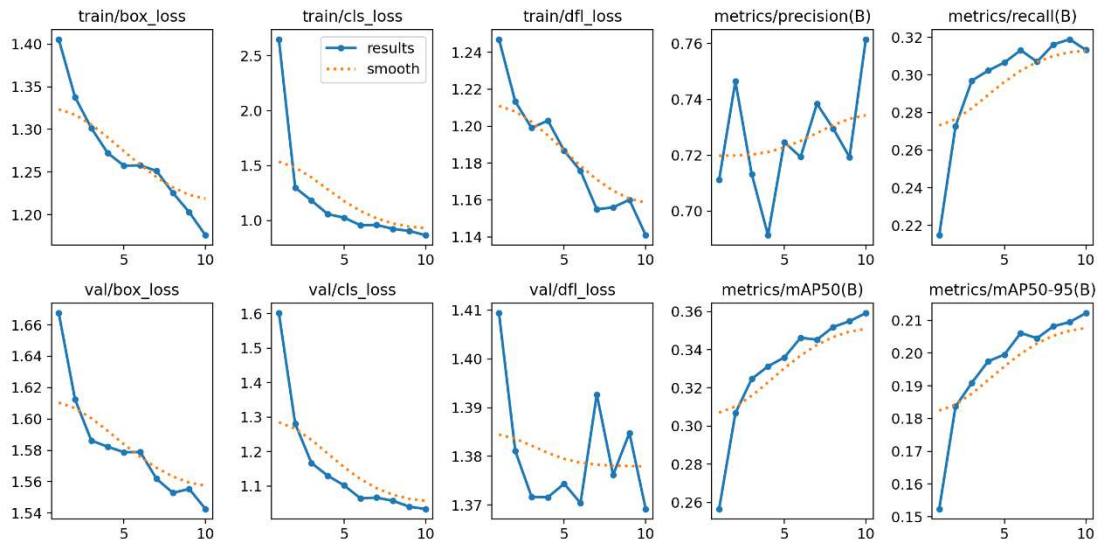


Рисунок 5.2 – Динаміка показників тренування та валідації за епохами для моделі YOLOv8 nano

Train/Box Loss і Val/Box Loss - показують втрати (loss) з прив'язки об'єктів (bounding box) для тренувального (таблиця 5.1). і валідаційного (таблиця 5.2) наборів даних відповідно. Втрати зменшуються з часом, що вказує на поліпшення точності прогнозування розташування об'єктів моделлю

Train/Cls Loss і Val/Cls Loss - показують втрати класифікації для тренувального і валідаційного наборів. Ці метрики вказують на здатність моделі правильно класифікувати об'єкти.

Train/Dfl Loss і Val/Dfl Loss - показують втрати в розпізнаванні деталей об'єкта для тренувального і валідаційного наборів.

Metrics/Precision(B) і Metrics/Recall(B) - точність і повнота моделі для тренувального набору даних. Точність показує відсоток правильно

ідентифікованих об'єктів серед усіх ідентифікованих, тоді як повнота показує відсоток правильно ідентифікованих об'єктів серед усіх реальних об'єктів.

Таблиця 5.1 – Показники тренування за епохами для моделі YOLOv8 nano

epoch	train/ box_loss	train/ cls_loss	train/ dfl_loss	metrics/ precision(B)	metrics/ recall(B)	metrics/ mAP50(B)	metrics/ mAP50- 95(B)
1	1.4056	2.6489	1.247	0.71117	0.21479	0.25661	0.15237
2	1.3377	1.2996	1.2133	0.74644	0.27267	0.3068	0.18366
3	1.3014	1.1818	1.199	0.71328	0.29691	0.32468	0.19092
4	1.2722	1.0585	1.203	0.69137	0.30222	0.33127	0.19749
5	1.2574	1.0262	1.1866	0.72471	0.30657	0.33594	0.19955
6	1.2577	0.95749	1.1758	0.71936	0.3131	0.34622	0.20614
7	1.2512	0.95977	1.1549	0.73844	0.30694	0.34522	0.20454
8	1.2255	0.9242	1.156	0.72954	0.31611	0.35186	0.20818
9	1.2029	0.90551	1.1601	0.71932	0.31888	0.35488	0.20947
10	1.1758	0.86639	1.1409	0.7615	0.31321	0.35914	0.21223

Таблиця 5.2 – Показники валідації за епохами для моделі YOLOv8 nano

epoch	val/ box_loss	val/ cls_loss	val/ dfl_loss	lr/pg0	lr/pg1	lr/pg2
1	1.6675	1.6016	1.4094	0.00016479	0.00016479	0.00016479
2	1.6124	1.2809	1.3811	0.00029865	0.00029865	0.00029865
3	1.5862	1.1665	1.3717	0.0003995	0.0003995	0.0003995
4	1.5823	1.13	1.3716	0.0003515	0.0003515	0.0003515
5	1.5788	1.1019	1.3744	0.0003515	0.0003515	0.0003515
6	1.5789	1.0644	1.3704	0.000302	0.000302	0.000302
7	1.5618	1.0664	1.3927	0.0002525	0.0002525	0.0002525
8	1.5529	1.0571	1.3762	0.000203	0.000203	0.000203
9	1.5554	1.0401	1.3848	0.0001535	0.0001535	0.0001535
10	1.5426	1.0334	1.3692	0.000104	0.000104	0.000104

Metrics/mAP50(B) і Metrics/mAP50-95(B) - середні точності по множині порогів IoU (Intersection over Union) від 0.5 до 0.95. Це загальні метрики для оцінки якості об'єктних детекторів, які враховують і точність, і повноту по широкому спектру порогів IoU.

Графік на рисунку 5.3 показує залежність точності від впевненості моделі при класифікації різних класів об'єктів. Точність (Precision) - це частка правильно ідентифікованих позитивних випадків серед усіх позитивних випадків, що були передбачені моделлю. Впевненість (Confidence) - це ймовірність, з якою модель вважає, що передбачення є правильним.

На графіку кожен колір відповідає різному класу об'єктів, таким як літаки, кораблі, автомобілі тощо. Криві показують, як точність змінюється при збільшенні порога впевненості, тобто при якому порозі впевненості модель вирішує, що вона достатньо впевнена в своєму передбаченні, щоб його розглядати як позитивний результат.

Лінія "all classes 1.00 at 0.977" показує середній показник точності по всіх класах при високій впевненості (0.977).

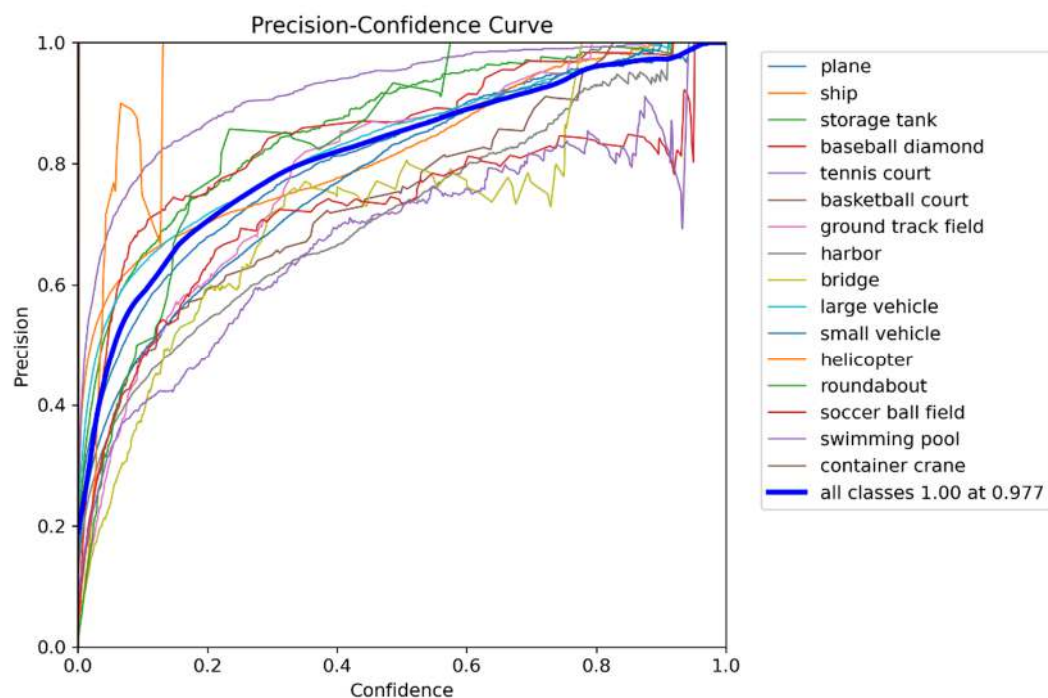


Рисунок 5.3 – Графік кривої точності-впевненості для YOLOv8 nano

На графіку відношення повноти до впевненості (рисунок 5.4), показано, як змінюється повнота детекції об'єктів (recall) при зміні порога впевненості детекції. Повнота вимірює частку правильно ідентифікованих позитивних випадків з усіх реальних позитивних випадків і є важливим показником для оцінки здатності моделі виявляти класи об'єктів. Лінія з підписом "всі класи" показує середню повноту для всіх класів при найнижчому порозі впевненості (у цьому випадку 0.0), що становить 0.42. Це означає, що в середньому модель правильно ідентифікує 42% існуючих об'єктів, коли вона взагалі не враховує впевненість своїх прогнозів.

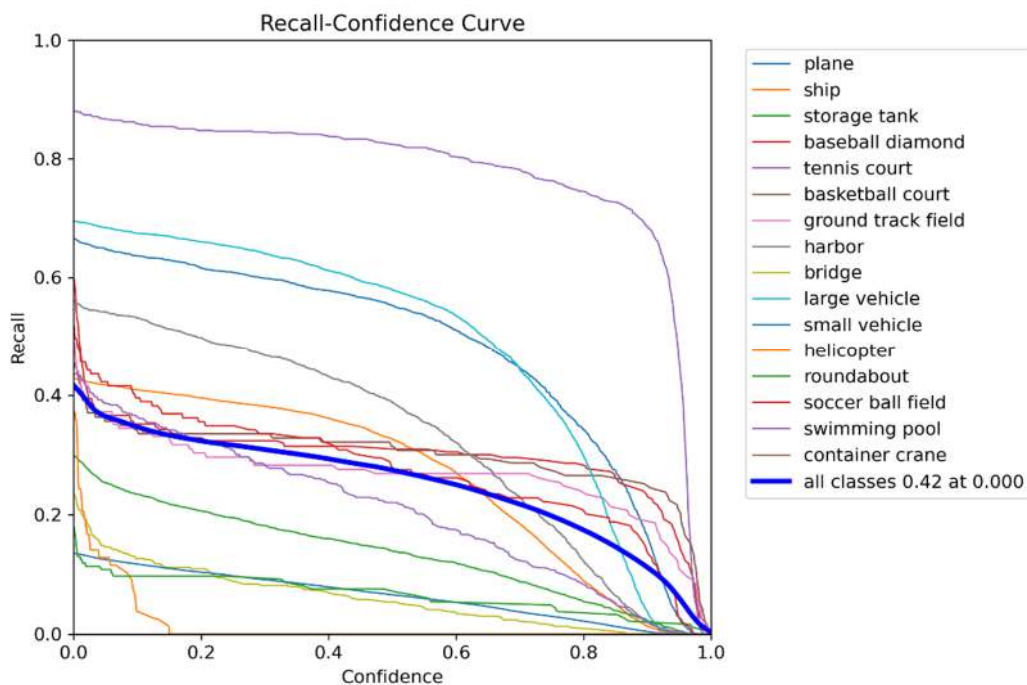


Рисунок 5.4 – Графік кривої відкликання-впевненості

Графік (рисунок 5.5) - це крива F1-впевненості, яка показує залежність F1-оцінки від рівня впевненості для різних класів об'єктів. F1-оцінка є гармонійним середнім значенням точності та повноти, забезпечуючи баланс між цими двома метриками. Це міра точності моделі, яка враховує як точність (кількість правильних позитивних результатів, поділена на кількість усіх позитивних результатів), так і повноту (кількість правильних позитивних результатів, поділена на кількість позитивних результатів, які

мали бути отримані). Лінія з позначкою "усі класи" вказує на середню F1-оцінку по всіх класах при певному рівні впевненості, який у цьому випадку становить 0.39 при рівні впевненості 0.257. Це означає, що в середньому модель має F1-оцінку 0.39, коли вона впевнена в своїх прогнозах на 25.7%.

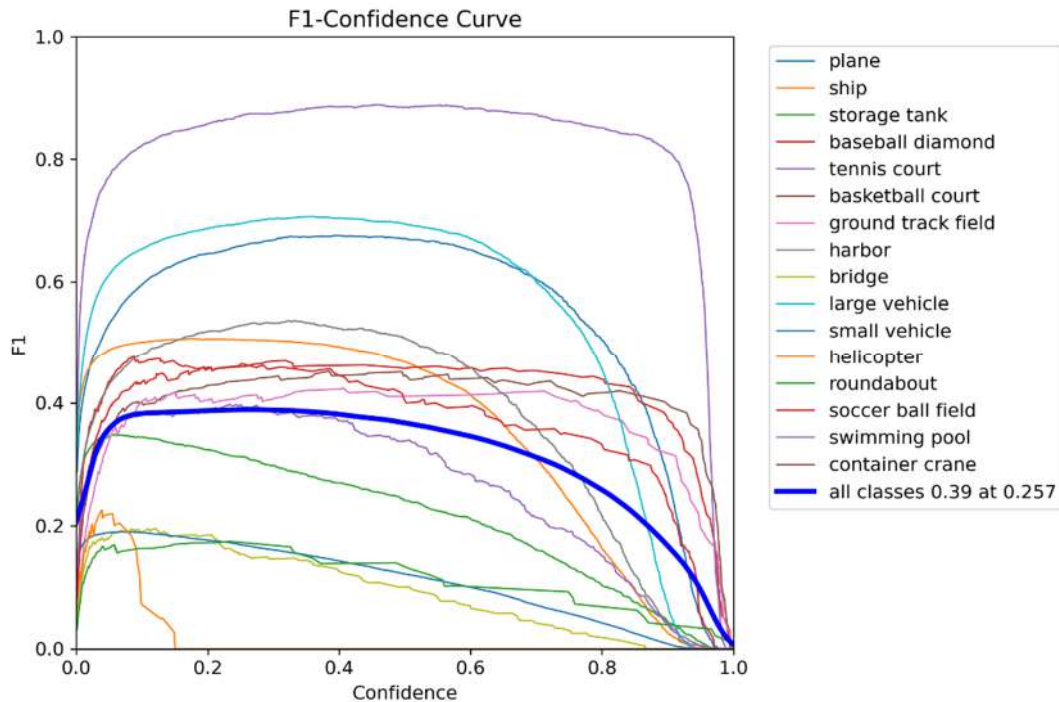


Рисунок 5.5 – Графік кривої F1 Confidence

Графік зображений на рисунку 5.6 - це крива точності-повноти, яка є важливим інструментом для визначення якості бінарних класифікаторів у задачі об'єктного виявлення. Кожна лінія на графіку відповідає одному класу об'єктів і показує залежність між точністю (precision) і повнотою (recall) при різних порогах впевненості.

Точність (відсоток правильно ідентифікованих об'єктів серед усіх об'єктів, що модель вважає ідентифікованими) і повнота (відсоток правильно ідентифікованих об'єктів серед усіх реальних об'єктів у даних) зазвичай мають обернену залежність: збільшення одного показника може призводити до зменшення іншого. На графіку також вказано загальний показник mAP (середня точність по всіх класах) при інтервалі впевненості 0.5, що в цьому

випадку складає 0.359. Це означає, що в середньому модель має точність близько 35.9% при ідентифікації об'єктів при впевненості 0.5.

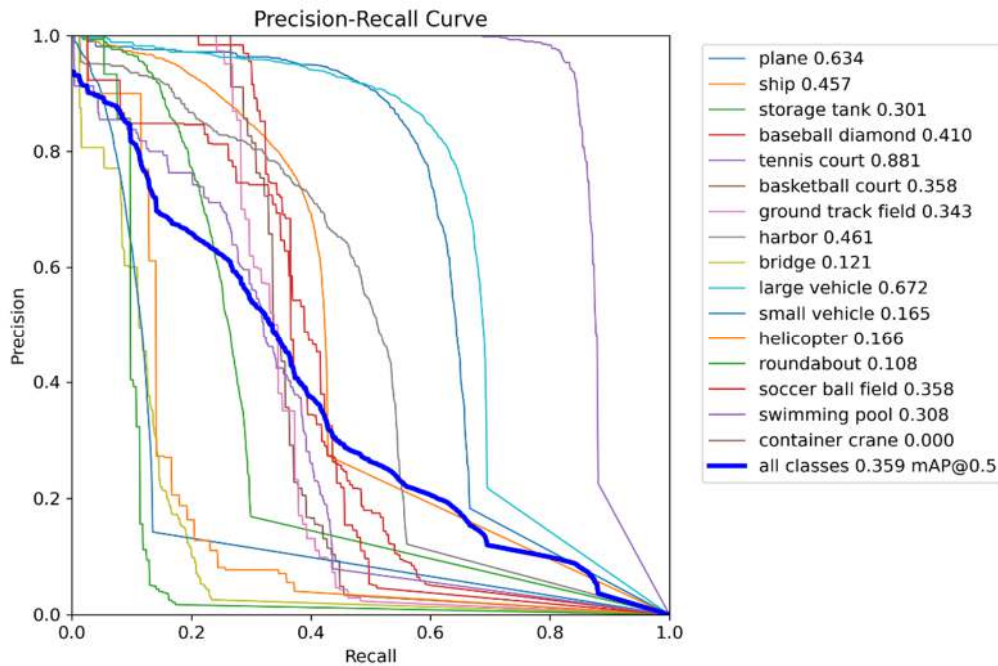


Рисунок 5.6 – Графік точності-повноти

5.2 Результати тренування YOLOv8 small

Тренування моделі YOLOv8 small на датасеті DOTA v1.5 за допомогою Tesla V100-SXM2-16GB, 16151MiB зайняло 0.473годин (рисунок 5.1).

```

Epoch   GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
20/20   12.1G    0.976    0.6382   1.099     17         640: 100%|██████████| 89/89 [00:48<00:00, 1.84it/s]
          Class  Images  Instances  Box(P  R  mAP50  mAP50-95): 100%|██████████| 29/29 [00:14<00:00, 2.06it/s]
          all    458    69565    0.78  0.372  0.424  0.255

20 epochs completed in 0.473 hours.
Optimizer stripped from runs/obb/train/weights/last.pt, 23.2MB
Optimizer stripped from runs/obb/train/weights/best.pt, 23.2MB

```

Рисунок 5.7 – Час тренування моделі YOLOv8 small на 20 епох

На рисунку 5.8 можна побачити як змінюються метрики впродовж тренування (таблиця 5.3) та валідації (таблиця 5.4) YOLOv8 small, що відбувалося на протязі 20 епох.

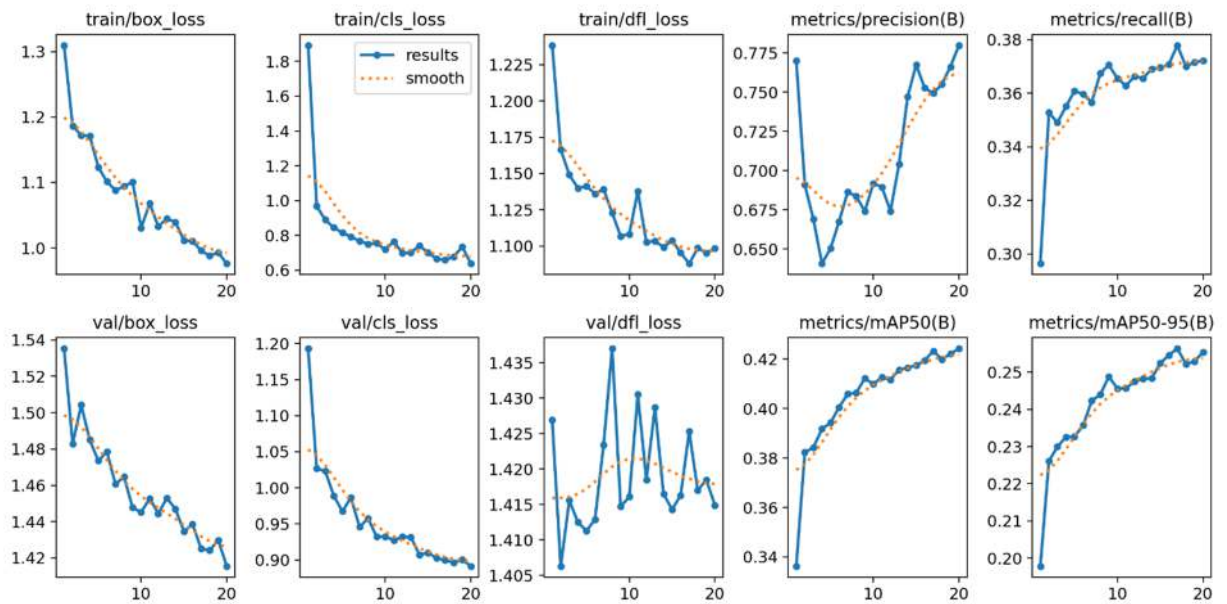


Рисунок 5.8 – Динаміка показників тренування та валідації за епохами для моделі YOLOv8 small

Порівняємо графіки моделей YOLOv8 Nano та YOLOv8 Small, проаналізуємо тенденції та значення для кожного показника за епохи. Розглянемо кожен графік та метрику по черзі:

`train/box_loss` і `val/box_loss` - Для YOLOv8 Nano втрата коробки зменшується стабільно протягом 10 епох як у тренуванні, так і у валідації, що свідчить про ефективне навчання моделі. Навпаки, YOLOv8 Small показує подібну тенденцію до зниження, але протягом 20 епох, що вказує на безперервне навчання та можливо більш стабільну конвергенцію завдяки більшій тривалості тренування.

`train/cls_loss` і `val/cls_loss` - Втрата класифікації для YOLOv8 Nano зменшується, але є деякі коливання у валідаційній втраті, що може вказувати на варіативність у навчанні або потребу в більшій кількості епох. YOLOv8 Small також демонструє зменшуючу тенденцію втрати класифікації. Коливання у валідаційній втраті менш виражені, можливо, це вказує на покращену стабільність із більшою кількістю епох тренування.

`train/dfl_loss` і `val/dfl_loss` - Втрата DFL для YOLOv8 Nano має помітні коливання, особливо у валідаційній втраті, що вказує на потенційне

перенавчання або на те, що модель ще не стабілізувалася. Втрата DFL для YOLOv8 Small здається стабільнішою та зменшується більш послідовно протягом 20 епох, що вказує на кращу узагальненість і навчання.

metrics/precision(B) і metrics/recall(B) - Для точності YOLOv8 Nano видно значну варіативність, що може вказувати на нестабільність у передбаченнях моделі. Метрика повернення більш стабільна і показує поступове поліпшення. YOLOv8 Small показує більш стабільне поліпшення і в точності, і в поверненні протягом 20 епох, що вказує на те, що вона навчається краще ідентифікувати відповідні інстанції з більшою послідовністю.

metrics/mAP50(B) і metrics/mAP50-95(B) - Значення mAP для YOLOv8 Nano зростають протягом епох із деякою варіативністю, що вказує на поліпшення точності моделі. Значення mAP для YOLOv8 Small також поліпшуються із часом, але тенденція гладша, що може свідчити про кращий загальний виступ і точність.

Висновок: хоча YOLOv8 Nano показує обнадійливі тенденції за 10 епох, YOLOv8 Small, натренована протягом 20 епох, демонструє більшу стабільність і послідовність у зниженні показників втрати та поліпшенні в точності, поверненні та mAP.

Це порівняння вказує на те, що YOLOv8 Small із додатковим часом тренування може бути більш надійною та ефективною для завдань виявлення об'єктів.

Таблиця 5.3 - Показники тренування за епохами для моделі YOLOv8 small

epoch	train/ box_loss	train/ cls_loss	train/ dfl_loss	metrics/ precision (B)	metrics/ recall (B)	metrics/ mAP50 (B)	metrics/ mAP50- 95(B)
1.0	1.3096	1.8927	1.2385	0.76996	0.29644	0.3362	0.19785
2.0	1.1865	0.96788	1.1665	0.69092	0.35278	0.3822	0.22606
3.0	1.1719	0.8897	1.1491	0.66897	0.3491	0.3842	0.22995
4.0	1.171	0.84462	1.14	0.64069	0.35514	0.3917	0.23256

Продовження таблиці 5.3

epoch	train/ box_loss	train/ cls_loss	train/ dfl_loss	metrics/ precision (B)	metrics/ recall (B)	metrics/ mAP50 (B)	metrics/ mAP50- 95(B)
5.0	1.1235	0.81603	1.1412	0.65033	0.3609	0.3942	0.23269
6.0	1.1012	0.79222	1.1361	0.66748	0.35964	0.4005	0.23569
7.0	1.0879	0.76822	1.1393	0.68637	0.35665	0.4058	0.24229
8.0	1.0941	0.75018	1.1228	0.68369	0.36748	0.4063	0.24396
9.0	1.1006	0.75518	1.1068	0.67405	0.37058	0.4122	0.24877
10.0	1.0308	0.71838	1.1083	0.69188	0.36558	0.4099	0.24558
11.0	1.0678	0.76526	1.1378	0.6895	0.36286	0.4126	0.24565
12.0	1.0328	0.69885	1.1027	0.67417	0.36636	0.4117	0.24748
13.0	1.0447	0.70159	1.1036	0.70391	0.36562	0.4156	0.24824
14.0	1.0395	0.74047	1.0992	0.74691	0.36913	0.4164	0.24833
15.0	1.0113	0.70101	1.1039	0.76742	0.36956	0.4174	0.25236
16.0	1.0107	0.66486	1.0954	0.75261	0.37083	0.4194	0.25454
17.0	0.99588	0.65923	1.0879	0.74925	0.37794	0.4233	0.25632
18.0	0.98803	0.67753	1.0987	0.75501	0.37003	0.4199	0.25218
19.0	0.99294	0.73492	1.0952	0.76623	0.37159	0.4221	0.25289
20.0	0.97602	0.63817	1.0986	0.77988	0.37224	0.4242	0.25534

Таблиця 5.4 – Показники валідації за епохами для моделі YOLOv8 small

epoch	val/ box_loss	val/ cls_loss	val/ dfl_loss	lr/pg0	lr/pg1	lr/pg2
1	1.5353	1.193	1.4269	0.00016479	0.00016479	0.00016479
2	1.4829	1.0266	1.4063	0.00031505	0.00031505	0.00031505
3	1.5042	1.023	1.4156	0.00044881	0.00044881	0.00044881
4	1.485	0.98877	1.4125	0.00042575	0.00042575	0.00042575
5	1.4735	0.96725	1.4113	0.00042575	0.00042575	0.00042575
6	1.4785	0.98633	1.4129	0.000401	0.000401	0.000401

Продовження таблиці 5.4

epoch	val/ box_loss	val/ cls_loss	val/ dfl_loss	lr/pg0	lr/pg1	lr/pg2
7	1.4607	0.94567	1.4234	0.00037625	0.00037625	0.00037625
8	1.4647	0.95757	1.437	0.0003515	0.0003515	0.0003515
9	1.4477	0.93199	1.4147	0.00032675	0.00032675	0.00032675
10	1.4452	0.9324	1.4161	0.000302	0.000302	0.000302
11	1.4528	0.92725	1.4305	0.00027725	0.00027725	0.00027725
12	1.4445	0.93202	1.4185	0.0002525	0.0002525	0.0002525
13	1.453	0.93142	1.4287	0.00022775	0.00022775	0.00022775
14	1.447	0.90708	1.4165	0.000203	0.000203	0.000203
15	1.4347	0.90986	1.4143	0.00017825	0.00017825	0.00017825
16	1.4386	0.90235	1.4163	0.0001535	0.0001535	0.0001535
17	1.4252	0.89972	1.4253	0.00012875	0.00012875	0.00012875
18	1.4242	0.89602	1.417	0.000104	0.000104	0.000104
19	1.4298	0.9004	1.4185	7.93E-05	7.93E-05	7.93E-05
20	1.4155	0.89126	1.4149	5.45E-05	5.45E-05	5.45E-05

Графік кривої точності-впевненості (рисунок 5.9) для різних класів об'єктів, які були ідентифіковані нейромережею. Кожен колір на графіку відповідає певному класу об'єкта, як наприклад літаки, кораблі, басейни тощо. Горизонтальна вісь показує рівень впевненості від 0 до 1, де 1 означає повну впевненість моделі в своїх передбаченнях. Вертикальна вісь показує точність від 0 до 1, де вищі значення вказують на більшу точність передбачень моделі.

На графіку залежності відкликання від впевненості (рисунок 5.10) криві показують, як відкликання змінюється зі зміною порога впевненості. При низькому порозі впевненості модель може мати високий відкликання, оскільки вона менш строга і може ідентифікувати більше об'єктів, але серед них можуть бути помилкові позитиви. Зі збільшенням порогу впевненості

відкликання зазвичай знижується, що означає, що модель стає більш селективною та зменшує кількість помилкових позитивів, але також може упустити деякі реальні об'єкти.

Криві F1-оцінки (рисунок 5.11) показують баланс між точністю та повнотою класифікації для кожного класу об'єктів. F1-оцінка варіюється від 0 до 1, де 1 означає найкращу можливу ефективність класифікації. Для кожної кривої можна спостерігати, що зі збільшенням довірчого інтервалу (відсотка довіри, що прогноз є правильним), F1-оцінка спочатку зростає, досягаючи піку, а потім знижується.

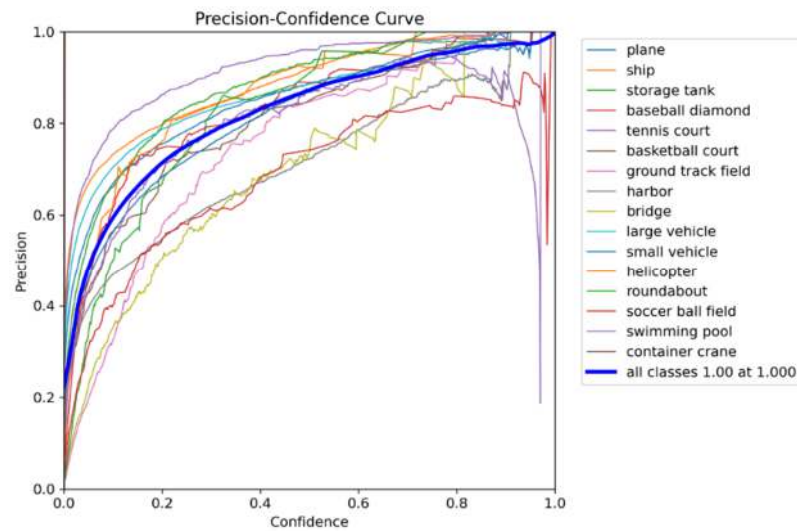


Рисунок 5.9 - Графік кривої точності-впевненості для YOLOv8 small

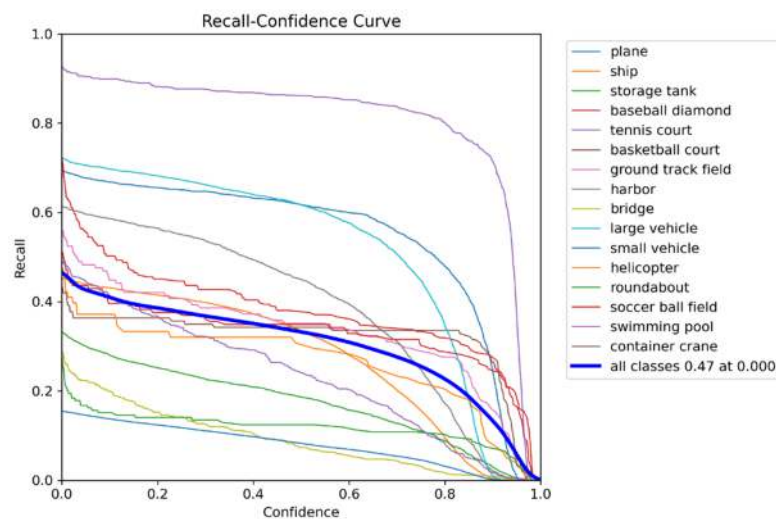


Рисунок 5.10 – Графік кривої відкликання-впевненості

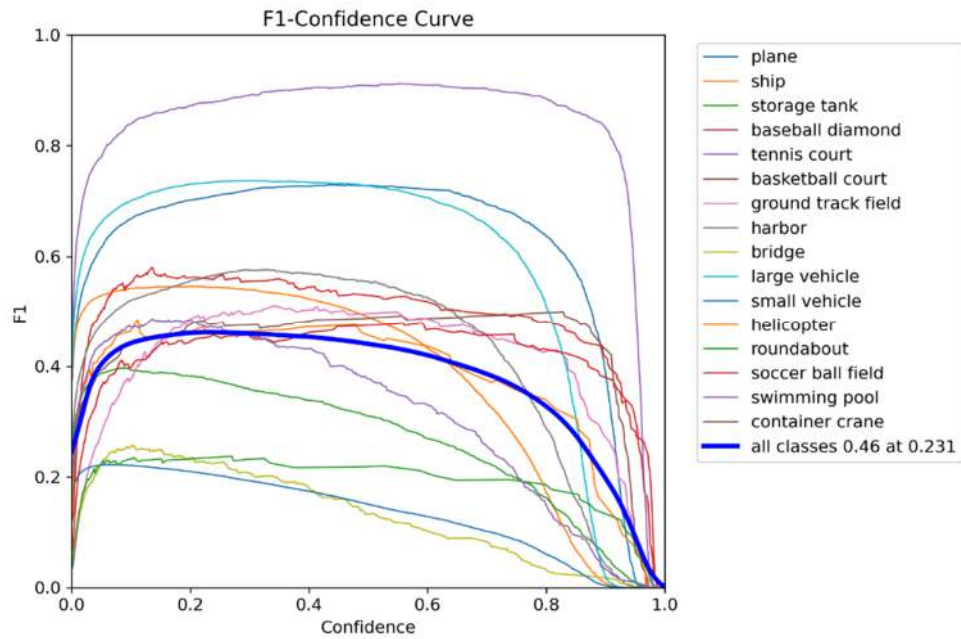


Рисунок 5.11 - Графік кривої F1 Confidence

На графіку точність-повнота (рисунок 5.12) Криві представляють наступні характеристики:

- точність на осі Y: Відсоток правильно ідентифікованих об'єктів з усіх об'єктів, що були класифіковані як позитивні;
- повнота на осі X: Відсоток правильно ідентифікованих об'єктів з усіх реально позитивних випадків у датасеті.

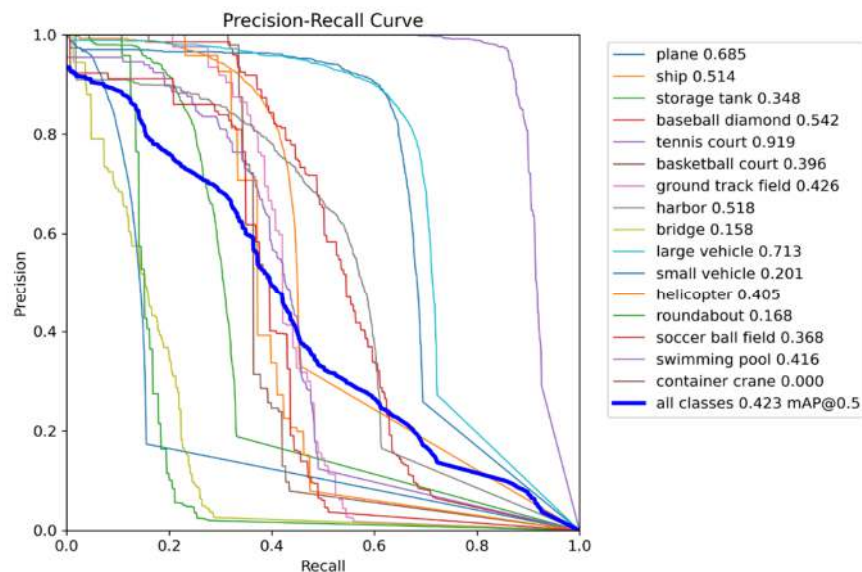


Рисунок 5.12 - Графік точності-повноти

5.3 Результати ідентифікації об'єктів на зображеннях повітряного оптичного моніторингу

Результати ідентифікації об'єктів на аерофотознімках моделями YOLOv8 nano та YOLOv8 small продемонстровані на рисунках 5.13, 5.14, 5.15, 5.16, 5.17, 5.18, 5.19, 5.20.

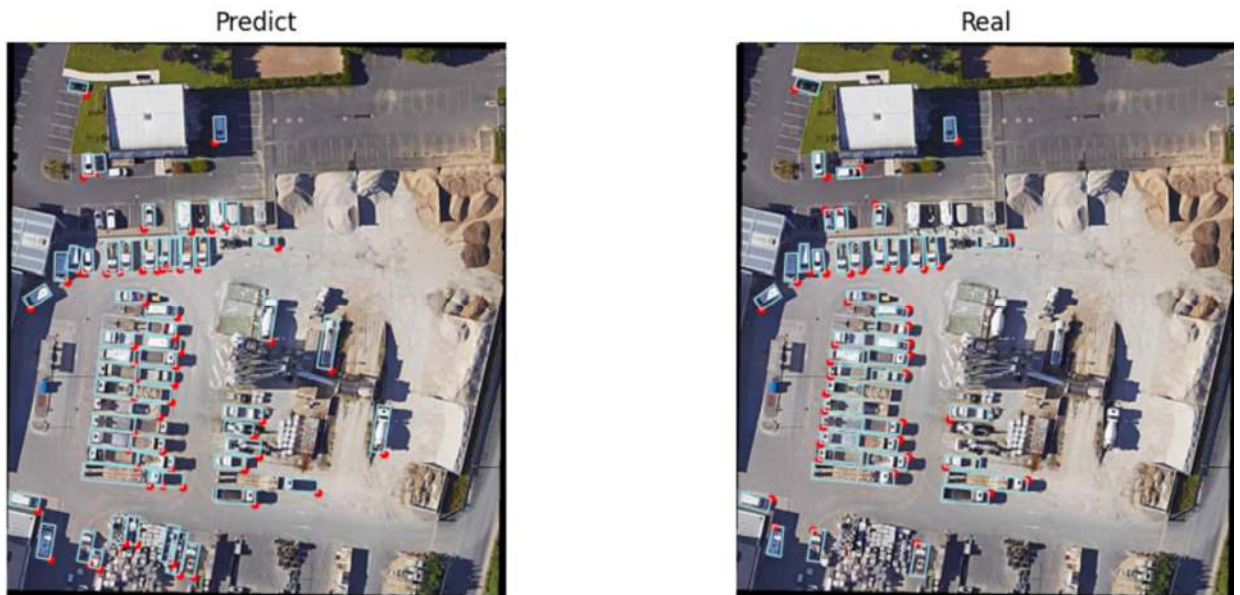


Рисунок 5.13 – Результат ідентифікації транспортних засобів YOLOv8 nano

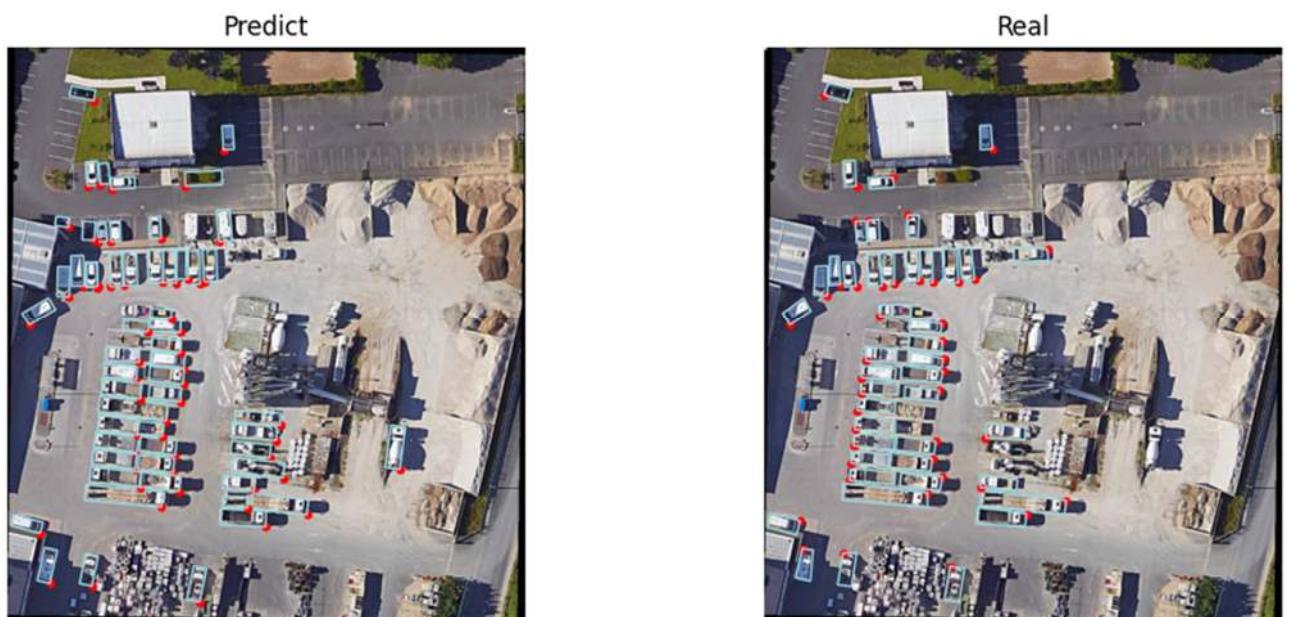


Рисунок 5.14 – Результат ідентифікації транспортних засобів YOLOv8 small

На лівому зображенні рамки відображають, де модель передбачає наявність об'єктів, тоді як на правому зображенні рамки показують фактичні місцезнаходження об'єктів.



Рисунок 5.15 – Ідентифікація тенісних кортів та автомобілей YOLOv8 nano



Рисунок 5.16 – Ідентифікація тенісних кортів та автомобілей YOLOv8 small

На основі проведеного аналізу тренувань моделей YOLOv8 nano та YOLOv8 small, які тренувалися відповідно на 10 та 20 епохах у середовищі Google Colab, можна зробити висновок про перевагу YOLOv8 small у

точності ідентифікації об'єктів. Додаткові епохи тренувань YOLOv8 small дозволили моделі краще адаптуватися до різноманітності даних, що проявилось у підвищенні метрик точності, відгуків та середньої точності по всіх класах (mAP). Особливо помітно це стало на графіках Precision-Confidence та Recall-Confidence, де лінії, що відповідають за YOLOv8 small, показують стабільніше зростання та вищі пікові значення.

Також важливо зазначити, що YOLOv8 small продемонстрував кращу здатність узагальнення, оскільки метрики валідації показали менше перекручування та надмірного підгону, що часто буває проблемою при недостатньому тренуванні. Це вказує на те, що YOLOv8 small має кращу здатність до запобігання перенавчання на тренувальних даних, що робить його більш надійним для використання у реальних умовах.

Окрім того, прогрес у точності та відгуку моделі YOLOv8 small може бути пов'язаний з її здатністю краще розпізнавати дрібні деталі та складності об'єктів, що особливо важливо у задачах ідентифікації об'єктів з різними масштабами. Загалом, YOLOv8 small продемонстрував більшу ефективність і надійність, що робить його переважним вибором для подальшого використання та розвитку.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи був проведений ретельний огляд сучасних методів та алгоритмів ідентифікації об'єктів на зображеннях, таких як SSD та YOLO. Були представлені їхні сильні та слабкі сторони, які висвітлюють компроміси між обчислювальною ефективністю та точністю.

Для роботи було обрано набір даних DOTA, який містить велику кількість зображень високої роздільної здатності, зроблених з повітря, що є викликаючим та різноманітним матеріалом для завдань виявлення об'єктів.

Завдяки потужним обчислювальним можливостям середовища Google Colab Pro були навчені моделі YOLOv8 nano та YOLOv8 small для ідентифікації об'єктів на зображеннях повітряного оптичного моніторингу. Ці моделі були спеціально вибрані за їх баланс продуктивності та ефективності, що робить їх добре пристосованими до обчислювальних вимог обробки зображень високої роздільної здатності.

Модель YOLOv8 nano продемонструвала надзвичайну швидкість і спритність у ідентифікації об'єктів, що робить її підходящою для реальних застосувань, де критичною є швидка обробка. З іншого боку, модель YOLOv8 small виявила вищу точність, що є важливішим у сценаріях, де точність є важливішою за швидкість.

Загалом, кваліфікаційна робота не тільки висвітлила можливості сучасних моделей виявлення об'єктів, але й надала ясний напрямок для майбутніх досліджень та розробок у аналізі аерофотознімків.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Зоу Ж., Чен К., Ші З., Гуо Ю., Є Дж. Виявлення об'єктів за 20 років: Огляд. В "Процедурах IEEE" (Proceedings of the IEEE). 2023. С. 257-276. DOI: 10.1109/JPROC.2023.3238524.
2. Гаріхаран Б., Арбелаез П., Гіршик Р., Малік Дж. Одночасне виявлення та сегментація. У збірнику: ECCV. Springer, 2014. С. 297–312.
3. Карпаті А., Фей-Фей Л. Глибокі візуально-семантичні вирівнювання для генерації описів зображень. У збірнику: CVPR. 2015. С. 3128–3137.
4. Канг К., Лі Х., Ян Дж., Зенг Х., Янг Б., Сяо Т., Чжан Ц., Ванг З., Ванг Р., Ванг Х. та ін. T-CNN: Tubelets with convolutional neural networks for object detection from videos. У "IEEE Transactions on Circuits and Systems for Video Technology", 2018, том 28, № 10, с. 2896–2907.
5. Кріжевський А., Суцкевер І., Гінтон Г. Е. Класифікація ImageNet з використанням глибоких згорткових нейронних мереж [Текст] / Кріжевський А., Суцкевер І., Гінтон Г. Е. – У матеріалах NIPS, 2012.
6. Гіршик Р., Донах'ю Дж., Даррелл Т., Малік Дж. Багаті ієрархії ознак для точного виявлення об'єктів та семантичної сегментації [Текст] / Гіршик Р., Донах'ю Дж., Даррелл Т., Малік Дж. – У матеріалах CVPR, 2014.
7. Розенблатт Ф. Перцептрон: сприймаючий та розпізнаючий автомат [Текст] / Розенблатт Ф. – Проект Пара, Лабораторія аеронавтики Корнелл, Буффало, 1957.
8. Мінський М., Паперт С. Перцептрон: введення до обчислювальної геометрії [Текст] / Мінський М., Паперт С. – Кембрідж, Массачусетс: MIT Press, 1969.
9. Мінський М.Л., Паперт С.А. Перцептрони: розширене видання [Текст] / Мінський М.Л., Паперт С.А. – Кембрідж, Массачусетс: MIT Press, 1988.

10. Ліннайнмаа С. Розклад Тейлора накопиченого округлення помилки [Текст] / Ліннайнмаа С. – BIT Numer Math, 1976. – Т.16, №2. – С.146–160..
11. Хохрайтер С., Шмідхубер Дж. Довга короткочасна пам'ять [Текст] / Хохрайтер С., Шмідхубер Дж. – Neural Comput, 1997. – Т.9, №8. – С.1735–1780.
12. Вакалопулу М., Христодулідіс С., Бургос Н., Колліот О., Лепетіт В. Основи глибинного навчання та згорткові нейронні мережі (CNN) / Марія Вакалопулу, Стергіос Христодулідіс, Нінон Бургос, Олів'є Колліот, Вінсент Лепетіт // Машинне навчання для розладів мозку. - Springer, 2023. - DOI: 10.1007/978-1-0716-3195-9_3. - HAL: hal-03957224v2.
13. Дхіллон А., Верма Г. К. Згорткова нейронна мережа: огляд моделей, методологій та застосувань до виявлення об'єктів // Прогрес в штучному інтелекті. – 2020. – Т. 9(85), С. 112. – DOI: 10.1007/s13748-019-00203-0.
14. Тервен Х., Кордова-Еспарза Д.М., Ромеро-Гонсалес Х.-А. (2023) Повний огляд архітектур YOLO у комп'ютерному зорі: від YOLOv1 до YOLOv8 та YOLO-NAS. Mach. Learn. Knowl. Extr., 2023, 5(4), с. 1680–1716. DOI: <https://doi.org/10.3390/make5040083>.
15. Собода А., Метод ідентифікації об'єктів по зображенням повітряного оптичного моніторингу // Modern problems of science, education and society. Proceedings of the 11th International scientific and practical conference. SPC “Sci-conf.com.ua”. Kyiv, Ukraine. 2024. с. 519-523. URL: <https://sci-conf.com.ua/xi-mizhnarodna-naukovo-praktichna-konferentsiya-modern-problems-of-science-education-and-society-8-10-01-2024-kiyiv-ukrayina-arhiv/> .
16. Xia G.-S., Bai X., Zhang L., Belongie S. DOTA: A Large-Scale Dataset for Object Detection in Aerial Images [Електронний ресурс] // CVPR 2018. – 2018. – DOI: 10.1109/CVPR.2018.00418. – Режим доступу до ресурсу: [веб-сайт].