

Національний аерокосмічний університет «Харківський авіаційний інститут»
Міністерство освіти і науки України

Національний аерокосмічний університет «Харківський авіаційний інститут»
Міністерство освіти і науки України

Кваліфікаційна наукова
праця на правах рукопису

ЦЕХМИСТРО РОСТИСЛАВ ВІКТОРОВИЧ

УДК 004.932:004.032.26

ДИСЕРТАЦІЯ

МЕТОДИ ТА ЗАСОБИ ЛОКАЛІЗАЦІЇ ТА КЛАСИФІКАЦІЇ ОБ'ЄКТІВ ЗА ДОПОМОГОЮ НЕЙРОННИХ МЕРЕЖ В УМОВАХ ПОВІТРЯНОЇ ЗЙОМКИ

122 Комп'ютерні науки
(шифр і назва спеціальності)

12 Інформаційні технології
(галузь знань)

Подається на здобуття наукового ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання чужих ідей, результатів і текстів мають посилання на відповідне джерело

Цехмистро Р. В.
(підпис, ініціали та прізвище здобувача)

Науковий керівник

Рубель Олексій Сергійович, кандидат технічних наук, доцент

Харків - 2025

АННОТАЦІЯ

Цехмистро Ростислав Вікторович. Методи та засоби локалізації та класифікації об'єктів за допомогою нейронних мереж в умовах повітряної зйомки. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація для здобуття ступеня доктора філософії з галузі знань 12 Інформаційні технології за спеціальністю 122 Комп'ютерні науки. - Національний аерокосмічний університет «Харківський авіаційний інститут», Харків, 2025.

Дисертаційна робота присвячена дослідженню та вдосконаленню методів та засобів локалізації та класифікації об'єктів на зображеннях що отримані в умовах повітряної зйомки та оцінці точності їх роботи в різних умовах, включаючи спотворення зображень шумом та несприятливі погодні умови. Об'єктом дослідження є процес локалізації та класифікації об'єктів на зображеннях, що отримані в умовах повітряної зйомки. Предметом дослідження є нейромережеві методи та засоби локалізації та класифікації об'єктів на зображеннях, що отримані в умовах повітряної зйомки.

У роботі виконано дослідження та модифікацію сучасних методів локалізації та класифікації об'єктів на зображеннях, що використовують нейронні мережі, а також проведено дослідження впливу різноманітних факторів на точність роботи цих методів. В ході досліджень проаналізовано точність роботи методів локалізації та класифікації для зображень звичайної якості, а також в умовах спотворення зображень шумом та впливу природніх факторів, таких як дощ чи туман. В процесі дослідження виявлено, що найбільший вплив може мати шум, що призводить до суттєвого зниження точності роботи досліджуваних методів. Вдосконалено методи локалізації та класифікації для підвищення точності локалізації та класифікації.

Проведено аналіз впливу розміру об'єкту (числа пікселів, що на нього приходяться) на характеристики локалізації та класифікації, визначено граничні

(мінімальні) значення площі об'єктів на зображеннях для точної їх класифікації та локалізації за допомогою нейромережових методів, що були дослідженні в даній роботі. Для підвищення точності роботи методів локалізації та класифікації з малорозмірними об'єктами використано метод обробки зображення частинами. Запропоновано метод визначення характеристик системи формування зображення на основі визначених параметрів розміру об'єкту, який може бути використаний як умова для вдосконаленого методу локалізації та класифікації.

Досліджено характеристики та інтенсивність шуму, що може бути присутній на зображеннях. Для цього було використано метод визначення кореляційних характеристик шуму, а також нейронну мережу для визначення інтенсивності шуму. Визначено, що вплив шуму на точність локалізації та класифікації є досить високим та збільшується зі збільшенням його інтенсивності. Для зменшення впливу шуму на точність локалізації та класифікації було вдосконалено метод локалізації та класифікації. Використовуючи нейронну мережу NoiseNet для визначення інтенсивності завад та фільтр на основі BM3D суттєво підвищено точність локалізації та класифікації.

Проведено дослідження впливу погодних факторів на точність локалізації та класифікації. Визначено, що дощ та туман не суттєво впливають на точність методів локалізації та класифікації. Запропоновано використання методу видалення туману, який має назву Dark Channel Prior, в методі локалізації та класифікації для підвищення точності. Досліджуючи використання методу не було отримано суттєвого покращення результатів прогнозувань.

За результатами проведеного аналізу в роботі поставлено та вирішено **актуальне науково-прикладне завдання**, а саме дослідження та вдосконалення методів та засобів локалізації та класифікації об'єктів на зображеннях, отриманих в умовах повітряної зйомки. Досліджено вплив розміру об'єктів (числа пікселів, що на них приходяться) на характеристики локалізації та класифікації, визначено граничні (мінімальні) значення площі об'єктів на зображеннях для їх класифікації та локалізації за допомогою нейромережових методів, що були дослідженні в даній роботі, із задовільними характеристиками.

Для підвищення точності роботи нейромережевих методів локалізації та класифікації з малорозмірними об'єктами досліджено метод обробки зображення частинами, а також запропоновано метод визначення характеристик системи формування зображень. Також досліджено тип та інтенсивність завад на зображеннях, що отримані в різних умовах та запропоновано вдосконалений метод локалізації та класифікації, що підвищує точність локалізації та класифікації на зображеннях, що спотворені шумом. Досліджено вплив природніх факторів на точність локалізації та класифікації.

Для вирішення наукових завдань використовувались наступні методи: математичне та чисельне моделювання, методи машинного навчання та нейронні мережі, методи математичної статистики.

Мета дисертаційної роботи полягає в дослідженні та вдосконаленні нейромережевих методів та засобів для підвищення точності локалізації та класифікації об'єктів на зображеннях, отриманих в умовах повітряної зйомки.

Відповідно до поставленої мети в дисертаційній роботі поставлено та вирішено такі основні **завдання**:

1. Провести аналіз відомих нейромережевих методів локалізації та класифікації при роботі з зображеннями, що отримані в умовах повітряної зйомки.
2. Експериментально дослідити точність роботи нейромережевих методів локалізації та класифікації при роботі з малорозмірними об'єктами та визначити мінімальні характеристики об'єкту (розмір в пікселях) для його достовірної локалізації та класифікації.
3. Вдосконалити метод локалізації та класифікації для підвищення точності локалізації та класифікації малорозмірних об'єктів. Експериментально дослідити вплив методу на точність локалізації та класифікації малорозмірних об'єктів.
4. Розробити метод визначення характеристик системи формування зображень враховуючи експериментально отримані результати про мінімально допустимі характеристики об'єкту (розмір в пікселях) для підвищення точності методу локалізації та класифікації.

5. Експериментально дослідити характеристики шуму на зображеннях, його тип та інтенсивність, визначити його вплив на точність локалізації та класифікації.

6. Вдосконалити метод локалізації та класифікації об'єктів на зображеннях, що спотворені шумом, шляхом використання методу оцінки інтенсивності шуму та методу фільтрації.

7. Експериментально дослідити вплив природніх факторів на точність локалізації та класифікації, а також дослідити можливість використання методів підвищення точності локалізації та класифікації в таких умовах.

Вперше запропоновано метод визначення характеристик системи формування зображень на основі експериментально визначених мінімально допустимих розмірів об'єктів (розміру в пікселях) на кольорових зображеннях, який відрізняється врахуванням статистичних результатів дослідження на базах великого розміру з широким спектром розмірів об'єктів, що дозволяє обґрунтувати характеристики польоту носія й встановленого на ньому сенсора для забезпечення точної роботи методів локалізації та класифікації малорозмірних об'єктів.

Отримав подальший розвиток метод та засіб оцінки статистичних та спектральних характеристик шуму на зображеннях всліпу, що дозволяє визначити тип шуму та його інтенсивність, який відрізняється використанням навченої нейромережі для оцінки інтенсивності шуму, що дозволило використовувати отриману інформацію для обґрунтованого застосування методу попередньої обробки зображення на основі BM3D-фільтра, що призводить до підвищення точності локалізації та класифікації об'єктів.

Вдосконалено метод тайлової локалізації та класифікації об'єктів з подальшою агрегацією результатів, що дозволяє підвищити точність при роботі з малорозмірними об'єктами, який на відміну від інших методів може використовувати інформацію про характеристики системи формування зображень та визначати доцільність тайлової обробки, що для певних типів

нейронних мереж дає змогу підвищити точність локалізації та класифікації без перенавчання моделі.

За матеріалами дисертації **опубліковано** 10 робіт, з яких 5 входять до переліку наукових фахових видань України, 2 з яких індексуються в Scopus (входять в Q3) та 5 доповідей на міжнародних конференціях, 4 з яких входять до Scopus.

Ключові слова: нейронні мережі, локалізація об'єктів, класифікація об'єктів, обробка зображень з БПЛА, ВМЗД, локалізація малорозмірних об'єктів, попередня обробка зображень, згорткові нейронні мережі, повітряна зйомка, видалення шуму, зображення з завадами.

Список публікацій здобувача за темою дисертації

Статті у наукових періодичних виданнях:

1. Tsekhmystro, R., Rubel, O., Lukin, V., “Study of methods for searching and localizing objects in images from aircraft using convolutional neural networks,” *Radioelectronic and Computer Systems*, vol. 2024, no. 1, P. 87–98, 2024. DOI: 10.32620/reks.2024.1.08 (Scopus).

2. Tsekhmystro, R., Rubel, O., Lukin, V., “Study of the dependence of accuracy in vehicles search on the size of the object using UAV images,” *Aerospace Technic and Technology*, no. 3, P. 89–98, 2024. DOI: 10.32620/aktt.2024.3.08 (фаховий).

3. Tsekhmystro, R., Rubel, O., Lukin, V., “Investigation of the effect of object size on accuracy of human localisation in images acquired from unmanned aerial vehicles,” *Aerospace Technic and Technology*, no. 2, P. 83–90, 2024. DOI: 10.32620/aktt.2024.2.09 (фаховий).

4. Tsekhmystro, R., Rubel, O., Prysiashniuk, O., Lukin, V., “Impact of distortions in UAV images on quality and accuracy of object localization,” *Radioelectronic and Computer Systems*, vol. 2024, no. 4, P. 59–67, 2024. DOI: 10.32620/reks.2024.4.05 (Scopus).

5. Цехмистро, Р. В., Абрамова, В. В., Рубель, А. С., Усс, М. Л., Проскура, Г. А., Рубель, О. С., “Оцінка характеристик шуму на реальних зображеннях з використанням згорткової нейронної мережі на мобільній платформі,” *Radioelectronic And Computer Systems*, no. 2, P. 60–70, Jun. 2019. DOI: 10.32620/reks.2019.2.05 (фаховий).

Наукові праці, які засвідчують апробацію матеріалів дисертації:

6. Rubel, O., Tsekhmystro, R., Lukin, V., Egiazarian, K., “Benchmark of Similar Blocks Search under Noisy Conditions,” *Electronic Imaging*, vol. 33, no. 10, P. 238-1–238, 2021. DOI: 10.2352/issn.2470-1173.2021.10.ipas-238. (Scopus).

7. Tsekhmystro, R., Rubel, O., Abramova, V., Zriakhov, M., Uss, M., "Mobile Deployment of NoiseNet: Noise Characteristics Assessment in Real-World Images," *2019 IEEE 2nd Ukraine Conference on Electrical and Computer Engineering (UKRCON)*, Lviv, Ukraine, 2019, P. 1112-1117. DOI: 10.1109/UKRCON.2019.8879797. (Scopus) .

8. Tsekhmystro, R., Oliinyk, V., Proskura, G., Rubel, O., "Web Assembled Benchmark for Image Visual Quality Assessment, Prediction and Improvement," *2020 IEEE 15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET)*, Lviv-Slavske, Ukraine, 2020, P. 791-795. DOI: 10.1109/TCSET49122.2020.235543. (Scopus).

9. Rubel, A., Rubel, O., Tsekhmystro, R., Rebrov, V., Lukin, V., “Automatic Decision Undertaking on Expedience of Image Denoising Based on Filter Efficiency Prediction,” *Springer Proceedings in Physics, Springer Nature Singapore*, P. 504–524, 2023. DOI: 10.1007/978-981-99-4098-1_44. (Scopus).

10. Tsekhmystro, R., Lukin, V., “Object localization and recognition in UAV-based noisy images,” *Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління*, vol. 3, P. 18-19, 2025.

ABSTRACT

Tsekhmystro Rostyslav. Methods and means of localization and classification of objects using neural networks in aerial surveys - Qualification scientific work, the manuscript.

Dissertation for the degree of Doctor of Philosophy in the field of knowledge 12 Information Technology, specialty 122 "Computer Science". - National Aerospace University "Kharkiv Aviation Institute", Kharkiv, 2025.

The dissertation is devoted to the study of methods and means for localizing objects in images obtained in aerial surveys and evaluating their performance in various conditions, including image distortion by noise and adverse weather conditions. The object of research is the process of localization and classification of objects in images obtained in aerial surveys. The subject of the study is neural network methods and tools for localization and classification of objects in images obtained in aerial survey conditions.

The research work includes the study and modification of modern methods of localization and classification of objects in images using neural networks, as well as the study of the influence of various factors on the accuracy of these methods. The research analyzed the accuracy of localization and classification methods for images of normal quality, as well as in conditions of image distortion by noise and the influence of natural factors such as rain or fog. The study revealed that noise can have the greatest impact, which leads to a significant decrease in the accuracy of the studied methods. The localization and classification methods were improved to increase the accuracy of localization and classification.

The influence of the object size (the number of pixels per object) on the localization and classification characteristics is analyzed, and the boundary (minimum) values of the area of objects in the images of their classification are determined with satisfactory characteristics using the neural networks studied in this paper. To improve the accuracy of the localization and classification methods with small-sized objects, the method of image processing by parts is used. A method for determining the characteristics of an

image capture system based on certain object size parameters is proposed, which can be used as a condition for an improved localization and classification method.

The characteristics and intensity of noise that may be present in images are investigated. For this purpose, a method for determining the correlation characteristics of noise was used, as well as a neural network for determining the noise intensity. It was determined that the impact of noise on the accuracy of localization and classification is quite high and increases with increasing noise intensity. To reduce the impact of noise on localization and classification accuracy, the localization and classification method was improved. By using the NoiseNet neural network to determine the intensity of the noise and a BM3D-based filter, the localization and classification accuracy was significantly improved.

The influence of weather factors on the accuracy of localization and classification was studied. It is determined that rain and fog do not significantly affect the accuracy of localization and classification methods. It is proposed to use a fog removal method called Dark Channel Prior in the localization and classification method to improve accuracy. The use of this method did not significantly improve the prediction results.

Based on the results of the analysis, the research sets and **solves an urgent scientific and applied task**, namely, the study and improvement of methods and means of localization and classification of objects in images obtained in aerial survey conditions. The influence of the size of objects (the number of pixels per object) on the characteristics of localization and classification was investigated, and the boundary (minimum) values of the area of objects in images for their classification and localization were determined using the neural network methods investigated in this work with satisfactory characteristics. To improve the accuracy of neural network methods for localization and classification with small-sized objects, a method of image processing in parts is investigated, and a method for determining the characteristics of an image capture system is proposed. Also, the type and intensity of noise in images obtained under different conditions are investigated, and an improved localization and classification method is proposed that improves the accuracy of localization and

classification in images distorted by noise. The influence of natural factors on the localization and classification accuracy is investigated.

The following methods were used to solve the scientific problems: mathematical and numerical modeling, machine learning methods and neural networks, methods of mathematical statistics.

The purpose of this thesis is to study and improve neural network methods and tools for improving the accuracy of object localization and classification in images obtained in aerial surveys.

In accordance with this goal, the following main tasks were set and solved in the dissertation:

1. To analyze the known neural network methods of localization and classification when working with images obtained in aerial surveys.
2. To experimentally investigate the accuracy of neural network methods of localization and classification when working with small-sized objects and determine the minimum characteristics of the object (size in pixels) for its reliable localization and classification.
3. Improve the localization and classification method to increase the accuracy of localization and classification of small objects. Experimentally investigate the effect of the method on the accuracy of localization and classification of small objects.
4. To develop a method for determining the characteristics of an image capture system, taking into account the experimentally obtained results on the minimum allowable characteristics of an object (size in pixels) to improve the accuracy of the localization and classification method.
5. Experimentally investigate the characteristics of noise in images, its type and intensity, and determine its impact on the accuracy of localization and classification.
6. Improve the method of localization and classification of objects in noise-distorted images by using the method of noise intensity estimation and filtering.
7. To experimentally investigate the influence of natural factors on the accuracy of localization and classification, as well as to investigate the possibility of using methods to improve the accuracy of localization and classification in such conditions.

For the first time, a method for determining the characteristics of an imaging system based on experimentally determined minimum allowable object sizes (size in pixels) in color images is proposed, which is distinguished by taking into account the statistical results of the study on large-size databases with a wide range of object sizes, which allows us to substantiate the flight characteristics of the carrier and the sensor installed on it to ensure accurate operation of methods for localizing and classifying small-sized objects.

The method and means for estimating statistical and spectral characteristics of noise in blind images **were further developed**, which allows determining the type of noise and its intensity, and is distinguished by the use of a trained neural network to estimate the noise intensity, which allowed using the information obtained for the justified application of the image preprocessing method based on a BM3D filter, which leads to an increase in the accuracy of object localization and classification.

The method of tile localization and classification of objects with subsequent aggregation of results **was improved**, which allows to increase the accuracy when working with small-sized objects, which, unlike other methods, can use information about the characteristics of the imaging system and determine the feasibility of tile processing, which for certain types of neural networks allows to increase the accuracy of localization and classification without retraining the model.

Based on the materials of the dissertation, 10 papers **were published**, 5 of which are included in the list of scientific professional publications of Ukraine, 2 of which are indexed in Scopus (included in Q3) and 5 reports at international conferences, 4 of which are included in Scopus.

Keywords: neural networks, object localization, object classification, UAV image processing, BM3D, localization of small objects, image preprocessing, convolutional neural networks, aerial photography.

ЗМІСТ

АННОТАЦІЯ.....	2
ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	15
ВСТУП.....	16
РОЗДІЛ 1. ЛОКАЛІЗАЦІЯ ТА КЛАСИФІКАЦІЯ ОБ’ЄКТІВ НА ЗОБРАЖЕННЯХ ПОВІТРЯНОЇ ЗЙОМКИ	24
1.1 Структура системи обробки зображення отриманих в умовах повітряної зйомки	24
1.2 Нейронні мережі для локалізації та класифікації об’єктів	27
1.2.1 Region-based Convolution Neural Network (R-CNN)	28
1.2.2 Fast R-CNN	30
1.2.3 Faster R-CNN	31
1.2.4 You Only Look Once (YOLO)	33
1.2.5 Single Shot Detector (SSD)	39
1.2.6 RetinaNet	40
1.2.7 Порівняння нейронних мереж за метриками	41
1.3 Процес навчання нейронних мереж.....	44
1.3.1 Функції втрат.....	45
1.3.2 Метрики якості прогнозування	48
1.4 Зображення, отримані за допомогою повітряної зйомки	52
1.5 Обробка зображень за допомогою нейронних мереж	56
1.6 Висновки і формулювання задач дослідження.....	58
РОЗДІЛ 2. ДОСЛІДЖЕННЯ РОБОТИ НЕЙРОННИХ МЕРЕЖ ДЛЯ ЛОКАЛІЗАЦІЇ ТА КЛАСИФІКАЦІЇ НА ЗОБРАЖЕННЯХ ПОВІТРЯНОЇ ЗЙОМКИ	61
2.1 Підготовка до навчання.....	61

	13
2.1.1 Вибір набору даних	63
2.1.2 Вибір фреймворку для навчання	69
2.2 Навчання нейронних мереж	73
2.2.1 Навчання Faster R-CNN.....	73
2.2.2 Навчання SSD.....	77
2.2.3 Навчання RetinaNet.....	80
2.2.4 Навчання YOLO.....	83
2.3 Порівняння навчених нейронних мереж	88
2.3.1 Оцінка точності прогнозувань.....	89
2.3.2 Оцінка розрахункової складності та кількості параметрів	96
2.4 Висновки до розділу.....	98
РОЗДІЛ 3. ОЦІНКА РОБОТИ НЕЙРОННИХ МЕРЕЖ З ОБ'ЄКТАМИ МАЛОГО РОЗМІРУ	101
3.1 Огляд набору даних для досліджень.....	101
3.2 Дослідження точності роботи з малорозмірними об'єктам	105
3.3 Метод підвищення точності локалізації та класифікації.....	117
3.4 Метод визначення характеристик об'єкту	128
3.5 Висновки до розділу.....	130
РОЗДІЛ 4. ДОСЛІДЖЕННЯ РОБОТИ НЕЙРОННИХ МЕРЕЖ В УМОВАХ СПОТВОРЕНЬ	132
4.1 Дослідження роботи нейронних мереж в умовах впливу шуму	133
4.1.1. Оцінка природи та параметрів шуму на зображеннях	134
4.1.2. Спотворення зображень шумом	139
4.1.3. Дослідження точності роботи нейронних мереж	141
4.1.4 Метод підвищення точності локалізації та класифікації в умовах шуму	144
4.2 Дослідження роботи нейронних мереж в поганих погодних умовах	150

4.2.1 Моделювання природніх погодних умов для зображень	150
4.2.2 Оцінка точності локалізації та класифікації за впливу природніх умов	153
4.2.3 Дослідження ефективності методу видалення туману	158
4.3 Висновки до розділу	163
ВИСНОВКИ	165
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	167
ДОДАТОК А СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА	184
ДОДАТОК Б АКТИ ВПРОВАДЖЕННЯ	186
ДОДАТОК В ПРОГРАМНИЙ КОД МЕТОДУ ПОТАЙЛОВОЇ ОБРОБКИ.....	187
ДОДАТОК Г ПРОГРАМНИЙ КОД МЕТОДУ З ФІЛЬТРАЦІЄЮ ШУМУ	191
ДОДАТОК Д ПРОГРАМНИЙ КОД МЕТОДУ DARK CHANNEL PRIOR.....	193

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

АБГШ – адитивний білий гаусів шум;

ЛА – літальний апарат;

БПЛА – безпілотний літальний апарат;

ДЗЗ – дистанційне зондування Землі;

ДЗ – дистанційне зондування;

СФЗ – система формування зображень;

ЗНМ – згортова нейронна мережа;

КМОН - комплементарна структура метал-оксид-напівпровідник;

НМ – нейронна мережа;

СКВ – середньоквадратичне відхилення;

BM3D – Block Matching and 3D filtering;

MSE – Mean Squared Error;

GPS - Global Positioning System;

JPEG - Joint Photographic Experts Group;

NMS - Non Max Suppression;

FLOPS – FLoating-Point Operation Per Second.

ВСТУП

Обґрунтування вибору теми дослідження. З розвитком технологій, які дозволяють отримувати більше інформації з зображень [1] зростають також і потреби в обробці цієї інформації та підвищенні ефективності такої обробки [2]. Локалізація та класифікація об'єктів [3-5] є, мабуть, одними з найпоширеніших задач отримання інформації з зображень. З розвитком нейронних мереж [6], що дозволяють досить надійно і швидко локалізувати та класифікувати об'єкти задачі набули ширшого використання та популярності. Отримана інформація про об'єкт на зображенні може бути використана в різноманітних сферах [7-10]. Це може бути екологічний моніторинг земної поверхні [11], забезпечення правопорядку [12], пошук вибухонебезпечних предметів [13] та інші. Така широта використання відображає потребу в точних та швидких методах локалізації та класифікації [14], які б дозволяли надійно виконувати цю задачу. Часто такі задачі вирішуються в цільових програмах досліджень які фінансуються як Україною так і європейськими державами. До прикладу можна привести програми, що були фінансовані державним бюджетом України та виконувались кафедрою інформаційно-комунікаційних технологій ім. О.О. Зеленського та кафедрою комп'ютерних систем, мереж і кібербезпеки Національного аерокосмічного університету «Харківський авіаційний інститут» протягом останніх років, це - «Методи та засоби виявлення вибухонебезпечних предметів з використанням багатофункційних інтелектуальних систем БПЛА» (ДР № 0123U101992) та «Інтелектуальні методи обробки багатоканальних зображень на основі карт візуальної уваги та баз зображень із спотвореннями» (ДР № 0124U001094). Обидві теми стосуються підвищення точності локалізації та класифікації об'єктів та розширення сфери застосувань таких методів.

На сьогоднішній день розв'язання задачі локалізації та класифікації об'єктів головним чином використовують нейронні мережі [3-5]. Існуючі методи дозволяють проводити локалізацію та класифікацію багатьох типів об'єктів різного розміру [3-5], але дослідження які проводяться стосуються загалом

точності нейронних мереж на широкому діапазоні розмірів [7-10]. Дослідники також пропонують методи підвищення точності локалізації малорозмірних об'єктів [14], але здебільшого вони стосуються саме процесу локалізації без визначення граничних значень розміру об'єкту. Тому є потреба в дослідженні методів локалізації та класифікації об'єктів на даних, що мають широкий діапазон розмірів об'єктів та визначення крайових значень розмірів об'єктів в пікселях, що дозволить використовувати метод обробки таких зображень з більшою точністю залежно від умов зйомки.

Також доведено, що шум може знижувати точність локалізації та класифікації [15], але не представлено наскільки такий вплив вагомий. Досить багато досліджень присвячено темі визначення необхідності застосування методів фільтрації при роботі з системами локалізації та класифікації, але досить мало інформації про визначення впливу таких методів в різних умовах. Тому важливим дослідженням є оцінка типу та інтенсивності завад, а також їх впливу на точність роботи методу локалізації та класифікації на основі нейронної мережі. Актуальним є також вдосконалення методів локалізації та класифікації шляхом застосування фільтрів та відповідно визначення впливу цих методів на точність локалізації та класифікації.

Ще одним фактором що також може вплинути на якість локалізації та класифікації, особливо в умовах повітряної зйомки, є погода, зокрема туман [16] та дощ [17]. Дослідження, які стосуються цього впливу не мають широкого поширення, а тому є важливою та актуальною задачею визначення на скільки сильним може бути вплив природніх факторів на якість локалізації та класифікації. Важливою темою також є дослідження методів видалення впливу таких факторів та вплив застосування таких методів на точність локалізації та класифікації.

Важливим параметром, особливо в умовах обмежених обчислювальних ресурсів, є обчислювальне навантаження [18] та кількість параметрів нейронної мережі [19], що використовується в методі локалізації та класифікації. В випадку з літальними апаратами, де важлива кожна характеристика при проектуванні його

літальних властивостей, важко підібрати потужні пристрої для виконання обчислень [20]. Використання ж важких нейронних мереж на потужних пристроях [21], що розміщені стаціонарно, є досить поширеною задачею, але не у всіх випадках може бути використаний такий підхід. Також можливим є використання підходу з розрахунком на пристрої управління літальним апаратом [22], часто цю роль виконує смартфон. Але такі пристрої також мають обмежену обчислювальну здатність.

Враховуючи цю інформацію, розгляд методів локалізації та класифікації об'єктів за допомогою нейронних мереж, а також методів для підвищення точності локалізації та класифікації, на зображеннях, отриманих в умовах повітряної зйомки є **актуальною науково-прикладною задачею**. Вирішення її в даній науковій роботі дозволить підвищити точність роботи методів локалізації та класифікації, що використовують нейронні мережі, та дослідити запропоновані методи покращення точності на практиці, а також використати отримані в процесі досліджень дані для проектування систем.

Об'єкт дослідження – процес локалізації та класифікації об'єктів на зображеннях, отриманих в умовах повітряної зйомки.

Предмет досліджень – нейромережеві методи та засоби локалізації та класифікації об'єктів на зображеннях, що отримані в умовах повітряної зйомки.

Мета і завдання дослідження. Мета дисертаційної роботи полягає в дослідженні та вдосконаленні нейромережевих методів та засобів для підвищення точності локалізації та класифікації об'єктів на зображеннях, отриманих в умовах повітряної зйомки.

Відповідно до поставленої мети в дисертаційній роботі сформульовано та вирішено такі основні **завдання**:

1. Провести аналіз відомих нейромережевих методів локалізації та класифікації при роботі з зображеннями, що отримані в умовах повітряної зйомки.

2. Експериментально дослідити точність роботи нейромережевих методів локалізації та класифікації при роботі з малорозмірними об'єктами та визначити

мінімальні характеристики об'єкту (розмір в пікселях) для його достовірної локалізації та класифікації.

3. Вдосконалити метод локалізації та класифікації для підвищення точності локалізації та класифікації малорозмірних об'єктів. Експериментально дослідити вплив методу на точність локалізації та класифікації малорозмірних об'єктів.

4. Розробити метод визначення характеристик системи формування зображень враховуючи експериментально отримані результати про мінімально допустимі характеристики об'єкту (розмір в пікселях) для підвищення точності методу локалізації та класифікації.

5. Експериментально дослідити характеристики шуму на зображеннях, його тип та інтенсивність, визначити його вплив на точність локалізації та класифікації.

6. Вдосконалити метод локалізації та класифікації об'єктів на зображеннях, що спотворені шумом, шляхом використання методу оцінки інтенсивності шуму та методу фільтрації.

7. Експериментально дослідити вплив природніх факторів на точність локалізації та класифікації, а також дослідити можливість використання методів підвищення точності локалізації та класифікації в таких умовах.

Методи досліджень. Для вирішення поставлених задач та проведення необхідних досліджень в роботі було використано методи машинного навчання. Також використано методи статистики для побудови графіків залежностей відповідно до досліджуваних сфер. Для дослідження роботи в умовах завад було використано методи математичного моделювання для моделювання шуму та створення тестових зображень. Також було використано методи фільтрації для покращення візуальної якості зображень та проведення досліджень.

Наукова новизна отриманих результатів:

– **Вперше запропоновано** метод визначення характеристик системи формування зображень на основі експериментально визначених мінімально допустимих розмірів об'єктів (розміру в пікселях) на кольорових зображеннях, який відрізняється врахуванням статистичних результатів дослідження на базах

великого розміру з широким спектром розмірів об'єктів, що дозволяє обґрунтувати характеристики польоту носія й встановленого на ньому сенсора для забезпечення точної роботи методів локалізації та класифікації малорозмірних об'єктів.

– **Отримав подальший розвиток** метод та засіб оцінки статистичних та спектральних характеристик шуму на зображеннях всліпу, що дозволяє визначити тип шуму та його інтенсивність, який відрізняється використанням навченої нейромережі для оцінки інтенсивності шуму, що дозволило використовувати отриману інформацію для обґрунтованого застосування методу попередньої обробки зображення на основі BM3D-фільтра, що призводить до підвищення точності локалізації та класифікації об'єктів.

– **Вдосконалено** метод тайлової локалізації та класифікації об'єктів з подальшою агрегацією результатів, що дозволяє підвищити точність при роботі з малорозмірними об'єктами, який на відміну від інших методів може використовувати інформацію про характеристики системи формування зображень та визначати доцільність тайлової обробки, що для певних типів нейронних мереж дає змогу підвищити точність локалізації та класифікації без перенавчання моделі.

Особистий внесок здобувача. Усі основні результати дисертаційної роботи було отримано автором самостійно. В публікації [1] було проведено дослідження нейронних мереж на предмет точності роботи в задачах локалізації та класифікації об'єктів на зображеннях, що отримані з БПЛА. У роботах [2, 3] було досліджено вплив розміру об'єкту на зображенні на якість його локалізації та класифікації, а також визначено мінімальні значення площі об'єкту для його локалізації та класифікації з достатньою та високою точністю. Робота [4] присвячена впливу шуму на зображення та на якість локалізації та класифікації об'єктів на таких зображеннях. Методи та засоби визначення типу та параметрів завад на реальних зображеннях представлено у роботах [5, 7-8], для цього було використано як класичні методи так і методи, що базуються на нейронних мережах. У роботі [6] представлено алгоритм придушення шуму на зображенні,

що базується на методі BM3D та використовує потужності відеокарти, а саме cuDNN для прискорення алгоритму пошуку схожих регіонів. Робота [9] присвячена оцінці покращення зображення після фільтрації без фактичного її проведення, що дозволяє оцінити доцільність фільтрації до її проведення. В роботі [10] розглянуто доцільність використання алгоритму фільтрації для підвищення якості локалізації та класифікації об'єктів на зображеннях, що спотворені шумом.

Список публікацій здобувача за темою дисертації представлено в Додатку А.

Апробація матеріалів дисертації.

Методи оцінки завад на зображеннях були представлені на конференції «Ukraine Conference on Electrical and Computer Engineering (UKRCON)» (2019, Львів, Україна), «*Electronic Imaging*» (2021, NYC, USA) «15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET)» (2020, Львів-Славське, Україна), «International Symposium of Space Optical Instruments and Applications» (2022, Beijing, China) та «Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління» (2025, Баку – Харків – Жиліна).

Зв'язок роботи з науковими програмами, планами, темами, грантами.

Дослідження, що представлені в дисертаційній роботі проводились на кафедрі інформаційно-комунікаційних технологій ім. О. О. Зеленського Національного аерокосмічного університету «ХАІ» і були відображені в наступних звітах про НДР: «Інтелектуальна система комплексної автоматичної обробки візуальних мультимедійних даних» (2020-2021, № держреєстрації 0119U100940), «Методи та засоби виявлення вибухонебезпечних предметів з використанням багатофункційних інтелектуальних систем БПЛА» (2024, № держреєстрації 0123U101992), «Інтелектуальні методи обробки багатоканальних зображень на основі карт візуальної уваги та баз зображень із спотвореннями» (2024-2026, № держреєстрації 0124U001094).

Практичне значення отриманих результатів:

– В результаті **проведеного порівняльного аналізу** основних характеристик чотирьох популярних типів різних архітектур ЗНМ з різними модифікаціями кожної з них у застосуванні до задач локалізації та класифікації характерних типів об'єктів, які можуть бути присутніми на зображеннях, отриманих в умовах повітряної зйомки було надано рекомендації по вибору ЗНМ в залежності від пріоритету вимог.

– В результаті **проведеного дослідження** точності локалізації та класифікації малорозмірних об'єктів на зображеннях з БПЛА, визначено мінімальний розмір об'єкту в пікселях, який дозволяє точно локалізувати та класифікувати об'єкт; а також наведено рекомендації щодо умов зйомки.

– В результаті **проведеного дослідження** характеристик локалізації та класифікації об'єктів в умовах впливу завад, визначено рівень завад, для яких починає спостерігатися різке погіршення точності локалізації та класифікації, що дозволяє виділити методи, які характеризуються більшою стійкістю до завад та спотворень, та рекомендувати їх до практичного застосування.

– В результаті **дослідження** впливу погодних факторів (дощу та туману) на характеристики локалізації та класифікації доведено, що негативний вплив зазвичай є невеликим, а відомі методи попередньої обробки для усунення негативного впливу не є ефективними.

Публікації. Основні результати дисертації опубліковані в 10 роботах, серед яких п'ять в наукових спеціалізованих виданнях України, що входять в наукометричні бази (Index Copernicus, Google Scholar), серед яких дві публікації в науковому спеціалізованому виданні України категорії «А» та індексуються в Scopus (квартиль Q3), та 5 доповідей, які надруковані в матеріалах міжнародних конференцій, 4 з них входять до Scopus.

Структура і обсяг дисертації. Дисертація складається з анотації, змісту, списку скорочень, вступу, чотирьох розділів, висновків, списку використаних джерел і додатків. Повний обсяг роботи становить 194 сторінки друкованого тексту, з яких анотація двома мовами — на 9 сторінок, зміст — на 3 сторінки, список скорочень — на 1 сторінку, основний текст — на 151 сторінку, список із

128 використаних джерел — на 17 сторінок , додатки — 11 сторінок. Дисертація містить 92 рисунки (5 з них на 5 окремих сторінках), 17 таблиць.

РОЗДІЛ 1.

ЛОКАЛІЗАЦІЯ ТА КЛАСИФІКАЦІЯ ОБ’ЄКТІВ НА ЗОБРАЖЕННЯХ ПОВІТРЯНОЇ ЗЙОМКИ

В розділі розглянуто структуру системи, які використовують літальні апарати для отримання зображень, а також їх обробки. Оглянуто основні характеристики нейронних мереж які можуть бути використані в методах локалізації та класифікації. Оцінено характеристики обраних нейронних мереж та доцільність їх використання в досліджуваній задачі. Також в розділі розглядається структура та характеристики зображень, що створені за допомогою дистанційних технологій, зокрема за допомогою літальних апаратів (ЛА). Приведена інформація про методи й засоби обробки таких зображень, а також про складнощі, що виникають в процесі обробки.

1.1 Структура системи обробки зображення отриманих в умовах повітряної зйомки

Для дослідження методів локалізації та класифікації на зображеннях, що отримані за допомогою безпілотних апаратів варто визначити будову таких систем та їх застосування [3]. Також необхідним є визначення основних факторів, які мають вплив на такі системи.

Можемо виділити 2 варіанти будови системи обробки зображень, які отримані за допомогою безпілотного чи пілотованого літального апарату (ЛА). Перший варіант досить поширений в задачах, де необхідна швидкість реакції самого пристрою (ЛА) та інформації про об’єкти на зображенні. В такому випадку намагаються використовувати методи обробки зображень, зокрема локалізації та класифікації, безпосередньо на самому пристрої [4, 5]. Візуальна інтерпретація такої системи показана на рисунку 1.1. Зокрема такі варіації системи можуть бути використані в задачах донаведення [7, 8] та слідкуванням

[9, 10] за об'єктом. В такій системі досить багато обмежень по використанню нейронних мереж, зокрема основне обмеження стосується кількості використовуваних ресурсів [18] та обчислювального навантаження [19]. Але для реалізації навантажених алгоритмів та нейронних мереж існують модулі розширення [20] які дозволяють запускати більш вимогливі до ресурсів методи [21]. Враховуючи навіть такий підхід дослідники намагаються зменшити навантаження та оптимізувати використовувані методи [23]. Тому при дослідженні різних методів варто враховувати також ті, які можуть бути використані безпосередньо на ЛА.

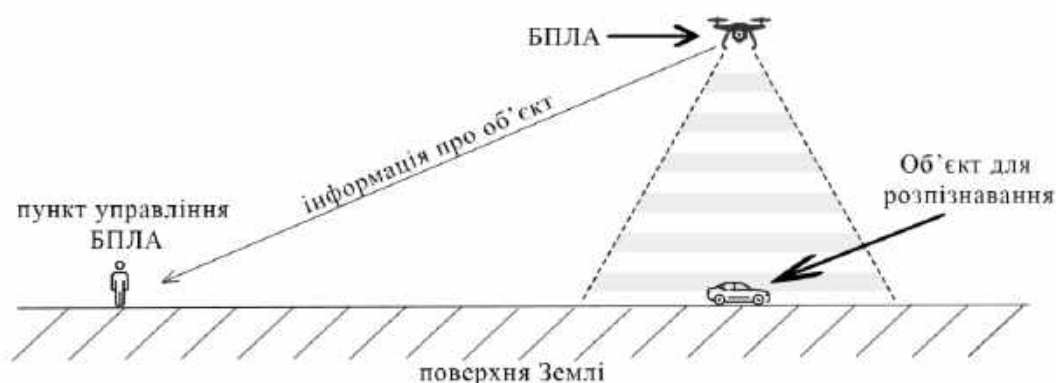


Рисунок 1.1 – Візуальна інтерпретація системи отримання інформації про об'єкти за допомогою ЛА

Для зменшення навантаження на сам пристрій (ЛА) використовують також другий варіант системи обробки зображень. В цього випадку для запуску методів локалізації та класифікації використовується зовнішній пристрій [22]. Він може бути різного типу, наприклад хмарний сервіс з відеокартою який має доступ через дистанційний зв'язок [24]. Варіант такої системи показаний на рисунку 1.2. Така варіація системи також має як плюси так і недоліки. До недоліків можна віднести затримку в каналах передачі даних, а тому рішення про переміщення такого пристрою приймаються з затримкою [25]. Але в сферах, де прийняття рішення не є важливим фактором, наприклад в сфері сільського господарства,

можливо використовувати такі системи. Їх використання дозволяє зменшити витрати на сам пристрій та зробити його дешевшим, адже зникає необхідність встановлення модулів для обчислення. Разом з тим такі системи можна зробити більш потужними з точки зору обчислювального навантаження та використати методи локалізації та класифікації які потребують більше ресурсів.

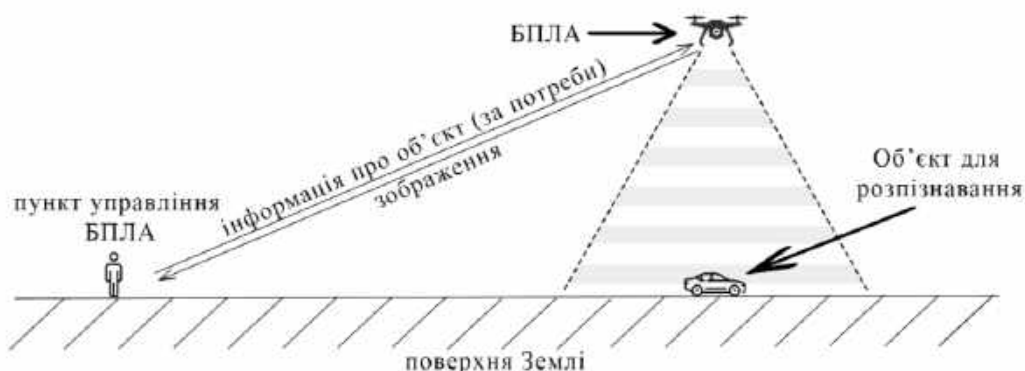


Рисунок 1.2 – Візуальна інтерпретація системи отримання інформації про об'єкти за допомогою БПЛА та відокремленого пристрою обробки

Враховуючи алгоритмічну схожість методів локалізації та класифікації на зображеннях з ЛА представимо послідовність дій такого методу (рисунок 1.3). Загалом він зводиться до використання нейронної мережі (НМ) для проведення локалізації та класифікації [26, 27]. Але враховуючи різноманітні фактори, такі як погана якість зображення чи робота в складних умовах, він може бути розширеним. Першим етапом алгоритму є отримання зображення за допомогою пристрою формування зображення безпосередньо на ЛА. В випадку з системами де обробка проводиться відразу на ЛА, зображення передається в метод локалізації та класифікації, якщо ж використовується віддалений пристрій то зображення передається за допомогою системи зв'язку. Алгоритм також перевіряє чи було отримано зображення та його валідність, щоб запобігти помилок в процесі виконання. Наступним кроком є сам процес локалізації та класифікації, який використовує нейронну мережу. Отримані дані про об'єкти опрацьовуються відповідно до необхідної задачі. Якщо має місце використання

методу, який працює на ЛА то внутрішні алгоритми аналізують подальшу його поведінку, а також інформація про об'єкти за потреби передається на пристрій керування. Якщо ж використовуються віддалені пристрої для обчислень то за потреби дані про об'єкти можуть бути передані на ЛА.

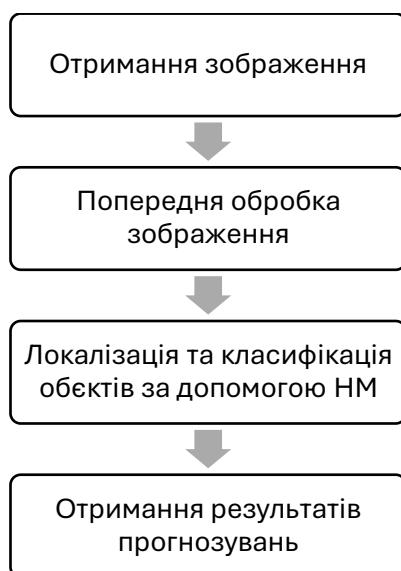


Рисунок 1.3 – Алгоритм локалізації та класифікації зображень з ЛА

Враховуючи інформацію про методи локалізації та класифікації, а також системи обробки зображень, в які вони входять можна зазначити що одним з головних елементів є нейронна мережа, яка використовується для локалізації та класифікації. Тому важливо дослідити такі методи та визначити доцільність їх використання в сфері локалізації та класифікації зображень з ЛА. Також важливо визначити інтенсивність впливу різних факторів на такі системи та розробити методи, що дозволять підвищити точність локалізації та класифікації.

1.2 Нейронні мережі для локалізації та класифікації об'єктів

Загалом архітектурно нейронні мережі можуть бути різного типу. Кількість різних нейронних мереж, що використовуються в області локалізації та

класифікації об'єктів [28], важко навіть перерахувати, але можна визначити декілька основних архітектур, які можуть бути використані в дослідженнях, це:

- Region-based Convolution Neural Network (R-CNN) [29] – нейронна мережа що базується на генеруванні регіон-кандидатів за допомогою алгоритму селективного пошуку (Selective Search);
- Fast R-CNN [30] – модифікація R-CNN, яка використовує нейронну мережу для екстракції ознак з всього зображення;
- Faster R-CNN [31] – наступна з модифікацій R-CNN мереж, в ній використовується нейронна мережа для генерування регіон-кандидатів;
- You Only Look Once (YOLO) [32] – нейронна мережа, яка вирішує задачу детектування як задачу регресії, що значно пришвидшує роботу мережі;
- Single Shot Detector (SSD) [33] – ця нейронна мережа може одночасно працювати на різних масштабах, що дозволяє краще працювати з об'єктами різного розміру;
- RetinaNet [34] – детектор, в якому запропоновано використовувати функцію втрат FocalLoss для компенсації дисбалансу класів;

Загалом кожна з цих архітектур принесла різні підходи в алгоритми локалізації та класифікації, тому буде вірним розібрати кожен з них детальніше та порівняти їх між собою за метриками якості локалізації та класифікації. Також варто визначити обчислювальне навантаження нейронних мереж які будуть використані для досліджень та порівняти їх між собою за цим параметром.

1.2.1 Region-based Convolution Neural Network (R-CNN)

Основою нейронної мережі з структурою R-CNN завжди є Region Proposal (RP) [29], тобто алгоритм для генерування пропозицій регіонів, які потрібно розглянути для класифікації. Список таких пропозицій складається за допомогою алгоритму «Вибіркового Пошуку» (Selective Search) [29]. Цей алгоритм приймає

на вхід зображення та на виході дає масив регіонів, в яких ймовірно знаходиться об'єкт.

Перший етап алгоритму – це генерування підсегментації вхідного зображення за допомогою алгоритму, який представлено в [35]. Наступний крок це об'єднання отриманих сегментаційних регіонів в більш глобальні використовуючи «жадібний» алгоритм [35]. Його логіка полягає в виборі двох найбільш схожих регіонів, комбінування їх в один, більший регіон. Алгоритм повторюється ітеративно для потрібної кількості ітерацій, кількості регіонів чи інших умов. На основі згенерованих регіонів генеруються результуючі кандидати, тобто масив регіонів, в яких, можливо, є об'єкт [36]. Приклад роботи такого алгоритму представлено на рисунку 1.4 для декількох ітерацій «жадібного» алгоритму. Загалом для класичної мережі R-CNN використовуються близько 2000 запропонованих регіонів.

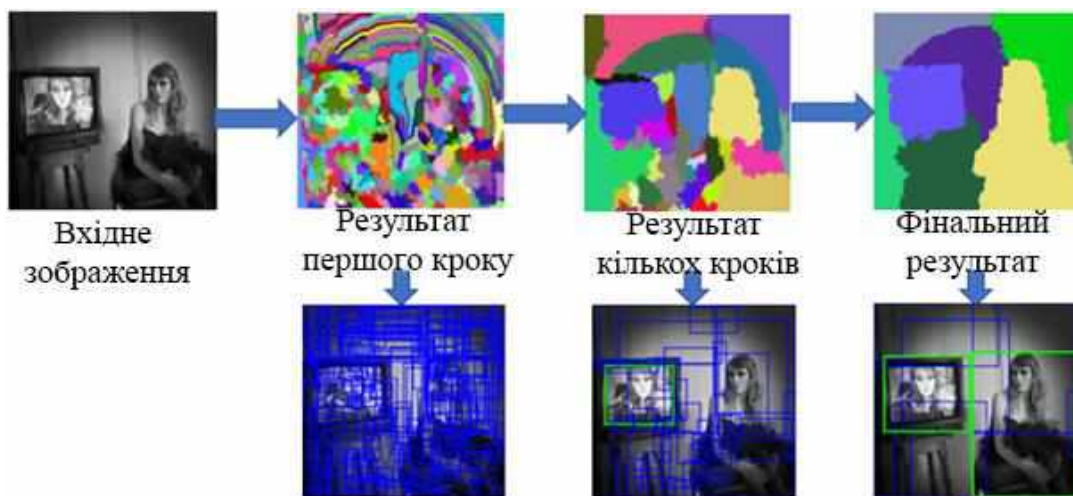


Рисунок 1.4 – Сегментація зображення та пропозиції регіонів за допомогою «жадібного» алгоритму

Наступний блок R-CNN це витягування карти ознак (feature map) [37] з кожного блоку. В публікації яка присвячена алгоритму роботи цієї мережі вказано використання глибокої згорткової нейронної мережі, що навчена на базі ImageNet [38, 39]. На практиці ж зустрічаються різні варіації екстрактора карти ознак, це

можу бути ResNet [40], MobileNet [41], VGG [42] та інші [43] в залежності від потрібної точності роботи, швидкості та інших параметрів.

Останній блок мережі – це класифікатор запропонованих регіонів на основі згенерованої карти ознак [29]. Цей блок відповідає також за фільтрацію результатів, тобто відкидає регіони в яких немає об'єкту чи ймовірність точної відповіді мала. Також іноді в цей блок включають алгоритм «Немаксимального придушення» (Non-Maximum Suppression) [44] для фільтрації регіонів, які накладаються. Типова архітектура нейронної мережі на базі регіон-кандидатів представлена на рисунку 1.5.

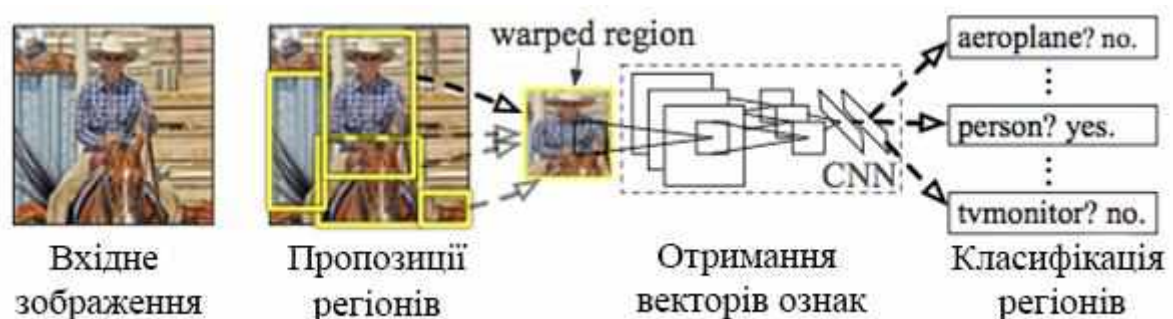


Рисунок 1.5 – Архітектура R-CNN

1.2.2 Fast R-CNN

Основним недоліком R-CNN є запуск згорткової нейронної мережі для екстракції карти ознак для кожного регіону. Звісно, це дається взнаки в швидкості роботи нейронної мережі. Саме цю проблему було вирішено в структурі Fast R-CNN [30], що є модифікацією R-CNN [29].

Модифікація пропонує запуск нейронної мережі для екстракції ознак, використовуючи згорткову нейронну мережу, для всього зображення [30]. Потім отримані запропоновані регіони накладаються на карту ознак. Тобто ознаки для регіону витягуються з отриманої заздалегідь карти ознак в потрібній точці.

Подальший алгоритм роботи нейронної мережі не модифікується, за виключенням додавання регресійної частини для поправок до отриманих регіонів. Архітектура нейронної мережі Fast R-CNN представлена на рисунку 1.6.

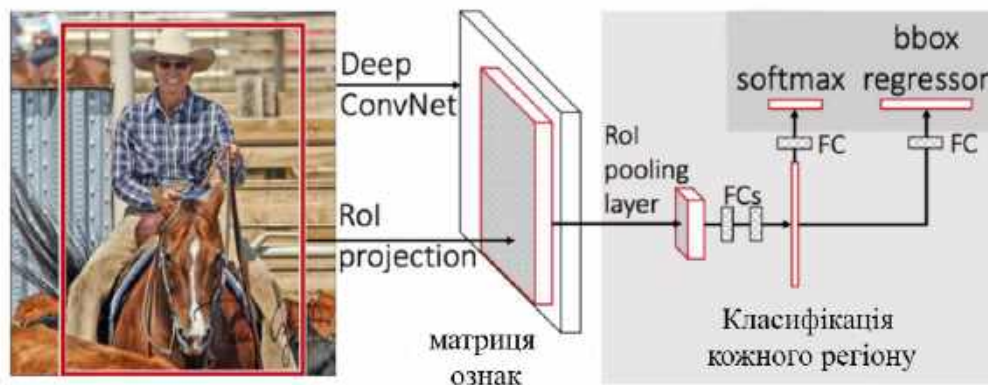


Рисунок 1.6 – Архітектура нейронної мережі Fast R-CNN

1.2.3 Faster R-CNN

З описаної архітектури Fast R-CNN [30] є зрозумілим, що вона також не позбавлена недоліків. Основним з них є алгоритм «Вибіркового Пошуку» (Selective Search) [29]. По-перше, він є алгоритмом, тому не навчається та відповідно не адаптується до потрібної задачі, враховуючи семантику. По-друге цей алгоритм є досить навантаженим по розрахунках, а тому займає багато часу. Щоб виправити ці проблеми було запропоновано нейронну мережу для генерування пропозиції регіонів (Region Proposal Network) [31].

Нейронна мережа для пропозиції регіонів навчається пропонувати регіони з об'єктами на основі карти ознак, що була згенерована для всього зображення за допомогою згорткової нейронної мережі. Для генерації таких регіонів опираються на якорі (anchors) – це заздалегідь заготовлені регіони [45], які використовуються як основа для вибору. Для цього кожен з якорів порівнюється до карти ознак та при відповідних умовах цей регіон вважається таким, в якому ймовірно є об'єкт. При навчанні нейронної мережі використовуються виходи з класифікації від RPN та регресії. Це дозволяє навчати нейронну мережу якісніше

ніж в складі всієї нейронної мережі. Алгоритм роботи нейронної мережі для пропозиції регіонів представлено на рисунку 1.7.

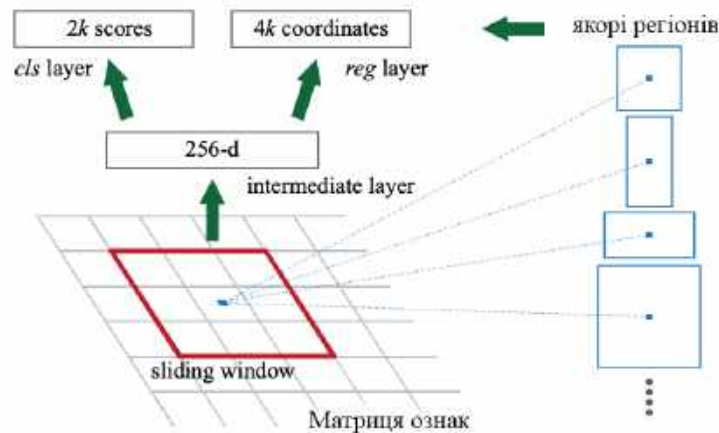


Рисунок 1.7 – Структура нейронної мережі для генерування пропозицій регіонів

Як і в випадку з попередніми архітектурами, R-CNN та Fast R-CNN, наступним є модуль класифікатора, який приймає запропоновані регіони та загальну карту ознак для всього зображення. Візуальна репрезентація архітектури представлена на рисунку 1.8.

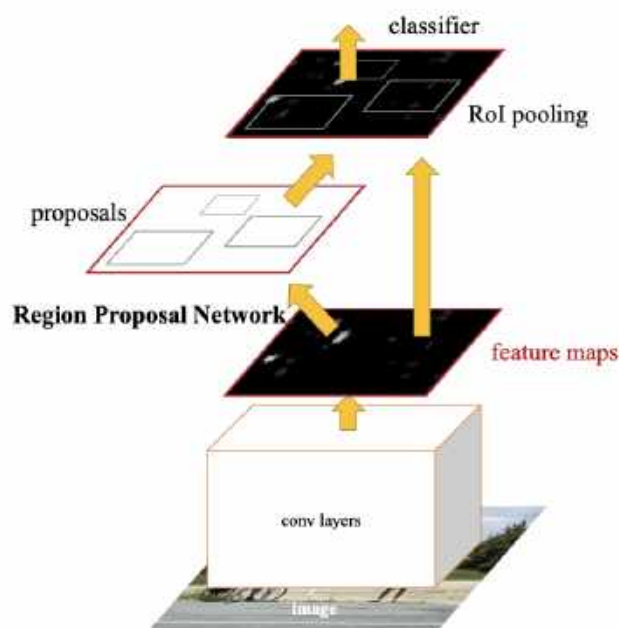


Рисунок 1.8 – Архітектура Faster R-CNN

1.2.4 You Only Look Once (YOLO)

Загалом зараз YOLO [32] – це велика архітектура, що дозволяє реалізовувати різні задачі, такі як локалізація та класифікація [46], сегментація [47], трекінг об'єктів [48] та багато інших задач. Але на початку розробки моделі це була архітектура для локалізації та класифікації об'єктів на зображеннях, яка вирішувала цю задачу як задачу регресії. Архітектура YOLO розвивалась з 2016 року, а тому є доцільним розглянути перший варіант архітектури та подальші зміни в ній.

Перша версія нейронної мережі [32] базувалась на розділенні зображення на 64 блоки та обробки кожного з цих блоків окремо. Цифра 64 обрана таким чином, що вхідне зображення мало розмір 448×448 пікселів, а розмір вхідного блоку в мережу був 7×7 елементів. В подальшому розділені блоки оброблялись окремо, на виході кожен з блоків мав свій клас та значення об'єктності. Об'єктність є значенням ймовірності наявності об'єкту в регіоні. Також на виході для кожного блоку було розраховано вектор належності до кожного класу. Загалом вихідний масив мав розмір 30 елементів для кожного блоку 7×7 . Цей масив складався з двох запропонованих регіонів по 5 елементів (4 – це координати регіону та 1 - об'єктність) та ймовірність для 20 класів. Архітектуру нейронної мережі представлено на рисунку 1.9.

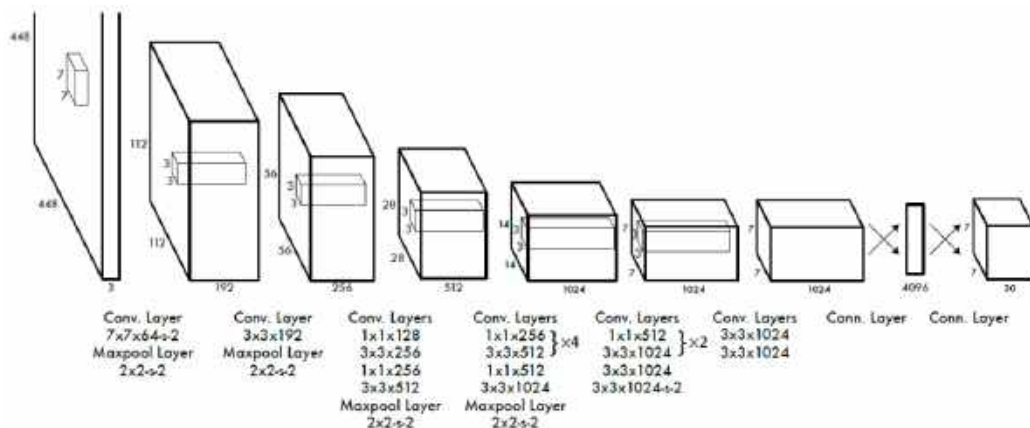


Рисунок 1.9 – Архітектура мережі YOLOv1

Також розглянемо функції втрат, що були запропоновані для навчання цієї нейронної мережі. Загалом вони є комбінацією класичних функцій втрат, але модифікують їх. Загальна функція втрат складається з декількох, це втрати регресії (regression loss) (1.1), втрати об'єктності (objectness loss) (1.2) та втрати класифікації (classification loss) (1.3):

$$\begin{aligned} \text{loss}_{\text{reg}} = & \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\ & + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{\text{obj}} \left[(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right], \end{aligned} \quad (1.1)$$

де x, y - істинні координати центру регіону;

$\hat{x}, \hat{y}$ - прогнозовані координати центру регіону;

w, h - істинні розміри регіону;

$\hat{w}, \hat{h}$ - прогнозовані розміри регіону;

λ_{coord} - задане значення поправки;

1^{obj} - масив об'єктів.

$$\text{loss}_{\text{conf}} = \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2 + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2, \quad (1.2)$$

де C - істинна класифікація;

$\hat{C}$ - прогнозована класифікація;

λ_{noobj} - задане значення поправки для непрогнозованих регіонів;

1^{noobj} - масив регіонів, що не визначені як об'єкт.

$$\text{loss}_{\text{class}} = \sum_{i=0}^{S^2} 1_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2, \quad (1.3)$$

де p - істинна ймовірність належності до класу;

$\hat{p}$ - прогнозована ймовірність належності до класу.

Кожна з функцій втрат має свій вплив на роботу моделі. Так втрати регресії – це фактично помилка в прогнозуванні позиції регіону. Помилка розраховується на основі попередньо розмічених регіонів та прогнозованих за допомогою мережі, варто зауважити що вона враховує тільки один прогнозований регіон, щоб не враховувати двічі одну і ту ж помилку. Також в другій частині для координат використовуються корені, це міра масштабування, яка дозволяє вірніше штрафувати малі регіони.

Класифікаційні втрати – це фактично середньоквадратична похибка, яка розраховується для ймовірностей належності до класу. Похибка розраховується відносно передбачених логітів до розміченого вектору належності до класу.

Втрати об'єктності відповідають за оцінку «впевненості» моделі в тому, що в прогнозованому регіоні є об'єкт неважливо якого класу. При цьому ймовірність виражається в прогнозуванні значення функції Intersection over Union (IoU) [49]. Так, при високому прогнозованому значенні метрики модель впевнена в тому, що в регіоні є об'єкт, при низькому значенні об'єкту, ймовірніше, немає в регіоні. Використана функція втрат вчить модель вірно прогнозувати параметр об'єктності як для регіонів з об'єктом, так і пустих регіонів, що зменшує кількість хибно позитивних результатів.

Результат роботи архітектури такий, що на першому етапі ми отримуємо масив з передбаченими результатами для 98 регіонів для частин зображення розміром 7x7 на вході, для кожної вхідної частини на виході є індекс початку та кінця. Для кожного з регіонів визначено локаційні параметри, параметр об'єктності та вектор належності до класу. Отримані регіони відфільтровуються по параметру об'єктності, який має перевищувати задану величину. Наступним

етапом є використання функції NMS для об'єднання регіонів з однаковим класом та близьких локаційно в один регіон, який вже відповідатиме всьому об'єкту.

Перша версія цієї архітектури була дуже швидкою, але мала проблеми з точністю прогнозування. В той час як Faster R-CNN мала точність 73,2%, YOLOv1 мала точність 63,4% [41].

Друга версія архітектури YOLO [50] зазнала перетворень. По-перше розробники замінили базову архітектуру на власну розробку Darknet-19 замість Google Net [51] в першій версії. Друга версія має менше згорткових шарів, а тому є швидшою. Друга модифікація – це зміна повнозв'язних шарів в кінці моделі на згортки. Також в архітектуру додали шари нормалізації, які добре впливають на збіжність моделі. З цією ж метою були додані і skip connection шари, які попереджають перенавчання моделі. Також великою зміною є зміна розширення вхідного зображення до розміру 416x416 пікселів, що дозволило розділити зображення на непарну кількість частин та в середині зображення використовувати накладання частин. Також збільшили кількість частин зображення на вході до 13x13. Зміни також зазнав вихідний вектор для кожної частини зображення, в новій версії до однієї частини відносилось 5 регіонів, які в свою чергу класифікувались окремо один від одного, тому кількість параметрів на виході виросла до розміру 13x13x125. Отримана архітектура показана на рисунку 1.10

Також в оновленій версії YOLO змінився підхід до локалізації об'єктів. Розробники надихнулись ідеєю з Faster R-CNN та додали в реалізацію якорі, які розраховували на основі датасету. Задача локалізації переросла в задачу визначення зміщення якоря відносно розрахованого вигляду. Тобто новій версії моделі не потрібно підбирати параметри регіону, а потрібно просто визначити параметри зміщення якоря відносно нульової точки. Така задача є простішою та надійнішою з точки зору точності відповідей. В процесі навчання змін не було, розраховувалась та ж сама функція втрат та в кінці використовувався алгоритм NMS.

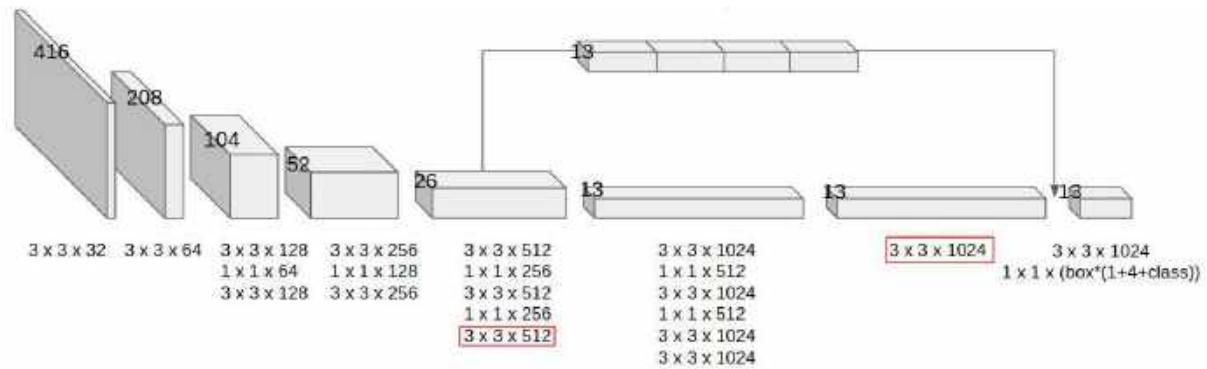


Рисунок 1.10 – Архітектура YOLOv2

В результаті отримана нейронна мережа мала досить високі показники точності роботи відносно її конкурентів, таких як Faster R-CNN та SSD. Також архітектура була досить швидкою в роботі та могла працювати в реальному часі. Але розробники через декілька років випустили також третю версію моделі.

Третя версія YOLO [52] отримала глибшу мережу в якості бекбону, в ній використали Darknet-53 замість Darknet-19, новий бекбон мав 53 згорткових шари проти 19 в старій версії. Через це швидкість роботи моделі впала, але разом з тим сильно виросла якість її прогнозувань. Також архітектура зазнала ще однієї глобальної зміни, автори використали підхід Feature Pyramid Networks (FPN) [53], ця нова архітектура має 3 виходи, які відрізняються по розміру, як показано на рисунку 1.11. Сенса зміни полягає в можливості знаходити об'єкти різного розміру, зокрема дуже дрібні. Це працює таким чином, що при більшій щільності ознак, ми можемо локалізувати більш дрібні об'єкти.

Також змін зазнала функція втрат, частину з втратами об'єктності модифікували, тепер вона не напряму відображає метрику IoU, а більше її модифікацію [54]. В модифікованому вигляді частина функції втрат для об'єктності (1.4) має становити 1 для об'єкту, який найімовірніше є в регіоні. Також зміни зазнала функція втрат для класифікації (1.5), на відміну від минулих версій використали крос-ентропію [55], яка краще підходить для даної задачі.

$$\begin{aligned} \text{loss}_{\text{conf}} = & \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{\text{obj}} \left[\hat{C}_i \log(C_i) + (1 - \hat{C}_i) \log(1 - C_i) \right] + \\ & + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{\text{noobj}} \left[\hat{C}_i \log(C_i) + (1 - \hat{C}_i) \log(1 - C_i) \right] \end{aligned} \quad (1.4)$$

$$\text{loss}_{\text{class}} = \sum_{i=0}^{S^2} 1_i^{\text{obj}} \sum_{c \in \text{classes}} \left[\hat{p}_i(c) \log(p_i(c)) + (1 - \hat{p}_i(c)) \log(1 - p_i(c)) \right]. \quad (1.5)$$

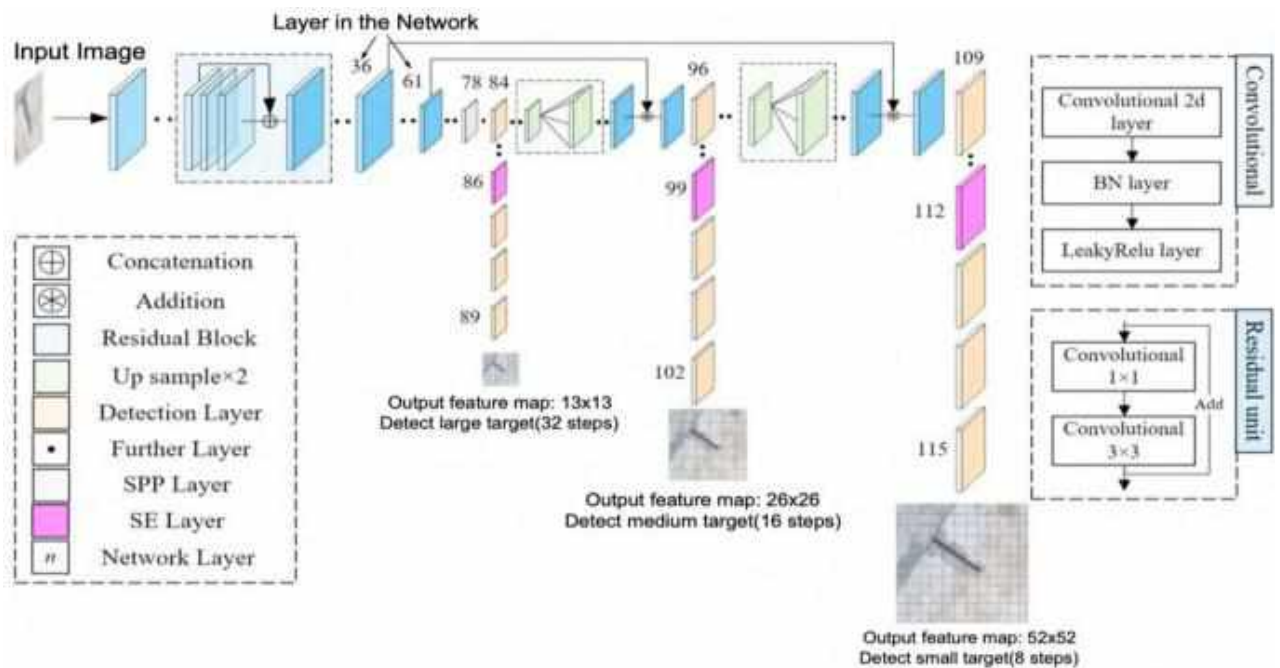


Рисунок 1.11 – Архітектура YOLOv3 з використанням FPN

Отримана архітектура не стала точнішою за аналоги, але є швидшою, ніж більшість конкурентів.

В 2020 році архітектуру продовжили модифікувати та вийшла версія YOLO v5 [56], яка використовує бекбон CSPDarknet53 як модифікацію. Архітектурно це такий же бекбон, як в попередній версії, але було змінено функції активації на Mish замість Relu, а також метод нормалізації було змінено на Cross Stage Partial. Також в якості шарів пропускового з'єднання було використано Spatial Pyramid Pooling (SPP) блоки. Паралельно з цим було додано шари насичення для більш

якісної роботи мережі для мапи ознак малого розміру. В процесі навчання також пройшли некритичні зміни, в якості аугментації даних автори запропонували метод мозаїки для склеювання декількох зображень та отримання так званого мозаїчного зображення.

Подальша версія YOLOv8 [57] зазнала зміни в архітектурі та ефективності. Зі сторони архітектури змінили шари CSP на EfficientRep блоки. Для згорткових шарів використано комбіновані шари згортка-нормалізація-активація (Conv-BN-ReLU). Також модифікували архітектуру бекбону, наразі він не використовує якорів регіонів для розрахунку, що спрощує навчання на об'єктах різного розміру. Загалом процес навчання залишився практично схожим на версію YOLOv5, але набув модифікацій в етапі обробки даних.

1.2.5 Single Shot Detector (SSD)

Будова моделі SSD [33] є досить схожою на раніше розглянуті варіанти. В якості екстрактора ознак з зображення використовується попередньо навчена нейронна мережа. Її архітектура може бути різною в залежності від потрібної точності та швидкості роботи моделі. Так це можуть бути ResNet [40], MobileNet [41] чи інші екстрактори [43]. Типовим рішенням є використання MoboleNetV2 [41] чи VGG16 [42] з вагами, які навчено на наборі даних ImageNet [38].

Наступними шарами є так звана голова SSD (SSD head), загальна архітектура моделі представлена на рисунку 1.12. Це основна логіка нейронної мережі, розглянемо її детальніше.

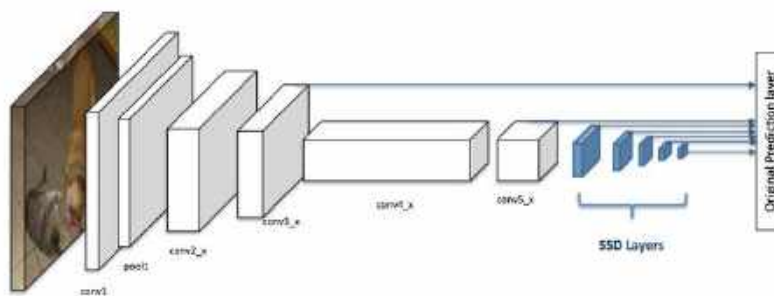


Рисунок 1.12 – Архітектура Single Shot Detector

Робота мережі подібна до роботи YOLO, в ній також використовується підхід розділення зображення на комірки, в результаті чого ми отримуємо так звану сітку. Для кожної комірки прогнозується наявність об'єкту та його клас. Також тут був використаний підхід з використанням якорів, які є попередньо розрахованими. В процесі роботи мережа намагається співставити якір та об'єкт з максимальним ступенем перекриття, де також враховується параметр співвідношення сторін, який в процесі навчання коригується. В параметрах моделі також є параметр приближення або масштабування, який вказує на скільки можна масштабувати якір для співставлення з об'єктом. Загалом архітектура є схожою на раніше розглянуті версії.

1.2.6 RetinaNet

Архітектурно RetinaNet [34] також має схожість з попередньо розглянутими версіями. В ній також використовується попередньо навчена згортова нейронна мережа в якості екстрактора ознак, тобто бекбону. Вона також має мультимасштабну архітектуру. В якості мережі, яка слугує для об'єднання ознак різного розміру, використовується Feature Pyramid Network (FPN) [53]. Вихідна частина має дві гілки, одна з яких це класифікаційна частина, інша – регресійна, що використовується для визначення регіону об'єкту. Загальна архітектура нейронної мережі представлена на рисунку 1.13.

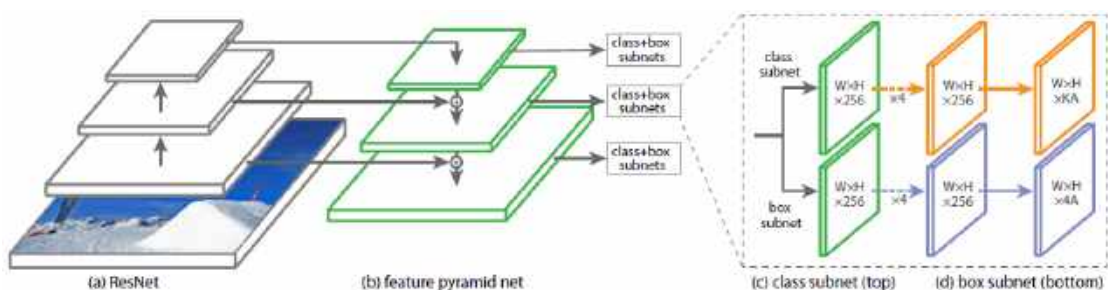


Рисунок 1.13 – Архітектура RetinaNet

Архітектура є досить точною та має велику схожість з попередньо розглянутими варіантами. Варто зазначити, що RetinaNet вийшла раніше ніж YOLOv3 [52], яка використовує схожий підхід, та моделі були конкуруючими з точки зору якості та швидкості роботи.

1.2.7 Порівняння нейронних мереж за метриками

Розглянуті нейронні мережі мають різні параметри, кількість шарів та будову, а тому досить сильно відрізняються між собою особливо в архітектурному плані. Використання нейронних мереж для більшості задач пов'язано з їх попередньою оцінкою на глобальних задачах, що схожі до тієї, в якій вони будуть використані. Враховуючи, що нейронні мережі будуть використані для задач локалізації та класифікації однією з таких оцінок можна вважати отримання метрик точності роботи, використовуючи попередньо натреновані ваги нейронних мереж, на наборі даних COCO [58]. Отримані характеристики дозволять визначити точність нейронних мереж, а також доцільність їх використання в даній задачі. Отримані в результаті оцінки значення метрик представлено в таблиці 1.1. Також на рисунку 1.14а представлено стопчикову діаграму значень метрик для порівняння нейронних мереж за точністю локалізації та на рисунку 1.14б – для точності класифікації.

Нейронні мережі показують досить високі значення метрики якості локалізації (IoU), яка для нейронних мереж YOLO сягає 0,85 - 0,87, для інших нейронних мереж лежить в діапазоні 0,77 – 0,79. Такі значення метрик можуть відображати високу точність нейронної мережі в задачах локалізації. Значення метрики класифікації для більшості досліджуваних нейронних мереж лежить в діапазоні 0,5 – 0,65, що є також досить хорошим показником, враховуючи що набір даних має 80 класів. Враховуючи це, можна підсумувати що нейронні мережі мають досить високу точність локалізації та класифікації та придатні для використання в подальших дослідженнях.

Таблиця 1.1 – Метрики точності локалізації та класифікації

Нейронна мережа	Точність локалізації (IoU)	Точність класифікації (F1)	FLOPS, Т	Кількість параметрів, млн
Faster R-CNN (ResNet50)	0,7769	0,6233	0,197	41,753
Faster R-CNN (ResNet101)	0,7863	0,6379	0,269	60,745
Faster R-CNN (ResNetXt)	0,7889	0,6515	0,273	60,374
RetinaNet (ResNet18)	0,7646	0,5753	0,179	21,41
RetinaNet (ResNet50)	0,7777	0,6008	0,227	37,969
RetinaNet (ResNet101)	0,7823	0,613	0,299	56,961
RetinaNet (ResNetXt)	0,7858	0,6262	0,303	56,589
SSD (MobileNetV2)	0,7709	0,5008	0,000774	4,229
SSD (VGG16)	0,78	0,5989	0,098629	36,038
YOLOv5n	0,8485	0,5503	0,0078	2,654
YOLOv5s	0,8585	0,6149	0,0242	9,153
YOLOv5m	0,8672	0,6534	0,0646	25,111
YOLOv5l	0,8687	0,6742	0,135	53,225
YOLOv8n	0,8518	0,5713	0,0089	3,157
YOLOv8s	0,8583	0,6209	0,0288	11,166
YOLOv8m	0,8669	0,6603	0,0793	25,902
YOLOv8l	0,8713	0,6779	0,165	43,691

Для задач, що пов'язані з використанням обмеженої кількості ресурсів, а локалізація та класифікація зображень з БПЛА є саме такою задачею, важливим є моніторинг кількості використовуваних ресурсів на етапі планування системи. Оцінка параметрів кількості операцій в секунду (FLOPS) та кількості параметрів представлено в таблиці 1.2. Враховуючи що система локалізації та класифікації може бути різного типу, тобто опрацьовувати зображення відразу на БПЛА чи

використовувати інші пристрої для обробки можемо використати кілька типів нейронних мереж. Вони відрізняються між собою кількістю параметрів та кількістю операцій в секунду. Потужні нейронні мережі, такі як Faster R-CNN, можуть бути використані в системі, де обробка проводиться на окремому, виділеному або хмарному, пристрої. Нейронні мережі, що мають менше навантаження, наприклад YOLO можуть бути використані відразу на БПЛА. На рисунку 1.15а представлено порівняння нейронних мереж за параметром кількості операцій в секунду, а на рисунку 1.15б – порівняння за кількістю параметрів.

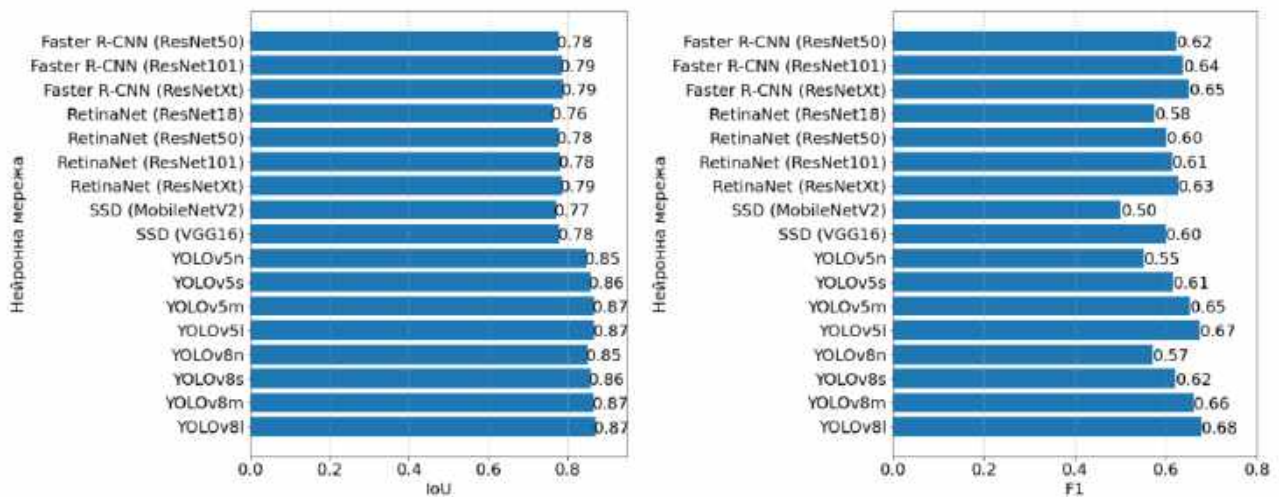


Рисунок 1.14 – Стопчикова діаграма метрик точності локалізації (а) та точності класифікації (б)

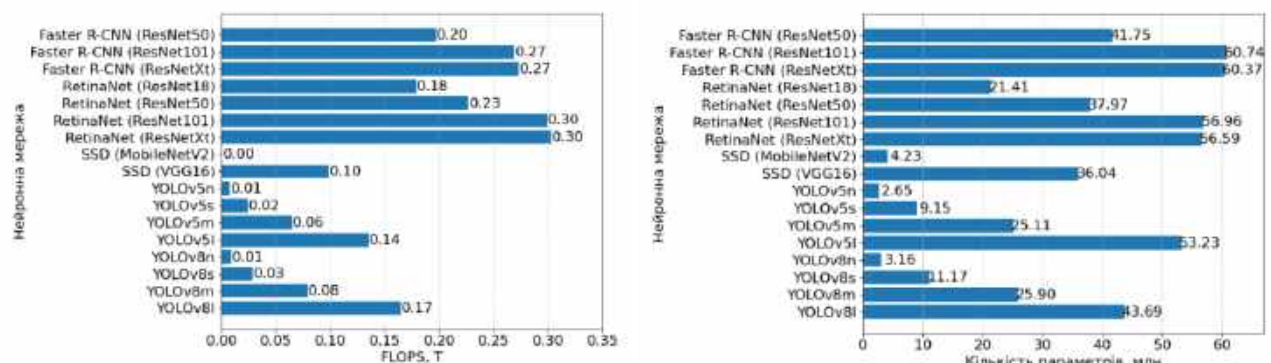


Рисунок 1.15 – Стопчикова діаграма обчислювальної складності (а) та кількості параметрів (б)

1.3 Процес навчання нейронних мереж

Процес навчання нейронних мереж може відрізнитись в залежності від структури нейронної мережі та її виду. Для нейронних мереж, які представлені в попередньому підрозділі, здебільшого використовується навчання методом зворотного поширення помилки [59, 60]. Робота метода полягає в обчисленні градієнтів вагових коефіцієнтів, оновлюючи їх за допомогою градієнтного спуску [61]. Для обчислення градієнтів вагових коефіцієнтів використовується правило ланцюгової похідної. В залежності від модифікації алгоритму функція втрат може бути обчислена для різного розміру вхідних даних. Так, в більшості випадків ця кількість залежить від розміру пакету (batch) даних, але може бути використаний метод з розрахунку функції втрат кожної епохи. Алгоритм навчання нейронної мережі з використання зворотного поширення помилки представлено на рисунку 1.16.



Рисунок 1.16 – Алгоритм навчання згорткової нейронної мережі

Враховуючи алгоритм роботи, важливим є розрахунок потрібного виду функції втрат. Від вибору цієї функції залежить якість навчання нейронної мережі, швидкість цього навчання та взагалі його можливість. В багатьох

випадках, особливо в комплексних задачах, використовують складні функції втрат, які складаються з декількох простіших функцій. Це потрібно для відображення втрат кожного виду, в випадку з представленими мережами це втрати локалізації та втрати класифікації (регресії). Функції втрат буде розглянуто в підрозділі 1.3.1.

Після навчання нейронної мережі або в процесі навчання в заданий період нейронна мережа оцінюється на предмет якості її прогнозувань. Здебільшого алгоритм навчання полягає в оцінці нейронної мережі кожної епохи, це дає можливість, опираючись на метрики, розуміти якість навчання та доцільність подальшого навчання нейронної мережі. Алгоритм оцінки нейронної мережі на предмет точності її прогнозувань представлено на рисунку 1.17.

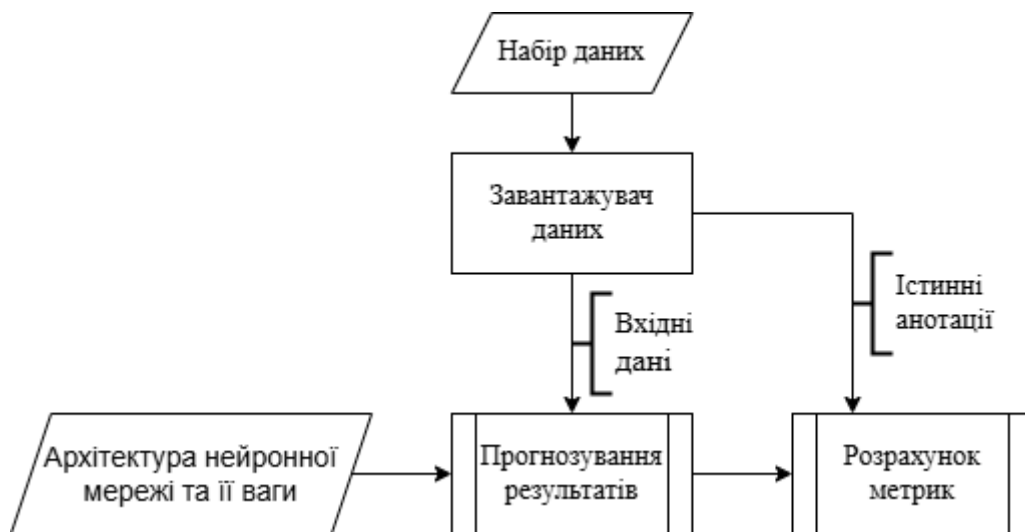


Рисунок 1.17 – Алгоритм оцінки нейронної мережі

1.3.1 Функції втрат

Функції втрат відіграють важливу роль в процесі навчання нейронної мережі. На їх основі вираховується градієнт вагових коефіцієнтів, що розповсюджується зворотнім поширенням помилки. В процесі навчання нейронних мереж для локалізації та класифікації здебільшого використовують комплексну функцію

втратах, що складається з декількох частин. Кожна з складових функції втрат відображає якість локалізації чи класифікації об'єкту, в комплексі вони представляють собою оцінку роботи всієї мережі.

Для втрат класифікації часто використовують перехресну ентропію (Cross-Entropy) [55], яка може мати різну модифікацію в залежності від кількості класів. Для двох класів використовують бінарну перехресну ентропію (Binary Cross-Entropy) [55], яку представлено в (1.6):

$$L = -\frac{1}{N} \sum_{i=0}^N [y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)], \quad (1.6)$$

де y - істинна класифікація;

$\hat{y}$ - прогнозована класифікація;

N - кількість елементів в пакеті даних.

Для задач класифікації на більше, ніж два класи, використовують категоріальну перехресну ентропію (Categorical Cross-Entropy [55]) (1.7), її використовують, коли об'єкт належить до одного з класів. Також модифікацією функції втрат є розріджена категоріальна крос-ентропія (Sparse Categorical Cross-Entropy) [55] в якій для розрахунку використовується не логіти, а індекси класів.

$$L = -\frac{1}{N} \sum_{i=0}^N \sum_{k=0}^K y_{ik} \log \hat{y}_{ik}, \quad (1.7)$$

де K – кількість класів.

Якщо розподіл класів є не рівномірним, то має місце використання фокусної функції втрат (Focal Loss) [34] (1.8). Вона дає змогу краще відображати втрати класифікації враховуючи балансувальний коефіцієнт для кожного з класів.

$$L = -\alpha(1 - \hat{y}_i)^\gamma y_i \log \hat{y}_i, \quad (1.8)$$

де α - балансувальних коефіцієнт;

γ - фокус-параметр.

Під час навчання нейронних мереж, які представлено в роботі, використовується категоріальна крос-ентропія для розрахунку втрат класифікації. Це зумовлено такими факторами, як її точність, розповсюдженість та швидкість розрахунку.

Переходячи до втрат локалізації варто зауважити, що локалізація об'єктів є задачею прогнозування числових значень, тобто регресії, тому для цієї задачі використовують регресійні функції втрат. В залежності від архітектури моделі ми можемо прогнозувати як зміщення від базового (якірного) регіону, так і сам регіон. Від типу прогнозування функції втрат фактично не залежать, так як враховують тільки різницю між прогнозованими значеннями та попередньо розміченими значеннями.

Типовою для таких задач є використання L1 функції втрат або середньої абсолютної похибки (Mean Absolute Error [62]) (1.9). Фактично функція втрат є середнім значенням абсолютної похибки для всіх прогнозованих регіонів, тому вона досить чутлива до викидів.

$$L = \frac{1}{N} \sum_{i=0}^N |y_i - \hat{y}_i|, \quad (1.9)$$

Також в задачах локалізації часто використовують L2 функцію втрат, яка відображає середньоквадратичну похибку (Mean Squared Error) [62] (1.10). Функція втрат є чутливою до великих значень похибки, що відображається на спотворенні результатів всього зображення.

$$L = \frac{1}{N} \sum_{i=0}^N (y_i - \hat{y}_i)^2, \quad (1.10)$$

Для максимізації впливу кожної з функції та позбавлення від типових проблем для кожної з них часто використовують згладжені втрати L1, або Huber Loss [63] (1.11). Вона враховує абсолютну різницю для кожного з регіонів та в залежності від її значення розраховує середньоквадратичну похибку чи абсолютну похибку.

$$L = \begin{cases} 0.5(y - \hat{y})^2, & \text{якщо } |y - \hat{y}| < 1 \\ |y - \hat{y}| - 0.5, & \text{інакше} \end{cases}, \quad (1.11)$$

1.3.2 Метрики якості прогнозування

В процесі навчання нейронної мережі важливо відслідковувати показники якості її роботи на кожному етапі. Звичайно, одним з показників є значення функцій втрат, яка слугує для розрахунку градієнту, але в багатьох випадках її недостатньо для вираження точності нейронної мережі. Для того, щоб якісніше виразити точність прогнозувань використовують метрики якості роботи нейронної мережі [64, 65]. Їх існує досить велика кількість, тому розглянемо метрики, які відповідають за точність локалізації та класифікації як окремо так і в загальному вигляді.

Найчастіше використовувані метрики для визначення точності класифікації та у випадку з задачами локалізації – кількості локалізованих регіонів, є точність (Precision) та повнота (Recall) [66]. Точність виражає частку об'єктів які є вірно класифіковано, від загальної кількості об'єктів, що були позначені як такі, що входять до класу. Повнота в свою чергу виражає частку об'єктів, що були віднесені до вірно класифікованих від загальної кількості об'єктів, що належать до цього класу. Для розрахунку цих метрик потрібно підрахувати кількість позитивно прогнозованих (True Positive), негативно прогнозованих (False Positive), позитивно не прогнозованих (True Negative) та негативно не прогнозованих (False Negative). Візуально такий розподіл можна представити за

допомогою графіку на рисунку 1.18. Так, в якості позитивно прогнозованих (True Positive) обрано ті об'єкти, які класифіковано вірно відносно розмітки. Для негативно прогнозованих (False Positive) обрано об'єкти, класифікація яких відповідає невірно позитивно класифікованим об'єктам досліджуваного класу. Загалом ці дві категорії складають собою загальну кількість об'єктів, які за прогнозами нейронної мережі віднесено до визначеного класу. Перейдемо до категорій непрогнозованих об'єктів, сюди належать позитивно непрогнозовані (True Negative) об'єкти, тобто ті, які не визначені в розмітці і не класифіковані нейронною мережею. Також до цієї категорії входять негативно непрогнозовані (False Negative), які позначають об'єкти, що були визначені в розмітці як належні до класу, але нейронною мережею прогнозовані як неналежні. Разом з позитивно прогнозованими (True Positive), негативно непрогнозовані (False Negative), складають собою категорію релевантних об'єктів, тобто елементів які визначені в розмітці.

Враховуючи отриманий розподіл можемо представити точність (Precision) (1.12) та повноту (Recall) (1.13) за допомогою математичних формул, для кращого розуміння:

$$P = \frac{\sum TP}{\sum TP + FP} \quad (1.12)$$

$$R = \frac{\sum TP}{\sum TP + FN} \quad (1.13)$$

Варто зауважити доцільність використання цих метрик для задач класифікації. Так повноту краще використовувати для задач, де важлива точність класифікації об'єктів, але не повнота класифікації. Повнота важливіша в задачах, де важливо мінімізувати пропущені об'єкти, але разом з тим кількість невірно класифікованих мало враховується. Тому для використання цих показників краще їх об'єднувати в метрику, яка буде відповідати за їх усереднення.

Однією з метрик, які дозволяють рахувати усереднену оцінку між точністю та повнотою є оцінка F1 (F1 score) [67], яку ще називають середнім гармонійним між точністю та повнотою. Метрика добре виражає точність класифікації, але потребує щоб класи були збалансовані по кількості об'єктів. Такий варіант на практиці трапляється рідко, але є можливість розрахунку метрики для кожного класу окремо та враховувати деяку кількість варіантів для оцінки точності. Математично метрику F1 [67] можна представити наступним чином (1.14):

$$F1 = 2 \frac{P \cdot R}{P + R} = \frac{2TP}{2TP + FP + FN} \quad (1.14)$$

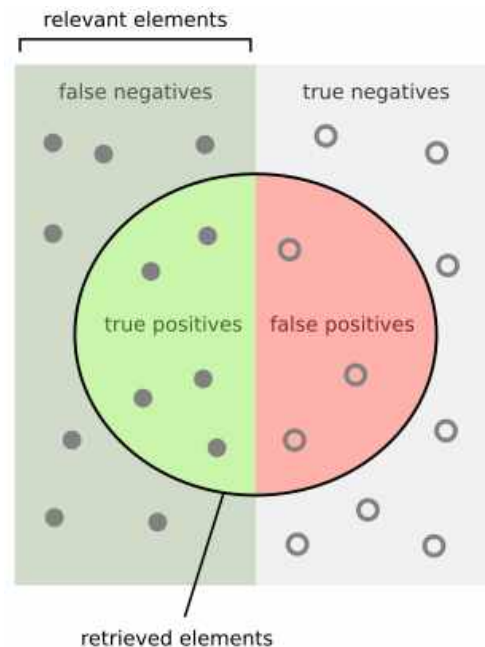


Рисунок 1.18 – Розподіл класифікованих об'єктів на категорії

Також для оцінки якості класифікації об'єктів може бути використана метрика середнього значення точності – mean Average Precision (mAP) [68]. З назви зрозуміло, що метрика виражає усереднене значення точності для всіх класів. Метрика є однією з найбільш використовуваних для задач локалізації та класифікації. Вона враховує тільки якість класифікації, але, модифікуючи її, можна опосередковано досягти відображення точності локалізації. Для того, щоб метрика відображала якість локалізації, використовують Intersection over Union

(IoU) [49]. Математично метрика є площею під кривою Precision/Recall для кожного з класів та їх усереднене значення. Значення площі під кривою Precision/Recall називають Average Precision (AP). Через велику кількість розрахунків метрика є досить ресурсозатратною, але чітко виражає точність класифікації об'єктів.

Попередньо згадані метрики виражають в більшості якість класифікації, хоч вона і частково виражає якість локалізації, через відображення кількості вірно класифікованих об'єктів. Для точнішого відображення якості локалізації часто використовують Intersection over Union (IoU) [49]. Метрика відображає точність локалізації об'єктів відносно попередньо розмічених регіонів. Також метрику часто використовують для пошуку регіонів, які відповідають анотованим. Для цього для кожного анотованого регіону розраховують значення метрики відносно кожного з передбачених та використовуючи порогове значення знаходять регіони які відповідають анотованим. Метрика досить добре виражає якість локалізації, адже є фактично відношенням площі перетину двох регіонів, анотованого та передбаченого, до їх загальної площі. Тобто, фактично метрика виражає відсоток перетину регіонів, де 1 відповідає 100% співпадінню. На рисунку 1.19 представлено графічний вигляд метрики IoU.



Рисунок 1.19 – Візуальна інтерпретація метрики Intersection over Union.

Описані варіанти метрик добре виражають оцінку точності класифікації та локалізації. Загалом процес визначення точності нейронної мережі за допомогою тестового набору даних можна описати наступним чином. Спочатку за допомогою порогового значення проводять відбір регіонів, для цього використовують значення об'єктності чи значення ймовірності класу, в залежності від архітектури мережі. Наступний етап - це розрахунок Intersection over Union, за допомогою метрики проводять оцінку передбачених блоків та вираховують оцінку точності локалізації. Також співставленні регіони обираються для оцінки точності класифікації в якості області класифікованих регіонів. Також неспівставні регіони беруть участь в оцінці точності класифікації, але в якості некласифікованих – в випадку розмічених регіонів, та невірно класифікованих – в випадку з передбаченими, як належні до класу. На основі цієї інформації проводять оцінку якості класифікації за допомогою розрахунку Precision, Recall, F1 Score та mean Average Precision [69].

Розглянути метрики оцінки якості роботи нейронних мереж було визначено, що для відображення точності локалізації зазвичай використовують метрику Intersection over Union або її модифікації. Це дає змогу оцінити точність локалізації, враховуючи кожен піксель. Саме цю метрику буде використано в роботі для оцінки точності прогнозувань нейронних мереж в розрізі якості локалізації. Що стосується класифікації - визначено, що метрика F1 досить часто використовується для оцінки якості класифікації. Вона враховує Precision та Recall, а тому може опосередковано відображати їх значення. Саме тому в роботі буде використано метрику F1 для оцінки точності класифікації об'єктів на зображеннях.

1.4 Зображення, отримані за допомогою повітряної зйомки

Дистанційна зйомка досить часто використовується для розв'язання різних типів задач [70, 71]. Це можуть бути задачі дослідження земної поверхні, пошуку

різних об'єктів [72, 73], екологічного моніторингу [11] і таке інше. Зображення, що отримані за допомогою дистанційної зйомки, можуть бути різними за типом, методом зйомки чи якістю [74]. Існує декілька основних напрямків дистанційної зйомки - це зйомка із супутників, що слугує в більшості випадків для дистанційного зондування Землі [75]. Або це може бути зйомка з різного типу літальних апаратів, як пілотованих так і безпілотних [76]. Дисертаційна робота присвячена зображенням, що отримані за допомогою різних типів авіаційних літальних апаратів, а тому є доцільним розгляд їх характеристик таких зображень детальніше.

Отримання зображень за допомогою повітряних літальних апаратів є досить широкою темою [77]. На даний час все частіше зображення отримують за допомогою безпілотних літальних апаратів (БПЛА) з використанням камер високої роздільної здатності. Це дозволяє отримувати зображення в автоматичному режимі з високою роздільною здатністю для регіонів великого розміру (рисунк 1.20а). Такий метод часто забезпечує кращу якість зображень, порівняно з супутниками, та точність геоприв'язки до визначеного регіону.

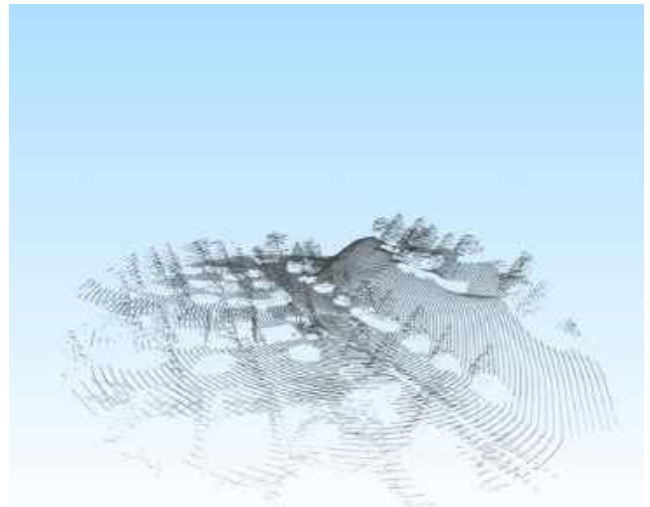
В якості девайсу, в основному, використовують квадрокоптер, через його маневровість та можливість до стабілізації при зміні обстановки чи погодних умов. В якості систем управління використовують різні типи блоків управління літальним апаратом, які в більшості працюють з системою GPS. Літальний апарат також оснащений камерою з сенсором типу КМОН (комплементарна структура метал-оксид-напівпровідник), який має високу роздільну здатність. Від роздільної здатності залежить можливість розрізнення об'єктів різного розміру, при високій роздільній здатності є можливість розрізняти об'єкти досить малого розміру.

Також існують технології спектральної зйомки за допомогою літальних апаратів [78]. В таких пристроях може використовуватись сенсор який захоплює ближній інфрачервоний діапазон. Це забезпечує можливість аналізу різних поверхонь та їх характеристик, які не досить помітні в умовах звичайної зйомки. Існують також інші типи сенсорів, які можуть бути встановлені на пристрій для

зйомки, та їх вибір залежить від потреб користувача. Це можуть бути сенсори з лідаром (рисунок 1.20б), стереосенсори [79] та інші.



а



б

Рисунок 1.20 – Приклади зображень, що отримані за допомогою безпілотного літального апарату в дійсному кольорі (а) та з датчику лідар (б)

В якості формату зображень використовуються тип зображень, що придатний до використання в поточній ситуації. Наприклад в ситуаціях, коли знімки швидко потрібно передати на пристрій керування чи обробки, використовують різні методи стискання з втратами, найчастіше JPEG. Якщо ж зображення опрацьовуються вже після їх зйомки, то часто використовують RAW формат даних, який дозволяє зберігати більше інформації.

Часто такі зображення використовують для побудови карт рельєфу, об'ємних карт місцевості [80], зокрема гірської, до якої є обмежений доступ. Досить часто такий метод отримання інформації застосовують в сільському господарстві, для оцінки якості рослинності, її ступені зрілості та інше.

Використання таких зображень в системах локалізації та класифікації також є поширеною темою [81, 82]. Їх застосовують для різноманітних задач, зокрема для локалізації різного типу об'єктів та визначення їх характеристик.

Все ж зображення, що отримані за допомогою БПЛА можуть бути спотворені різними факторами, що мають вплив на точність роботи методів локалізації та класифікації. Зокрема такими факторами може бути шум різної інтенсивності та природи [15]. Дослідження що стосуються підвищення точності роботи різних методів також є досить широкими, в них використовуються різні методи придушення шуму. Зазвичай дослідження не враховують природу шуму та його інтенсивність, а також можуть бути недостатніми дослідження величини покращень яких можна досягнути за допомогою попередньої обробки зображень. Тому дослідження природи та інтенсивності шуму на зображеннях, а також його впливу на точність роботи нейронних мереж є досить актуальною задачею. Також актуальним є дослідження впливу фільтрації на підвищення точності роботи методів локалізації та класифікації об'єктів на зображеннях.

Шум є не єдиним фактором, що може впливати на якість зображень з БПЛА, можуть мати вплив також погодні умови, зокрема такими є дощ та туман [16, 17]. Вплив таких природніх факторів мало оцінений в дослідженнях, але такі зображення можуть мати досить низьку якість і відповідно вплив на точність роботи нейронних мереж. Враховуючи це задача оцінки якості роботи методів локалізації та класифікації в умовах впливу природніх факторів є важливо та актуальною. Також актуальним є дослідження роботи методів локалізації та класифікації після застосування методів видалення, або зменшення, впливу природніх умов.

Враховуючи що дослідження стосується зображень з БПЛА важливо дослідити точність роботи нейронних мереж в умовах коли об'єкт має малий розмір на зображенні [14]. Розмір можна визначати за допомогою розрахунку кількості пікселів що припадають на цей об'єкт. В випадку з зображеннями з БПЛА це є важливим фактором, адже є ймовірність зйомки з досить великих висот, що відповідно може зменшувати точність нейронних мереж. Загалом дослідники визначають методи покращення точності роботи в таких умовах шляхом використання потайлової обробки. Але важливо також розуміти доцільність такої обробки на етапі виконання алгоритму. А тому важливим є не

тільки дослідження підвищення якості роботи нейронних мереж в умовах локалізації та класифікації малорозмірних регіонів, а і визначення граничних розмірів таких регіонів. Тому задача дослідження та визначення розміру регіону в пікселях, якого достатньо для достовірної локалізації та класифікації об'єкту, є актуальною та важливою задачею. Також важливо розуміти певні характеристики системи захвату зображень, а саме відстань до об'єкту, для того щоб будувати систему локалізації та класифікації відповідно до необхідних умов. А тому задача набуває ще більшої актуальності. Необхідним є також розробка методу визначення висоти враховуючи на отримані в процесі досліджень результати.

1.5 Обробка зображень за допомогою нейронних мереж

З ростом популярності нейронних мереж їх використання для обробки зображень різного типу постійно зростає. Вже досить довго нейронні мережі використовуються для обробки зображень різного типу [83]. Часто нейронні мережі використовуються для детектування різного типу об'єктів на зображеннях, прогнозування їх характеристик та іншого. Також нейронні мережі знайшли своє використання в методах зменшення шумових завад на зображеннях, зменшення впливу навколишніх факторів, таких як туман чи дощ. Існує багато методів для зменшення впливу змазу та дефокусу на зображеннях, що отримані в несприятливих умовах чи пошкодженні іншими факторами. Частим методом використання нейронних мереж також є їх стиснення чи прогнозування параметрів стиснення [84].

Задача обробки зображень за допомогою нейронних мереж є широко використовуваною, але часто потребує також попередньої обробки цих зображень [85]. Існує декілька видів такої обробки і всі вони залежать від задачі та технологій що використовуються. Загалом алгоритм попередньої обробки зображення складається з таких етапів:

- розділення на частини (тайли), це дає можливість обробляти зображення набагато більшої розмірності з обмеженими ресурсами. Такий алгоритм може бути використаний в попередній обробці, але не є обов'язковим;

- нормалізація, необхідний крок для обробки зображення нейронною мережею. Це може бути звичайна нормалізація до інтервалу від 0 до 1 (просте ділення на максимальне значення, зазвичай 255). Також можуть бути використані мінімаксна нормалізація чи інші типи нормалізацій.

- перетворення в різні області, наприклад в частотну; не є обов'язковим кроком, але має місце використання в деяких нейронних мережах.

Зворотне перетворення залежить від використання попередньої обробки та типу нейронної мережі або типу отримуваних даних. Якщо на виході ми отримуємо оброблене зображення, то зазвичай в якості зворотного перетворення використовується зворотна нормалізація та, якщо потрібно, складання зображення в одне ціле з частин. Якщо ж вихідні дані не є зображенням, то їх перетворення залежить від типу таких даних. Так для вектору ймовірності належності до класу в задачах класифікації застосовують пошук індексу максимального елементу. Або ж визначення декількох індексів елементів, при багатокласовій класифікації, враховуючи порогові значення. Для задач локалізації використовують алгоритм NMS, для видалення регіонів, що повторюються.

Враховуючи це, можна визначити, що вхідна нормалізація є важливим кроком в обробці зображення за допомогою нейронних мереж. Розглянемо основні типи нормалізації, яка часто використовується в мережах різного типу. Перша нормалізація, яка використовується досить рідко, але є найпростішою, це звичайне ділення на максимально можливий елемент, беручи до уваги що це зображення, маємо значення 255. Після ділення на 255 всі елементи зображення будуть мати значення від 0 до 1. Також може використовуватись розширення діапазону, за допомогою множення на 2 та віднімання 1. В такому випадку отримуємо діапазон $[-1; 1]$.

Мінімаксна нормалізація [86] є широко використовуваним методом нормалізації. Її принцип роботи спирається на знаходженні мінімального та максимального елементу у вхідній послідовності. Після чого застосовується перетворення з формули (1.15):

$$\text{img}_{\text{norm}} = \frac{\text{value} - \min}{\max - \min}, \quad (1.15)$$

де value – значення поточного пікселя,

min – мінімальне значення всього зображення,

max – максимальне значення всього зображення.

Також часто в нейронних мережах використовується нормалізація з використанням попередньо зазначених значень середнього значення та стандартного відхилення [86]. Це дозволяє нормалізувати зображення відповідно до тренувального набору даних, що зменшує викиди даних у вхідних послідовностях. Формула нормалізації представлена в (1.16):

$$\text{img}_{\text{norm}} = \frac{\text{value} - \mu}{\sigma}, \quad (1.16)$$

де value – значення поточного пікселя,

μ – середнє значення пікселів зображення,

σ – середньоквадратичне відхилення.

Розглянуті види нормалізацій найчастіше використовуються в нейронних мережах. Використання інших методів також можливе, але зазвичай вони використовуються рідко та для специфічних нейронних мереж [87].

1.6 Висновки і формулювання задач дослідження

Враховуючи розглянуті методи та засоби локалізації та класифікації об'єктів на зображеннях повітряної зйомки можна зробити висновок що задача є досить

поширеною. Здебільшого для методів локалізації та класифікації використовують нейронні мережі різного типу. Досить багато сфер, які використовують отриману інформацію про об'єкти для досліджень. Це можуть бути аграрні сфери використання, для моніторингу якості посіву, виду рослинності, ступенів її зрілості. Також це може бути сфера дослідження земної поверхні, класифікація її на частини (види земної поверхні), пошук аномалій. Ще однією сферою де такі методи здобули широкого використання є військова сфера або забезпечення правопорядку. Враховуючи це можна зробити висновок, що методи локалізації та класифікації на основі нейронних мереж є досить актуальними та потребують досліджень для підвищення їх точності.

Розглядаючи фактори впливу, які можуть впливати на точність роботи нейронних мереж та загалом методів локалізації та класифікації визначено що шум та природні фактори впливають на якість зображень. Впливаючи на візуальну складову зображення вони теоретично можуть спотворювати результати локалізації та класифікації. Загалом існує досить багато досліджень що стосується факторів впливу на точність роботи нейронних мереж, а також методів запобігання такому впливу та підвищенню точності роботи нейронних мереж. Здебільшого такі методи потребують додаткових доопрацювань. З необхідного можна виділити дослідження роботи методів локалізації та класифікації з малорозмірними об'єктами, а також в умовах шуму та впливу природніх факторів, а також визначення граничних значень характеристик об'єкту в таких умовах.

Розглядаючи нейронні мережі які використовуються для локалізації та класифікації визначено що їх існує досить велика кількість, але обрано обмежений перелік таких нейронних мереж для подальших досліджень. Вони є різними по структурі, часу роботи, точності, але виконують одну і ту ж задачу. Обраний перелік нейронних мереж визначений таким чином, щоб їх властивості дозволяли визначити параметри точності залежно від потреб системи. Тобто нейронні мережі в списку мають різне обчислювальне навантаження та кількість параметрів, для можливості їх застосування як на літальних апаратах так і на

віддалених пристроях для обчислень. Відповідно нейронні мережі, що будуть розглянуті в дослідженнях мають різну точність локалізації та класифікації.

З огляду на отриману інформацію та на вимоги до методів локалізації та класифікації основними задачами дослідження є:

- Провести аналіз відомих нейромережових методів локалізації та класифікації при роботі з зображеннями, що отримані в умовах повітряної зйомки.
- Експериментально дослідити точність роботи нейромережових методів локалізації та класифікації при роботі з малорозмірними об'єктами та визначити мінімальні характеристики об'єкту (розмір в пікселях) для його достовірної локалізації та класифікації.
- Вдосконалити метод локалізації та класифікації для підвищення точності локалізації та класифікації малорозмірних об'єктів. Експериментально дослідити вплив методу на точність локалізації та класифікації малорозмірних об'єктів.
- Розробити метод визначення характеристик системи формування зображень враховуючи експериментально отримані результати про мінімально допустимі характеристики об'єкту (розмір в пікселях) для підвищення точності методу локалізації та класифікації.
- Експериментально дослідити характеристики шуму на зображеннях, його тип та інтенсивність, визначити його вплив на точність локалізації та класифікації.
- Вдосконалити метод локалізації та класифікації об'єктів на зображеннях, що спотворені шумом, шляхом використання методу оцінки інтенсивності шуму та методу фільтрації.
- Експериментально дослідити вплив природніх факторів на точність локалізації та класифікації, а також дослідити можливість використання методів підвищення точності локалізації та класифікації в таких умовах.

РОЗДІЛ 2.

ДОСЛІДЖЕННЯ РОБОТИ НЕЙРОННИХ МЕРЕЖ ДЛЯ ЛОКАЛІЗАЦІЇ ТА КЛАСИФІКАЦІЇ НА ЗОБРАЖЕННЯХ ПОВІТРЯНОЇ ЗЙОМКИ

Розглянувши архітектури нейронних мереж, що можуть бути використані для локалізації та класифікації об'єктів, отримуємо досить велику різноманітність. Але вибір потрібної нейронної мережі залежить від багатьох факторів, які залежать від задачі та техніки, що використовується. Нейронні мережі відрізняються за обчислювальним навантаженням, тому зазвичай, в випадку з БПЛА, вибір нейронної мережі сильно залежить від цього фактору. Важливим критерієм є також точність, тож в багатьох випадках потрібно обирати оптимальний варіант з точки зору точності роботи та навантаження. Тому розділ присвячено оцінці точності локалізації та класифікації та обчислювального навантаження досліджуваних нейронних мереж, які працюють з зображеннями, що отримані за допомогою безпілотних літальних апаратів.

В розділі приведено і проаналізовано метрики точності роботи нейронних мереж, враховуючи навчання на наборі даних з безпілотних літальних апаратів. Враховуючи те, що швидкість обробки зображення є важливим критерієм, буде представлено порівняння нейронних мереж за параметром обчислювального навантаження та кількості параметрів, що впливає на швидкість роботи нейронної мережі. Відповідно розраховано ресурси, які потрібні для запуску нейронної мережі, адже це є також важливим фактором в виборі архітектури.

2.1 Підготовка до навчання

Зараз процес навчання нейронної мережі з точки зору теорії не є важкою задачею. Для спрощення процесу навчання існують бібліотеки для python, які дозволяють організовувати цей процес швидко та з потрібною якістю. До таких бібліотек відносяться Torch [88], TensorFlow [89], Keras [90] та інші. В роботі

використано бібліотеку PyTorch [88] через те, що вона має зрозумілу документацію, велику різноманітність шарів та реалізованих архітектур нейронних мереж та стрімко розвивається. Також за допомогою цієї бібліотеки реалізовано досить багато архітектур в сторонніх репозиторіях.

На процес навчання нейронної мережі та її точність впливають декілька факторів:

- По-перше, це архітектура нейронної мережі, при виборі недостатньо точної архітектури чи такої архітектури, яка не підходить під специфіку задачі, є висока ймовірність отримати поганий результат, тому дослідження нейронних мереж потрібно проводити для кожної конкретної задачі, враховуючи її специфіку.

- Другий важливий фактор - це набір даних, що використовується в процесі навчання. При використанні набору даних з недостатньо точною розміткою чи недостатньою якістю зображень отриманий результат може бути недостатньо точним навіть для архітектур, які дають високу точність для схожих задач.

- Третій важливий фактор – це вибір функцій втрат, які відображають специфіку задачі, а також їх параметрів, що впливають на процес навчання. Так, параметри функції втрат можуть відрізнятись для збалансованих та незбалансованих наборів даних.

- Ще один з важливих факторів - це підбір оптимізатора та параметрів навчання, таких як темп навчання (learning rate) [91], а також планувальник темпів навчання (learning rate scheduler) [92]. Ці параметри дозволяють керувати зміщенням вагів, їх підбором в процесі навчання, і, відповідно, впливають на кінцеву точність нейронної мережі.

Оскільки першому розділі вже обрано набір обмежений перелік нейронних мереж, які можуть бути використані для досліджуваної задачі, то подальша робота полягає в їх дослідженні перших етап якого полягає в навчання нейронних мереж на наборі даних з БПЛА. А тому першим важливим етапом підготовки до навчання нейронних мереж є вибір набору даних для навчання та оцінки в процесі навчання.

2.1.1 Вибір набору даних

Як було вже згадано вище, вибір якісних даних для навчання є дуже важливим пунктом. Набір даних сильно впливає на якість отримуваних прогнозувань. Тому на етапі дослідження даних для тренування потрібно мати основні вимоги до набору даних, щоб запобігти проблем з низькою точністю при навчанні моделі.

Мабуть найважливішою вимогою в виборі набору даних для локалізації та класифікації є якість розмітки, або анотацій. Дані розмічаються анотаторами за допомогою використання різного типу програм. В випадку з анотацією даних для локалізації та класифікації важливою є саме розмітка регіону з об'єктом, яка повинна мати відповідні до параметрів характеристики. Цими характеристиками є розмір регіону, його внутрішні відступи, необхідність розмітки малорозмірних об'єктів. Також важливим параметром якості розмітки є точна класифікація об'єктів за визначеним набором класів. Такий набір класів може бути як абстрактним, в якому класи виражають великі категорії об'єктів, наприклад «транспорт», «людина», «будівля». Або ж набір класів може бути розширеним, маючи градації для типів транспорту, будівель та іншого. Ширина такого розширення може бути досить різною, враховуючи контекст та потреби задачі.

Важливим пунктом є не тільки точність розмітки або анотацій, а і якість вхідних зображень. Так, при низькій роздільній здатності зображення локалізація малорозмірних об'єктів є досить складною задачею, яка, в принципі, не буде показувати високу точність прогнозувань. Також в такому випадку безпомилкова класифікація об'єктів є майже неможливою, адже на об'єкт маємо невелику кількість пікселів (див. розділ 3). Разом з цим, в задачах повітряної зйомки зображення можуть бути спотворені завадами, природними явищами чи іншими факторами, тому задача набуває ще більшої складності (див. розділ 4). Якість зображення також опосередковано впливає на ширину класифікації, так при низькому розширенні краще використовувати більш абстрактну класифікацію, так як візуальне розрізнення об'єктів малого розміру є важкою задачею.

Враховуючи ці фактори, тренувальні зображення повинні бути високої якості з якомога кращим просторовим розрізненням.

Ще один з пунктів, які мають вплив на результат навчання нейронної мережі, це кількість даних. Існує декілька методів, які дозволяють визначити мінімально необхідну кількість даних для навчання, опираючись на кількість параметрів. Одним з таких є метод Вапника-Червоненкіса [93] яка ще має назву ВЧ-теорія. Вона дозволяє визначити ВЧ-розмірність нейронної мережі (2.1) та за допомогою неї розрахувати необхідну кількість параметрів (2.2).

$$VC - \dim \approx O(P \log P), \quad (2.1)$$

де P – кількість параметрів нейронної мережі.

$$N > \frac{1}{\varepsilon} (VC - \dim \log \frac{1}{\varepsilon} + \log \frac{1}{\delta}), \quad (2.1)$$

де N – кількість прикладів для навчання;

ε - допустима помилка;

δ - ймовірність помилки.

Враховуючи параметри нейронних мереж, які будуть обрані для дослідження варто розрахувати ВЧ-розмірність [93] для кожної з нейронних мереж і відповідно визначити мінімально допустиму кількість даних для навчання. Отримані результати представлено в таблиці 2.1. Враховуючи що для деяких нейронних мереж було проведено донавчання, використовуючи попередньо навчені нейронні мережі для екстракції ознак, для них буде використана кількість параметрів, яка відповідає тільки частині яка тренується. Для більшості нейронних мереж необхідна досить велика кількість даних для навчання. Досягти цього набором даних досить важко, а тому в процесі навчання будуть використані попередньо натреновані ваги нейронних мереж з замороженням

перших шарів, а також алгоритми аугментації даних, щоб запобігти перенавчанню нейронних мереж.

Таблиця 2.1 – Кількість параметрів нейронних мереж та оцінена кількість необхідних даних для навчання

Назва нейронної мережі	Кількість параметрів, млн	ВЧ-розмірність	Кількість прикладів для навчання, млн
Faster R-CNN (ResNet18)	16,631	120,0911	1200,911
Faster R-CNN (ResNet50)	15,799	113,7311	1137,312
Faster R-CNN (ResNet101)	15,892	114,4411	1144,411
Faster R-CNN (ResNetXt)	16,298	117,5434	1175,434
RetinaNet (ResNet18)	8,279	57,27394	572,7394
RetinaNet (ResNet50)	10,937	76,98443	769,8443
RetinaNet (ResNet101)	11,029	77,67213	776,7213
RetinaNet (ResNetXt)	11,436	80,71842	807,1842
SSD (MobileNetV2)	0,829	4,722472	47,22473
SSD (VGG16)	6	40,66891	406,6891
YOLOv5n	2,51	16,06318	160,6318
YOLOv5s	9,126	63,51952	635,1952
YOLOv5m	25,07	185,4968	1854,968
YOLOv5l	53,171	410,7819	4107,819
YOLOv8n	3,012	19,51431	195,1431
YOLOv8s	11,139	78,49482	784,9482
YOLOv8m	25,862	191,7063	1917,063
YOLOv8l	43,637	333,3803	3333,803

Підсумовуючи, хочеться виділити три основні пункти, які впливають на вибір набору даних та, власне, на кінцеву точність нейронної мережі. До них входять:

- Точність розмітки, яка виражається в потрібній кількості класів, розташуванні регіонів та їх розмірі.

- Якість зображень в наборі даних, зображення з кращим просторовим розрізненням дають кращий результат локалізації та класифікації, а також впливають на можливість локалізації малорозмірних об'єктів.

- Кількість даних також має великий вплив на швидкість навчання та точність нейронної мережі.

Враховуючи приведені фактори, проведемо вибір набору даних для навчання, використовуючи відкриті джерела та попередньо розмічені набори даних. Дослідимо декілька варіантів таких наборів, один з яких буде використано для навчання нейронних мереж та подальшого дослідження їх точності та впливу різних факторів.

Перший досліджуваний набір даних – це Unmanned Aerial Vehicles Detection and Tracking (UAVDT) [94]. Набір даних створено для навчання та тестування нейронних мереж для локалізації та класифікації об'єктів на зображеннях з БПЛА. Автори набору були зосереджені на створенні такого набору даних, який покривав би більшість варіантів використання таких зображень. Це використання даних в умовах зйомки з великої висоти, несприятливих погодних умов, недостатньої видимості та інше. Загалом в набір даних було обрано 80000 кадрів, які відповідали визначеним ознакам. Кадри послідовностей можна об'єднати в відео, що дозволяє використовувати набір даних в задачах супроводу об'єкту. В наборі даних було проведено розмітку для задач локалізації та класифікації, відстеження одного чи декількох об'єктів. Беручи до уваги факт того, що кадри набору даних це кадри з 10 різних відео з частотою 30 кадрів/с маємо високу надлишковість даних.

В наборі даних анотовано 80000 кадрів та виділено на них близько 840 тисяч регіонів, з нерівномірним розподілом між об'єктами. Загалом на зображеннях анотовано три класи техніки, це автомобілі, грузовики та автобуси, розподіл представлено на графіку 2.1. Розподіл класів є недостатнім для проведення досліджень, яким присвячено дану дисертаційну роботу.

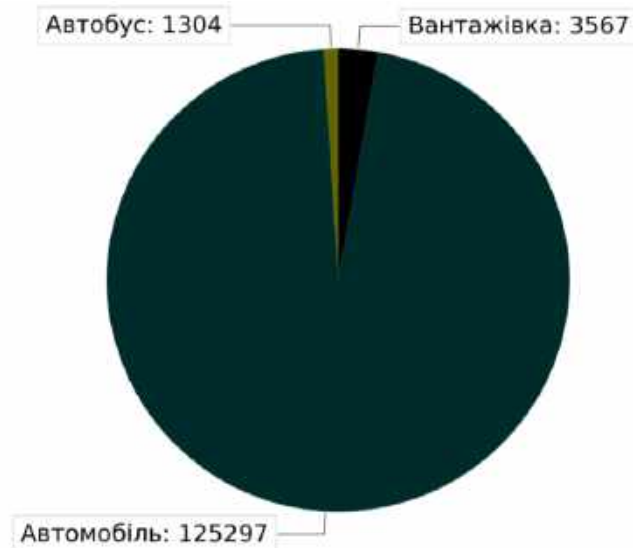


Рисунок 2.1 – Розподіл класів в наборі даних UAVDT

Також мінусом набору даних є додавання регіонів, в яких є неанотовані дані. Ці регіони позначені як ті, що повинні бути ігноровані, що також не відповідає потребам дослідження. Приклад такої розмітки представлено на рисунку 2.2. Помітно, що об'єкти анотовані тільки в зоні дороги, але на зонах, що виходять за їх межі, об'єкти ігноруються.

Враховуючи ці фактори даний набір даних не підходить для поставленої задачі перевірки точності нейронних мереж на даних різних категорій. Але набір даних може бути використаний для розширення набору даних шляхом дорозмітки, а також тестування нейронних мереж в ручному режимі.



Рисунок 2.2 – Приклад анотації в наборі даних UAVDT

Ще один набір даних - це VisDrone [95]. Цей набір даних також може використовуватись для задач локалізації та класифікації, а також для задач відстеження об'єктів. Зображення бути відзняті в різних регіонах, з різною щільністю транспорту та людей, а також різним оточенням та умовами зйомки. Загалом набір даних складається з 263 відео та 10209 зображень, які не перетинаються між собою. Загальна кількість кадрів, враховуючи відео та зображення, складає 179 264 зображень, в яких є також дані з високою надлишковістю. Враховуючи той фактор, що набір даних містить 10209 зображень з низькою надлишковістю то такий набір даних є кращим варіантом для задачі нашого дослідження.

Зображення в наборі даних було анотовано на регіони, розділяючи їх на 10 класів, які можна об'єднати в декілька абстрактних категорій. За класами набір даних розділяється на класи «пішохід», «людина», «велосипед», «автомобіль», «фургон», «грузовик», «трицикл», «трицикл з тентом», «автобус», «мото». Розподіл класів в наборі є досить нерівномірним. Загалом класи можна об'єднати в декілька категорій, це «людина», «автомобіль», «вантажівка», «автобус», «трицикл», «велосипед». Розподіл за категоріями представлено на рисунку 2.3.

Загалом набір даних містить майже 340 тисяч анотованих регіонів, що є достатнім для якісного навчання нейронних мереж. Дані регіони анотовані на зображеннях різного типу, різних погодних умов, різного нахилу камери відносно поверхні землі та інших факторів. Також, на відміну від попереднього набору даних, розмітка проведена для всіх об'єктів на зображеннях, враховуючи також малорозмірні об'єкти. Приклад розмітки приведено на рисунку 2.4.

Враховуючи проведений аналіз наборів даних, а також параметри, які необхідні для проведення дослідження, було обрано набір даних VisDrone. Основними факторами вибору VisDrone є його розмір та якість анотацій. Також важливим фактором, що вплинув на вибір, є розширена кількість класів та можливість їх об'єднання в більш абстрактні категорії. Різноманітність параметрів зйомки та великий розмір зображень є також важливим фактором, який є позитивним для даного набору даних.

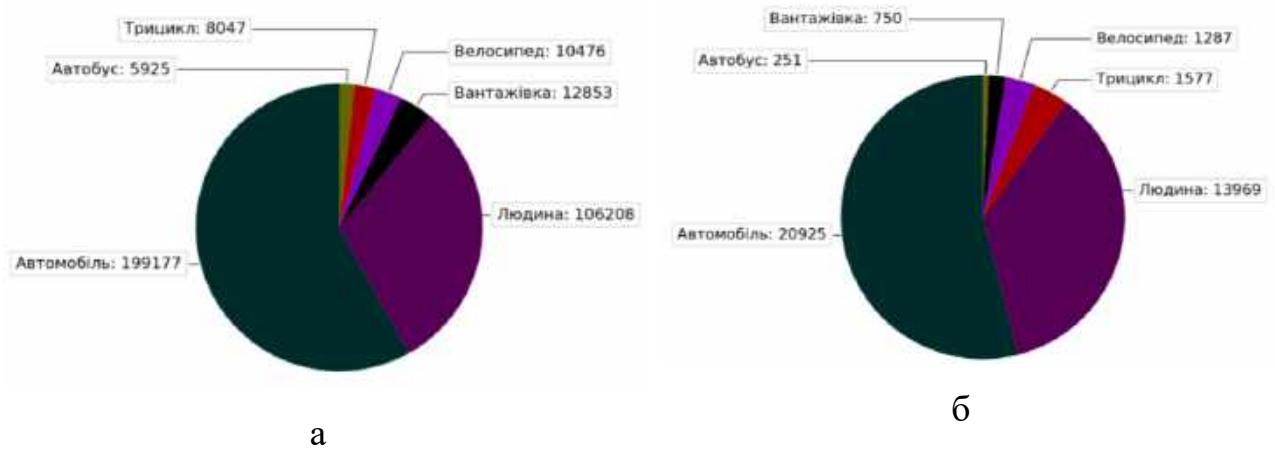


Рисунок 2.4 – Приклад анотацій в наборі даних VisDrone

2.1.2 Вибір фреймворку для навчання

Важливим пунктом при навчанні нейронних мереж є також вибір фреймворку для навчання. Існує декілька фреймворків, які дозволяють реалізовувати процес навчання нейронних мереж. В більшості вони реалізують однаковий функціонал, але мають також і різні функції, які можуть спростувати процес розробки нейронної мережі та її навчання. Розглянемо найпопулярніші фреймворки для навчання нейронних мереж - це Torch [88], TensorFlow [89], Keras [90], JAX [96] та MXNet [97].

Torch [88] є одним з найпопулярніших фреймворків для навчання нейронних мереж, який був розроблений компанією Facebook. Його реалізація дозволяє проводити навчання на Python, з використанням бібліотеки PyTorch [88]. Свою популярність він отримав завдяки широкому поширенню серед дослідників, адже має досить легкий інтерфейс та забезпечує високу гнучкість у розробці моделей. PyTorch [88] підтримує динамічні графи, а тому є можливою зміна структури моделі під час виконання. Основною мовою програмування для використання фреймворку є Python, але також є інтерфейс для C++, що дає змогу запускати навчені моделі на багатьох пристроях, враховуючи підтримку мобільних пристроїв через Torch mobile [88].

Перевагами фреймворку є:

- динамічні графи обчислень, що дозволяє реалізовувати різноманітну логіку в структурі нейронної мережі;
- велика кількість реалізованих архітектур нейронних мереж як в самому фреймворку так і в сторонніх репозиторіях;
- простота синтаксису, яка виражається в інтуїтивно зрозумілому API, що значно спрощує розробку.

Недоліками порівняно з іншими фреймворками є:

- менша масштабованість – розподілення навчання на декількох пристроях є досить обмеженим;
- обмеженість інструментів для візуалізації саме в фреймворку, але є можливість використання tensorboard та візуалізація за допомогою сторонніх бібліотек.

TensorFlow [89] також є дуже поширеним фреймворком для навчання нейронних мереж. Розроблений він був компанією Google та має широке поширення серед науковців та розробників вже декілька років. Базою моделей, що реалізовані за допомогою TensorFlow [89], є графи, що дозволяє будувати складні алгоритми машинного навчання. Також, як і Torch [88], TensorFlow [89] використовує мову програмування Python як основну для використання свого API, але також має реалізації для C++, JavaScript (JS) та інших. Разом з цим

існують бібліотеки для різних мов та платформ, такі як TensorFlow.js, для JS, та TensorFlow Lite для мобільних платформ.

Перевагами TensorFlow [89] є:

- обчислювальні графи, що дозволяє скомпілювати граф перед виконанням моделі та оптимізувати продуктивність навчання;
- широке поширення в науковій спільноті, що виражається в кількості реалізованих нейронних мереж за допомогою цього фреймворку;
- масштабованість, яка виражається в можливості навчання на декількох девайсах;
- інструменти для візуалізації, такі як tensorboard, завдяки яким можна зручніше відслідковувати процес навчання.

Але все ж в фреймворку є також і недоліки, серед яких:

- крута крива навчання, яка виражається в складності вивчення функціоналу через складність API;
- висока вимогливість до ресурсів, при базовому використанні фреймворк використовує більше ресурсів, ніж аналоги;
- часті оновлення API, які в більшості випадків є плюсом, але інколи інтерфейс змінюється критично, що вносить проблеми в підтримку версій.

Keras [90] є скоріше високорівневим API, який використовує низькорівневі фреймворки, такі як TensorFlow [89] чи Theano [98], для реалізації алгоритмів навчання. Основна мета фреймворку - зробити процес побудови нейронної мережі та її навчання простішим та швидшим. Він орієнтований більше на швидке прототипування та простоту використання. Серед особливостей можна відмітити модульну архітектуру та структурованість проектів, що створені за допомогою Keras.

Перевагами фреймворку є:

- простота використання, а саме зрозумілий інтерфейс та синтаксис;
- швидке прототипування завдяки зручному API.

До недоліків використання Keras можна віднести:

- менша гнучкість, яка виражається в обмеженій можливості реалізації більш складних та нетрадиційних архітектур;

- залежність від інших фреймворків, адже keras є просто високорівневим інтерфейсом для низькорівневих фреймворків, тому це також впливає на функціональність.

JAX [96] – це новий фреймворк від Google, який призначений для автоматичного диференціювання та високопродуктивних обчислень. До особливостей можна віднести використання компілятора XLA для оптимізації обчислень та легку реалізацію складних математичних моделей з підтримкою градієнтів.

До переваг фреймворку можна віднести:

- високу продуктивність, яка досягається за допомогою компілятору та оптимізації обчислень;

- підтримка динамічного створення графів та автоматичного диференціювання.

Недоліками фреймворку JAX [96] є:

- низька популярність та поширення, ймовірно через новизну;
- обмежена документація, що також впливає на популярність та розуміння фреймворку.

MXNet [97] є відкритим фреймворком для глибокого навчання від Apache. Він підтримує гібридну архітектуру моделей, тобто графи можуть бути статичні та динамічні, що дозволяє обирати кращий варіант для своїх потреб та максимально контролювати обчислення. Також цей фреймворк підтримує досить велику кількість мов програмування, такі як Python, R, Scala та інші.

Перевагами MXNet [97] є:

- підтримка великої кількості мов програмування, що робить його кроссплатформенним рішенням;

- підтримка розподілених обчислень, що дозволяє масштабувати проекти.

До недоліків відносяться:

- менша популярність, ніж в попередньо розглянутих варіантів;

– складність API, що додає складності в вивченні та роботі з фреймворком.

Враховуючи вказані переваги та недоліки кожного з розглянутих фреймворків можна зробити вибір фреймворку для власних досліджень. Так як дослідження включають досить багато різних архітектур, важливо, щоб фреймворк був популярним, та мав вже реалізовані версії досліджуваних архітектур або підтримував їх швидку реалізацію. Також важливою є можливість навчання з використанням графічних відеокарт, але не важливим є використання розподілених систем, адже досліджувані архітектури не є гіпернавантаженими. Важливо щоб фреймворк підтримував швидку реалізацію потрібного функціоналу та модифікацію нейронних мереж. Враховуючи всі потреби можна визначити що під задачі дослідження підходить фреймворк Torch, враховуючи його позитивні та негативні якості, якими можна знехтувати.

2.2 Навчання нейронних мереж

Для навчання нейронних мереж вже було обрано набір даних, метрики та фреймворк, в якому будуть реалізовані архітектури та процес навчання. Всі нейронні мережі в подальшому будуть оцінені на предмет точності та швидкості роботи за допомогою однакових метрик. Функції втрат для кожної нейронної мережі було обрано відповідно до вказаних в публікаціях щодо цих нейронних мереж. Цей підхід дозволяє навчати нейронні мережі з кращими параметрами, що притаманні кожній архітектурі.

2.2.1 Навчання Faster R-CNN

Для дослідження сімейства архітектур R-CNN [29] було обрано саме Faster R-CNN [31] через її новизну, кращу якість та швидкість. Для навчання обрано Faster R-CNN [31], для якої в якості екстрактора ознак (бекбону) було обрано декілька нейронних мереж. До них відноситься ResNet18, ResNet50, ResNet101 та

ResNetXt [40]. Використання різних бекбонів потрібне для отримання кращої статистики роботи моделі в задачах локалізації та класифікації, а також оцінки впливу бекбону на якість прогнозувань нейронної мережі.

Процес навчання нейронної мережі представлений на рисунку 1.16 та має таку ж структуру для даної нейронної мережі. Враховуючи процес навчання, головним фактором є вибір функції втрат. Як було вказано раніше, функція втрат є такою, яка була використана в роботі, що присвячена даній нейронній мережі [31]. В випадку з Faster R-CNN [31] функції втрат розділяються на дві частини, це функції втрат для мережі прогнозування регіонів (Region Proposal Network) [31] та для отриманих результатів прогнозування. Втратами мережі прогнозування регіонів є бінарна перехресна ентропія (БПЕ) [55] для втрат об'єктності та згладжені втрати L1 [63] в якості втрат локалізації, які підсумовуються для розрахунку загальних втрат мережі прогнозування регіонів. Для результатів прогнозування використовується категоріальна перехресна ентропія (КПЕ) [46], для втрат класифікації та також згладжені втрати L1 для розрахунку втрат локалізації, які також підсумовуються в загальну функцію втрат. Значення функцій втрат підсумовуються для визначення загальних втрат нейронної мережі та використовуються для розрахунку градієнту.

Під час навчання всі функції втрат моделі логувались, усереднюючись за ітераціями, тому можна побудувати графік зміни функцій втрат в залежності від ітерації навчання. Через різницю в розмірі пакету (batch) даних, а відповідно і кількості ітерацій на епоху результати представлено на різних графіках. На рисунку 2.5 представлено графік втрат для архітектури з використанням ResNet18. Результати функцій втрат для нейронної мережі з ResNet50 представлено на рисунку 2.6. На рисунку 2.7 представлено результати функцій втрат для архітектури ResNet101 та на рисунку 2.8 представлено результати для архітектури з ResNetXt для задач локалізації та класифікації.

Помітно, що результати отримані для ResNet50 мають інший вигляд порівняно з іншими, це зумовленою більшою кількістю ітерацій та меншим розміром пакету даних для навчання. Через це результати функції було отримано

з більшою частотою, відповідно вони мають більшу частоту зміни даних. Все ж результати функцій втрат для всіх модифікацій нейронної мережі є досить схожими між собою, а саме тим, що значення функцій втрат класифікації та локалізації попарно збігаються для втрат з кінцевих виходів нейронної мережі, а також з мережі для пропозицій регіонів. Така поведінка свідчить про високу залежність результатів прогнозів нейронної мережі від мережі пропозицій регіонів, яка є архітектурно важливою. Також є помітним різкий підйом функції втрат локалізації для результатів прогнозування. Це пов'язано з покращенням самої мережі для пропозиції регіонів. Після визначеної кількості ітерації ця нейронна мережа починає генерувати більшу кількість валідних регіонів.

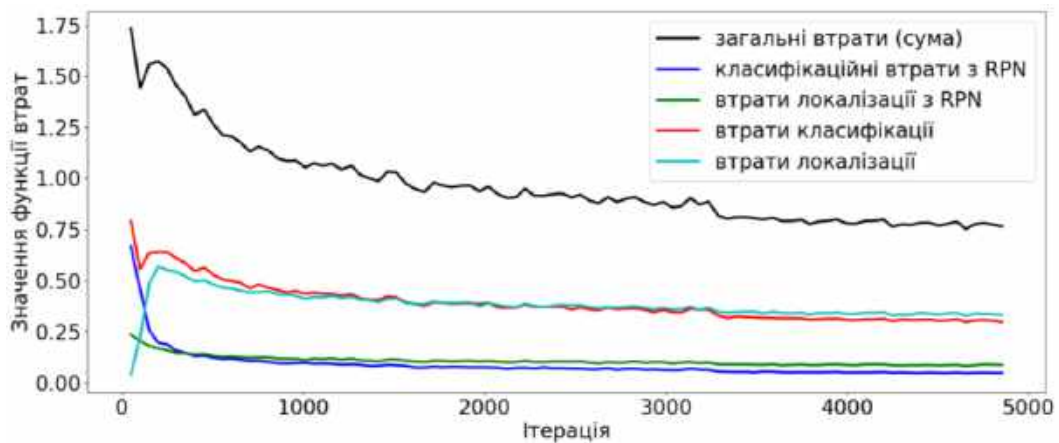


Рисунок 2.5 – Зміна значення функції втрат в залежності від ітерації для Faster RCNN з ResNet18

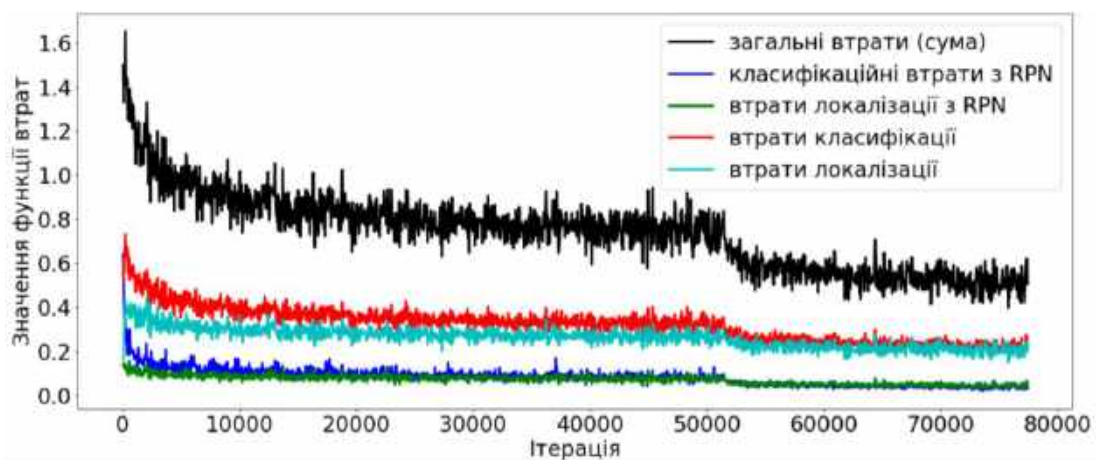


Рисунок 2.6 – Зміна значення функції втрат в залежності від ітерації для Faster RCNN з ResNet50

Загалом, аналізуючи значення функцій втрат, можна сказати що мережа має хорошу збіжність на наборі даних, що був використаний для навчання. А також помітне зменшення спаду функції втрат на кінцевих ітераціях, що може свідчити про вихід на плато, і, відповідно, подальше навчання нейронної мережі є недоцільним.

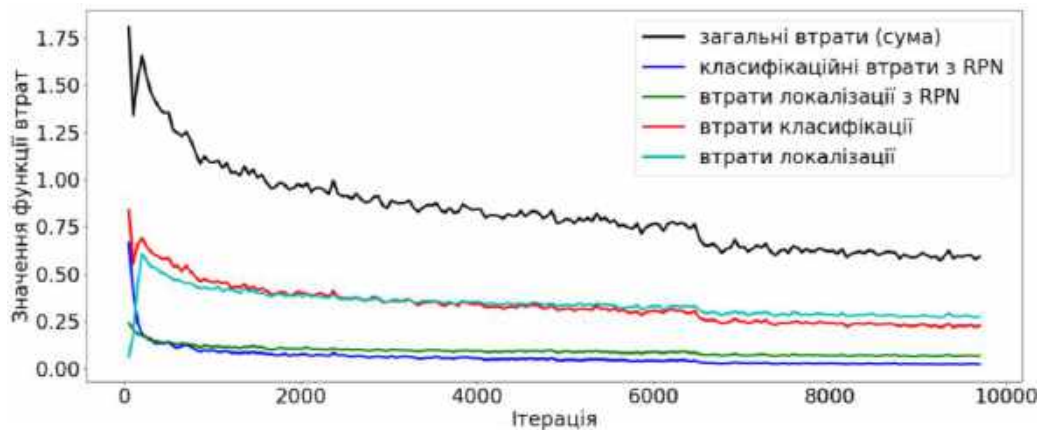


Рисунок 2.7 – Зміна значення функції втрат в залежності від ітерації для
Faster RCNN з ResNet101

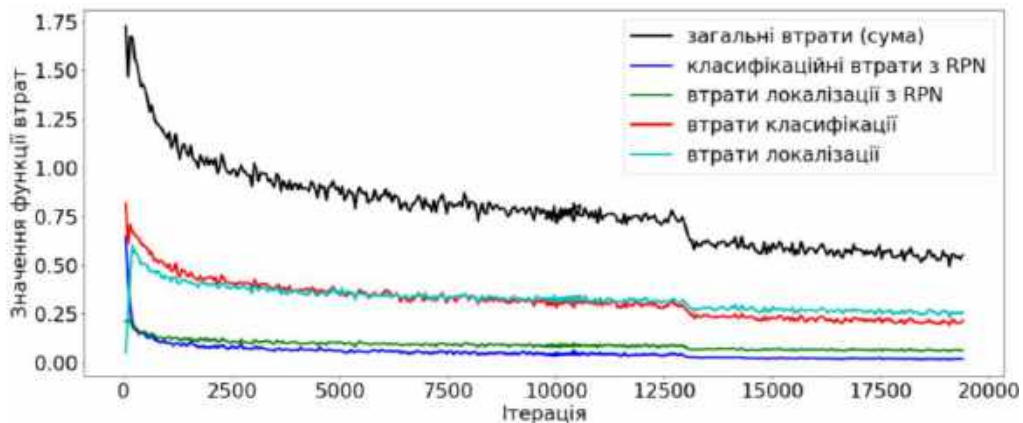


Рисунок 2.8 – Зміна значення функції втрат в залежності від ітерації для
Faster RCNN з ResNetXt

В якості оптимізатора було використано оптимізатор з алгоритмом стохастичного градієнтного спуску (SGD) [61], який досить добре підходить для задач даного типу. Робота оптимізатора залежить від параметрів швидкості навчання (learning rate), моменту (momentum) та спад ваг (weight decay). Від цих параметрів залежить якість навчання та його швидкість, а також взагалі збіжність

моделі. Якщо параметри обрані не вірно, то модель може не досягти кращих параметрів або ж взагалі не збігтись. Для оптимізатора було обрано параметри швидкості навчання рівний 0,02, момент 0,9 та спад ваг 0,0001.

Параметр швидкості навчання є одним з найвпливовіших на збіжність, тому використання планувальника швидкості навчання (learning rate scheduler) позитивно впливає на збіжність моделі. Використання планувальника полягає в зміні параметру швидкості навчання за допомогою зазначених умов. Існують різні варіанти таких алгоритмів, в випадку з Faster RCNN було використано LinearLR [99], який реалізує лінійну зміну параметру швидкості навчання, яка змінюється кожної зазначеної ітерації в умові, а також MultiStepLR [99], який працює протягом всього навчання та на визначених ітераціях перемножує параметр швидкості навчання на параметр gamma, що заданий в конфігурації планувальника. Фактично LinearLR [99] працює перші 500 ітерацій та плавно збільшує параметр швидкості навчання від 0,001 до 0,02. А MultiStepLR [99] під час закінчення 8 та 11 епохи навчання (ітерації залежать від епохи) множить встановлене значення швидкості навчання на 0,1, відповідно після 8 епохи швидкість навчання становить 0,002, а після 11 – 0,0002.

2.2.2 Навчання SSD

Single Shot Detector (SSD) [33] є архітектурою, яка може бути використана в різних реалізаціях. Загалом зберігається основний алгоритм роботи нейронної мережі, який представлений в 1.2.5, але може бути змінений бекбон або екстрактор матриці ознак. Для SSD [33] було проведено декілька експериментів з використанням різних екстракторів матриці ознак, але зі збереженням загальної структури мережі. Цими екстракторами є VGG16 [42] та MobileNetV2 [41].

Процес навчання є аналогічним до інших нейронних мереж. В якості функцій втрат було використано згладжені втрати L1 (smooth L1) [63] для розрахунку втрат локалізації та крос-ентропію (cross-entropy) [55] в якості функції втрат для класифікації. На відміну від Faster RCNN, при навчанні SSD не

використовуються функції втрат для екстрактора матриці ознак, нами було використано тільки втрати класифікації та локалізації для всієї моделі. За отриманими даними втрат при навчанні було побудовано графік зміни функцій втрат від ітерації для різних реалізацій нейронної мереж, тобто для різних бекбонів. Графік зміни функцій втрат для архітектури, основою якої є нейронна мережа MobilenetV2, представлена на рисунку 2.9. На рисунку 2.10 представлено графік залежності функцій втрат від ітерації для архітектурі на основі нейронної мережі VGG.

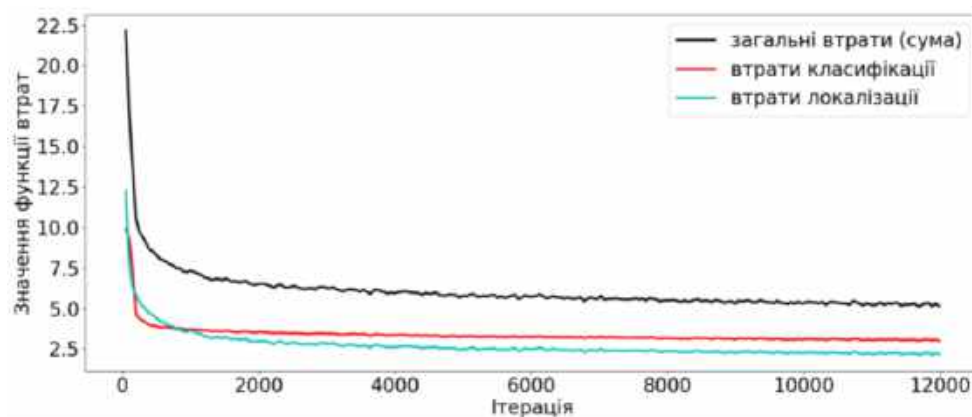


Рисунок 2.9 – Зміна функцій втрат в залежності від ітерації для архітектури з використанням MobilenetV2

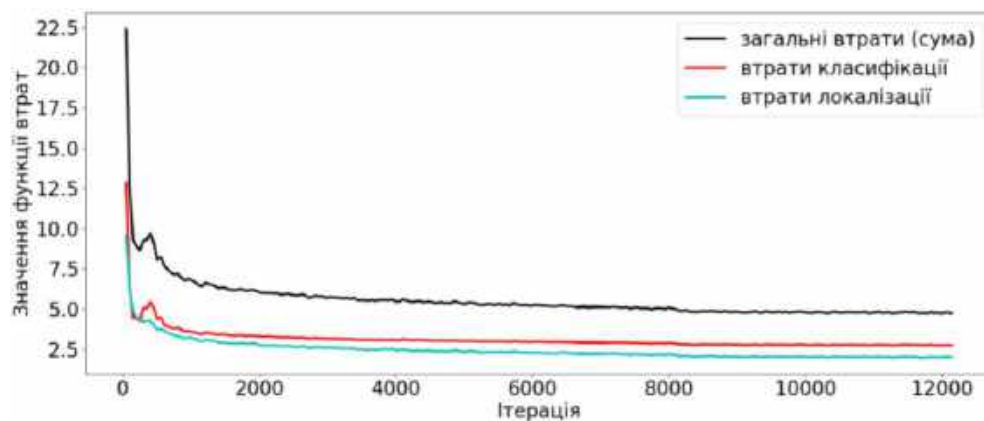


Рисунок 2.10 – Зміна функцій втрат в залежності від ітерації для архітектури з використанням VGG

Аналізуючи графіки, помітною є тенденція швидкого спаду значення функції втрат на перших ітераціях та стабілізація значень ближче до кінцевих ітерацій, що може означати вихід на плато. Помітно, що всі функції втрат мають схожу поведінку, а тому можна зробити висновок, що нейронна мережа придатна до навчання на обраному датасеті з різними варіаціями екстрактора матриці ознак (бекбону).

В якості оптимізатора в процесі навчання було використано стохастичний градієнтний спуск (SGD) [61], який має параметри швидкості навчання (learning rate) рівний 0,002, момент 0,9 та спад ваг 0,0004 для нейронної мережі на базі VGG16, а також параметри швидкості навчання (learning rate) рівний 0,015, момент 0,9 та спад ваг 0,00004 для нейронної мережі на базі MobilenetV2. Також в процесі навчання було використано планувальник параметру швидкості навчання, який змінює параметр в процесі навчання НМ в залежності від заданих умов.

Планувальник параметру швидкості навчання також був використаний для навчання нейронної мережі. Загалом його структура схожа до планувальника, який був використаний для нейронної мережі Faster RCNN [31], тобто використано мультимодульну структуру планувальника. Для нейронної мережі на базі VGG16 [42] його структура не відрізняється, а поведінка є такою, що протягом перших 500 ітерацій швидкість навчання поступово збільшується від 0,0002 до значення, що встановлено в оптимізаторі, тобто 0,002. Наступним етапом є зменшення швидкості навчання 8 та 11 епохи, за допомогою перемноження на параметр gamma, який рівний 0,1. Для нейронної мережі з екстрактором ознак MobilenetV2 [41] було використано змінену структуру планувальника, першим кроком виступає все такий же лінійний планувальник, який змінює параметр швидкості навчання від 0,0015 до значення 0,015. Наступним етапом є використання планувальника CosineAnnealingLR [99], який поступово зменшує параметр швидкості навчання до мінімально заданого за косинусним законом. Мінімально заданий параметр швидкості навчання задано на рівні 0.

Помітно що використання різного типу планувальників швидкості навчання не сильно вплинуло на швидкість навчання нейронної мережі та на якість прогнозованих результатів. Це можуть забезпечувати декілька факторів, один з них - це швидкий вихід моделі на плато через однотипність навчальних даних або ймовірність поганого навчання на досліджуваному наборі даних. Отримані нейронні мережі буде проаналізовано на якість роботи в наступному підрозділі.

2.2.3 Навчання RetinaNet

Нейронна мережа з архітектурою RetinaNet [34] відповідно до попередньо описаних нейронних мереж має модульну реалізацію. Детальний опис архітектури представлено в підрозділі 1.2.6. Варіативність архітектури забезпечуються за допомогою використання різного типу бекбонів, тому в якості досліджень також буде використано різні бекбони. Для тестування було обрано ResNet18, ResNet50, ResNet101 та ResNetXt [40] в якості екстрактора матриці ознак. За допомогою порівнянь точності роботи архітектури на основі цих нейронних мереж можемо обрати кращу версію за параметрами точності, яку буде використано для подальших досліджень.

В якості функції втрат архітектура RetinaNet [34] використовує FocalLoss [34] для розрахунку втрат класифікації та L1 [63] для втрат локалізації. Загальні втрати відповідають сумі втрат локалізації та класифікації. Отримані результати відображені на графіках, але бути розділені за використанням бекбоном через різницю в кількості ітерацій. Кількість ітерацій залежна від розміру пакету даних таким чином, що при зменшенні пакету зростає кількість ітерацій. На рисунку 2.11 представлено зміну значення функції втрат під час навчання для нейронної мережі RetinaNet [34] з використанням ResNet18 в якості бекбону. Для архітектури з ResNet50 таку ж залежність представлено на рисунку 2.12. Графік на рисунку 2.13 відображає залежність значення функції втрат від ітерації для архітектури з бекбоном ResNet101. Для ResNetXt залежність представлено на рисунку 2.14. Відразу помітно, що результат навчання нейронної мережі з

використанням екстрактору ознак ResNetXt має погану збіжність при використаних параметрах навчання.

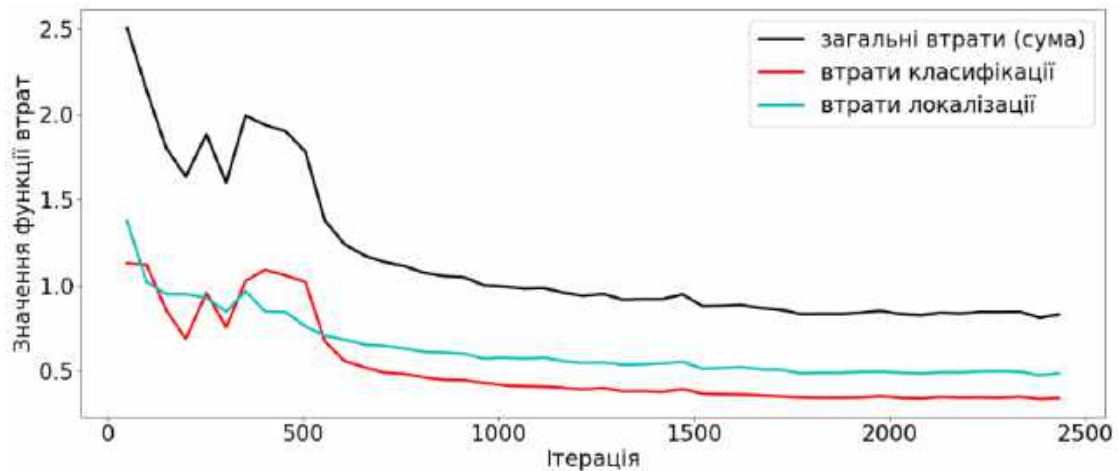


Рисунок 2.11 – Зміна функції втрат локалізації в залежності від епохи для RetinaNet з ResNet18

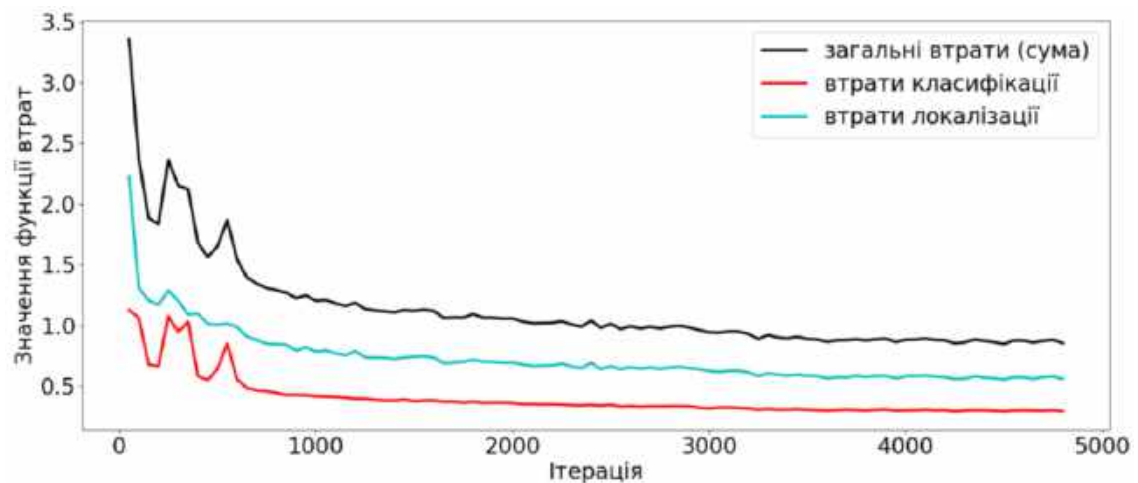


Рисунок 2.12 – Зміна функції втрат локалізації в залежності від епохи для RetinaNet з ResNet50

Аналізуючи отримані результати, також помітна спільна динаміка поведінки функції втрат для версій з ResNet18, ResNet50 та ResNet101. Так, помітний різкий спад на перших ітераціях та плавна зміна на більш пізніх етапах. Такий вигляд функції втрат може відображати процес навчання, який має досить хорошу збіжність, на відміну від ResNetXt на рисунку 2.14. Візуально графік функції

вtrat досить схожий з графіком для нейронної мережі SSD, але має різницю в числових значеннях функції втрат.

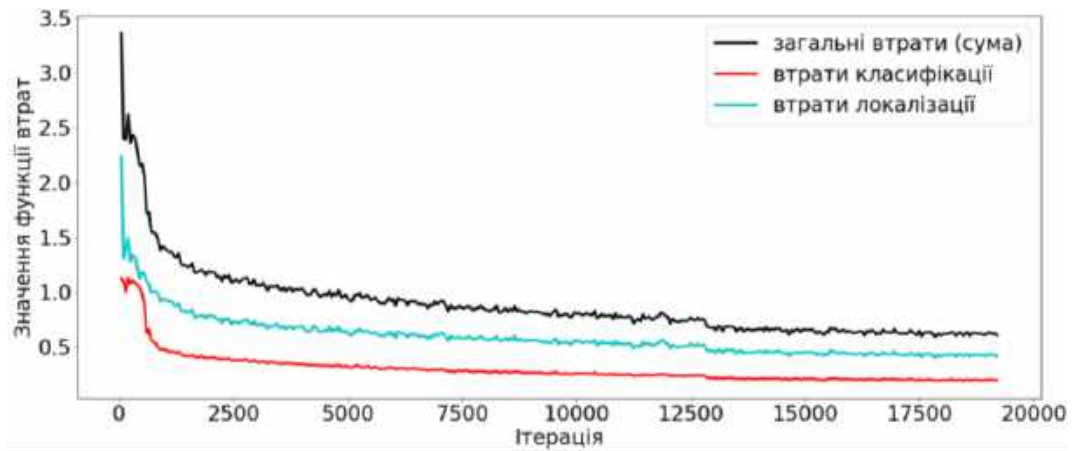


Рисунок 2.13 – Зміна функції втрат локалізації в залежності від епохи для RetinaNet з ResNet101

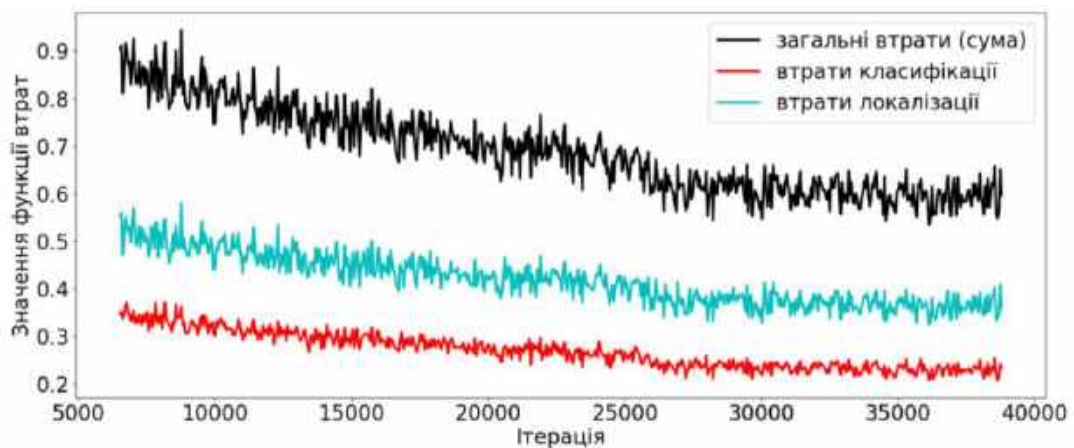


Рисунок 2.14 – Зміна функції втрат локалізації в залежності від епохи для RetinaNet з ResNetXt

Оптимізатором для моделі виступає стохастичний градієнтний спуск (SGD) [61] як і для попередніх архітектур. Параметри оптимізатора схожі до інших моделей, це параметри швидкості навчання рівний 0,01, момент 0,9 та спад ваг 0,0001.

Планувальник параметру швидкості навчання також є комплексним, як і в попередніх архітектурах. Для досліджуваної нейронної мережі він є ідентичним до нейронної мережі Faster R-CNN [31]. Таким чином, початкове значення

параметру визначено як 0,001 та поступово, протягом 500 ітерацій, воно зростає до 0,01, тобто до значення що встановлено в оптимізаторі. Постійний планувальник змінює параметр швидкості навчання після завершення 8-ї та 11-ї епох, перемножуючи його на γ , яка дорівнює 0,1.

2.2.4 Навчання YOLO

Архітектура YOLO [32], яка представлена в підрозділі 1.2.4, має дещо іншу структуру на відміну від попередньо представлених нейронних мереж. Ця архітектура не передбачає використання бекбону в якості екстрактору матриці ознак. Враховуючи це варіаціями для навчання було обрано різний розмір нейронної мережі, який представлено розробникам. Це - YOLOv5n (nano), YOLOv5s (small), YOLOv5m (medium) та YOLOv5l (large) [56]. Відповідно до назв архітектури відрізняються розміром, швидкістю та точністю роботи. Також варіаціями є навчання паралельно архітектур YOLOv8n (nano), YOLOv8s (small), YOLOv8m (medium) та YOLOv8l (large) [57], які є більш новою розробкою авторів 5-ї версії.

Алгоритм навчання нейронної мережі також має деякі відмінності від класичного, а саме в методі попередньої обробки даних. Для підвищення якості прогнозувань в процесі навчання використовується мозаїчна аугментація [52]. Її робота полягає в компонуванні декількох зображень в одне, розташовуючи їх на одному зображенні. Всі об'єкти, що потрапляють в видиму область, обробляються та додаються в анотації. Цей метод використовується для розширення набору даних, а також для запобігання перенавчанню.

Також в якості однієї зі змін в процес навчання була внесена можливість закінчення навчання, якщо функції втрат та метрики, що рахуються в процесі навчання, не змінюються протягом декількох епох. Такий підхід дозволяє не витрачати час, якщо модель не змінює своїх характеристик протягом декількох ітерацій, адже подальше навчання не є доцільним. Зупинка навчання може бути пов'язана з декількома факторами. Перший – це навчання моделі, тобто

досягнення максимальної якості що можлива з поточними параметрами для даної архітектури. Другий фактор – це потрапляння в локальний мінімум, що також відображає недоцільність подальшого тренування з поточними параметрами.

В якості функції втрат використовується комбінована функція втрат, в якій розраховано повні втрати IoU [49], які враховують також неперекриті регіони що підвищує точність відображення результатів, в якості втрат локалізації. В якості втрат класифікації використано бінарну крос-ентропію [55], де в якості класу використано ймовірність (0 чи 1) належності для кожного з можливих класів. Як втрати об'єктності використано бінарну крос-ентропію в класичному її вигляді. Загальна функція втрат виглядає як сума всіх втрат, а саме об'єктності, класифікації та локалізації. Для кожної з досліджуваних версій нейронної мережі було побудовано графіки залежності кожної з функцій втрат, що представлені на рисунках. Так, на рисунку 2.15 представлено графік залежності функцій втрат від епохи для YOLOv5n. На рисунку 2.16 представлено графік функції втрат для YOLOv5s. Для нейронної мережі YOLOv5m графік представлено на рисунку 2.17, для YOLOv5l – на рисунку 2.18.

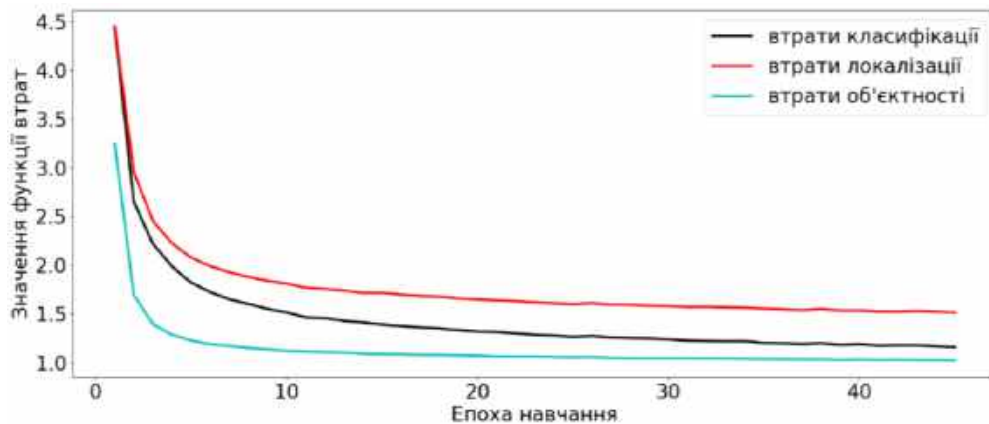


Рисунок 2.15 – Зміна функції втрат для нейронної мережі YOLOv5n в залежності від епохи

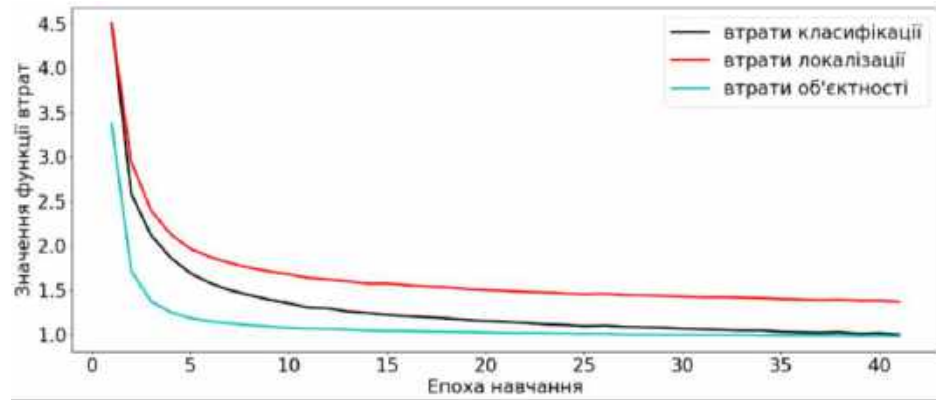


Рисунок 2.16 – Зміна функції втрат для нейронної мережі YOLOv5s в залежності від епохи

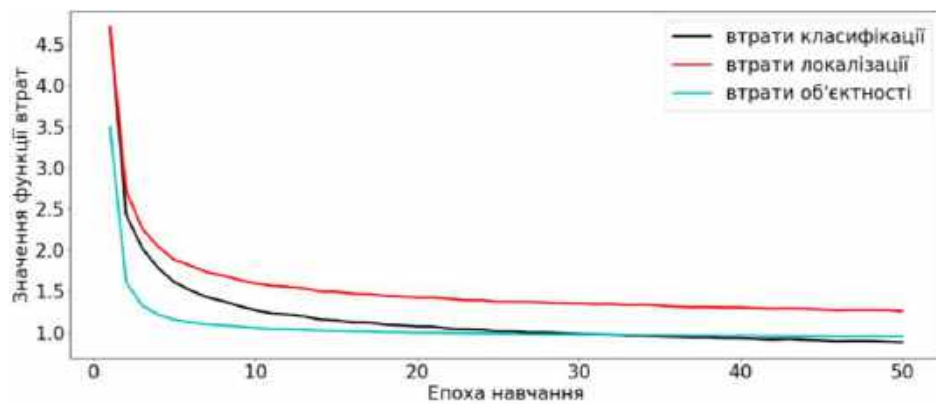


Рисунок 2.17 – Зміна функції втрат для нейронної мережі YOLOv5m в залежності від епохи

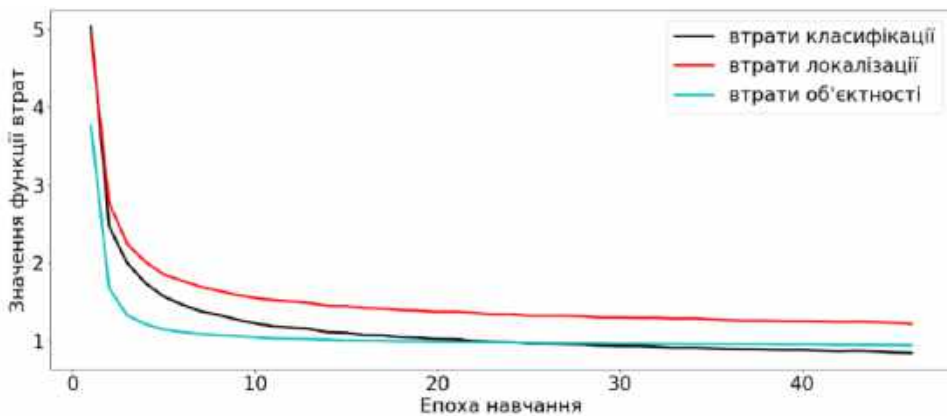


Рисунок 2.18 – Зміна функції втрат для нейронної мережі YOLOv5l в залежності від епохи

Судячи з отриманих даних можна зробити висновки по швидкості навчання досліджуваних нейронних мереж п'ятої серії YOLO. Майже всі нейронні мережі

навчались близько 50-ти епох. А саме найменша версія (YOLOv5n) навчалась 45 епох, наступна по розміру YOLOv5s навчалась 41 епоху, середня версія (YOLOv5m) навчалась протягом 50-ти епох та найбільша версія що використовувалась для досліджень (YOLOv5l) – 46 епох.

Далі розглянемо зміну значень функцій втрат в процесі навчання для восьмого покоління моделей YOLO. А саме YOLOv8n, графік функції втрат якої представлено на рисунку 2.19. Для YOLOv8s графік зміни функції втрат представлено на рисунку 2.20. Відповідно графіки зміни функції втрат для нейронних мереж YOLOv8m та YOLOv8l представлено на рисунках 2.21 та 2.22 відповідно.

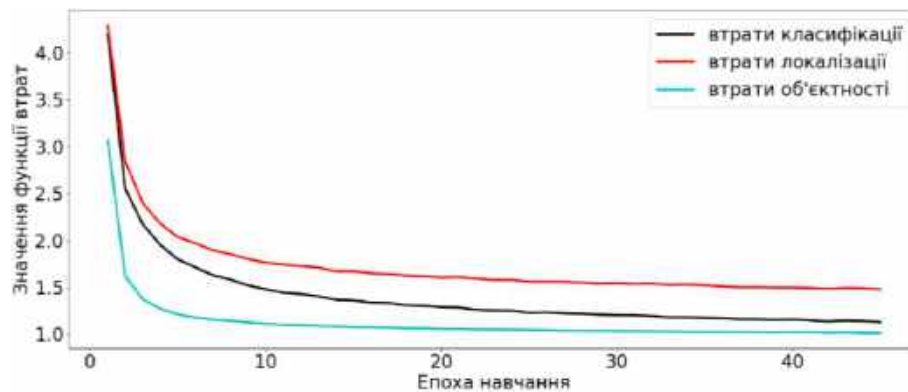


Рисунок 2.19 – Зміна функції втрат для нейронної мережі YOLOv8n в залежності від епохи

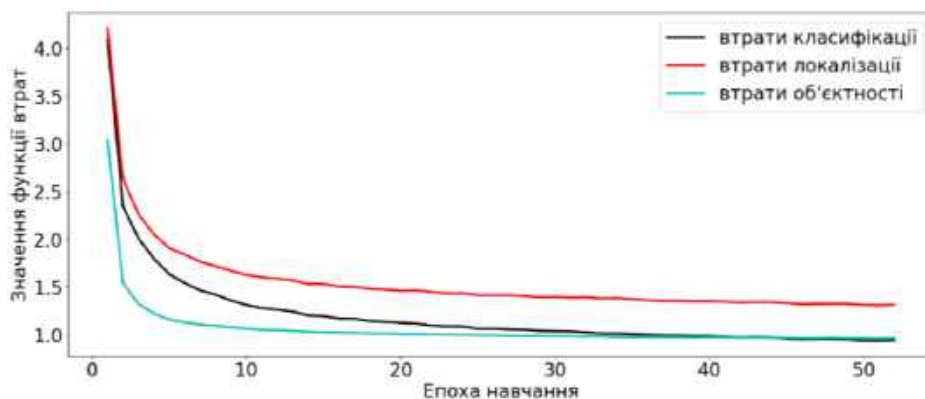


Рисунок 2.20 – Зміна функції втрат для нейронної мережі YOLOv8s в залежності від епохи

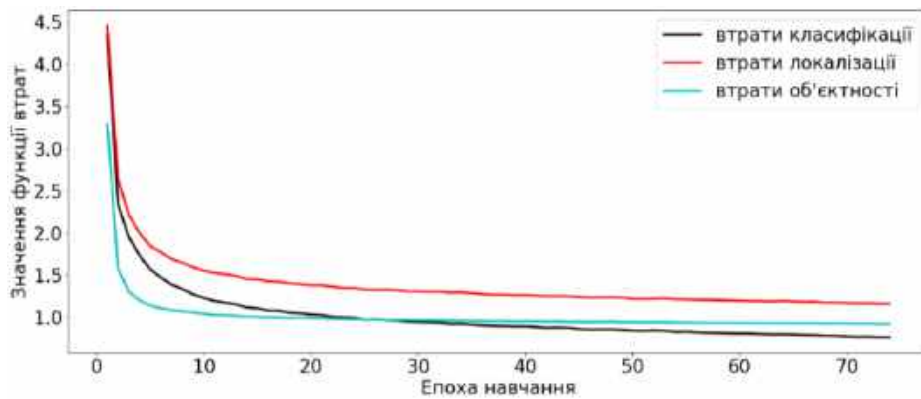


Рисунок 2.21 – Зміна функції втрат для нейронної мережі YOLOv8m в залежності від епохи

Аналізуючи отримані результати, також можна підсумувати кількість епох, протягом яких навчались нейронні мережі. Загалом найменша версія, а саме YOLOv8n, навчалась впродовж 45 епох, що відповідає показнику попереднього покоління (YOLOv5n). Наступна за розміром версія (YOLOv8s) навчалась 52 епохи, що є більшим значенням ніж для попереднього покоління. Середня версія (YOLOv8m) навчалась протягом 74 епох, а найбільша досліджувана версія (YOLOv8l) – 68 епох, що також є більшими показниками за попередню версію. Зміна кількості епох навчання є не критичною та може бути результатом впливу різних факторів та початкових значень вагів моделі.

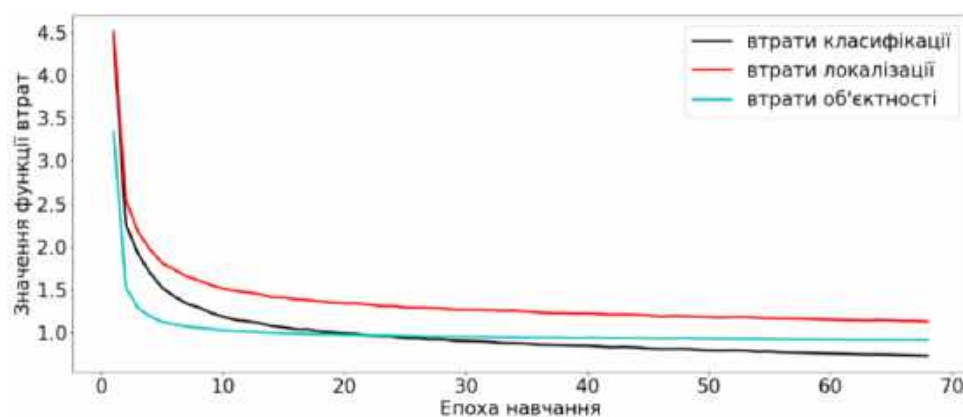


Рисунок 2.22 – Зміна функції втрат для нейронної мережі YOLOv8l в залежності від епохи

Аналізуючи отримані значення маємо досить хороші показники збіжності всіх версій та поколінь моделі. При аналізі помітно, що всі моделі мають різкий спад функції втрат на початку та плавний спад на більш пізніх ітераціях.

Оптимізатором для нейронної мережі YOLO також слугує стохастичний градієнтний спуск. Його параметри встановлені наступним чином: параметр швидкості навчання має значення 0,01, момент 0,937 та спад ваг 0,0005. Ці параметри дозволяють навчати модель з достатньою точністю.

В якості планувальника параметру швидкості навчання використано OneCycleLR [99] з параметром фінального значення швидкості навчання рівного 0,001. В процесі навчання значення параметру плавно спадають, але ближче до фінальних ітерацій маємо агресивне зменшення параметру швидкості навчання, що позитивно впливає на якість навчання та збіжність моделі.

2.3 Порівняння навчених нейронних мереж

Важливою характеристикою для використання нейронної мережі в задачах локалізації та класифікації є точність її прогнозувань. Дослідження точності прогнозувань важливо проводити на одному і тому ж наборі даних для всіх досліджуваних кандидатів. Це дає змогу отримати більш надійну оцінку точності роботи нейронної мережі в порівнянні з іншими. Також важливою є специфіка та якість даних, за допомогою яких проводиться порівняння нейронних мереж. В ідеальних умовах дані повинні бути схожі на дані з пристроїв, за допомогою яких будуть отримані зображення, або ж взагалі зображення що отримані за допомогою таких пристроїв. Це дозволяє якісніше оцінити точність роботи на специфічних даних та обрати нейронну мережу, яка буде відповідати потрібним характеристикам. Важливо, щоб провести автоматизоване тестування, щоб дані були анотовані. В дослідженнях, що представлені в цій роботі дані будуть отримані з тестової частини набору даних VisDrone [95]. Ці дані схожі з

тренувальними та мають однакову специфіку, а тому порівняння нейронних мереж буде проходити в однакових умовах.

Враховуючи, що задача пов'язана з безпілотними літальними апаратами, а тому, в більшості, має обмежені ресурси для запуску нейронних мереж, важливим є також критерій розміру моделі та її розрахункова складність. Параметр розрахункової складності використовують для визначення мінімальних вимог до пристрою на якому буде проведено запуск нейронної мережі. Параметр розміру моделі також використовується для визначення вимог до пристрою, але він враховує мінімально необхідну кількість оперативної пам'яті для запуску нейронної мережі. Розрахункова здатність пристрою залежна від необхідної кількості кадрів, що опрацьовуються за 1 секунду, в той же час кількість необхідного розміру оперативної пам'яті є сталим та практично не залежить від кількості кадрів.

2.3.1 Оцінка точності прогнозувань

Переходячи до оцінки точності прогнозувань розпочнемо з розгляду набору даних для тестування. Як було сказано раніше буде використано тестову частину набору даних VisDrone [95], інформацію про який приведено в розділі 2.1.1. Даних, що представлені в наборі достатньо для проведення тестувань та отримання стійкої оцінки точності прогнозувань. Набір даних має таку ж кількість класів як і тренувальна частина, а тому повністю збігається з параметрами моделі, без необхідності проведення маніпуляцій з підведенням класів. Це є хорошим моментом, адже дозволяє працювати з даними без попередньої обробки, що впливає на швидкість проведення експерименту та мінімізує помилки, що можуть бути допущені в процесі тестування.

Проведемо оцінку точності класифікацій та локалізації для кожної з навчених моделей та їх модифікацій. Для цього тестові зображення було оброблено за допомогою кожної з нейронних мереж та отримано результат. Отримані результати пройшли порівняння та за допомогою метрик було оцінено якість та

точність локалізації та класифікації. Деякі метрики залежні від класу та відображають не тільки кількість локалізованих об'єктів і точність локалізації, а і точність класифікації локалізованих об'єктів. Для відображення точності локалізації було обрано метрику Intersection Over Union (IoU) [49], принцип роботи якої було описано в розділі 1.3.2. Для відображення точності класифікації використана метрика оцінки точності класифікації F1 [67]. Для визначення повноти отримуваних прогнозувань було пораховано відсоток прогнозованих об'єктів відносно оригінальної розмітки.

Розглянемо окремо візуальний результат локалізації для різних мереж. Першою досліджуваною нейронною мережею є Faster RCNN візуальна інтерпретація прогнозувань якої представлена на рисунку 2.23.

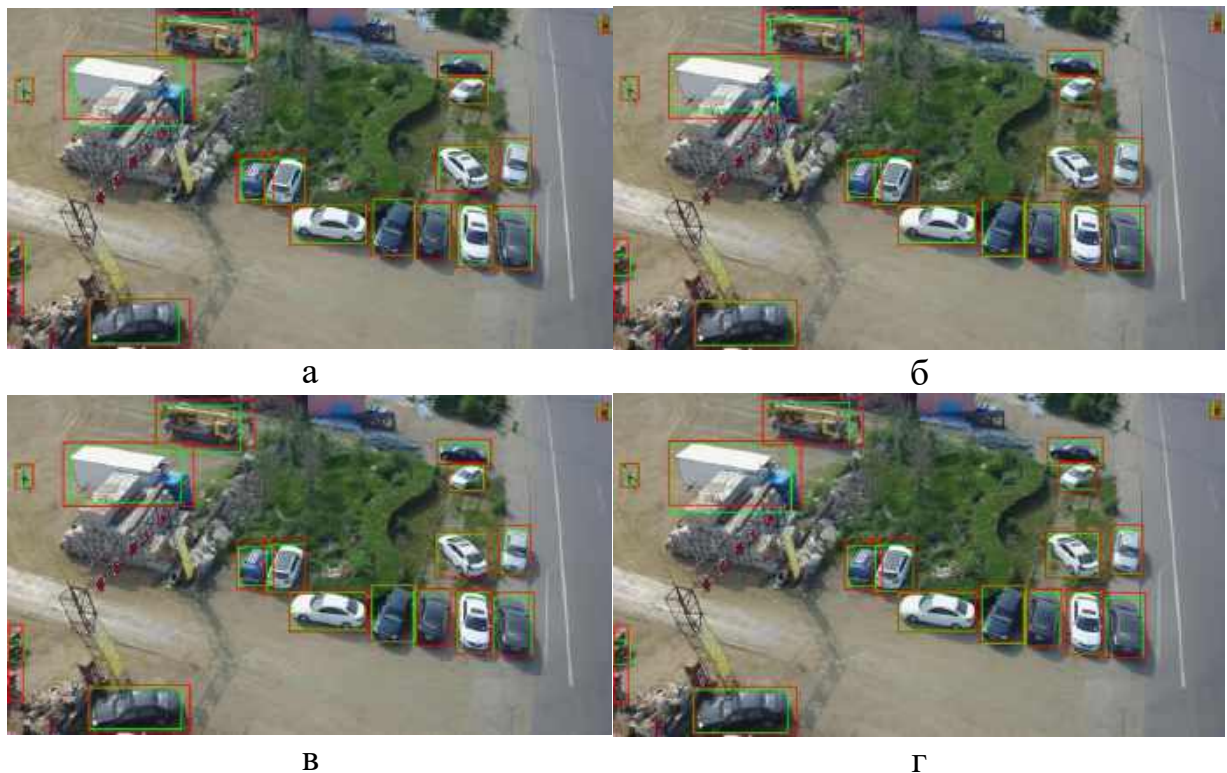


Рисунок 2.23 – Результат прогнозувань нейронної мережі Faster RCNN з ResNet18 (а), ResNet50 (б), ResNet101 (в), ResNetXt (г)

Представлені результати на рисунку 2.23, візуалізовані для кожної з модифікацій нейронних мереж, що були навчені. На зображеннях червоним кольором зображено прогнозований регіон, зеленим – попередньо анотований.

Аналізуючи отримані результати помітно, що всі модифікації нейронної мережі досить добре локалізують об'єкти, навіть невеликого розміру. Всі об'єкти що визначені в анотованих даних є в прогнозованій вибірці, результати які виражено в числових даних представлено в таблиці 2.1.

Аналізуючи результати, що прогнозовані за допомогою нейронної мережі RetinaNet, які представлено на рисунку 2.24, помітно що більшість результатів досить непоганої якості. Основна частина регіонів має хороший показник перекриття, але в нейронної мережі невеликого розміру, з бекбоном ResNet18, маємо досить великі регіони, які захоплюють не тільки об'єкт. Очікувано, що метрика локалізації для цієї нейронної мережі матиме низький результат. Результати метрик для нейронних мереж з архітектурою RetinaNet також представлено в таблиці 2.1.

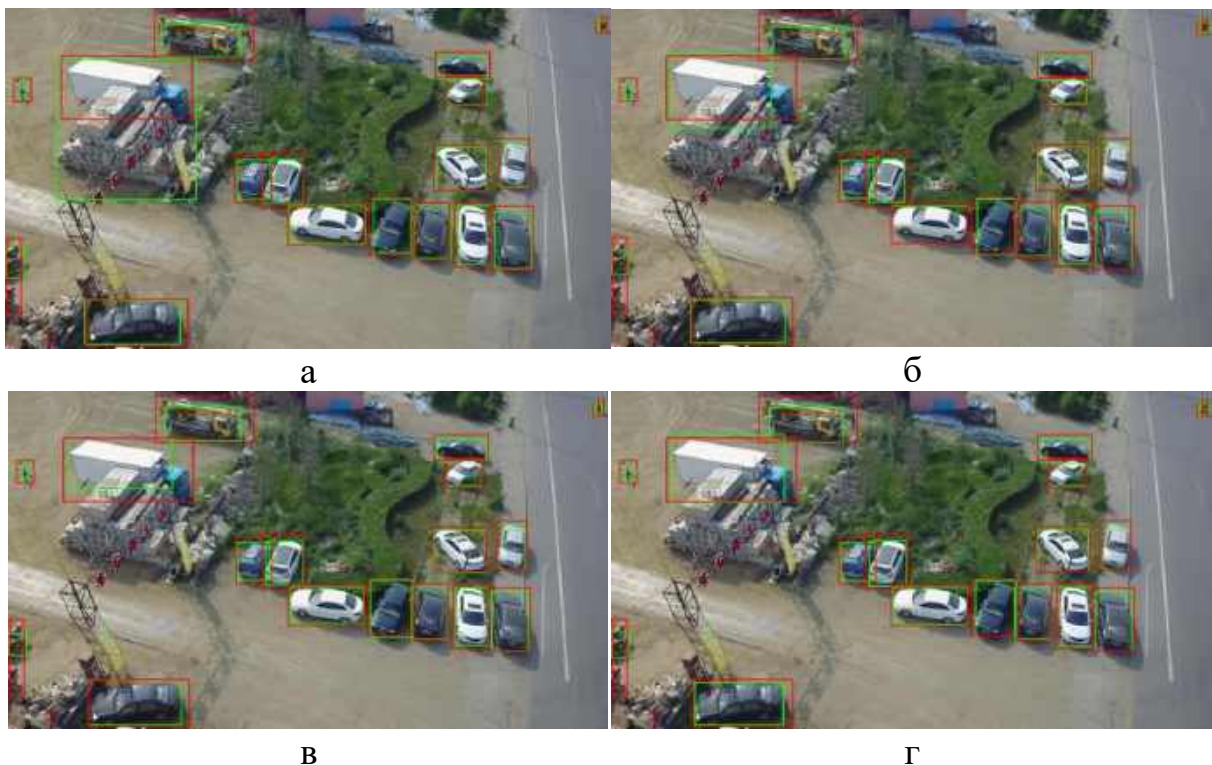


Рисунок 2.24 – Результат прогнозувань нейронної мережі RetinaNet з ResNet18 (а), ResNet50 (б), ResNet101 (в), ResNetXt (г)

Проведемо аналіз нейронних мереж з архітектурою SSD. Алгоритм оцінки точності локалізації відповідає попереднім моделям. Результат прогнозувань

представлено на рисунку 2.25, який включає в себе дві модифікації архітектури RetinaNet, з MobileNetV2 та VGG енкодерами. Загалом можна зробити висновок, що модель досить непогано локалізує об'єкти та має високий рівень точності прогнозувань. Більшість об'єктів на тестовому зображенні локалізована з високою точністю; метрики якості, що відображають більш об'єктивну картину представлені для порівняння в таблиці 2.1.

Аналізуючи результати для нейронної мережі YOLO 5-ї серії, які представлено на рисунку 2.26, помітно, що маємо падіння точності класифікації та кількості прогнозованих об'єктів. В більшості випадків нейронна мережа досить добре прогнозує об'єкти різного розміру, але кількість прогнозованих регіонів є меншою за попередньо розглянуті нейронні мережі. Значення метрик представлені в таблиці 2.1.



Рисунок 2.25 – Результат прогнозувань нейронної мережі SSD з VGG(a), MobileNetV2 (б)

Остання з аналізованих нейронних мереж – це YOLO 8-ї серії, результати якої представлені на рисунку 2.27. В більшості випадків результати прогнозування схожі з прогнозуваннями для 5-ї серії, але мають більше прогнозованих результатів а також, візуально, кращі показники точності локалізації. Нейронна мереж локалізує як об'єкти великого розміру, так і малорозмірні об'єкти. Значення метрик для нейронної мережі представлені також в таблиці 2.1.

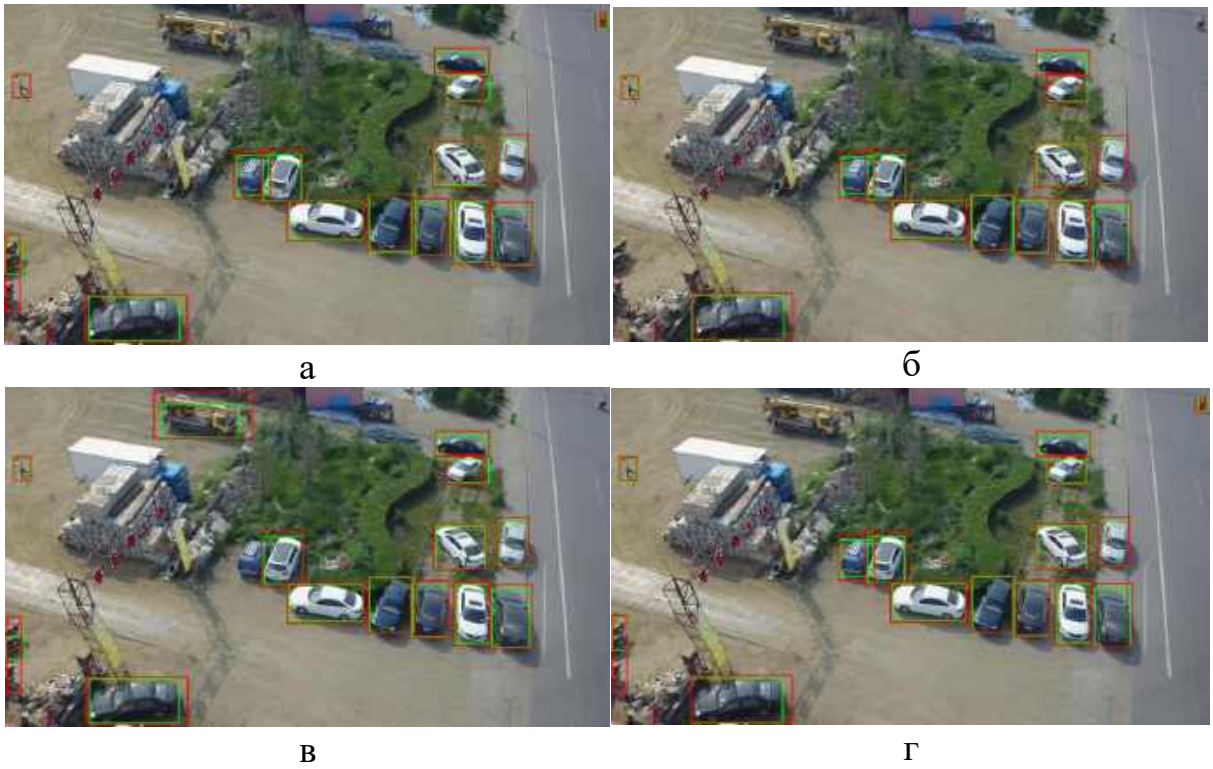


Рисунок 2.26 – Результат прогнозувань для нейронних мереж YOLOv5n (а), YOLOv5s (б), YOLOv5m (в), YOLOv5l (г)

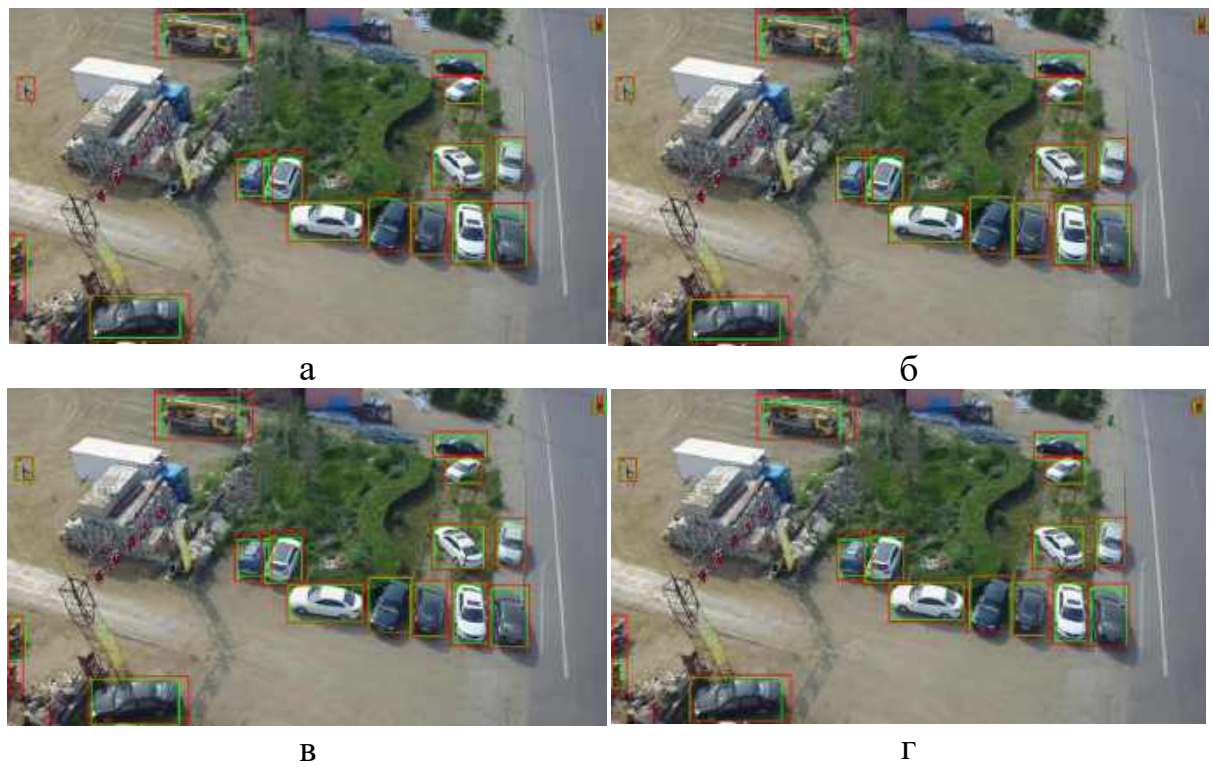
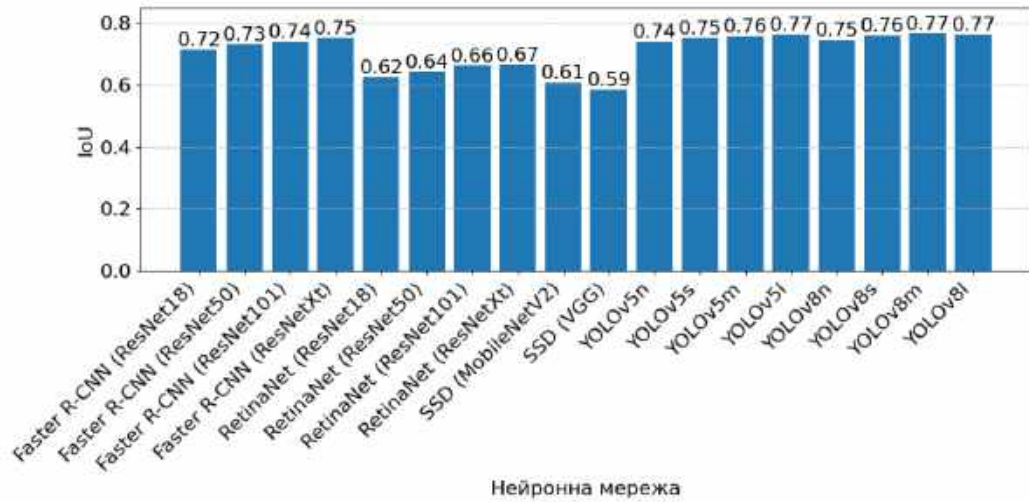


Рисунок 2.27 – Результат прогнозувань нейронних мереж YOLOv8n (а), YOLOv8s (б), YOLOv8m (в), YOLOv8l (г)

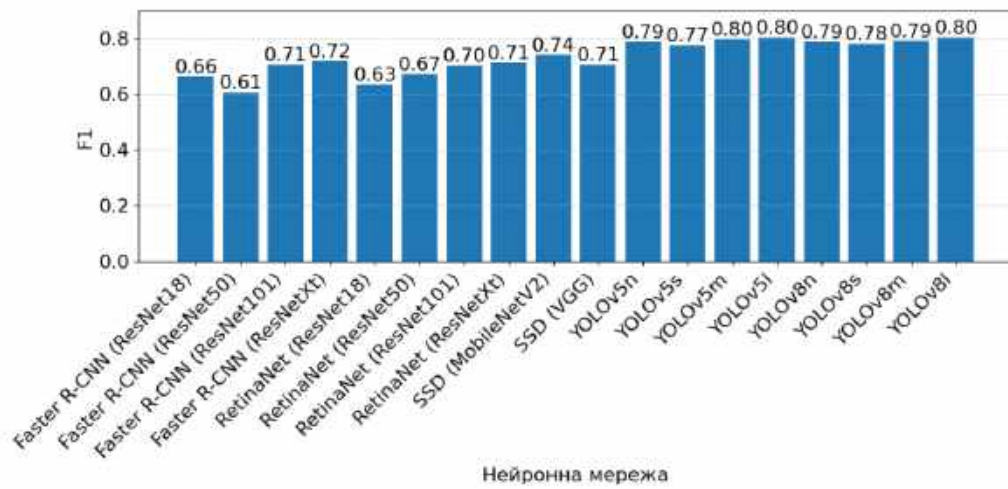
Дані, які визначають точність локалізації та класифікації представлені в таблиці 2.2, а також на стопчикових діаграмах на рисунку 2.29а для метрики якості локалізації, на 2.29б для метрики якості класифікації та на рисунку 2.29в для кількості вірно прогнозованих регіонів. Аналізуючи дані в таблиці 2.2 помітно що найменший відсоток вірно прогнозованих регіонів мають нейронні мережі YOLO з модифікацією nano. Результат є очікуваним, адже нейронна мережа позиціонується як швидке рішення з достатньою точністю. Повертаючись до візуалізованих результатів також бачимо перетин характеристик. Загалом оцінюючи відсоток вірно прогнозованих регіонів бачимо, що найкращий результат дають RetinaNet та SSD, а нейронні мережі YOLO мають найменші значення метрики.

Аналізуючи інші метрики, насамперед точність локалізації (IoU) помітна зворотна тенденція до кількості прогнозованих регіонів. Таким чином нейронна мережа YOLO має найвищі показники точності локалізації, несуттєво їй поступається нейронна мережа Faster RCNN незалежно від модифікацій. Найменший показник має нейронна мережа SSD. З метрикою F1 ситуація схожа. Найкращий показник має нейронна мережа YOLO, особливо найбільших модифікацій (large), наступна FasterRCNN, яка є не критично гіршою.

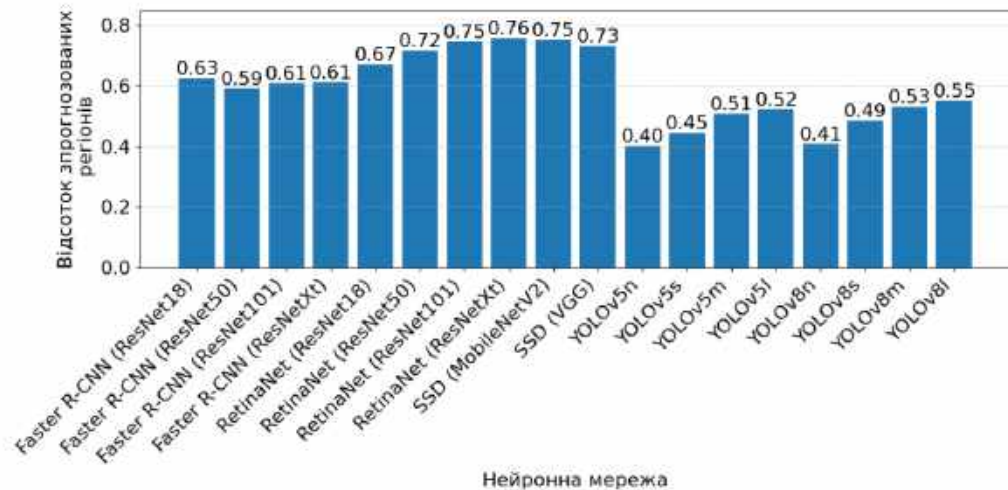
Таким чином проаналізувавши отримані результати маємо показники, які дозволяють зробити вибір потрібної нейронної мережі, за потреби. Якщо система повинна давати велику кількість вірно прогнозованих регіонів то кращим вибором буде нейронна мережа SSD, але в такому разі ми нехтуємо точністю локалізації та класифікації. Якщо ж нам критична точність локалізації то відмінним вибором буде архітектура YOLO, модифікацію якої можна обрати в залежності від розрахункової здатності пристрою. Гарним вибором за всіма критеріями є архітектура Faster RCNN, особливо з ResNet101, вона має досить високий відсоток вірно прогнозованих регіонів а також досить високе значення метрик локалізації та класифікації.



а



б



в

Рисунок 2.29 – Стопчикові діаграми метрик якості локалізації (а), класифікації (б) та кількості прогнозованих регіонів (в)

Таблиця 2.2 – Метрики якості прогнозувань нейронних мереж

Назва нейронної мережі	IoU	F1	Відсоток прогнозованих регіонів
Faster R-CNN (ResNet18)	0,716	0,662	0,627
Faster R-CNN (ResNet50)	0,733	0,607	0,592
Faster R-CNN (ResNet101)	0,740	0,707	0,609
Faster R-CNN (ResNetXt)	0,753	0,720	0,612
RetinaNet (ResNet18)	0,625	0,633	0,672
RetinaNet (ResNet50)	0,642	0,672	0,717
RetinaNet (ResNet101)	0,662	0,702	0,747
RetinaNet (ResNetXt)	0,665	0,712	0,759
SSD (MobileNetV2)	0,609	0,740	0,752
SSD (VGG16)	0,586	0,707	0,732
YOLOv5n	0,740	0,786	0,401
YOLOv5s	0,753	0,774	0,446
YOLOv5m	0,757	0,796	0,507
YOLOv5l	0,765	0,803	0,523
YOLOv8n	0,746	0,789	0,408
YOLOv8s	0,760	0,781	0,486
YOLOv8m	0,766	0,792	0,531
YOLOv8l	0,765	0,803	0,552

2.3.2 Оцінка розрахункової складності та кількості параметрів

Для досліджуваних нейронних мереж також було проведено оцінку розміру моделей. Ця характеристика є важливим показником нейронної мережі. Адже при виборі нейронної мережі потрібно враховувати також вартість запуску цієї нейронної мережі.

В більшості випадків для вибору оптимальної нейронної мережі за співвідношенням точності роботи до швидкості використовують параметр

FLOPS (FLoating-point OPerations per Second), що відображає кількість операцій з плаваючою комою за секунду. Фактично цим параметром відображають мінімальну розрахункову здатність пристрою, що використовується для запуску нейронної мережі. Параметр є мінімальним для пристрою який може обробляти 1 кадр (1 зображення) за 1 секунду, але для використання нейронної мережі в реальному часі (не менше 24 кадрів за секунду) потрібно враховувати фактор кількості кадрів, перемноживши його на розрахункову важкість обробки одного кадру. Отримана величина є мінімально допустимою для запуску нейронної мережі з потрібними параметрами, але краще використовувати пристрої з більшою розрахунковою здатністю та досліджувати їх на предмет тротлінгу [100] з плином часу використання.

Також не менш важливим є показник кількості параметрів в нейронній мережі, він відображає кількість елементів у вагових коефіцієнтах моделі. Параметр є залежним до параметру FLOPS, та за допомогою нього можна розрахувати мінімально необхідну кількість оперативної пам'яті для запуску нейронної мережі. В таблиці 2.3 представлено результати FLOPS та кількості параметрів для всіх досліджуваних нейронних мереж.

Отримані характеристики можуть бути використані для вибору найкращих нейронних мереж за співвідношенням якості роботи до використовуваних ресурсів. Аналізуючи отримані дані можна визначити що найменшу розрахункову навантаження мають нейронні мережі YOLO особливо малих версій, таких як nano та small. Також невеликою є потреба в ресурсах для нейронних мереж, в яких екстрактором векторів ознак є нейронна мережа ResNet18. Найбільшу кількість параметрів та розрахункову навантаження має нейронна мережа Faster RCNN з екстрактором векторів ознак ResNetXt. Але враховуючи отриману раніше інформацію про якість роботи нейронних мереж можна визначити, що більша кількість параметрів в архітектурі нейронної мережі може допомогти краще вирішувати поставлену задачу. Вибір тієї чи іншої нейронної мережі для задач локалізації неможливо базувати тільки на параметрах

розрахункової складності чи точності роботи, це повинно бути комплексне рішення з використанням всіх раніше досліджених параметрів.

Таблиця 2.3 – Параметри нейронних мереж

Назва нейронної мережі	FLOPS, T	Params, M
Faster R-CNN (ResNet18)	0,069	28,331
Faster R-CNN (ResNet50)	0,091	41,399
Faster R-CNN (ResNet101)	0,121	60,392
Faster R-CNN (ResNetXt)	0,184	99,298
RetinaNet (ResNet18)	0,063	19,979
RetinaNet (ResNet50)	0,083	36,537
RetinaNet (ResNet101)	0,114	55,529
RetinaNet (ResNetXt)	0,176	94,436
SSD (MobileNetV2)	0,0027	3,18
SSD (VGG16)	0,139	25,861
YOLOv5n	0,007	2,510
YOLOv5s	0,024	9,126
YOLOv5m	0,064	25,070
YOLOv5l	0,135	53,171
YOLOv8n	0,008	3,012
YOLOv8s	0,028	11,139
YOLOv8m	0,079	25,862
YOLOv8l	0,165	43,637

2.4 Висновки до розділу

Для оцінки точності роботи нейронних мереж в задача локалізації та класифікації було проведено їх навчання та досліджено точність їх роботи. Враховуючи отримані дані можна зробити деякі висновки.

По-перше, більшість з досліджуваних нейронних мереж донавчаються досить швидко та дозволяють швидко коригувати якість їх роботи, змінюючи параметри навчання та перезапускаючи процес донавчання. Досліджуючи процес навчання було визначено, що використання різного типу планувальників параметру швидкості навчання (learning rate) дає позитивний результат щодо швидкості навчання. В багатьох випадках його використання дає змогу вийти оптимізатору з локального мінімуму в процесі навчання. Також в процесі навчання було визначено, що всі досліджувані нейронні мережі вийшли на плато. Це показує, що використання нейронних мереж для дослідження є доцільним, адже їх точність є задовільною для заданих умов.

Другий висновок стосується якості роботи нейронних мереж. Фаворитом за кількістю прогнозованих регіонів є RetinaNet. Але YOLO показує кращі результати для точності локалізації та класифікації, її регіони є більш схожими до анотованих, але поступається в параметрі кількості прогнозованих регіонів. Інші досліджувані нейронні мережі хоч і мають дещо нижчі значення метрик, все ж показують досить високі результати, а тому також є актуальними до використання в подальших дослідженнях.

Третій висновок стосується обчислювальної складності нейронних мереж, а також кількості параметрів. Цей фактор є досить важливим, особливо в задачах де потужність пристроїв обмежена, а відповідно БПЛА є пристроями з обмеженою обчислювальною здатністю. Також використання занадто потужних плат для розрахунків на БПЛА значно збільшує їх вагу, а відповідно й інші характеристики. Найменшу обчислювальну складність мають нейронні мережі YOLO, особливо малого розміру, а також SSD з MobileNetV2. Найбільшу розрахункову складність мають нейронні мережі Faster R-CNN, але вони показують кращий результат в кількості локалізованих об'єктів.

Враховуючи отримані дані, можна стверджувати, що кожна з нейронних мереж має свої плюси та мінуси, але в загальному кожна з них є досить точною в задачах локалізації та класифікації. Так, деякі архітектури дозволяють краще локалізувати об'єкти, але поступаються в кількості локалізованих регіонів. Деякі

нейронні мережі, хоч і мають посередні показники точності локалізації та класифікації, але дозволяють проводити розрахунки на пристроях з обмеженою потужністю.

РОЗДІЛ 3.

ОЦІНКА РОБОТИ НЕЙРОННИХ МЕРЕЖ З ОБ'ЄКТАМИ МАЛОГО РОЗМІРУ

В задачах локалізації та класифікації об'єктів на зображеннях, що отримані за допомогою БПЛА, важливим є показник точності роботи нейронної мережі для малорозмірних об'єктів (наприклад, людей чи мін). В багатьох випадках зйомка проводиться з досить великої висоти для отримання більшої кількості даних за один фрейм. Також через особливість (наприклад, нерівності) рельєфу та в залежності від задач досліджень висота зйомки може критично відрізнятись. Тому дослідження роботи нейронних мереж на предмет якості локалізації та класифікації малорозмірних об'єктів є важливою задачею.

Розділ присвячено дослідженню точності роботи методів локалізації та класифікації в умовах зйомки з різних висот, а також точності локалізації та класифікації об'єктів різного розміру, зокрема малорозмірних. Для досліджень буде використано попередньо навчені нейронні мережі з розділу 2.

3.1 Огляд набору даних для досліджень

Для дослідження якості роботи нейронної мережі важливим є вибір правильних даних. Дані повинні бути відповідними до предмету досліджень, тобто в даному випадку покривати широкий діапазон висот, а також мати велику кількість зображень. Тренувальний набір даних не може бути використаний для оцінки, адже нейронні мережі мають здатність до запам'ятовування та загалом показують високі результати на даних, за допомогою яких навчались. Така поведінка нейронних мереж може привести до спотворення результатів досліджень. Тестова частина даних з набору VisDrone не підходить під характеристики. Вона має досить малу кількість зображень, що не дає можливість побудувати стійку залежність між розміром об'єкту та точністю

локалізації та класифікації. Також набір даних VisDrone не має визначених висот зйомки, хоч і має досить широкий діапазон розмірів об'єктів. Враховуючи ці фактори необхідним є вибір іншого набору даних для проведення досліджень.

Набором даних, який може покривати потреби, що зазначені раніше, а саме визначені висоти зйомки, велика кількість зображень та різноманітність висот зйомки, є набір AU-AIR [101]. Він має визначені характеристики висоти зйомки, що дає змогу будувати залежності точності локалізації від висоти зйомки. Відповідно діапазон зйомки є досить широким, з максимальним значенням 30,625 метрів, що є допустимою висотою для досліджень. Розподіл висот в наборі даних представлено на рисунку 3.1. Помітно декілька глобальних викидів на висотах близьких до 12 метрів, 21 метру та два викиди в значеннях 28 та 29 метрів. На основі цих значень можна зробити висновок, що за характеристиками розподілу набір даних покриває необхідні параметри.

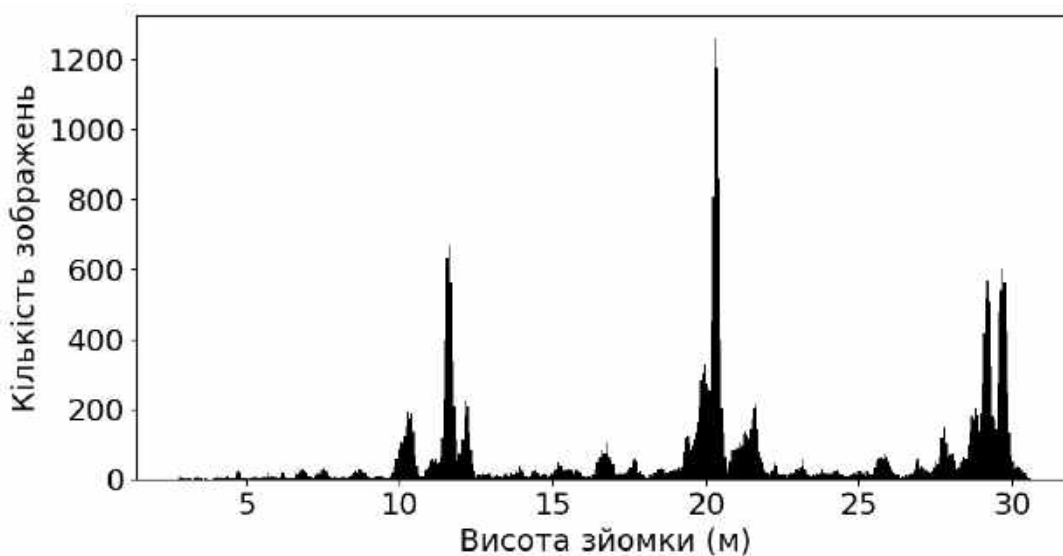


Рисунок 3.1 – Розподіл висот зйомки в наборі даних AU-AIR

Наступною кроком є визначення характеристик анотованих об'єктів. Загалом структура класів є схожою з тренувальним набором даних, а саме список категорій є таким: людина, авто, вантажівка, фургон, мотоцикл, велосипед, автобус, трейлер. Більшість категорій співпадають з тренувальним набором даних і це є ще одним показником до використання даного набору даних в якості

основного для досліджень. Розподіл класів в наборі даних представлено на рисунку 3.2. З розподілу помітно, що більшість регіонів зосереджені в класі «Авто», а також в класах «Фургон» та «Вантажівка», що дає можливість дослідити точність локалізації техніки різного розміру. Також досить велика кількість даних зосереджена в класі «Людина», цей клас також важливий для досліджень, адже саме людину важко визначити на зображеннях з БПЛА. Загалом розподіл класів є відповідним до задачі та дозволяє використовувати набір даних для подальших досліджень.

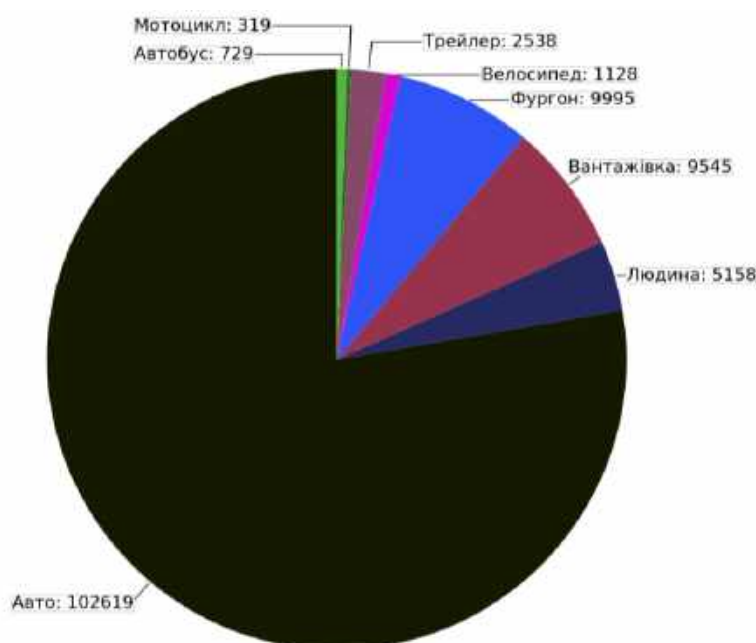


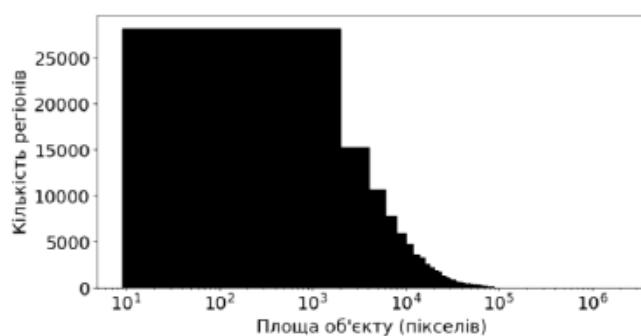
Рисунок 3.2 – Розподіл класів в наборі даних AU-AIR

Також важливим етапом дослідження набору даних для нашої задачі є визначення мінімального та максимального розміру об'єкту для кожного з класів, а також побудова розподілу кількості регіонів різного розміру. Для цього першим етапом визначено розмір об'єктів для кожного класу та визначено граничні значення (мінімальну та максимальну площу регіону), а також середнє значення площі регіону, отримані дані занесено в таблицю 3.1. Також побудовано розподіл площ регіонів, в логарифмічному масштабі враховуючи великий розкид даних, для важливих класів, а саме для класів «Авто», «Людина», «Вантажівка» та «Фургон», який представлено на рисунку 3.3. Враховуючи отримані значення

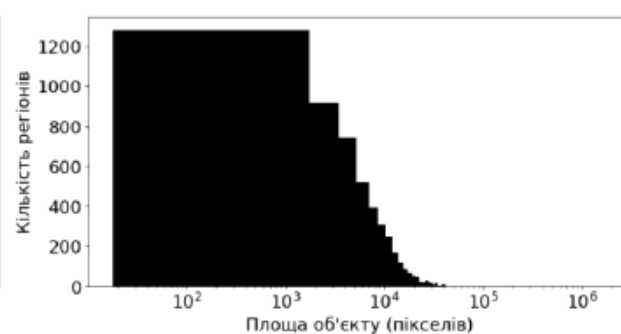
можна зробити висновок, що в наборі даних площі об'єктів є досить широкими, що помітно з графіків, але більшість об'єктів агреговані в зоні досить малих розмірів, окрім класу фургонів, що загалом відповідає умовам досліджень.

Таблиця 3.1 – Граничні та середні значення площ регіонів за класами

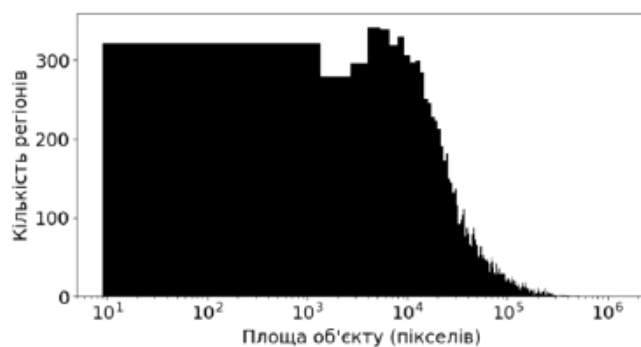
Клас	Мінімальна площа, пікселів	Максимальна площа, пікселів	Середня площа, пікселів
Авто	9	2046720	13007,94
Людина	18	1720512	7761,87
Вантажівка	9	1338750	46561,6
Фургон	9	504458	20695,7
Велосипед	15	593640	7659,13
Трейлер	9	1361723	41174,84
Мотоцикл	9	185760	8924,15
Автобус	120	676476	47124,11



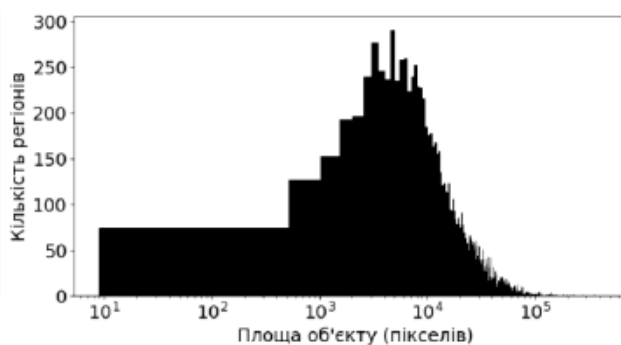
а



б



в



г

Рисунок 3.3 – Розподіл площ регіонів для класів «Авто» (а), «Людина» (б), «Вантажівка» (в) та «Фургон» (г)

Також, досліджуючи набір даних, важливо розуміти висоти зйомки для кожного з класів. Ця інформація слугує для розуміння кількості малорозмірних об'єктів в кожному класі, а також відображає об'єктивність отриманих даних в подальших дослідженнях. На рисунку 3.4 представлено розподіл висот зйомки для кожного з чотирьох досліджуваних класів. Отримані графіки розподілу показують, що висоти зйомки регіонів розподілені близько до загального розподілу набору даних. З цього випливають два основних висновки. По-перше, розподіл регіонів за класами досить широкий на одному зображенні, що робить результати більш стійкими. Другий висновок - це можливість застосування даного набору даних для проведення досліджень, адже він відповідає всім необхідним характеристикам.

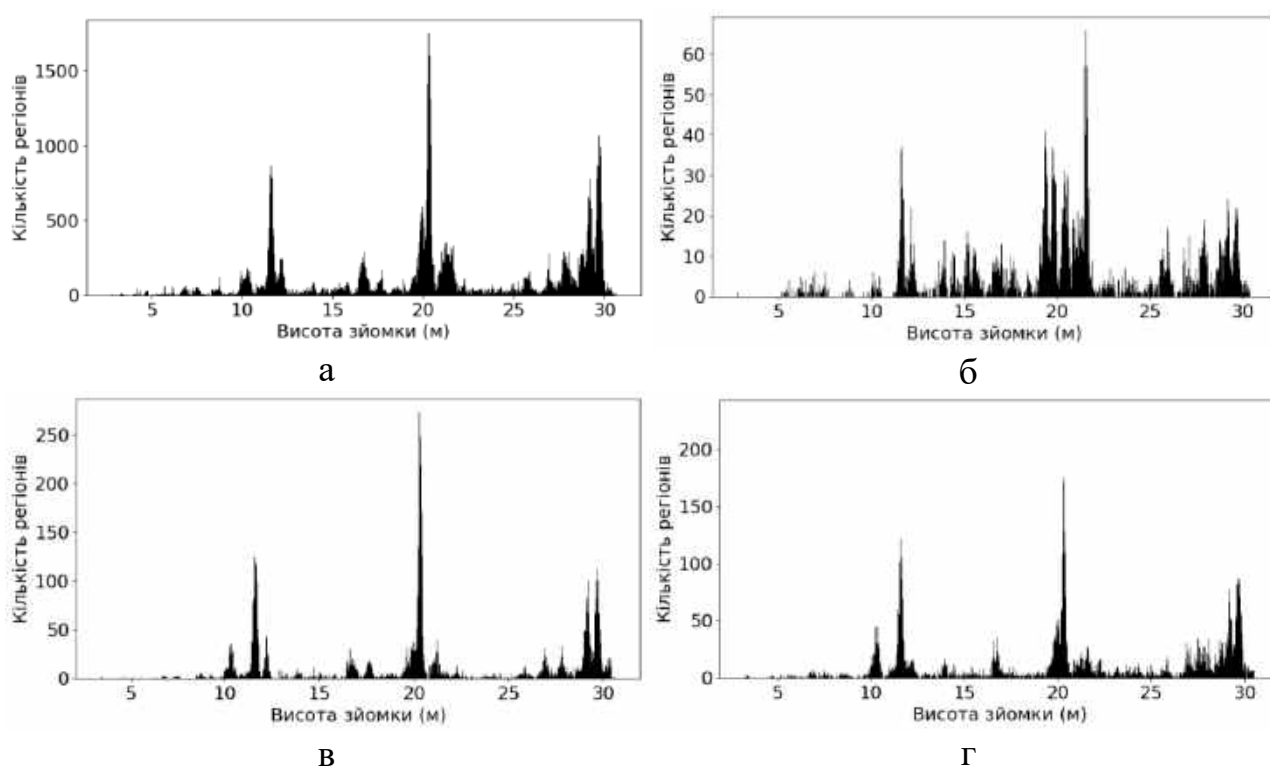


Рисунок 3.4 – Розподіл висот зйомки для класів «Авто» (а), «Людина» (б), «Вантажівка» (в) та «Фургон» (г)

3.2 Дослідження точності роботи з малорозмірними об'єктам

Як було сказано раніше, об'єкти малого розміру представляють особливу цікавість на зображеннях, що отримані за допомогою безпілотних літальних

апаратів. Цікавість до таких об'єктів виражається в можливості їх розпізнавання та дослідження в умовах, коли об'єкт займає декілька сотень пікселів, адже висоти зйомки з БПЛА можуть бути досить великими. Висота зйомки може варіюватись в залежності від потрібної задачі. Так, в цивільній зйомці з БПЛА можуть бути досить різні висоти, навіть досить мала дистанція до землі, що становить кілька метрів [102, 103]. В задачах військового призначення, в більшості випадків, висота зйомки є зазвичай досить суттєвою для забезпечення безпечності літального апарату [104, 105]. Так як дослідження більше присвячені цивільній зйомці з БПЛА, буде розглянуто широкий діапазон висот, де максимальна висота зйомки становить 30 метрів, мінімальна 5 метрів.

Дослідження якості роботи нейронних мереж, що були навчені в розділі 2, буде проходити в декілька етапів. Першим етапом буде досліджено точність роботи нейронних мереж, яка буде оцінена за значеннями метрик для різного розміру площі регіонів. До цих метрик будуть входити IoU [49], яка показує якість локалізації регіонів, а також метрика якості класифікації F1 [67] та відсоток вірно прогнозованих регіонів. Значення будуть визначені для декількох розмірів регіонів, список яких є наступним:

- до 100 пікселів - регіони, розмір яких є критично малим;
- від 101 до 200 пікселів – регіони, що мають допустиму площу;
- від 201 до 500 пікселів – регіони з достатньою площею;
- більше 500 пікселів – регіони звичайного розміру.

Представлені розміри регіонів визначені за допомогою перерахунку в розмір, що буде передано до нейронної мережі, щоб отримати більш стійку статистику. Варто зауважити, що різні нейронні мережі мають дещо різний вхідний розмір зображення, але враховуючи те, що навчання проходило з такими ж характеристиками розмірностей, цим параметром фактично можна знехтувати. Також ще один фактор можливості неврахування значень зміни вхідного розміру полягає в незначній різниці між входами в різних нейронних мереж.

Почнемо із загальної оцінки якості роботи нейронної мережі на тестовому наборі даних. Для цього проведемо звичайну оцінку метрик якості роботи

нейронної мережі, які представлено на рисунку 3.5 для метрики відсотку прогнозованих регіонів, на рисунку 3.6 - для метрики IoU та на рисунку 3.7 для метрики F1. Враховуючи отримані дані, можна зробити висновок, що більшість нейронних мереж мають досить високі значення цих метрик. Більшість нейронних мереж мають відсоток кількості прогнозованих регіонів більше за 0,75, що є досить хорошим показником. Також показник точності локалізації IoU має задовільні значення, які є дещо меншими, ніж на наборі даних VisDrone. Показник точності класифікації F1 для більшості нейронних мереж має досить високі значення та дозволяє добре класифікувати більшість регіонів.



Рисунок 3.5 – Стопчикова діаграма метрики відсотку прогнозованих регіонів

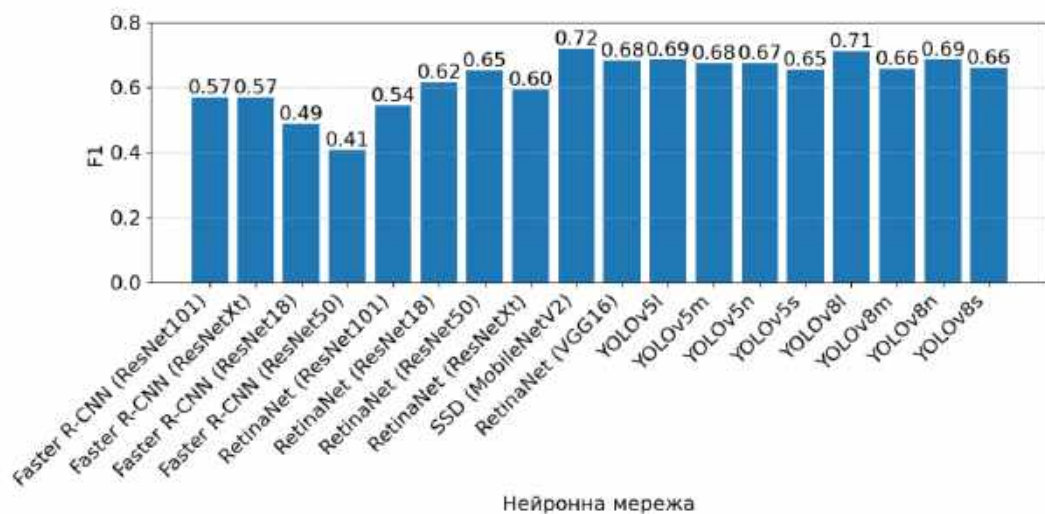


Рисунок 3.6 – Стопчикова діаграма метрики F1

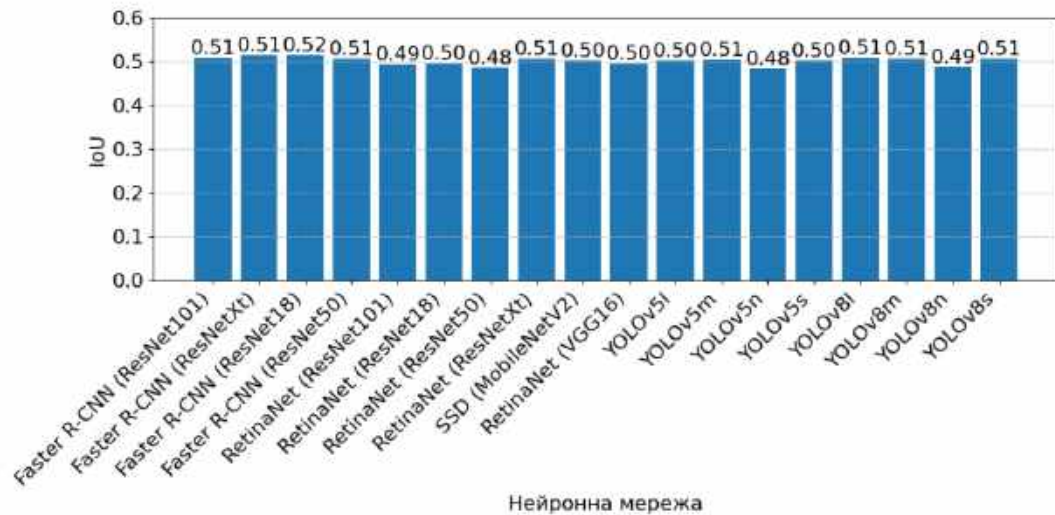
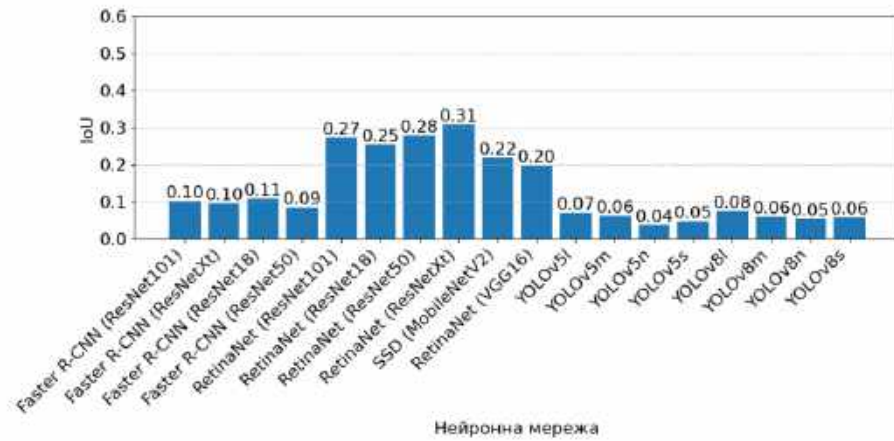


Рисунок 3.7 – Стопчикова діаграма метрики IoU

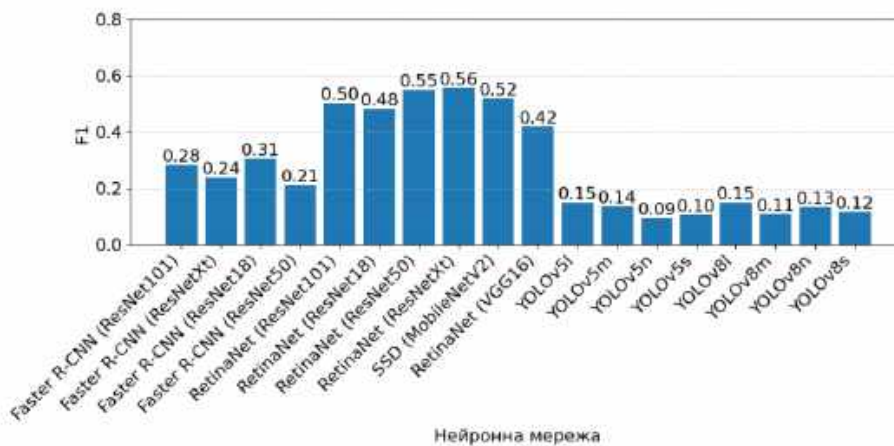
Проведемо дослідження для регіонів площею до 100 пікселів, для розміру зображення в 1920x1080 пікселів ця площа є досить критичною, що становить менше одного відсотка, а саме 0.04% від загальної площі зображення. Якщо враховувати розмір зображення, що передається на вхід нейронної мережі, то варто зауважити, що площа регіону буде мати менше значення. Такі регіони досить важко локалізувати та безпомилково класифікувати, але для об'єктивності дослідження такі дані потрібно досліджувати в випадку з зображеннями з БПЛА. Результати, що отримані при класифікації таких регіонів представлені в таблиці 3.2. На рисунку 3.8 представлено стопчикову діаграму метрик точності локалізації та класифікації для регіонів розміром до 100 пікселів.

З аналізу отриманих даних видно, що частина нейронних мереж досить добре локалізує та класифікує об'єкти навіть малого розміру, в даному випадку площею до 100 пікселів на вході нейронної мережі. Не зважаючи на те, що нейронні мережі SSD мають найменший вхідний розмір об'єктів, вони показують досить хороший результат кількості локалізованих об'єктів. Також метрики точності локалізації та класифікації для нейронних мереж цього типу мають досить високе значення, тому можна вважати що нейронна мережа SSD може локалізувати об'єкти малого розміру. Також хороший результат показують нейронні мережі RetinaNet та Faster RCNN, кращою з них є RetinaNet. Це пов'язано з

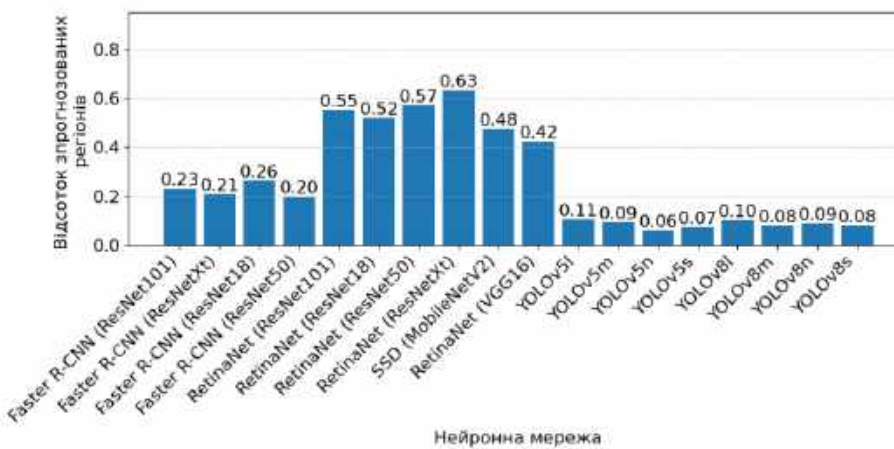
використанням архітектури, що має декілька виходів, які мають різний розмір ознак. В свою чергу така архітектура дозволяє локалізувати об'єкти різного розміру, що підтверджується дослідженням на об'єктах малого розміру.



а



б



в

Рисунок 3.8 – Стопчикові діаграми метрик якості локалізації (а), класифікації (б) та кількості прогнозованих регіонів для регіонів розміром до 100 пікселів

Таблиця 3.2 – Результати прогнозувань для регіонів з площею до 100 пікселів

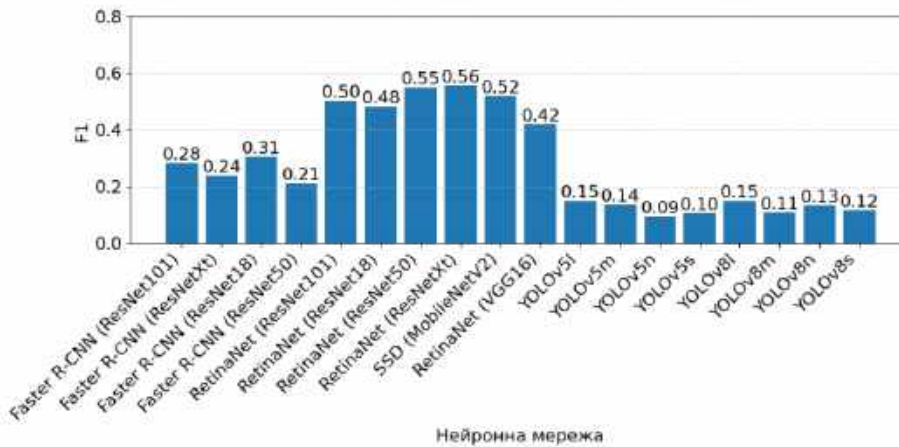
Назва нейронної мережі	F1	IoU	Відсоток прогнозованих регіонів
Faster R-CNN (ResNet18)	0,305	0,109	0,262
Faster R-CNN (ResNet50)	0,211	0,085	0,200
Faster R-CNN (ResNet101)	0,284	0,103	0,232
Faster R-CNN (ResNetXt)	0,239	0,095	0,209
RetinaNet (ResNet18)	0,485	0,254	0,521
RetinaNet (ResNet50)	0,549	0,279	0,571
RetinaNet (ResNet101)	0,504	0,273	0,553
RetinaNet (ResNetXt)	0,557	0,309	0,633
SSD (MobileNetV2)	0,519	0,219	0,476
SSD (VGG16)	0,419	0,198	0,424
YOLOv5n	0,095	0,037	0,062
YOLOv5s	0,105	0,048	0,073
YOLOv5m	0,135	0,063	0,093
YOLOv5l	0,152	0,072	0,105
YOLOv8n	0,134	0,054	0,090
YOLOv8s	0,118	0,059	0,081
YOLOv8m	0,109	0,061	0,082
YOLOv8l	0,150	0,075	0,104

Наступним кроком проведемо дослідження точності локалізації на об'єктах більшого розміру, а саме площею від 101 до 200 пікселів в перерахунку на вхідне зображення. Отримані дані відображені в таблиці 3.3 для всіх досліджуваних нейронних мереж. Також результати представлено у вигляді стопчикових діаграм на рисунку 3.9. Загалом результат досить схожий з локалізацією об'єктів меншого розміру, спостерігається такий же розподіл нейронних мереж за точністю. Але при цьому всі досліджувані нейронні мережі мають кращі показники точності роботи порівняно з дослідженням на об'єктах площею до 100 пікселів. Також помітно, що нейронні мережі з архітектурою RetinaNet показують досить хороший результат кількості прогнозованих блоків, що досягає 91%. Це відображається також і в параметрі точності локалізації, який має досить високі показники для цієї нейронної мережі. Точність локалізації все ж має досить низькі значення. Це пов'язано з неможливістю точно локалізувати регіони з

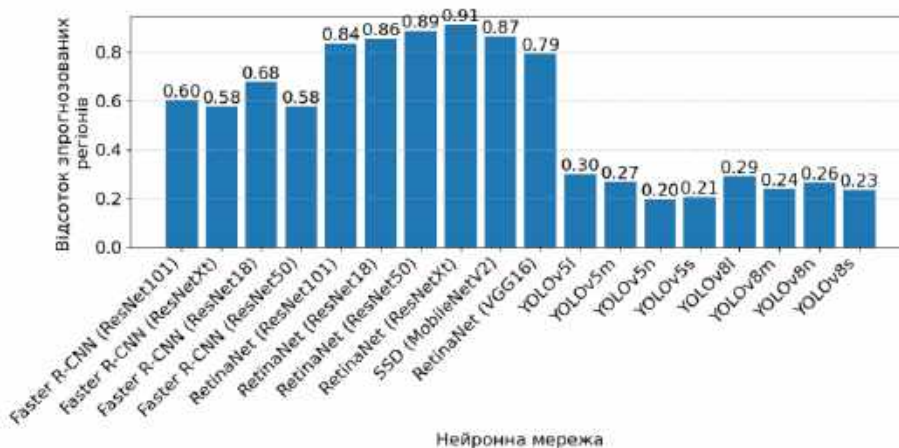
досить малою площею, адже кількість пікселів є малою та навіть невелике зміщення границі вносить велику помилку.



а



б



в

Рисунок 3.9 – Стопчикові діаграми метрик якості локалізації (а), класифікації (б) та кількості прогнозованих регіонів для регіонів розміром від 101 до 200 пікселів

Таблиця 3.3 – Результати прогнозувань для регіонів з площею від 101 до 200 пікселів

Назва нейронної мережі	F1	IoU	Відсоток прогнозованих регіонів
Faster R-CNN (ResNet18)	0,568	0,361	0,678
Faster R-CNN (ResNet50)	0,491	0,323	0,578
Faster R-CNN (ResNet101)	0,575	0,352	0,604
Faster R-CNN (ResNetXt)	0,532	0,345	0,579
RetinaNet (ResNet18)	0,753	0,440	0,856
RetinaNet (ResNet50)	0,797	0,461	0,888
RetinaNet (ResNet101)	0,716	0,441	0,837
RetinaNet (ResNetXt)	0,753	0,492	0,912
SSD (MobileNetV2)	0,798	0,396	0,865
SSD (VGG16)	0,714	0,375	0,794
YOLOv5n	0,287	0,138	0,198
YOLOv5s	0,279	0,154	0,208
YOLOv5m	0,346	0,191	0,268
YOLOv5l	0,373	0,210	0,299
YOLOv8n	0,355	0,184	0,264
YOLOv8s	0,306	0,172	0,235
YOLOv8m	0,282	0,178	0,239
YOLOv8l	0,366	0,212	0,291

Розглянемо також локалізацію та класифікацію регіонів, що мають розмір від 201 до 500 пікселів в перерахунку на вхідний розмір нейронної мережі. Судячи з попередніх досліджень такі регіони повинні бути локалізовані з досить високою точністю, особливо за допомогою RetinaNet та SSD. Результати, що були отримані в процесі досліджень представлені в таблиці 3.4, а також на рисунку 3.10.

При площі регіону об'єкту від 201 до 500 пікселів більшість нейронних мереж мають досить хороші показники якості локалізації та класифікації. Нейронні мережі RetinaNet, яка прогнозує найбільше регіонів, та Faster RCNN, а також SSD показують хороший результат. Вони прогнозують високий відсоток регіонів, який досягає $\approx 90\%$. Це показує, що нейронна мережа покриває вимоги задачі для локалізації такого типу об'єктів. Також нейронні мережі YOLO в такому діапазоні площ регіонів мають результат, що є вищим за попередні дослідження.

З цього можна зробити висновок, що нейронні мережі YOLO добре працюють з об'єктами середнього розміру. Отримані результати дозволяють зробити висновок, що при розмірі площі об'єкту від 201 пікселів можна гарантувати досить високу точність локалізації та класифікації за допомогою досліджуваних нейронних мереж.

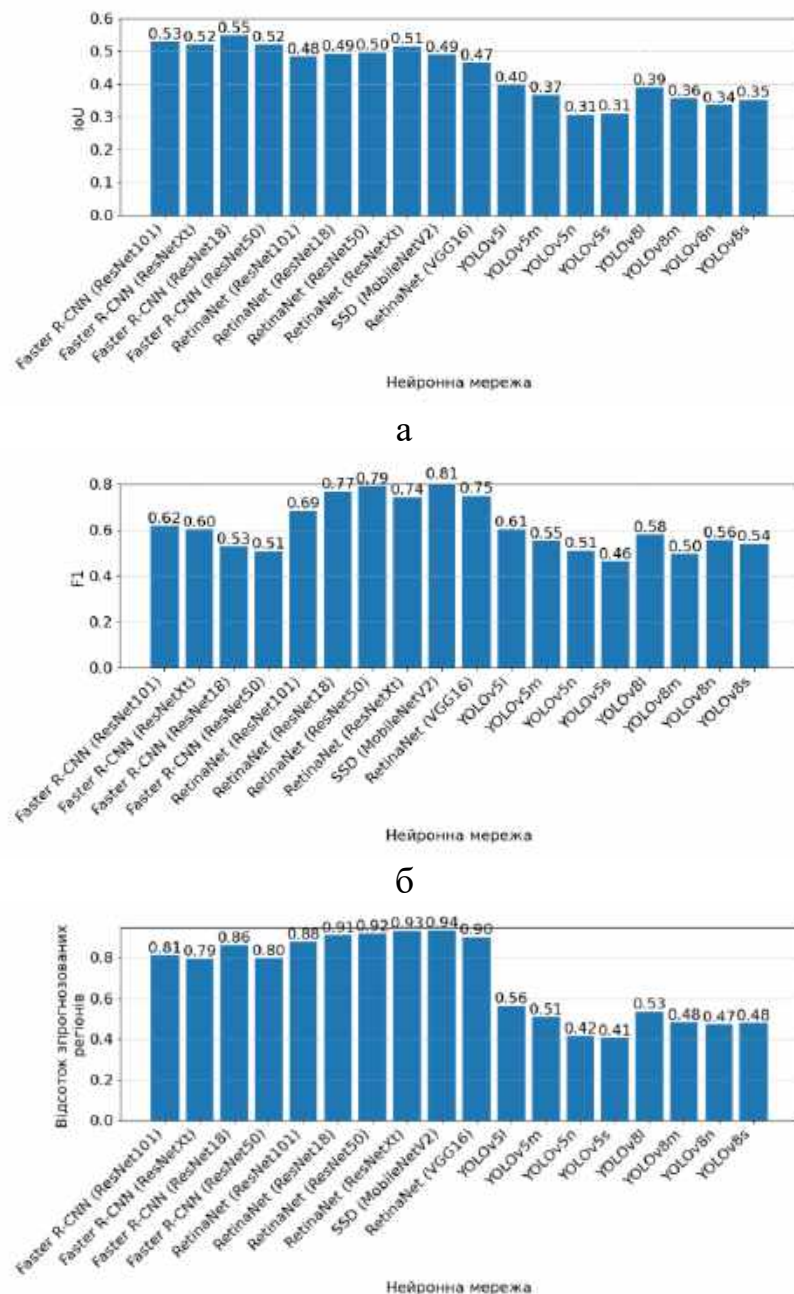


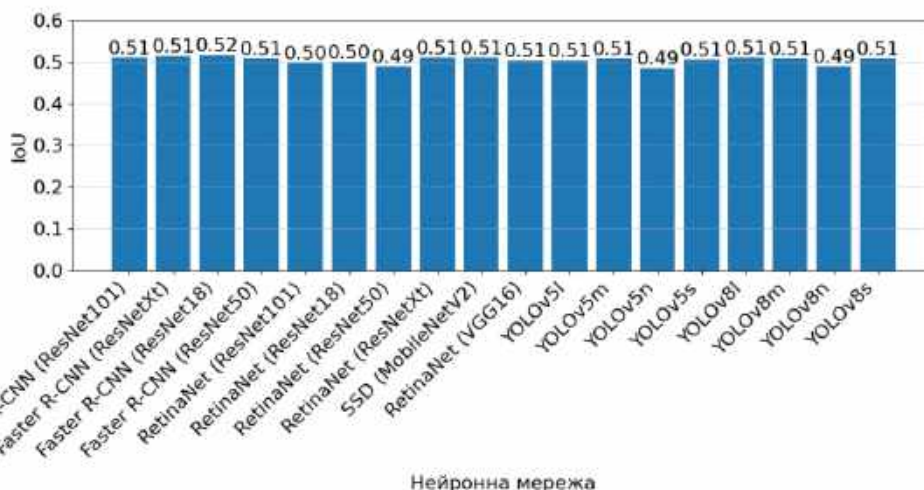
Рисунок 3.10 – Стопчикові діаграми метрик якості локалізації (а), класифікації (б) та кількості прогнозованих регіонів для регіонів розміром від 201 до 500 пікселів

Таблиця 3.4 – Результати прогнозувань для регіонів з площею від 201 до 500 пікселів

Назва нейронної мережі	F1	IoU	Відсоток прогнозованих регіонів
Faster R-CNN (ResNet18)	0,529	0,548	0,862
Faster R-CNN (ResNet50)	0,508	0,522	0,796
Faster R-CNN (ResNet101)	0,619	0,528	0,811
Faster R-CNN (ResNetXt)	0,603	0,521	0,795
RetinaNet (ResNet18)	0,769	0,492	0,913
RetinaNet (ResNet50)	0,793	0,497	0,919
RetinaNet (ResNet101)	0,686	0,482	0,879
RetinaNet (ResNetXt)	0,743	0,514	0,931
SSD (MobileNetV2)	0,815	0,489	0,937
SSD (VGG16)	0,750	0,466	0,901
YOLOv5n	0,510	0,307	0,415
YOLOv5s	0,465	0,310	0,405
YOLOv5m	0,553	0,366	0,508
YOLOv5l	0,605	0,397	0,562
YOLOv8n	0,557	0,336	0,472
YOLOv8s	0,542	0,352	0,479
YOLOv8m	0,495	0,356	0,482
YOLOv8l	0,580	0,389	0,534

Також дослідження якості роботи нейронних мереж було проведено і для регіонів з площею від 500 пікселів. Результат розрахованих метрик представлено в таблиці 3.5 та на рисунку 3.11.

Отримані дані дозволяють зробити висновок, що показники для більшості нейронних мереж метрики зросли порівняно з регіонами від 201 до 500 пікселів. Це може свідчити про те, що дані нейронні мережі добре працюють з об'єктами площею від 500 пікселів, та дозволяють використовувати їх в таких умовах. Має місце збільшення метрик для нейронних мереж YOLO, що відповідає попереднім припущенням, що дані нейронні мережі можуть якісно прогнозувати головним чином регіони середньої та великої площі.



а



б



в

Рисунок 3.11 – Стопчикові діаграми метрик якості локалізації (а), класифікації (б) та кількості прогнозованих регіонів для регіонів розміром понад 500 пікселів

Таблиця 3.5 – Результати прогнозувань для регіонів з площею понад 500 пікселів

Назва нейронної мережі	F1	IoU	Відсоток прогнозованих регіонів
Faster R-CNN (ResNet18)	0,477	0,517	0,837
Faster R-CNN (ResNet50)	0,394	0,508	0,828
Faster R-CNN (ResNet101)	0,560	0,510	0,824
Faster R-CNN (ResNetXt)	0,562	0,515	0,826
RetinaNet (ResNet18)	0,600	0,501	0,828
RetinaNet (ResNet50)	0,638	0,489	0,813
RetinaNet (ResNet101)	0,524	0,498	0,808
RetinaNet (ResNetXt)	0,582	0,510	0,826
SSD (MobileNetV2)	0,712	0,511	0,858
SSD (VGG16)	0,676	0,505	0,841
YOLOv5n	0,664	0,485	0,682
YOLOv5s	0,647	0,506	0,720
YOLOv5m	0,667	0,508	0,744
YOLOv5l	0,679	0,505	0,761
YOLOv8n	0,674	0,489	0,696
YOLOv8s	0,652	0,508	0,740
YOLOv8m	0,655	0,508	0,765
YOLOv8l	0,707	0,511	0,767

Аналізуючи отримані результати, можна підвести підсумки результатів точності роботи нейронних мереж на регіонах різного розміру, зокрема малого. Помітно, що для нейронних мереж Faster RCNN, RetinaNet та SSD метрика кількості прогнозованих регіонів зростає до регіонів розміром від 200 пікселів. Для регіонів, площа яких більше, ніж 200 пікселів, значення метрики є близькими до сталих, що означає гарну здатність даних нейронних мереж до детекції більшої кількості об'єктів на зображеннях. Показники точності нейронних мереж YOLO продовжують рости, а тому можна стверджувати, що для локалізації більшої кількості об'єктів необхідно за допомогою цих нейронних мереж, щоб ці об'єкти були досить великого розміру.

Аналізуючи метрику F1 та співставляючи її з метрикою кількості прогнозованих регіонів, можна помітити, що значення цієї метрики є найвищими для нейронної мережі RetinaNet. Враховуючи також дані з кількості

прогнозованих регіонів, можна стверджувати, що цей тип нейронних мереж дозволяє класифікувати об'єкти з нижчим відсотком помилок ніж інші нейронні мережі. Це може бути пов'язано саме з архітектурою нейронної мережі, адже будова нейронних мереж RetinaNet має кількарозмірну структуру, що виражається в декількох виходах в різних масштабах (scales). Така будова дає змогу розпізнавати регіони навіть на зображеннях низького розрізнення.

Аналізуючи метрику IoU, можна помітити, що для площ регіонів більше за 500 пікселів для всіх нейронних мереж значення майже однакове. Але для регіонів меншого розміру RetinaNet показує краще результати, який для регіонів більше 100 пікселів є близьким до максимального.

З отриманих результатів можна зробити висновок, що нейронна мережа RetinaNet має змогу розрізняти об'єкти навіть невеликого розміру та локалізувати і класифікувати їх з меншою помилкою. Нейронні мережі YOLO можуть надійно прогнозувати тільки об'єкти з площею більше 500 пікселів, а при об'єктах розміром більше 201 пікселів локалізувати та класифікувати їх з достатньою точністю.

Враховуючи отримані дані та обмежену можливість локалізації об'єктів з малим розміром на зображеннях за допомогою нейронних мереж [106, 107], доцільним є подальше проведення досліджень з застосуванням засобів попередньої обробки зображень.

3.3 Метод підвищення точності локалізації та класифікації

Отримані в попередньому підрозділі результати, що стосуються точності локалізації та класифікації показують, що є доцільним використання попередньої обробки зображення для підвищення точності локалізації та класифікації.

Загалом способів підвищення локалізації та класифікації є декілька, можна представити деякі з них:

- донавчання нейронних мереж на зображеннях низької розрізняючої здатності з об'єктами малого розміру;
- проведення попередньої обробки зображення шляхом збільшення зображення (upscale) [108];
- обробка зображення частинами (тайлами) [109].

Ці способи є універсальними, але для подальшого дослідження було обрано саме обробку зображення частинами, тобто тайлами. На це є декілька причин.

Перше, це роздільна здатність нейронної мережі [110]. Більшість архітектур мають фіксований вхідний розмір та використання збільшення зображення не зможе збільшити вхідне розширення нейронної мережі. Збільшити роздільну здатність на вході нейронної мережі можливо, але це підвищує її обчислювальне навантаження та в деяких випадках знижує точність [111].

Друге - це використання більшої кількості зображень з малими об'єктами. Такий варіант є можливим, коли ми маємо змогу обробляти зображення з високою роздільною здатністю. Використовуючи ж обмежений розмір вхідних зображень, можемо мати недопустимо малий розмір регіонів. Відповідно ці регіони будуть відкинуті в процесі навчання через недоцільність їх використання для навчання.

Третє - це поділ зображення на частини та проведення послідовної чи паралельної обробки для кожної з них. Це дає можливість покращити ефективну вхідну роздільну здатність для нейронної мережі. Якщо розділити зображення на 4 частини, то вхідна роздільна здатність еквівалентно покращиться приблизно в два рази по кожній з координат. Відповідно такий підхід дає змогу обробляти зображення з розрізняючою здатністю, близькою до оригіналу [112]. Також при кращій розрізняючій здатності можна використовувати поділ зображення на більшу кількість частин. Але такий спосіб також підвищує кількість зображень, що необхідно обробити, тож зростає обчислювальне навантаження нейронної мережі.

Алгоритм методу обробки зображення частинами представлено на рисунку 3.12. Відносно оригінального методу в нього додається частина з розділенням

зображення на частини (тайли) та агрегація результатів. Розділення зображення на частини проводиться шляхом визначення середніх значень розмірності по кожній з координат, тобто в більшості випадків відношення сторін не змінюється. В процесі розділення зображення також зберігаються координати кожної з частин для подальшої агрегації результатів. Агрегація проводиться шляхом додавання зсувів до кожного з визначених координат, які відповідають одній з частин. Програмний код, який реалізує такий метод представлено в Додатку В.

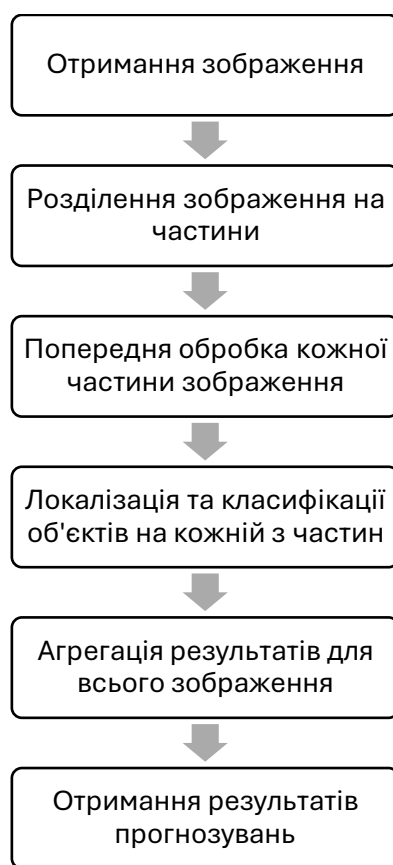
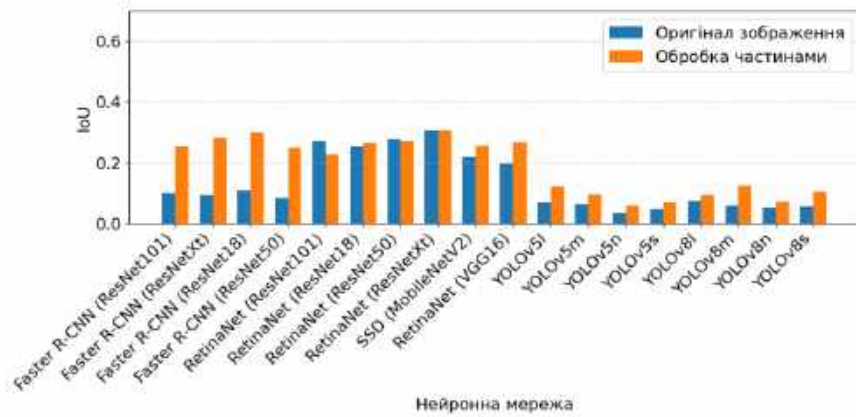


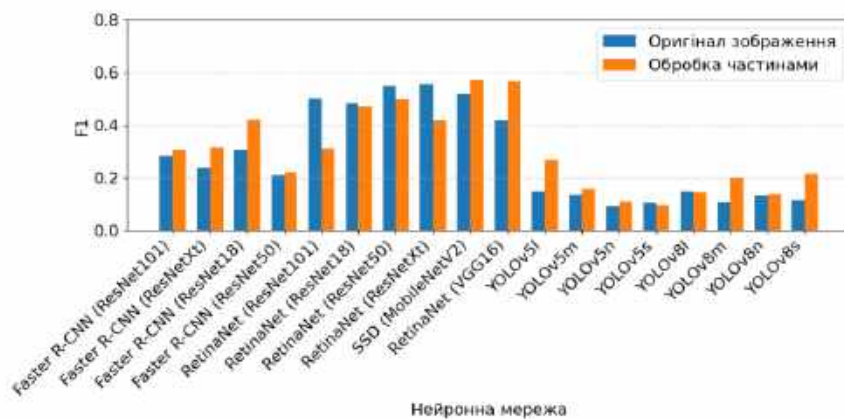
Рисунок 3.12 – Використання обробки зображення частинами в методів локалізації та класифікації

Для визначення доцільності використання даного методу для локалізації та класифікації малорозмірних об'єктів було проведено дослідження для регіонів різного розміру, відповідно до частини 3.2. Відповідно до послідовності дій в попередньому розділі, та для співставлення результатів, першим пунктом розрахуємо метрики точності локалізації та класифікації для регіонів всіх

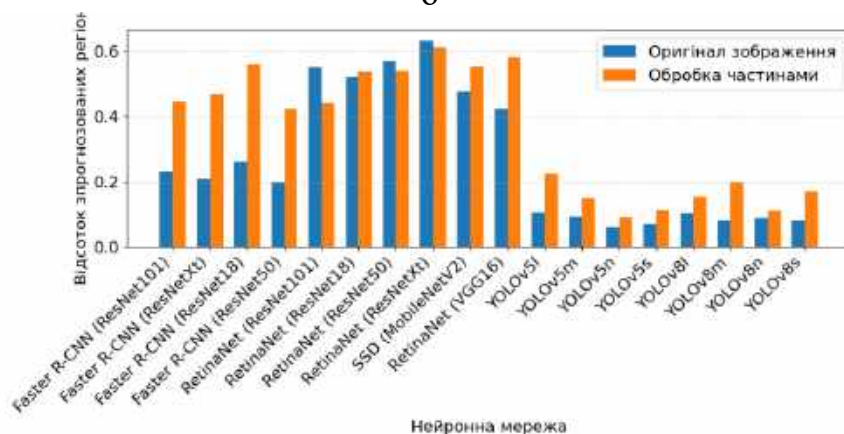
розмірів. Отримані дані представлено аналогічно до відповідних значень з попереднього підрозділу. На рисунку 3.13а представлено результат для метрики якості локалізації, на рисунку 3.13б - результат для метрики якості класифікації та на рисунку 3.13в - результат для кількості прогнозованих регіонів.



а



б



в

Рисунок 3.13 – Стопчикові діаграми метрик якості локалізації (а), класифікації (б) та кількості прогнозованих регіонів для регіонів розміром до 100 пікселів

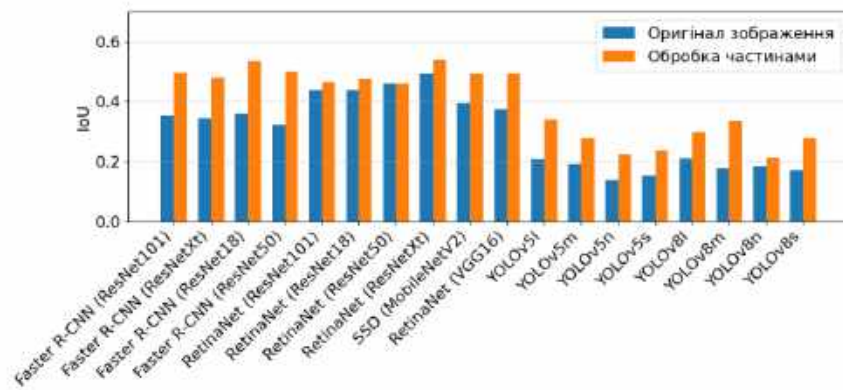
Враховуючи отримані результати та порівнявши їх з результатами, отриманими в розділі 3.2 (таблиця 3.2 та рисунок 3.8) маємо кращі показники. При аналізі помітно, що кількість прогнозованих регіонів зростає для більшості нейронних мереж, зокрема зростання становить від 0,2 до 0,4 пунктів. Також помітний зріст є для метрики якості локалізації, яка зростає на 0,1-0,2 пункти. Метрика якості класифікації зростає не суттєво. Числові значення метрик для регіонів такого розміру представлено в таблиці 3.6.

Таблиця 3.6 – Результати прогнозувань для регіонів з площею до 100 пікселів

Назва нейронної мережі	F1	IoU	Відсоток прогнозованих регіонів
Faster R-CNN (ResNet18)	0,488	0,515	0,823
Faster R-CNN (ResNet50)	0,408	0,507	0,806
Faster R-CNN (ResNet101)	0,570	0,509	0,805
Faster R-CNN (ResNetXt)	0,571	0,514	0,805
RetinaNet (ResNet18)	0,617	0,495	0,825
RetinaNet (ResNet50)	0,652	0,485	0,812
RetinaNet (ResNet101)	0,543	0,493	0,804
RetinaNet (ResNetXt)	0,595	0,507	0,826
SSD (MobileNetV2)	0,720	0,503	0,853
SSD (VGG16)	0,684	0,496	0,833
YOLOv5n	0,674	0,482	0,641
YOLOv5s	0,655	0,503	0,674
YOLOv5m	0,676	0,505	0,704
YOLOv5l	0,686	0,502	0,722
YOLOv8n	0,686	0,488	0,659
YOLOv8s	0,662	0,506	0,696
YOLOv8m	0,659	0,506	0,719
YOLOv8l	0,712	0,509	0,727

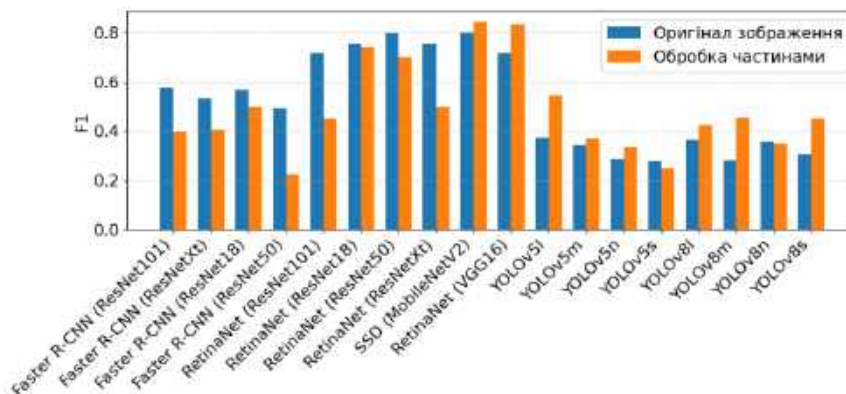
Наступною є оцінка метрик якості локалізації та класифікації для регіонів, що мають розмір від 101 до 200 пікселів. Отримані результати представлено в таблиці 3.7. Також на стопчикових діаграмах на рисунку 3.14 представлено порівняння метрик якості локалізації при використанні запропонованого методу та обробкою оригінального зображення. Отримані результати також показують

зростання. Кількість прогнозованих регіонів зростає в середньому на 0,2 пункти. Метрика якості локалізації також показує зростання на 0,15-0,2 пункти. Якість класифікації для деяких нейронних мереж знижується, але для YOLO та SSD показує зростання.



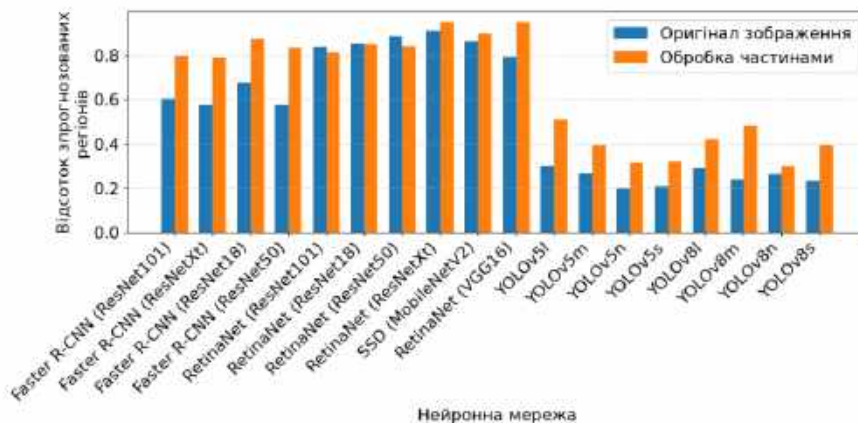
Нейронна мережа

а



Нейронна мережа

б



Нейронна мережа

в

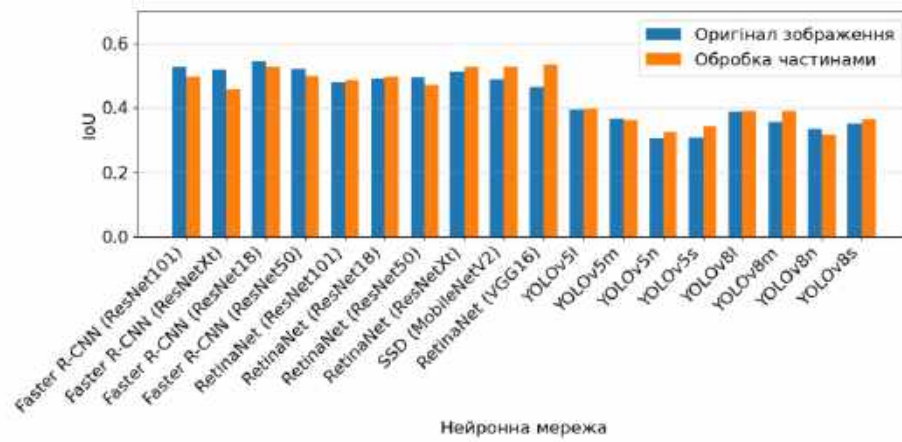
Рисунок 3.14 – Стопчикові діаграми метрик якості локалізації (а), класифікації (б) та кількості прогнозованих регіонів для регіонів розміром від 101 до 200 пікселів

Таблиця 3.7 – Результати прогнозувань для регіонів з площею до 200 пікселів

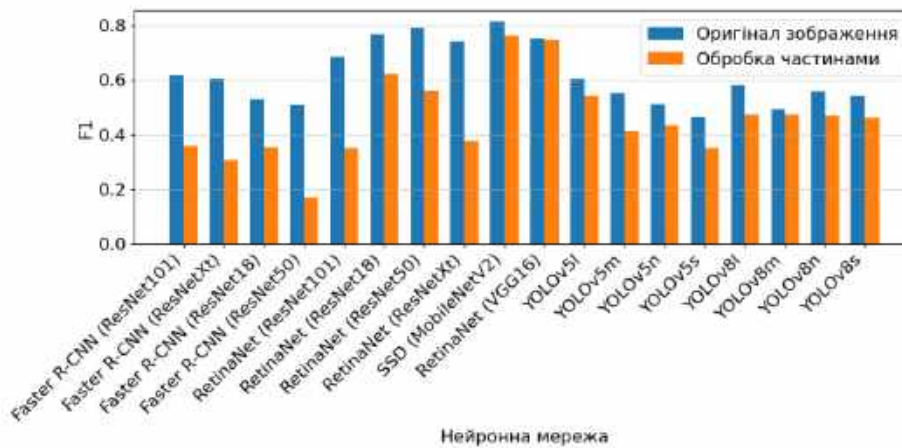
Назва нейронної мережі	F1	IoU	Відсоток прогнозованих регіонів
Faster R-CNN (ResNet18)	0,498	0,534	0,874
Faster R-CNN (ResNet50)	0,224	0,501	0,833
Faster R-CNN (ResNet101)	0,399	0,496	0,801
Faster R-CNN (ResNetXt)	0,408	0,481	0,789
RetinaNet (ResNet18)	0,739	0,477	0,850
RetinaNet (ResNet50)	0,701	0,461	0,840
RetinaNet (ResNet101)	0,452	0,467	0,815
RetinaNet (ResNetXt)	0,496	0,539	0,952
SSD (MobileNetV2)	0,844	0,493	0,900
SSD (VGG16)	0,833	0,493	0,951
YOLOv5n	0,337	0,223	0,319
YOLOv5s	0,249	0,239	0,322
YOLOv5m	0,373	0,279	0,396
YOLOv5l	0,544	0,340	0,511
YOLOv8n	0,350	0,214	0,302
YOLOv8s	0,453	0,279	0,394
YOLOv8m	0,455	0,336	0,482
YOLOv8l	0,425	0,299	0,423

Далі проведемо дослідження регіонів з площею від 201 до 500 пікселів. Значення метрик представлено в таблиці 3.8. Також на рисунку 3.15 представлено стопчикові діаграми результатів які отримані після застосування запропонованого методу та обробки цілого зображення. В випадку з дослідженнями, коли оброблялось усе зображення одночасно, було визначено, що цей діапазон є переломним, та починаючи з такої площі маємо хороші показники якості роботи нейронних мереж. В випадку з дослідженням, коли зображення оброблялось частинами, не маємо особливого приросту кількості прогнозованих регіонів, в деяких випадках маємо також незначний спад метрики. Для метрики якості локалізації також має місце спад для більшості нейронних мереж. Також метрика якості класифікації досить сильно знижується для нейронних мереж Faster R-CNN. Така поведінка метрик при використанні

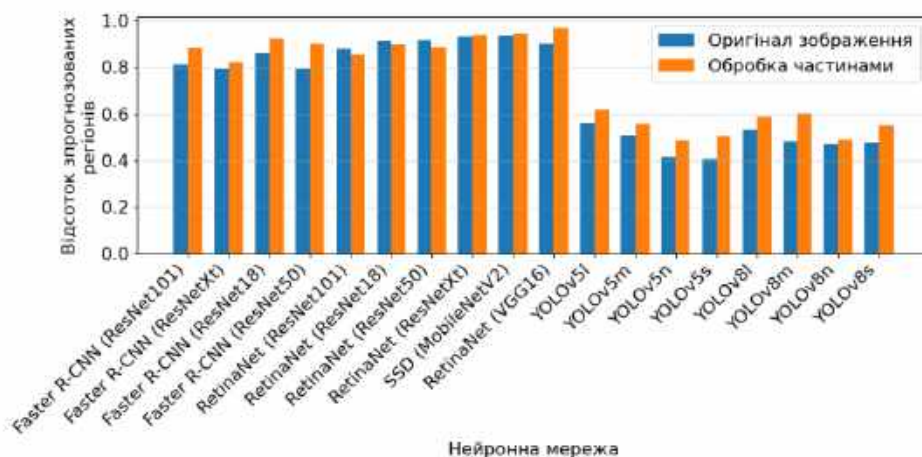
запропонованого методу відображає проблематику обробки регіонів великого розміру за допомогою навчених нейронних мереж.



а



б



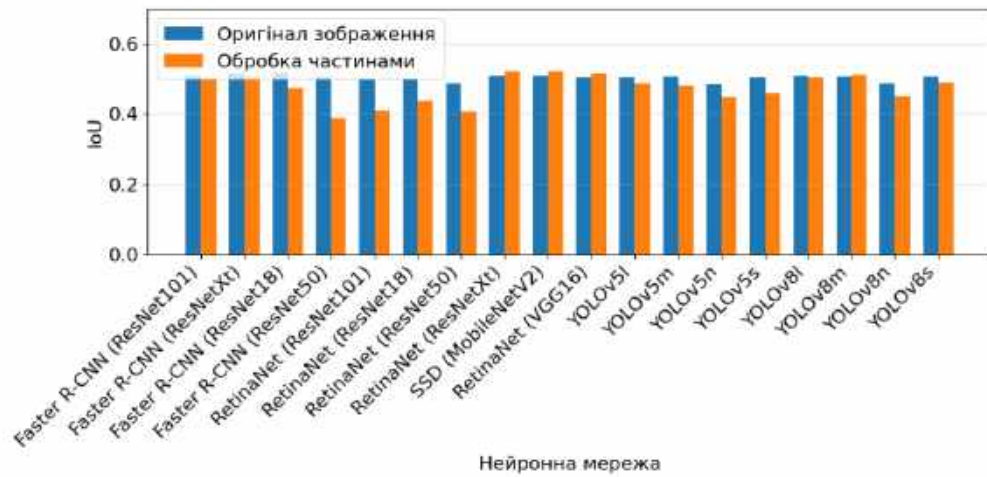
в

Рисунок 3.15 – Стопчикові діаграми метрик якості локалізації (а), класифікації (б) та кількості прогнозованих регіонів для регіонів розміром від 201 до 500 пікселів

Таблиця 3.8 – Результати прогнозувань для регіонів з площею до 500 пікселів

Назва нейронної мережі	F1	IoU	Відсоток прогнозованих регіонів
Faster R-CNN (ResNet18)	0,352	0,527	0,926
Faster R-CNN (ResNet50)	0,170	0,502	0,901
Faster R-CNN (ResNet101)	0,358	0,498	0,884
Faster R-CNN (ResNetXt)	0,307	0,460	0,823
RetinaNet (ResNet18)	0,624	0,497	0,899
RetinaNet (ResNet50)	0,562	0,471	0,886
RetinaNet (ResNet101)	0,351	0,487	0,856
RetinaNet (ResNetXt)	0,375	0,526	0,940
SSD (MobileNetV2)	0,764	0,527	0,945
SSD (VGG16)	0,749	0,535	0,970
YOLOv5n	0,435	0,327	0,485
YOLOv5s	0,350	0,343	0,507
YOLOv5m	0,412	0,363	0,557
YOLOv5l	0,543	0,398	0,618
YOLOv8n	0,472	0,317	0,488
YOLOv8s	0,463	0,365	0,551
YOLOv8m	0,474	0,391	0,600
YOLOv8l	0,474	0,392	0,589

Також було проведено дослідження для регіонів великого розміру, що більше за 500 пікселів. Результати метрик, отриманих в цьому дослідженні, представлено в таблиці 3.9. Також на рисунку 3.16 представлено порівняння метрик точності локалізації та класифікації відносно результатів що отримані при обробці повнорозмірного зображення. Отримані результати кількості прогнозованих регіонів при використанні запропонованого методу є меншими ніж при обробці повнорозмірного зображення, зниження становить приблизно 0,1 пункт. Також спостерігається зниження метрики якості локалізації, особливо для нейронних мереж Faster R-CNN та RetinaNet, для яких зниження метрики становить приблизно 0,1 пункт. Якість класифікації знижується досить сильно, зниження метрики становить 0,2-03 пункти, що є досить високим значення. З цього випливає висновок про неможливість деяких нейронних мереж точно локалізувати та класифікувати регіони великого розміру.



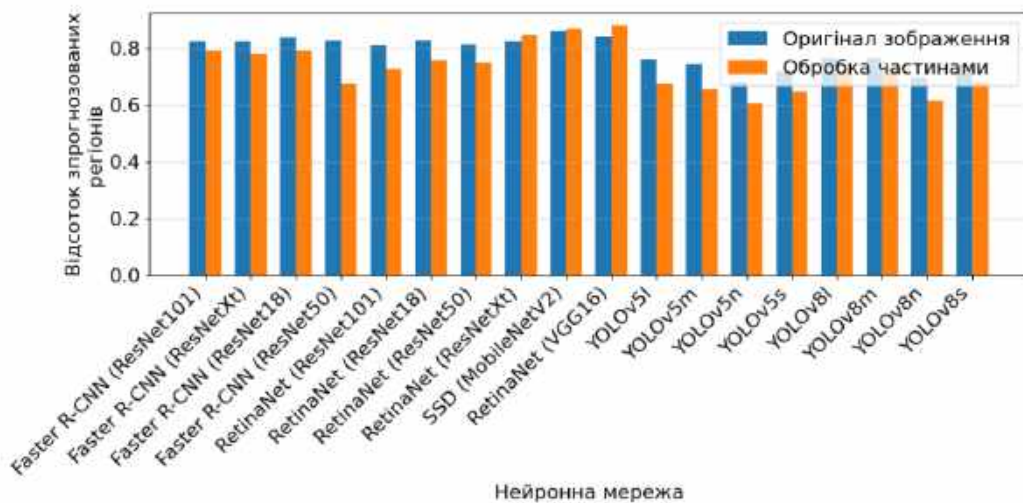
Нейронна мережа

а



Нейронна мережа

б



Нейронна мережа

в

Рисунок 3.16 – Стопчикові діаграми метрик якості локалізації (а), класифікації (б) та кількості прогнозованих регіонів для регіонів розміром від 500 пікселів

Таблиця 3.9 – Результати прогнозувань для регіонів з площею більш 500 пікселів

Назва нейронної мережі	F1	IoU	Відсоток прогнозованих регіонів
Faster R-CNN (ResNet18)	0,160	0,474	0,791
Faster R-CNN (ResNet50)	0,125	0,393	0,678
Faster R-CNN (ResNet101)	0,241	0,498	0,787
Faster R-CNN (ResNetXt)	0,190	0,502	0,773
RetinaNet (ResNet18)	0,390	0,438	0,759
RetinaNet (ResNet50)	0,283	0,409	0,750
RetinaNet (ResNet101)	0,216	0,413	0,728
RetinaNet (ResNetXt)	0,282	0,521	0,848
SSD (MobileNetV2)	0,532	0,518	0,865
SSD (VGG16)	0,524	0,510	0,880
YOLOv5n	0,442	0,450	0,584
YOLOv5s	0,495	0,458	0,620
YOLOv5m	0,586	0,479	0,633
YOLOv5l	0,534	0,484	0,656
YOLOv8n	0,591	0,451	0,588
YOLOv8s	0,506	0,489	0,656
YOLOv8m	0,518	0,508	0,701
YOLOv8l	0,595	0,504	0,700

Аналізуючи отримані результати можемо зробити висновок, що метод обробки зображення частинами є досить дієвим, але для обмеженого діапазону застосування. Застосування методу підвищує кількість прогнозованих регіонів та якість локалізації для регіонів розміром до 200 пікселів. Для регіонів від 201 пікселів для багатьох випадків маємо сталу метрику або ж зниження результатів, але досить не суттєве для більшості нейронних мереж. В випадку з регіонами розміром більше 500 пікселів маємо різке зниження якості класифікації, а також зниження кількості прогнозованих регіонів та метрики якості локалізації. Враховуючи це важливим є використання умови обробки. Тобто визначати розмір об'єктів, які необхідно локалізувати та класифікувати, в пікселях використовуючи дані про висоту польоту та характеристики системи захвату зображень.

3.4 Метод визначення характеристик об'єкту

Враховуючи, що в попередньому розділі результати використання методу обробки зображення частинами дають суттєве підвищення точності локалізації та класифікації регіонів малого розміру, а також зниження точності роботи в умовах об'єктів великого розміру важливим є дослідження та визначення граничних характеристик польоту. Для цього використовуючи отриману інформацію про досліджені граничні розміри регіону в пікселях, а саме 200 пікселів для достатнього рівня класифікації без використання методу обробки частинами визначимо граничні висоти польоту. Графічно вигляд системи формування зображення з використанням БПЛА можна представити за допомогою зображення на рисунку 3.17.

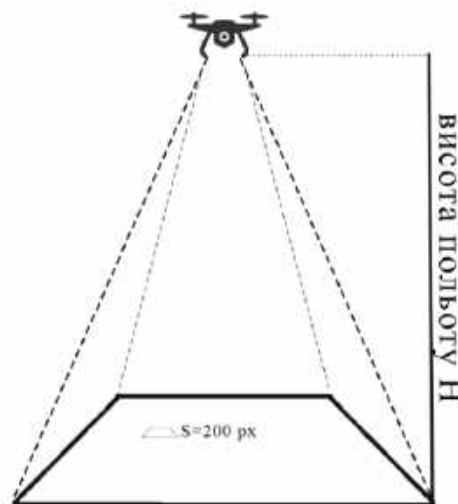


Рисунок 3.17 – Графічне відображення системи формування зображення з використанням БПЛА

Для цього визначимо Δ як лінійну роздільну здатність системи формування зображення. Тоді $\delta\Theta$ можемо визначити як кутову роздільну здатність, яка виражена в радіанах. Враховуючи, що H – відстань до об'єкту польоту, можемо

визначити лінійну роздільну здатність використовуючи кутову роздільну здатність та висоту польоту (3.1):

$$\Delta = H \times \delta\Theta, \quad (3.1)$$

Враховуючи, що кутова роздільна здатність пов'язана з кутом огляду камери (Θ) та кількістю пікселів на зображенні, можемо виразити площу об'єкту за допомогою формули (3.2):

$$\frac{S}{\Delta^2} = \frac{S}{(H^2 \Theta^2 / N^2)} = \frac{S \times N^2}{H^2 \Theta^2}, \quad (3.2)$$

де N^2 – це кількість пікселів,

S – площа об'єкту.

Використовуючи отриману формулу можемо виразити відношення до граничного розміру регіону (3.3):

$$\frac{S \times N^2}{H^2 \Theta^2} = 200, \quad (3.3)$$

За отриманими рівняннями можемо вивести кінцеву формулу для визначення висоти польоту БПЛА (3.4):

$$H = NS^{0,5} / 14,1\Theta, \quad (3.4)$$

За допомогою отриманої формули визначення висоти польоту, яка враховує визначені в підрозділі 3.3 граничні значення при яких є доцільним використання методу обробки зображення частинами розрахуємо висоту польоту. Якщо площа об'єкту становить $0,3 \text{ м}^2$ та розмір зображення за однією стороною, що

отримується за допомогою системи формування зображення становить 640 пікселів, а кут зйомки рівний $\frac{\pi}{6}$ то можемо визначити висоту польоту (3.5):

$$H = \frac{640 \times 0,3^{0,5}}{14,1 \times 0,52} \approx 47,8 \text{ м}, \quad (3.4)$$

З отриманого результату можемо зробити висновок, що при розмірі об'єкту в 0,3 метри, що відповідає 200 пікселів на зображенні, для досягнення високого рівня точності локалізації та класифікації варто щоб зйомка проводилась з висоти не більше ніж 47,8 метри. При зйомці з більшої висоти варто використовувати метод обробки зображення частинами.

3.5 Висновки до розділу

В розділі було досліджено роботу нейронних мереж для локалізації та класифікації при роботі з об'єктами малого розміру. Всі об'єкти було розділено на 4 основних групи: 1) об'єкти площею до 100 пікселів; 2) об'єкти з площею від 101 до 200 пікселів; 3) об'єкти, що мають площу від 201 до 500 пікселів, для яких було доведено, що цього розміру достатньо для допустимої точності локалізації та класифікації; 4) об'єкти з площею більше 500 пікселів, що є достатнім рівнем для локалізації та класифікації.

Дослідження проводились на наборі даних, що має досить широкий діапазон розміру об'єктів, а також схожу специфіку з тренувальними даними і велику кількість даних. Це дозволило дослідити нейронні мережі в широкому діапазоні розмірів об'єктів, а також визначити граничні розміри об'єкту в пікселях для достатньої точності локалізації та класифікації.

На основі отриманих даних запропоновано використання методу обробки зображення частинами для підвищення точності локалізації та класифікації малорозмірних об'єктів. Для регіонів, що мають розмір до 200 пікселів

використання цього методу для деяких нейронних мереж значно покращує метрики, зокрема кількість локалізованих регіонів збільшується в середньому на 0,15 пунктів. Точність локалізації також може бути підвищена в середньому на 0,2 пункти в залежності від нейронної мережі. Враховуючи такі отримані дані можна зробити висновок, що даний метод підвищує точність локалізації та класифікації об'єктів, але на визначеному діапазоні розмірів об'єктів, зокрема визначено що до 200 пікселів.

Враховуючи, що 200 пікселів є граничним значенням, об'єкти розміром більші за яке можуть бути локалізовані точніше без використання методу обробки частинами було запропоновано метод розрахунку висоти польоту опираючись на характеристики системи формування зображення та розмір об'єкту. Він враховує реальний розмір об'єкту, просторове розрізнення зображення та кутову роздільну здатність. Визначення параметрів польоту враховуючи даний метод є досить точним та може бути використаний для методу локалізації та класифікації як умова вибору методу обробки. Враховуючи даний параметр та характеристики системи формування зображення можна визначити граничні висоти польоту для об'єктів різного розміру.

Отримані результати також показують що в випадку з обробкою малорозмірних регіонів найбільшу точність мають нейронні мережі Retina Net які використовують FPN в своїй архітектурі.

Проведені в розділі дослідження було відображено у наукових працях [107, 108].

РОЗДІЛ 4.

ДОСЛІДЖЕННЯ РОБОТИ НЕЙРОННИХ МЕРЕЖ В УМОВАХ СПОТВОРЕНЬ

Вплив різного типу завад на зображення, зокрема шуму, може сильно спотворювати зображення, що ускладнює його обробку. В випадку з даними, що отримані за допомогою безпілотних літальних апаратів, досить часто ми отримуємо зображення, що спотворені шумом. Це пов'язано з кількома факторами, через які може проявлятися шум на зображеннях. Перше, це вплив внутрішніх чинників камери, таких як перегрів чи наведення. Перегрів часто трапляється в модулях малого розміру, через близьке розташування різних елементів. Другий фактор - це вплив зовнішніх факторів, які також можуть впливати як на перегрів, так і на наведення, наприклад підвищена вологість. Третім фактором виникнення шуму на зображеннях є використання довгих витримок, через що на сенсор камери може потрапляти багато світла, що створює шум. Стиснення з втратами також призводить до спотворень, які, за певних умов, подібні до адитивного гаусова білого шуму [113].

Враховуючи ймовірність спотворення зображення шумом дослідження точності роботи нейронних мереж на таких зображеннях є важливою задачею. Також важливим є етап дослідження можливих способів підвищення точності роботи нейронних мереж в таких умовах. Саме тому розділ присвячено дослідженню впливу шумових завад на якість локалізації та класифікації об'єктів за допомогою нейронних мереж.

Існує досить багато методів оцінки шуму на зображеннях, більшість з них базуються на отриманні статистичних характеристики зображення [114]. Також методи визначення характеристик шуму можуть бути побудовані на базі нейронних мереж [115, 116]. Ці методи дають оцінку шуму на зображеннях, та дозволяють застосовувати різні алгоритми фільтрації для придушення шуму. Знаючи параметри шуму ми можемо застосовувати алгоритми фільтрації з більшою точністю [117]. Також маючи дані про параметри та природу шуму ми

можемо оцінити ефективність фільтрації та її вплив на зображення, не проводячи сам процес фільтрації [118], що суттєво зменшує навантаження на пристрій.

Аналізуючи дослідження, які проводили інші вчені можемо визначити, що вплив фільтрації на якість локалізації та класифікації є досить високим [119, 120] та вимагають попередньої обробки в вигляді видалення шуму на зображеннях [121, 122]. Враховуючи ці фактори подальші наші дослідження повинні включати дослідження впливу шуму на якість локалізації та класифікації та вплив фільтрації на покращення точності роботи нейронних мереж.

Також факторами, що можуть вплинути на якість локалізації та класифікації, є вплив погодних умов на зображення. Це можуть бути різного типу природні явища, такі як туман чи дощ. Наприклад щільний туман унеможлиблює точну локалізацію та класифікацію, або навіть візуальне розрізнення об'єктів. Дощ може впливати на зображення по різному, в залежності від налаштувань камери та від її розташування на БПЛА. Тому друга частина розділу присвячена дослідженню роботи нейронних мереж в умовах завад, що спричинені погодними факторами.

4.1 Дослідження роботи нейронних мереж в умовах впливу шуму

Для дослідження впливу шуму різної інтенсивності на якість роботи нейронних мереж обрано метод штучного додавання адитивного гаусового шуму різної інтенсивності до зображень, що вважаються безшумними. В подальшому на згенерованих зображеннях проведено оцінку метрик візуальної якості та метрик точності локалізації та класифікації. За допомогою такого підходу можна визначити інтенсивність шумових завад, який може суттєво впливати на якість роботи нейронних мереж. Відповідні результати порівнюємо з результатами, що отримані в розділі 3.1, де визначено точність локалізації об'єктів різного розміру. Отриману статистику проаналізовано на якість локалізації та класифікації при впливі завад різної інтенсивності, що дає змогу оцінити стійкість нейронних

мереж до впливу завад. Також в розділі запропоновані методи покращення якості зображення, шляхом видалення шуму за допомогою фільтру BM3D [118], які базуються на попередньому визначенні типу та характеристик завад.

4.1.1. Оцінка природи та параметрів шуму на зображеннях

Для проведення досліджень, що стосуються визначення точності роботи нейронних мереж для локалізації та класифікацій об'єктів є необхідним визначення типу та природи шуму [123], а також його параметрів на реальних зображеннях. Це необхідно для генерації штучного набору даних, який буде містити попередньо розмічені регіони та зображення будуть спотворені шумом необхідної інтенсивності. Алгоритм дослідження можна описати за допомогою блок-схеми, яка представлена на рисунку 4.1.

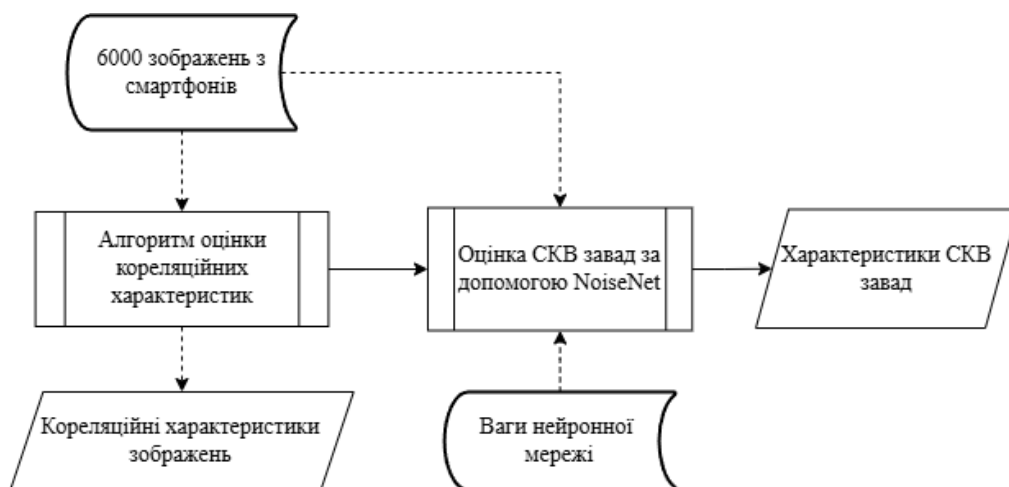


Рисунок 4.1 – Блок-схема алгоритму дослідження шуму

Проведення досліджень, що стосуються природу шуму, будемо проводити на зображеннях, що отримані за допомогою камер телефонів. Всі отримані зображення збережені відразу з камери, не проходячи етапу обробки та покращення якості. Це дозволяє отримати зображення як є та проводити дослідження з сирими даними. Для цього було розроблено додаток для системи Android, який дозволяє отримувати зображення безпосередньо з камери

телефону та зберігати в форматі tiff, який не використовує стиснення та обробки, що в свою чергу дає змогу аналізувати сирі дані. Загалом, за допомогою методу було отримано 6000 зображень з 6 різних пристроїв, з різними сенсорами та різних виробників. Також зображення були отримані в різному освітленні, різну пору доби та умовах, а також в приміщенні та на вулиці. Такий підхід дозволяє проаналізувати широкий набір зображень та отримати вірну оцінку типу шуму на зображеннях. Приклади зображень, що були отримані в процесі збору вибірки представлено на рисунку 4.2.

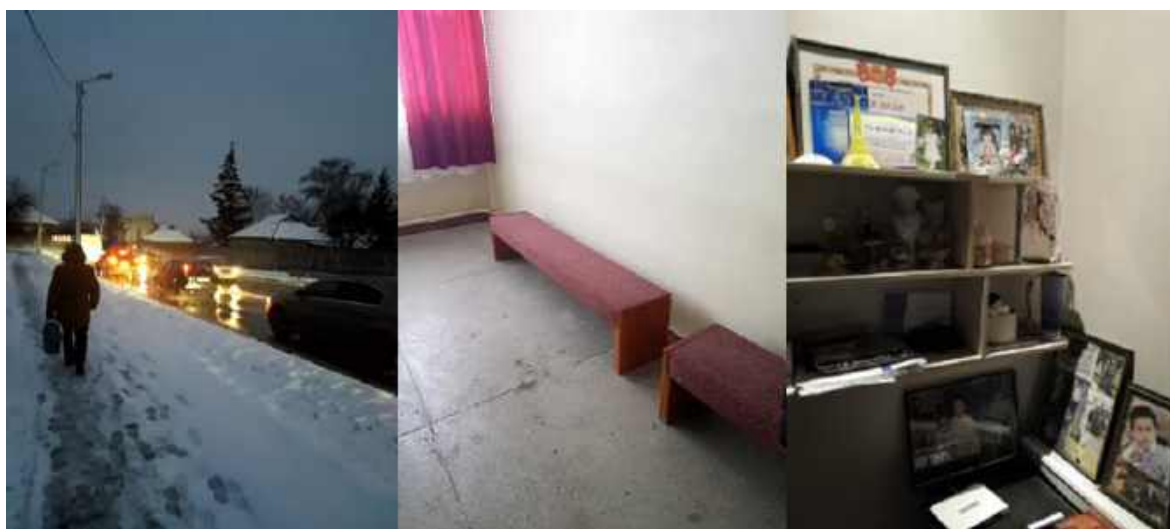


Рисунок 4.2 – Приклади зображень що були використані для оцінки параметрів шуму

Першим етапом було досліджено кореляційні характеристики шуму за допомогою автоматичного методу визначення кореляційних характеристик завад [124]. Отримані результати показують що шум є загалом просторово корельованим (з невеликим ступенем кореляції) та залежить від умов освітленості. Так, при хорошому вуличному освітленні маємо низьку просторову кореляцію, в приміщенні при денному світлі та хорошому штучному освітленні маємо трохи більший рівень кореляції. На зображеннях, що отримані при штучному освітленні на вулиці, а також при поганому освітленні в приміщенні, ступінь кореляції шуму є ще більшою. Враховуючи отримані дані та те, що сенсори є оптичними, можемо припустити що завади є сумішшю адитивних та

квазі-пуасонівських завад [114-116], які можна представити за допомогою формули (4.1):

$$\sigma_n^2(I) = \sigma_{SI}^2 + k_{sd}I, \quad (4.1)$$

де σ_{SI}^2 – дисперсія сигнально-незалежної компоненти шуму;

I – справжнє значення пікселя зображення;

k_{sd} – це коефіцієнт пропорційності сигнальнозалежної компоненти шуму (коефіцієнт підсилення квазі-пуасонівського шуму); дисперсія сигнальнозалежної компоненти шуму буде відповідно дорівнювати $k_{sd}I$.

Отримані дані дозволяють застосувати метод [124], що призначений для визначення параметрів завад, що описані формулою (4.1). Метод полягає в побудові скатерограми пар локальних оцінок дисперсії і середнього, отриманих для однорідних ділянок зображення, визначенні центрів кластерів скатерограми робастним методом і вписуванні регресійного полінома першого порядку за цими центрами з використанням двічі зваженого метода найменших квадратів. Параметри цього полінома є оцінками характеристик сигнально-незалежної і сигнальнозалежної складових завади. Отримані результати представлено в таблиці 4.1 в залежності від умов зйомки. Отримані дані показують, що на зображеннях дійсно присутня суміш адитивних та квазі-пуасонівських завад іноді з превалюванням сигнально-залежної складової. Отримана оцінка показує, що результати збігаються з результатами інших дослідників [125], а тому отримані результати можна вважати вірними.

Отримані дані про природу шуму дозволяють застосувати NoiseNet [116, 126], яка розроблена для визначення параметрів просторово-некорельованих завад. Нейронна мережа дозволяє ітеративно, тобто послідовно для кожного з блоків, прогнозувати параметри шуму. Загалом дана нейронна мережа працює з блоками 32x32 та для кожного з таких блоків прогнозує значення СКВ шуму. Також для кожного з прогнозованих параметрів додається оцінка, за допомогою якої можна

визначити на скільки даний блок репрезентативний та може бути використаний для загальної оцінки СКВ шуму.

Нейронну мережу NoiseNet [116, 126] було навчено з допомогою вибірки зображень в кількості 158 штук, які було зроблено з допомогою дзеркальної камери Nikon D80. Для кожного зображення було отримано 3120 блоків 32×32 , які не перекриваються. На кожному з блоків було оцінено програмним методом основне значення дисперсії шуму за формулою (4.1). Для перевірки точності отримання значення дисперсії було проведено оцінку також з накладанням блоків $s = 2$ та 4 . В результаті було помічено що значення залежить від кількості накладених пікселів, можливо це зв'язано з ефектом кліпінгу. Саме тому для точної оцінки дисперсії шуму було обрано блоки з накладанням 9 пікселів. В результаті оцінки та збору навчальних даних було отримано 690116 блоки 32×32 , серед яких 552092 (80%) використовувались для навчання та 138024 (20%) для тестування навченої мережі.

Для перевірки точності визначення рівня шумових завад, точніше їх СКВ, було проведено перевірку нейронної мережі на штучно спотворених зображеннях. В якості даних для перевірки було використано базу зображень, що складається з 300 високоякісних зображень, які мають власне СКВ шуму рівне 0 . Всі зображення було спотворено адитивним білим гаусовим шумом з двома рівнями інтенсивності 5 та 10 . На спотворених зображеннях, що були отримані, було визначено СКВ шуму з допомогою NoiseNet [126]. В результаті проведення тестування були отримані такі результати:

- для фрагментів однорідних блоків що були спотворені шумом з СКВ 5 , отримано значення СКВ $5,0013$;
- для фрагментів текстурних блоків, які були спотворені шумом з СКВ 5 , отримано значення СКВ $5,39$;
- для однорідних фрагментів зображення з реальним СКВ 10 , було отримано результат СКВ рівний $10,05$;
- для текстурних блоків, які були спотворені адитивним шумом з СКВ 10 , отримано СКВ рівне $11,18$.

Враховуючи отримані результати можна зробити висновок, що навчена нейронна мережа NoiseNet [116, 126] може бути використана для точної оцінки інтенсивності завад на зображеннях.

Таблиця 4.1 – Ступінь просторової кореляції шуму в залежності від умов зйомки

Освітлення та умови зйомки	Ступінь просторової кореляції	Рекомендований коефіцієнт проріджування
Денне вуличне світло	Слабка	2
	В хорошому освітленні відсутня	1
Денне світло в приміщенні	Слабка	2
Штучне освітлення на вулиці	середня	3
Штучне освітлення в приміщенні	Слабка	2
	При поганому освітленні сильна	3

Результати, отримані в процесі досліджень з зображеннями, що отримані за допомогою смартфонів, представлено в таблиці 4.2. В результаті аналізу всіх зображень, що отримані з мобільних пристроїв було визначено, що більшість зображень мають середнє значення СКВ, завад рівне 2 [127]. Деякі з зображень, особливо ті що отримані в умовах штучного вуличного світла та в приміщенні, мають середнє значення СКВ завад рівне 3. Для зображень що отримані в умовах денного світла максимальне значення СКВ досягає 7. З цього випливає, що основна маса отриманих зображень має середнє значення СКВ на рівні 2-7. Це дає змогу враховувати реалістичність подальших досліджень та орієнтуватись на ці дані в якості інформації про спотворення на зображеннях. Втім, до вказаних спотворень у вигляді шуму можуть додатись спотворення внаслідок стиснення з втратами. Більш того, на БПЛА можуть бути встановлені сенсори більш низької

якості, ніж в досліджених вище мобільних телефонах. Тож має сенс розглянути й більші значення СКВ шуму.

Таблиця 4.2 – Результати оцінки СКВ завад на зображеннях в залежності від умов зйомки

	Денне вуличне світло	Денне світло в приміщенні	Штучне освітлення на вулиці	Штучне освітлення в приміщенні
Середнє значення СКВ	3	7	4	5

4.1.2. Спотворення зображень шумом

Для дослідження якості роботи нейронних мереж в умовах, коли зображення спотворені шумом, важлива генерація такої вибірки. Зазвичай для досліджень впливу шуму використовують адитивний білий гаусовий шум (АБГШ). Генерація такого типу шуму є достатньою для визначення впливу завад на якість локалізації та класифікації. Також важливими аргументами в користь АБГШ є простота розуміння природи шуму та швидкість генерації зображень, а також операція використання варіаційно-стабілізуючих перетворень для конверсії сигнально-залежного шуму в адитивний, яка часто використовується для фільтрації та стиснення з втратами зображень, що спотворені сигнально-залежним шумом [128, 129]. Ще одним аргументом є те, що навіть якщо завади в сирих даних є сигнально-залежними, то вони стають більш близькими до адитивних після виконання так званої гамма-корекції, яка зазвичай використовується всередині мобільних пристроїв формування оптичних зображень [130].

Для генерування даних, що спотворені АБГШ було використано набір даних AU-AIR [101], який був використаний для досліджень в розділі 3.1. Це дозволяє провести оцінку відносно вже прорахованих метрик та порівняти значення.

Зображення, що містяться в наборі даних було спотворено адитивним білим гаусовим шумом з різним значенням середньоквадратичного відхилення (СКВ). Всі канали зображення були спотворені шумом з однаковою інтенсивністю, але з різним розподілом, що дозволяє генерувати більш реалістичні зразки. Загалом було обрано декілька рівнів шуму, значення СКВ для яких є 3, 5, 7, 10, 15, 20 та 25. Отримані зображення представлено на рисунку 4.3.

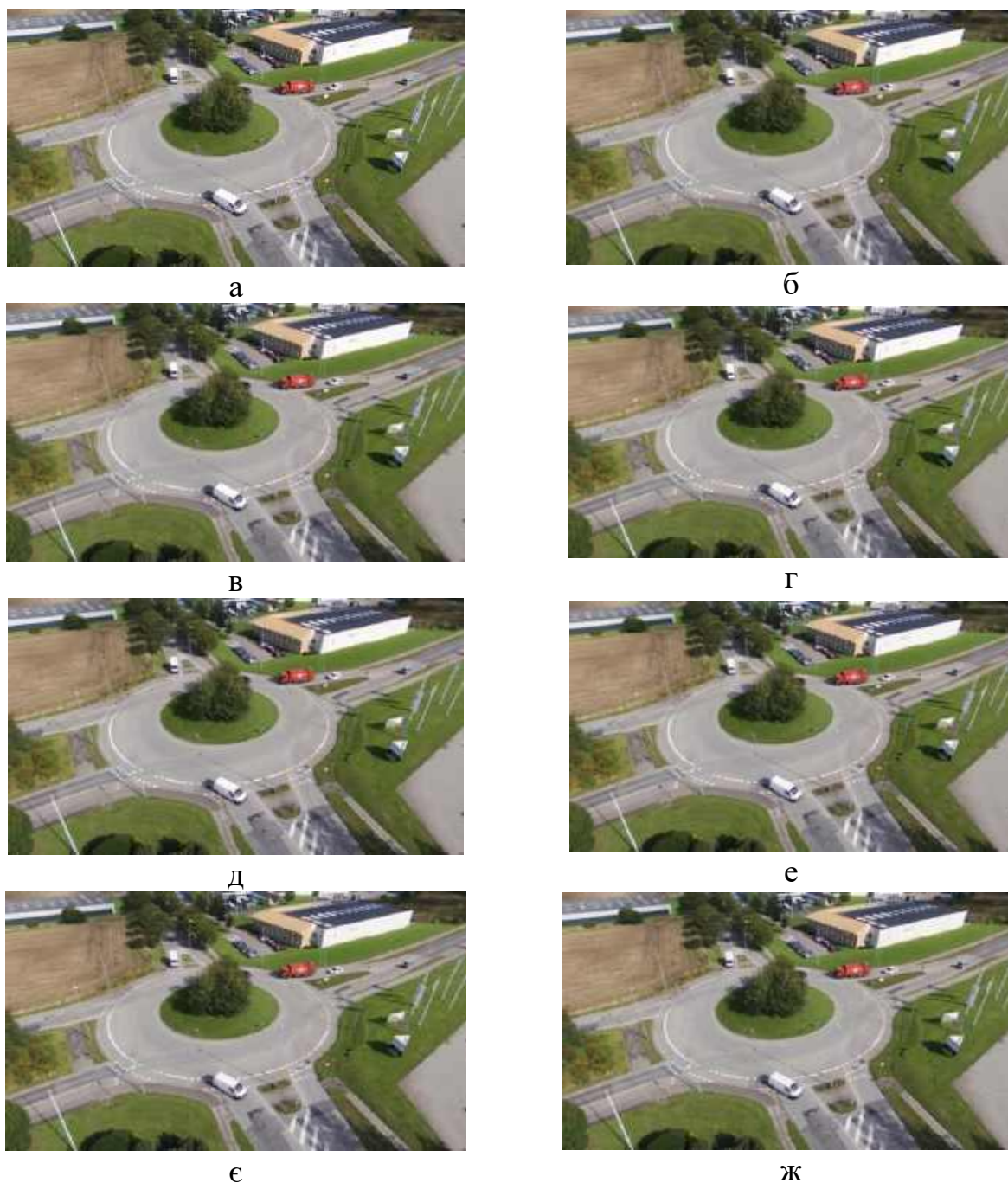


Рисунок 4.3 – Зображення за набору даних (а) та спотворене АБГШ з СКВ 3(б), 5(в), 7(г), 10(д), 15(е), 20(є) та 25 (ж)

Аналізуючи отримані дані на доцільність обраного діапазону значень СКВ помітно, що при СКВ 3 та 5 маємо досить низький рівень шуму, який є майже непомітним. Теоретично такий рівень завад не повинен сильно впливати на якість роботи нейронних мереж. При значенні СКВ=7 маємо більший рівень завад, який, можливо, ще також не є критичним. Для СКВ=10 отримуємо помітний шум, що може впливати на якість локалізації малих об'єктів, а також об'єктів специфічної форми та розміру. Значення СКВ>20 відповідають дуже високому рівню шуму. Об'єкти на таких зображеннях можуть бути спотворені, може бути змінена їх форма, колір, границі та текстура, що сильно впливає на якість роботи нейронних мереж.

Для оцінки рівня завад та відповідно ступеню спотворення було використано також метрику PSNR – пікове відношення сигналу до шуму. Розраховані значення метрики представлені в таблиці 4.3. Візуально аналізуючи спотворені шумом зображення, можна зрозуміти, що більшість тестових зображень не сильно спотворені шумом, якщо СКВ не більше, ніж 5 або 7, але використання більших значень СКВ дозволяють відслідкувати більш серйозні спотворення та можливості покращення якості локалізації.

Таблиця 4.3 Результати метрики PSNR для різних значень СКВ

СКВ	3	5	7	10	15	20	25
PSNR, dB	42,52	36,48	33,61	30,69	27,41	25,07	23,24

4.1.3. Дослідження точності роботи нейронних мереж

Як було визначено в попередніх розділах СКВ завад на згенерованих зображеннях має значення 3, 5, 7 (враховуючи оцінку зображень зі смартфонів) та 10, 15, 20, 25. Використовуючи отримані набори даних, потрібно провести дослідження ефективності методів локалізації та класифікації. Для цього проведено оцінку методів на основі нейронних мереж за параметрами кількості

прогнозованих регіонів, якості локалізації та якості класифікації. Отримані результати порівняно для кожного рівня інтенсивності завад.

Для порівняння значень метрик між різними нейронними мережами було побудовано стовпчикові діаграми значень цих метрик. На рисунку 4.4 представлено результати для відсотку прогнозованих регіонів, який відображає кількість вірно прогнозованих регіонів відносно розмітки. Оцінюючи отримані дані, можна помітити загальну тенденцію до зниження значення метрики для більших значень СКВ. Ця поведінка метрики досить прогнозована, адже при більшій інтенсивності завад об'єкти є менш помітними. Також варто зауважити, що для нейронних мереж YOLO спад метрики є більш стрімким, що може показувати гіршу якість роботи з спотвореними зображеннями. Метрики для нейронних мереж Faster RCNN та RetinaNet є майже незмінними, що свідчить про високу стійкість нейронної мережі до впливу завад. Отримана статистика відображає досить високий рівень стійкості нейронних мереж до впливу шуму.

Також було пораховано значення метрики IoU. Отримана статистика представлена на рисунку 4.5. Варто врахувати, що для розрахунку метрики не застосовуються не прогнозовані регіони, через що метрика відображає тільки точність прогнозування регіонів, не враховуючи їх кількість. Проаналізувавши отримані дані, є помітною практична сталість метрики. Для більших значень СКВ метрика зменшується, але на досить невелике значення, в діапазоні 0,01-0,05. Тож можна зробити висновок, що нейронні мережі досить добре локалізують об'єкти на зображеннях, що спотворені шумом. В більшості випадків інтенсивність завад не сильно впливає на точність локалізації.

Відповідні дослідження також було проведено для точності класифікації, яка відображена за допомогою метрики F1. Отримані стовпчикові діаграми представлені на рисунку 4.6. Аналізуючи отримані дані, помітно, що точність класифікації має найбільший спад у порівнянні з іншими метриками. Так, маємо різкий спад метрик для нейронних мереж YOLO, відповідний результат було отримано також і для кількості прогнозованих регіонів. Також отримані дані показують, що є різкий спад метрики для інших нейронних мереж з

екстракторами карти ознак ResNet18 та ResNet50. Ймовірно, що такий результат було отримано через досить невелику кількість параметрів в цих нейронних мережах, а тому нижчу можливість до аналізу різнотипних даних. Отримані значення показують досить низьку стійкість нейронних мереж до впливу завад при класифікації об'єктів.

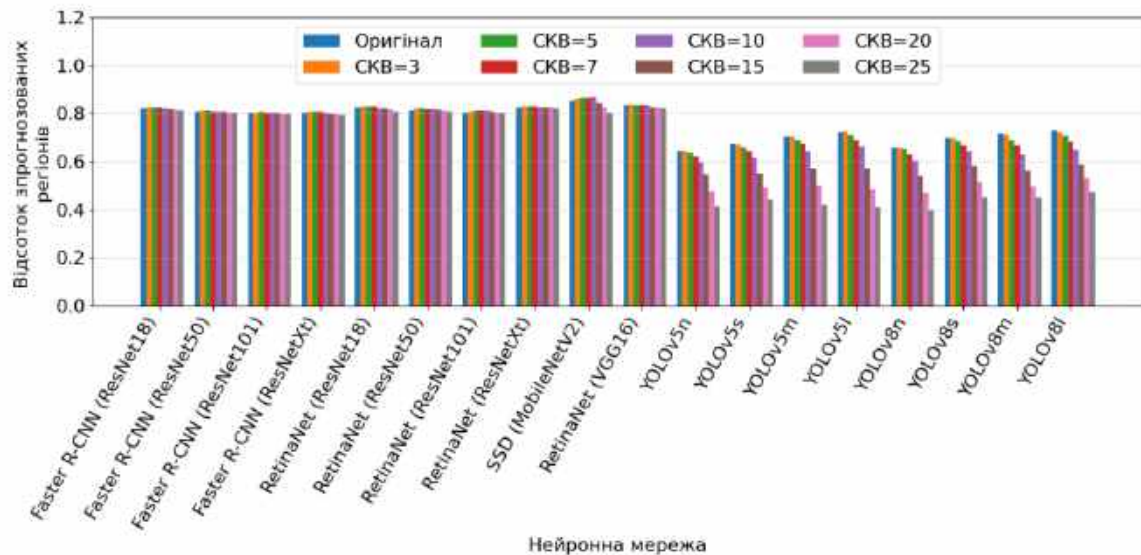


Рисунок 4.4 – Стопчикова діаграма відсотку прогнозованих регіонів в залежності від інтенсивності шуму

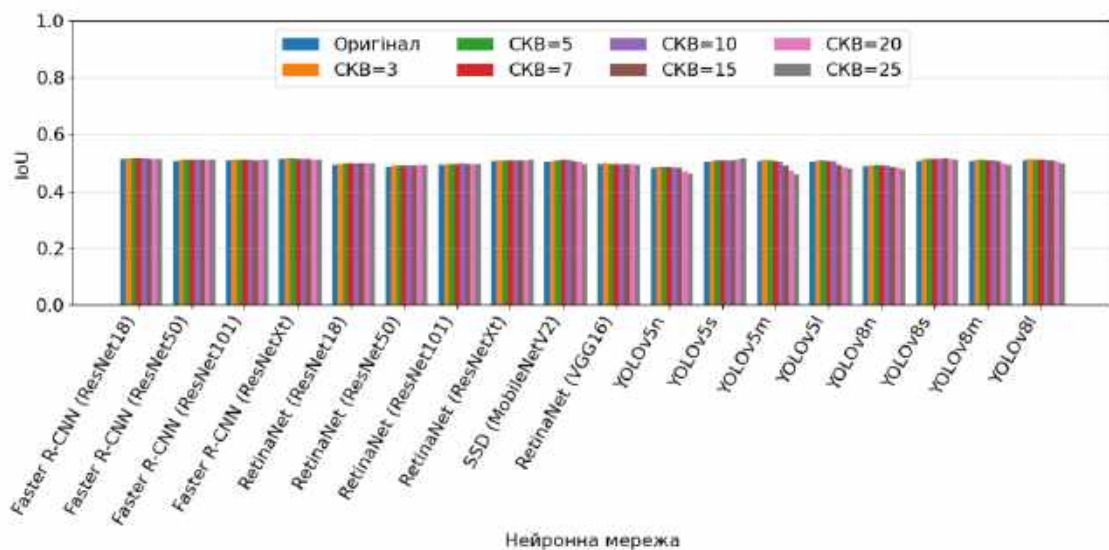


Рисунок 4.5 – Стопчикова діаграма метрики якості локалізації (IoU) в залежності від інтенсивності шуму

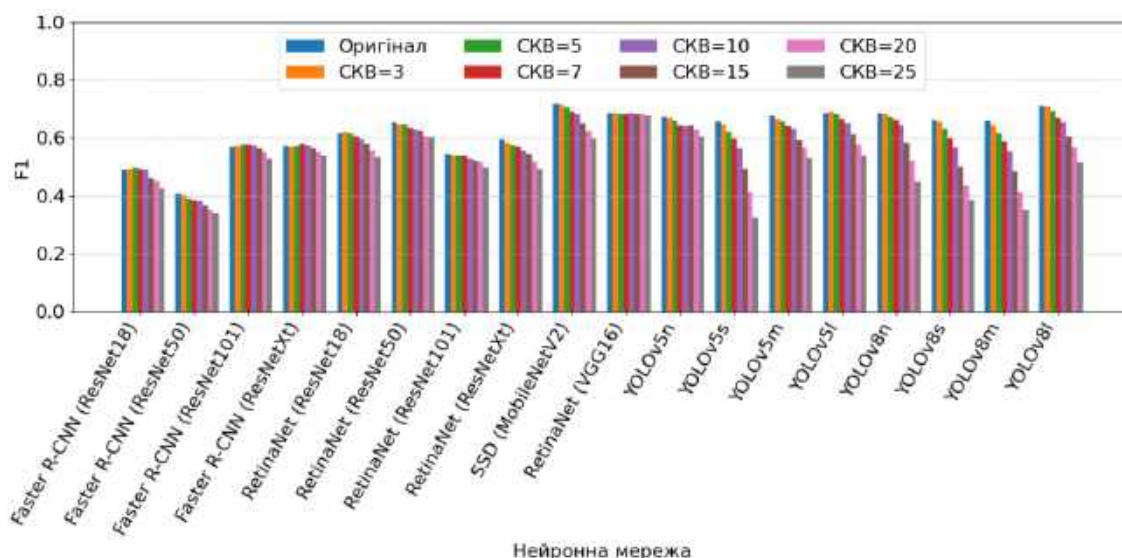


Рисунок 4.6 – Стопчикова діаграма метрики якості класифікації (F1) в залежності від інтенсивності шуму

Аналізуючи отримані результати загалом, можна зробити деякі проміжні висновки. По-перше, якість роботи нейронних мереж в умовах спотворення шумовими завадами є досить прогнозованою. Так, зазвичай маємо сильний спад метрики класифікації та кількості прогнозованих регіонів при $СКВ \geq 10$. Це може означати досить низьку стійкість нейронних мереж до впливу завад, а тому є потреба в попередній обробці даних для зменшення впливу спотворень. Також можна зауважити, що метрика локалізації є досить сталою для всіх значень СКВ, але це здебільшого пов'язано з розрахунком метрики, яка не враховує регіони, що не були прогнозовані. Враховуючи отримані результати, можна припустити, що попередня обробка є важливим етапом для задач локалізації та класифікації об'єктів на зображеннях, що спотворені шумом.

4.1.4 Метод підвищення точності локалізації та класифікації в умовах шуму

Одним із способів попередньої обробки є придушення завад. Для цього використаємо класичний метод зменшення шуму на зображенні, такий як BM3D [110]. Послідовність операцій методу представлено на рисунку 4.7.

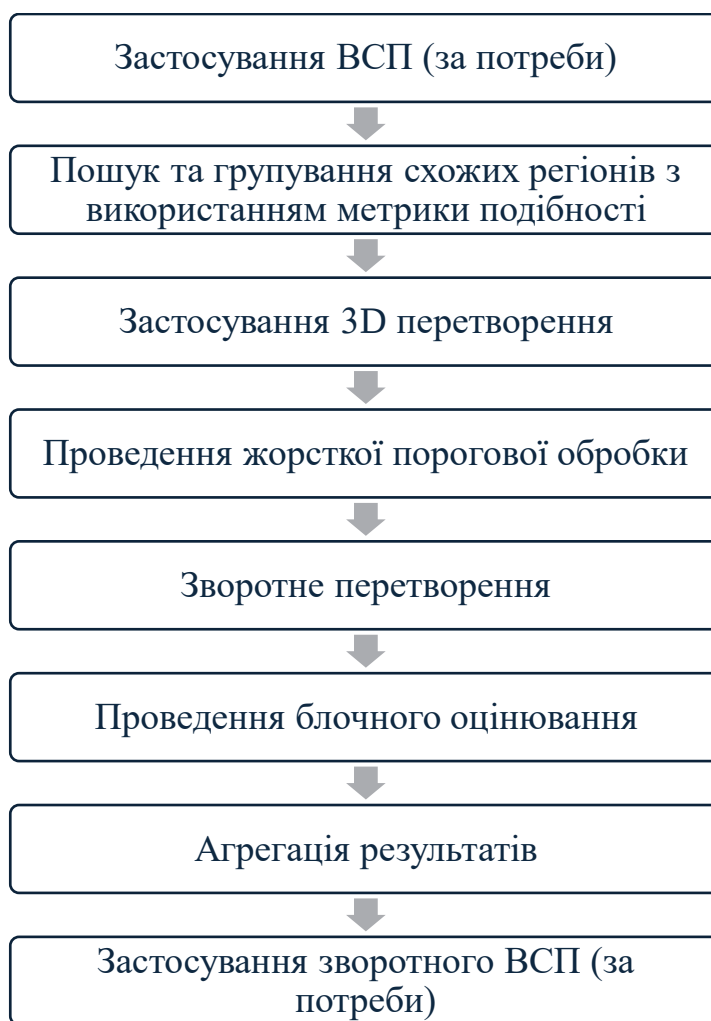


Рисунок 4.7 – Послідовність операцій в методі придушення шуму BM3D

Робота методу BM3D [118] базується на пошуку схожих блоків на зображеннях, подальшому їх групуванні та застосуванні фільтрації для тривимірних груп подібних патчів (блоків). Метод фільтрації може бути трохи різним, але зазвичай використовують фільтр Вінера та ДКП. Такий метод дозволяє застосовувати різну силу фільтрації для регіонів різної текстурованості та локальної активності. Це дає змогу краще зберігати деталі на зображеннях та, відповідно, залишати більше інформації для подальшої локалізації та класифікації. Зазначимо також, що існують варіанти BM3D для сигнально-залежних завад [131], а також для просторово-корельованих завад (де модифікації внесені як в алгоритм пошуку подібних блоків, так і в розрахунок порогів, які є адаптованими до просторового спектру завад [131]). Алгоритм методу локалізації та класифікації з використанням методу придушення шуму

представлено на рисунку 4.8, а програмна реалізація такого методу представлена в Додатку Г.

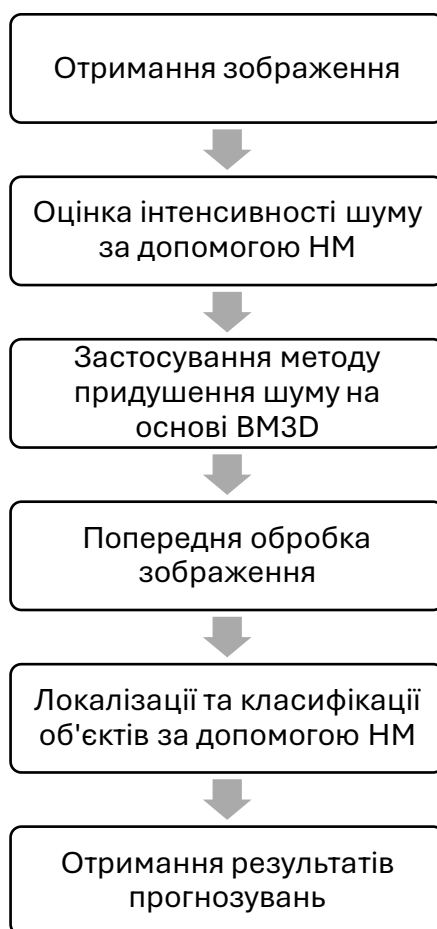


Рисунок 4.8 – Методу локалізації та класифікації з використанням алгоритму придушення шуму.

Для проведення дослідження всі зображення, що були спотворені шумом в розділі 4.1.2, було відфільтровано та отримано результат – зображення із придушеним шумом. Для отриманих зображень також було розраховано метрику PSNR, щоб мати можливість оцінити ефективність фільтрації. Отримані значення представлено в таблиці 4.4. Проаналізувавши отримані результати можна зробити висновок, що алгоритм фільтрації досить добре підвищує загальну якість зображень, а тому отримані зображення можуть бути використані для подальшого дослідження. Чим нижче якість первинного зображення з шумом, тим більш високою є ефективність фільтрації у сенсі покращення PSNR.

Таблиця 4.4 Значення метрики PSNR, (дБ) для різних значень СКВ усереднені по всіх зображеннях

СКВ	3	5	7	10	15	20	25
PSNR до фільтрації, дБ	42,52	36,48	33,61	30,69	27,41	25,07	23,24
PSNR після фільтрації, дБ	43,27	40,35	39,21	38,16	36,50	35,17	34,12

Метод локалізації та класифікації об'єктів з використанням алгоритму фільтрації можна представити в вигляді блок-схеми яку зображено на рисунку 4.9. Його робота полягає в попередній обробці зображення, а саме в застосування методу фільтрації BM3D. Отримане зображення, на якому видалено шум передається в нейронну мережу для прогнозування регіонів та їх класифікації. Отримані результати використовуються в подальшій обробці.

Важливо також дослідити ефективність роботи такого методу в задачах локалізації та класифікації. Для цього було проведено прогнозування регіонів з використанням алгоритму фільтрації використовуючи метод який було описано раніше. Для всіх отриманих даних було проведено розрахунок відсотку прогнозованих регіонів, метрики локалізації IoU [48] та класифікації F1 [64]. На рисунку 4.10 представлено стопчикову діаграму для відсотку прогнозованих регіонів. Аналізуючи отримані результати, є помітним, що метрика є досить стабільною на всіх значеннях СКВ завад та для всіх нейронних мереж. Спостерігаються деякі варіації в значеннях метрики, але вони не більше, ніж 0,02. Також варто зауважити, що для нейронних мереж YOLO значення метрик залишаються стабільними, на відміну від даних для випадку впливу шуму (див. попередній підрозділ). Це є досить хорошим показником, який визначає доцільність фільтрації, як методу попередньої обробки даних.

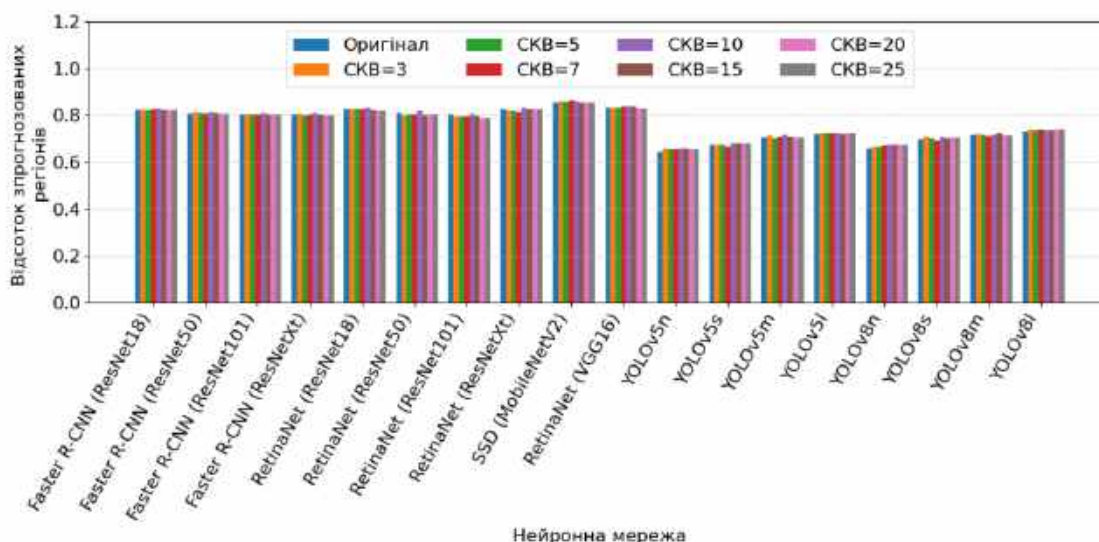


Рисунок 4.9 – Стопчикова діаграма відсотку прогнозованих регіонів відфільтрованих зображень відносно оригінального використовуючи метод з використання фільтра

На рисунку 4.10 представлено стопчикову діаграму значень метрики локалізації IoU для всіх значень інтенсивності шуму. При аналізі отриманих даних помітна також стабільність значень, що може означати слабку залежність точності роботи НМ від інтенсивності шуму. Варто зауважити, що якість локалізації є близькою до того ж рівня, що і для даних з шумом (рис. 4.5), тож для IoU вплив фільтрації є невеликим.

Аналогічно до попередніх досліджень було пораховано значення F1. Отримані результати представлено на рисунку 4.11 у вигляді стопчикової діаграми для всіх значень СКВ. Отримані значення є стабільними, що відображає практично однакову точність нейронних мереж для всього діапазону досліджуваних СКВ. За отриманими даними можна також зробити висновок про доцільність фільтрації, принаймні якщо $СКВ > 5$ (зазначимо, що СКВ чи дисперсія завад можуть бути досить точно автоматично оцінені [126, 127], про що було згадано в підрозділі 4.1.1). Так, результати є більш стабільними та мають дещо більші значення, ніж значення метрик до фільтрації. Також етап попередньої обробки даних в вигляді фільтрації позитивно впливає на якість класифікації за допомогою нейронної мережі YOLO, яка дає суттєво кращий результат для

великих інтенсивностей завад. Загалом всі нейронні мережі дають більші (кращі) значення метрик, особливо на зображеннях, що мали більше значення СКВ до фільтрації. Враховуючи отриманий результат, можна зробити висновок, що попереднє усунення шуму на зображеннях збільшує точності класифікації об'єктів за допомогою нейронних мереж.

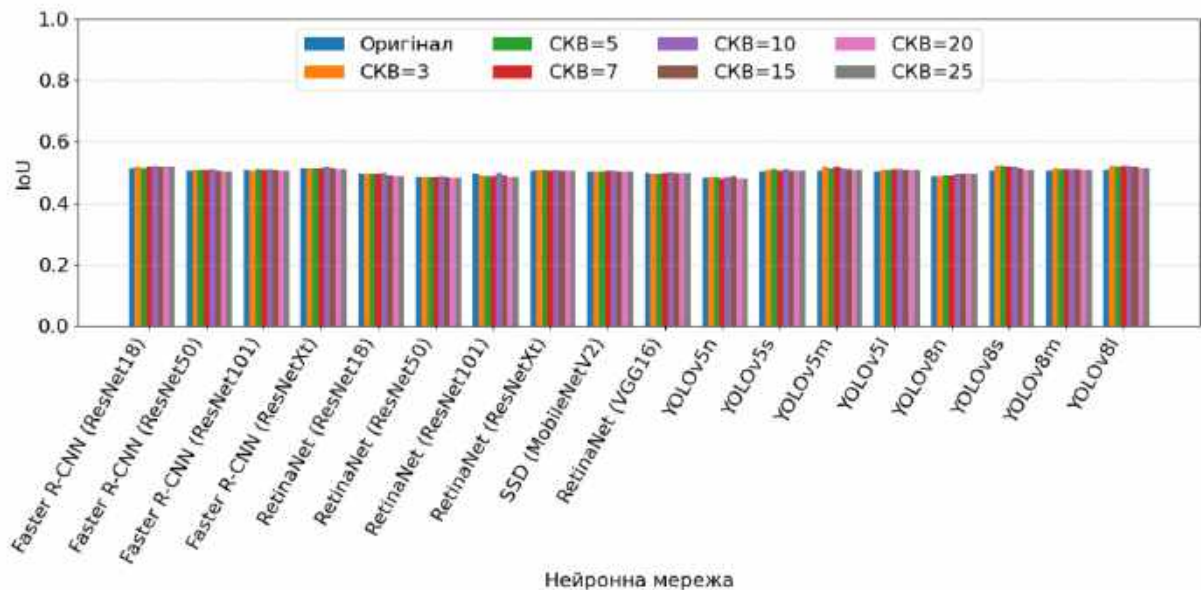


Рисунок 4.10 – Стопчикова діаграма метрики IoU відфільтрованих зображень відносно оригінального використовуючи метод з використання фільтру

Враховуючи отримані дані, а саме покращення метрик після фільтрації, можна зробити висновок, що попередня фільтрація є доцільною та корисною для задач локалізації та класифікації за допомогою нейронних мереж (ці результати узгоджуються з результатами нашої роботи [132], де аналізується доцільність використання фільтрації для підвищення візуальної якості зображень). Отримані результати показують збільшення метрик класифікації та кількості локалізованих регіонів, що є основними показниками якості роботи нейронної мережі. Особливо позитивно застосування попередньої фільтрації впливає на методи що використовують нейронну мережу YOLO, підвищуючи якість прогнозованих результатів.

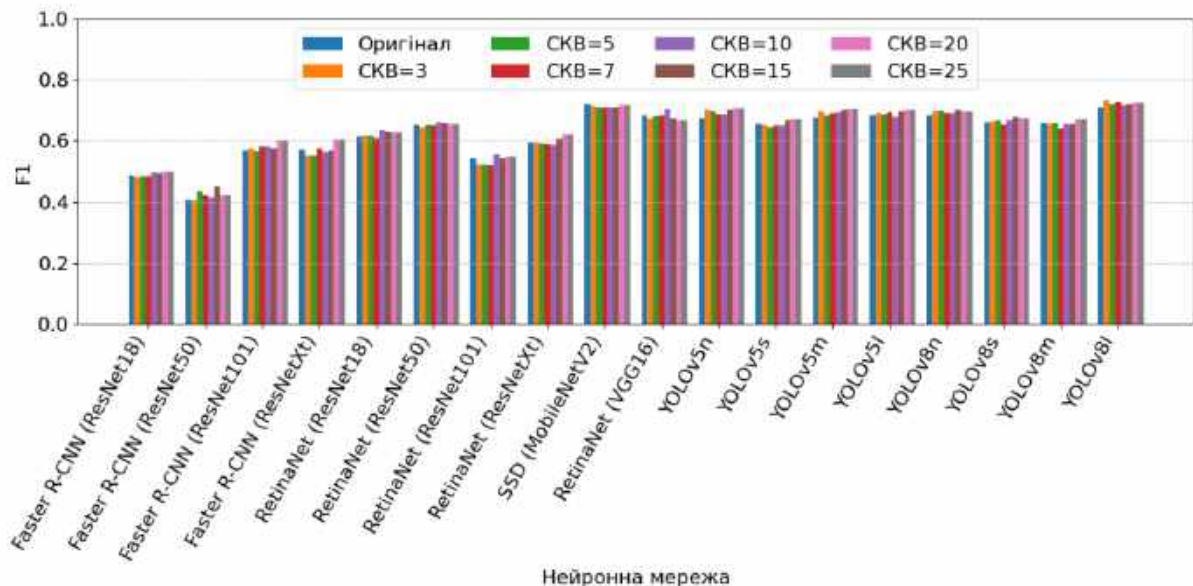


Рисунок 4.11 – Стопчикова діаграма для метрики F1 відфільтрованих зображень відносно оригінального використовуючи метод з використання фільтру

4.2 Дослідження роботи нейронних мереж в поганих погодних умовах

Отримання даних з безпілотних літальних апаратів не завжди є простою задачею. Часто на якість зображення впливають такі фактори, як туман чи дощ. В залежності від інтенсивності впливу таких завад можемо мати різні показники локалізації та класифікації. Залежність від цих факторів може бути високою, адже туман може маскувати об'єкти, що унеможлиблює їх локалізацію на деяких висотах чи при його високій щільності. Тому дослідження впливу цих явищ на якість локалізації та класифікації є доцільним та має практичну потребу.

4.2.1 Моделювання природніх погодних умов для зображень

Для дослідження якості роботи нейронної мережі в умовах туману та дощу є необхідною генерація відповідного набору даних. Такі дані повинні відповідати декільком умовам. Перше – це реалістичність даних, що дозволить оцінити вплив природніх факторів на якість локалізації та класифікації. Друге – це варіативність

даних, яка досягається великою кількістю зображень та широким діапазоном щільності туману та інтенсивності дощу.

Генерація такого типу даних розподілена на 2 частини. Перша частина присвячена моделюванню туману на зображеннях, друга частина – моделюванню дощу різної інтенсивності. Для моделювання туману було використано метод додавання туману на зображення, що використовує карту глибини [133]. Для цього було застосовано алгоритм, що зображений на рисунку 4.12.



Рисунок 4.12 – Алгоритм додавання туману на зображення

Розберемо алгоритм додавання туману детальніше. Першим кроком алгоритму є генерації карти глибини за допомогою одного зображення. Для цього було використано нейронну мережу DepthAnything [134], яка має відкритий вихідний код та дозволяє отримувати карту глибини високої якості за допомогою одного зображення. Наступним етапом є додавання туману до фото, для цього було згенеровано параметр щільності, що має діапазон від 0 до 1. Цей параметр дозволяє регулювати щільність туману шляхом зміни прозорості масиву з туманом, що додається. Результатом є зображення, що візуально спотворене туманом різної інтенсивності, варіативність якої досягається за допомогою генерації параметру щільності. Отримані результати представлено на рисунку 4.13 в порівнянні з оригінальним зображенням. Помітно, що зображення, яке спотворене туманом, має білий відтінок та розмиття об'єктів невеликого розміру.

Перейдемо до генерації даних, що моделюють вплив дощу. Для цього було використано також алгоритм оцінки глибини для зображень та подальше додавання дощу на зображення з поправкою на дальність від камери [135]. В

якості параметрів, які можуть регулювати інтенсивність впливу дощу використано параметр величини крапель, який виражається в ширині полоси чи діаметру точки на зображенні. Також в якості параметру було використано витримку, в залежності від величини якої можуть бути різні прояви дощу на фото. Так, при низькій витримці дощ має форму краплі, при великих значеннях витримки маємо довгі полоси, що відображають падіння краплі. Ці параметри дозволяють зробити вибірку більш варіативною та відповідно отримати більш реалістичний результат. Отримані зображення представлено на рисунку 4.14 порівняно з оригіналом.



Рисунок 4.13 – Результат зображень, що спотворені туманом (праворуч) різної інтенсивності, порівнюючи з оригіналом (ліворуч)

Аналізуючи отримані дані помітно, що при зйомці згори-вниз маємо низький вплив дощу на зображення, який може зменшити точність локалізації та класифікації, тільки перекривши об'єкт. На зображеннях, що зняті під кутом до земної поверхні, вплив дощу є сильнішим, адже він створює шлейф проходження краплі, який має різну довжину, яка залежить від витримки. Отримані дані є

досить варіативними та достатніми для проведення досліджень, присвячених точності локалізації та класифікації на зображеннях. Можна зробити припущення, що вплив туману буде сильнішим ніж вплив дощу, адже зображення є більш візуально деградованими.



Рисунок 4.14 – Результат моделювання зображень, що піддані впливу дощу (праворуч) різної інтенсивності порівнюючи з оригіналом (ліворуч)

4.2.2 Оцінка точності локалізації та класифікації за впливу природніх умов

Оцінку впливу природніх умов на локалізацію та класифікацію об'єктів на зображенні буде проведено таким же чином, як і для попередніх досліджень. Тобто для зображень, що були згенеровані в розділі 4.2.1, було виконано прогнозування за допомогою попередньо навчених нейронних мереж. Відповідно були розраховані метрики, які відображають якість локалізації та класифікації. Оцінку отриманих результатів можна робити відносно даних, що були отримані для цих же даних в розділі 3.2.

Перейдемо до відповідного аналізу даних, що були отримані на зображеннях, що спотворені туманом. Для таких даних результат представлено в вигляді стопчикової діаграми на рисунку 4.15 для відсотку прогнозованих регіонів. З її аналізу та порівняння з даними на рисунку 3.5 можна зробити висновки, що кількість регіонів, прогнозованих нейронною мережею знизилась. Але це зниження не є критичним, в деяких випадках становить близько 4-5% від загальної кількості анованих регіонів на зображеннях. Найбільше зниження спостерігається для нейронних мереж YOLO, що відображає низьку стійкість цих нейронних мереж до впливу завад в вигляді туману.

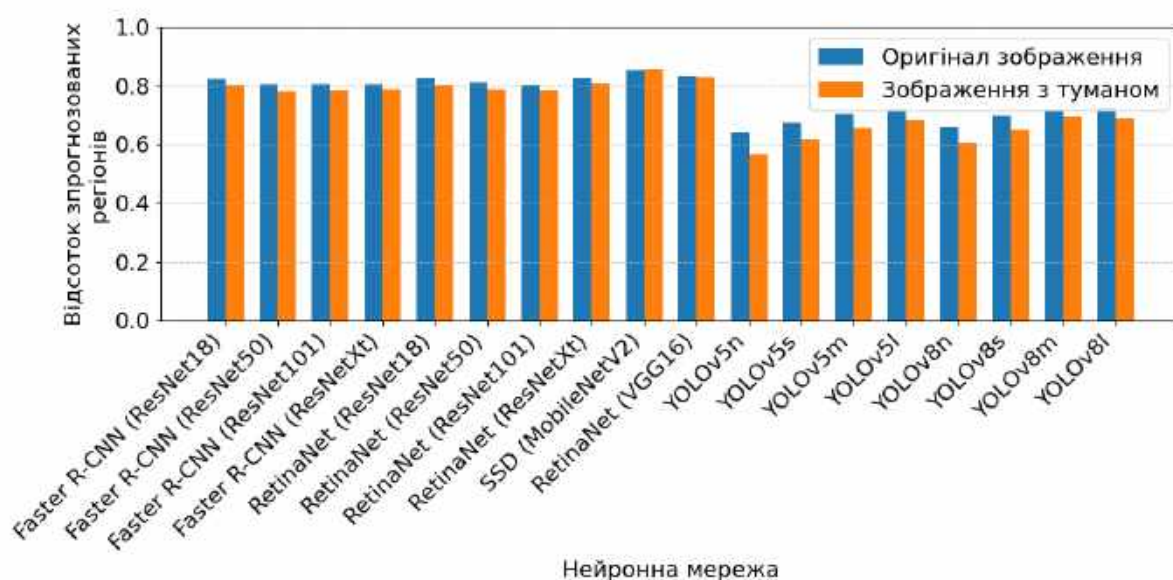


Рисунок 4.15 – Стопчикова діаграма для відсотку прогнозованих регіонів на зображеннях, що моделюють вплив туману

Результати для метрики IoU представлено на рисунку 4.16. Порівняємо отримані результати з результатами, що були розраховані для оригінальних зображень та представлені на рисунку 3.6. Отримані результати показують близькі значення до тих, що були розраховані раніше, а тому можна стверджувати що вплив туману на якість локалізації не є критичними та нейронні мережі добре локалізують об'єкти на таких зображеннях. За допомогою аналізу отриманих

даних можна стверджувати, що вплив туману на якість локалізації зазвичай є невеликим (окрім випадків густого туману).

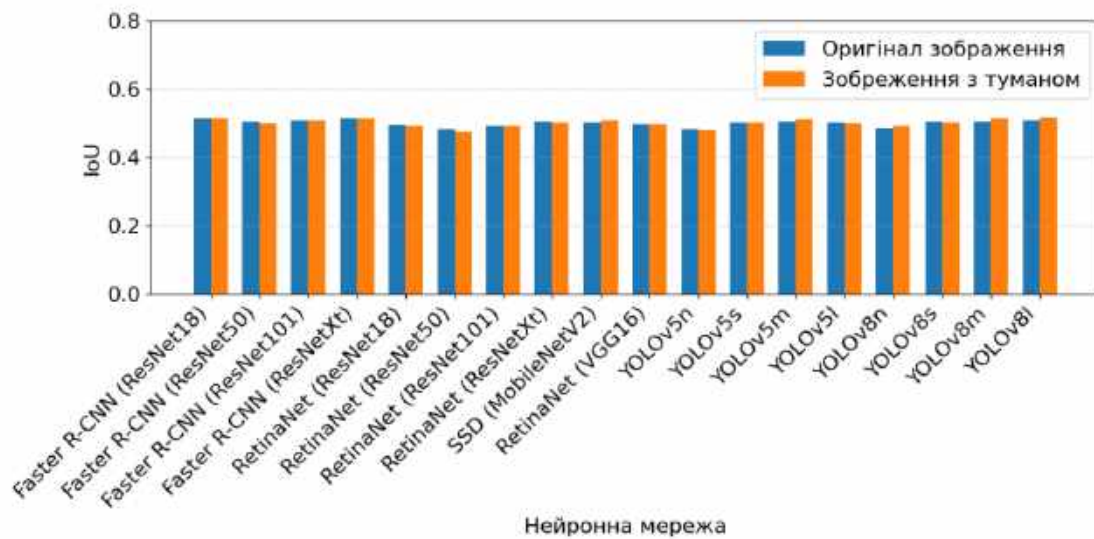


Рисунок 4.16 – Стопчикова діаграма для метрики IoU на зображеннях, що моделюють вплив туману

Отримані результати для метрики F1 представлена на рисунку 4.17. При порівнянні з даними на рисунку 3.7 є помітним невелике падіння метрики. Цей фактор може вказувати на недостатність інформації для безпомилкової класифікації об'єктів, адже при додаванні туману на зображенні змінились характеристики об'єкту. Така зміна може виражатись в спотворенні меж об'єкту чи його кольору, а також частковому чи повному перекритті об'єкту. Відповідно до отриманих результатів можна зробити висновок, що туман загалом, як і очікувалось, впливає на якість класифікації об'єктів на зображеннях, але вплив не є суттєвим (за виключенням спеціальних випадків дуже густого туману).

Проведемо дослідження впливу дощу на якість локалізації та класифікації об'єктів на зображеннях. Для цього використаємо дані, що були згенеровані в розділі 4.2.1. Для всіх зображень порахуємо відповідні метрики, а саме відсоток прогнозованих регіонів, точність локалізації IoU та точність класифікації F1. Отримані дані для наглядності також представимо в вигляді стопчикових діаграм та порівняємо з результатами, що отримані в розділі 3.2.

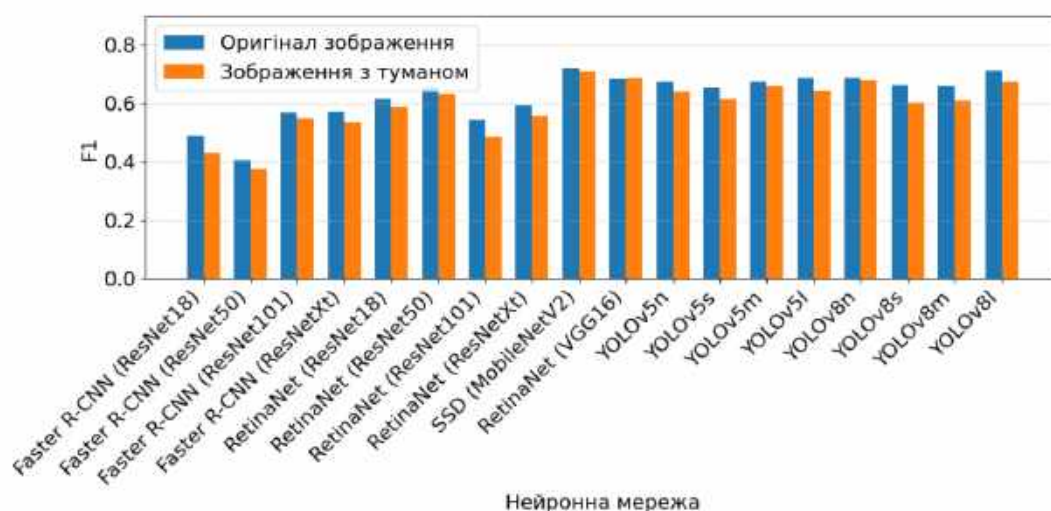


Рисунок 4.17 – Стопчикова діаграма метрики F1 на зображеннях, для яких змодельовано вплив туману

Для початку дослідимо відсоток прогнозованих регіонів. Отримана стопчикова діаграма представлена на рисунку 4.18. Аналізуючи отримані результати та порівнюючи їх з результатами на рисунку 3.5, можна помітити невелике погіршення, яке складає близько 1%. Також порівнявши отриманий результат з результатом, що був розрахований для впливу туману, відзначимо помітно більші значення показників при впливі дощу. Враховуючи отримані дані, можна зробити висновок, що негативний вплив дощу на роботу нейронних мереж є дуже малим та дозволяє локалізувати велику кількість об'єктів. Це дає змогу використовувати нейронні мережі без попередньої обробки зображення.

Проаналізуємо також точність локалізації на даних, що моделюють вплив дощу. Для цього розрахуємо метрику IoU, що представлено на рисунку 4.19 для всіх нейронних мереж. Порівняємо отримані значення із значеннями метрики, що отримані для зображень без завад, яку представлено на рисунку 3.6. Отримані дані є співрозмірними, з незначними відхиленнями, що можна віднести до статистичних відхилень. Також співставивши дані, що отримані при спотворенні зображень туманом, помітно, що результат є схожим. Враховуючи ці фактори, можна визначити, що дощ майже не впливає на якість локалізації об'єктів.

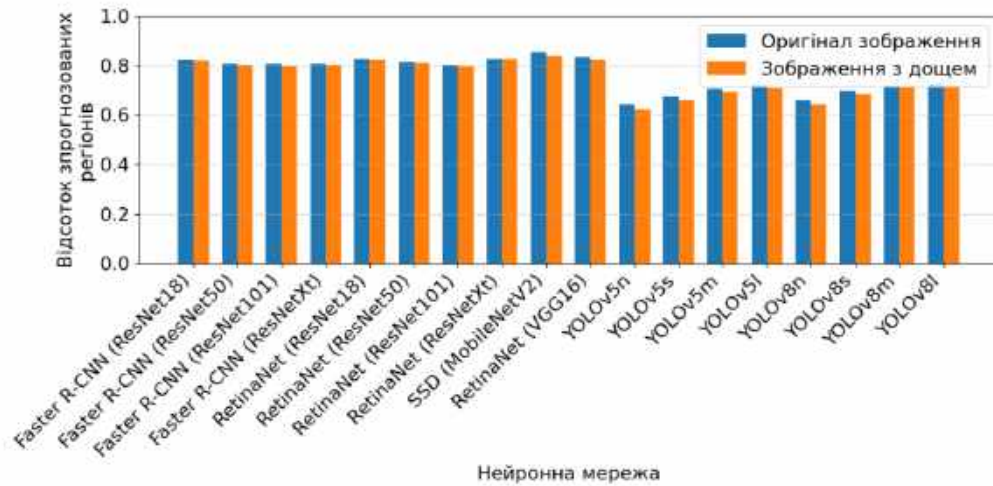


Рисунок 4.18 – Стопчикова діаграма для відсотку прогнозованих регіонів на зображеннях, для яких змодельовано вплив дощу

Проведемо аналіз метрики точності класифікації F1 для всіх нейронних мереж на зображеннях, де змодельовано вплив дощу. Для цього побудовано стопчикову діаграму значень метрики, яку зображено на рисунку 4.20. Аналізуючи отримані результати та співставивши їх з результатами, що зображені на рисунку 3.7, помітно, що для більшості нейронних мереж значення метрики залишається на тому ж рівні. Відповідно до цих результатів можна зробити висновок, що дощ мало впливає на якість класифікації об'єктів за допомогою нейронних мереж.

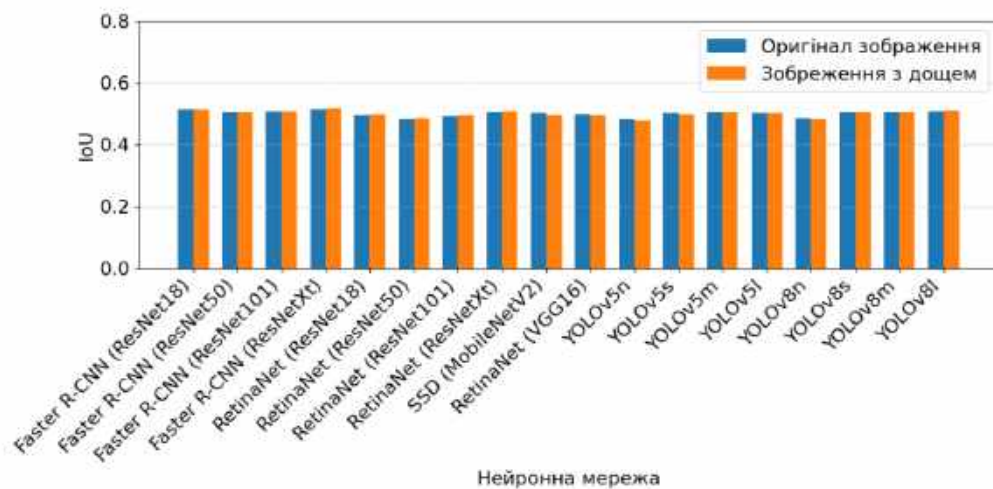


Рисунок 4.19 – Стопчикова діаграма для метрики IoU на зображеннях, для яких змодельовано вплив дощу

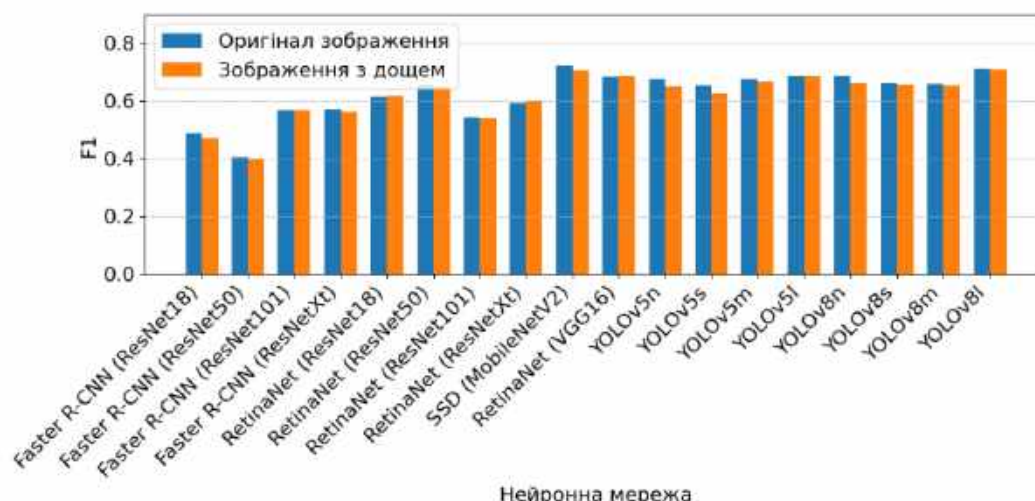


Рисунок 4.20 – Стопчикова діаграма для метрики F1 на зображеннях, для яких змодельовано вплив дощу

Враховуючи отримані дані та провівши оцінку результатів, можна зробити загальні висновки щодо впливу природних явищ на якість локалізації та класифікації за допомогою нейронних мереж. Отримані результати показують, що вплив дощу на якість локалізації та класифікації зазвичай не є критичним та характеризується відносно невеликою різницею. Така різниця може виникнути через декілька факторів, одним з яких може бути внесені спотворення в процесі моделювання зображень. Тому можна стверджувати, що зображення з дощем, як правило, не потребують попередньої обробки.

Проаналізувавши вплив туману на якість роботи нейронних мереж помітно, що дещо втрачається якість класифікації та кількість прогнозованих регіонів. Ці метрики співзалежні. Втрата приблизно 6% регіонів при прогнозуванні може бути в деяких випадках критичною, тому доцільними є дослідження ефективності методів усунення впливу туману на зображеннях.

4.2.3 Дослідження ефективності методу видалення туману

Для підвищення якості роботи нейронних мереж проведено дослідження пропонується використовувати метод покращення якості зображення шляхом

видалення туману. Алгоритм методу представлено на рисунку 4.21. Для видалення туману застосовано метод, який базується на алгоритмі «Dark Channel Prior» (DCP) [136]. Цей метод досить часто використовується на практиці, програмна реалізація методу представлена в Додатку Д. Для усунення туману за допомогою даного методу було оброблено всі зображення та отримано результати вивлення та розпізнавання.

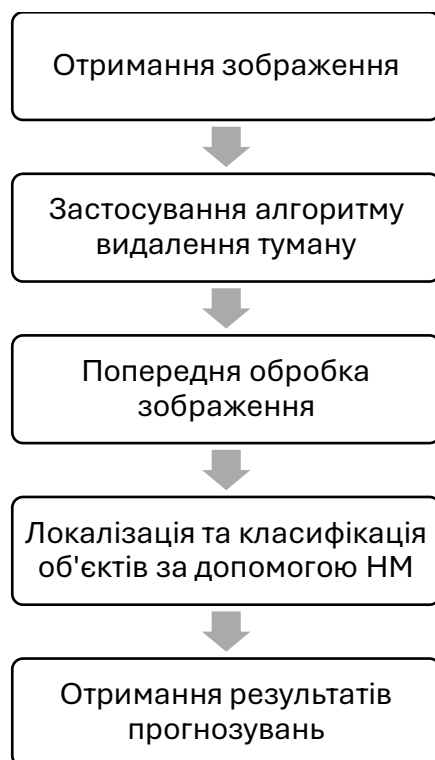


Рисунок 4.21 – Блок-схема методу локалізації та класифікації з використанням методу видалення туману

Детальний алгоритм роботи методу видалення туману представлено на рисунку 4.22. На першому етапі на вхідному зображенні проводиться оцінка каналу яскравості за допомогою визначення характеристик кожного пікселя. Наступним етапом є визначення оцінки атмосферного світла на карті, що отримана за допомогою оцінки каналу яскравості. Наступний етап полягає в оцінці інтенсивності світла, що доходить до камери використовуючи карту каналу яскравості. Отримані карти враховуються при обробці зображення. Загалом метод збільшує інтенсивність каналі яскравості, що в свою чергу

зменшує вплив туману на зображення. Приклади оброблених зображень представлено на рисунку 4.23. Візуально помітно, що результат є більш яскравим та чітким, ніж зображення з впливом туману (рис. 4.13), що, як ми сподіваємось, дасть змогу краще оцінити об'єкти.

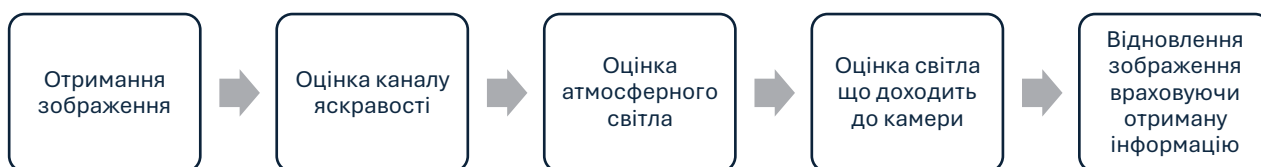


Рисунок 4.22 – Алгоритм методу видалення туману



Рисунок 4.23 – Результат зображень, що відновлені від впливу туману (праворуч) порівнюючи з оригіналом (ліворуч)

Для дослідження, чи підвищив метод усунення туману якість роботи нейронних мереж, проведемо дослідження аналогічні до попередніх, а саме розрахуємо відсоток прогнозованих регіонів, IoU та F1. Отримані результати порівняємо з даними, що були отримані для зображень з туманом в підрозділі 4.2.2. Почнемо з відсотку прогнозованих регіонів, результат представлено на рисунку 4.24. Аналізуючи отримані дані та порівнюючи з результатом що був отриманий для зображення з туманом, помітно, що дані мають близькі значення. Для деяких нейронних мереж, а саме Faster RCNN та RetinaNet метрика незначно зросла, на 0,01, але це можна вважати некритичним покращенням. Також варто зауважити, що для нейронних мереж YOLO в декількох випадках спостерігається погіршення метрики, що відображає вплив високого контрасту на якість роботи

нейронних мереж. Отримані дані показують, що критичного приросту якості роботи не виникає, а в деяких випадках навіть виникають погіршення якості роботи нейронних мереж.

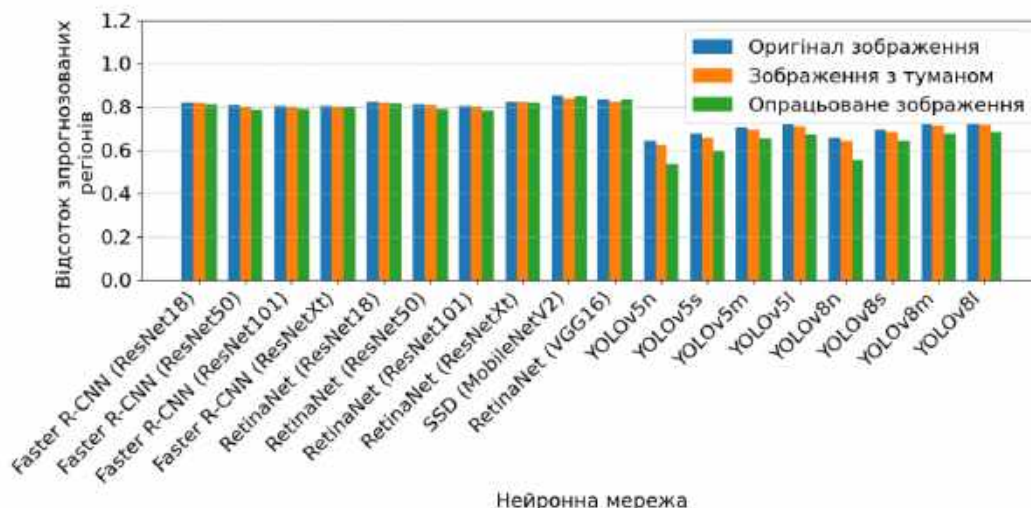


Рисунок 4.24 – Стопчикова діаграма для відсотку прогнозованих регіонів на зображеннях що відновленні після впливу туману

Дослідимо також значення метрики якості локалізації IoU, стопчикова діаграма якої представлена на рисунку 4.25. Для більшості нейронних мереж значення метрик співпадають із значеннями, які розраховані для нейронних мереж при спотворенні зображень туманом. Для Faster RCNN маємо підвищення на 0,01, для YOLOv8n зниження метрики на 0,02. Це підтверджує гіпотезу про негативний вплив переко́нтрастування на якість роботи нейронних мереж YOLO.

Проведемо дослідження показників метрики F1. На рисунку 4.26 представлено стопчикову діаграму значень метрики для кожної з нейронних мереж. Співставивши отримані дані з результатами, що отримані для зображень з туманом, стопчикова діаграма яких представлена на рисунку 4.13, можемо зробити висновок. Для всіх нейронних мереж метрика якості класифікації є нижчою, ніж в умовах туману (тобто без корекції) на 0,01-0,03. Тож виконання відновлення, принаймні для розглянутого метода, не має сенсу.

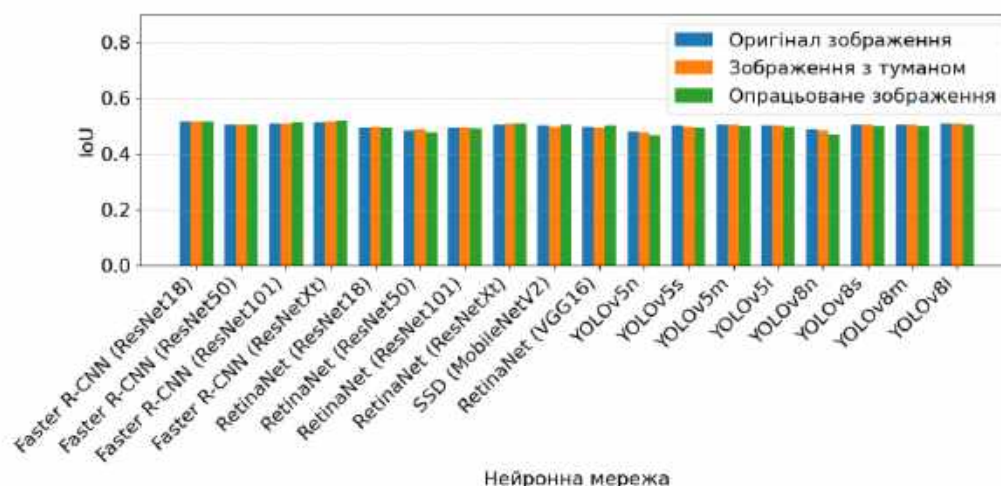


Рисунок 4.25 – Стопчикова діаграма для метрики IoU на зображеннях що відновленні після впливу туману

Тож при використанні методу DCP характеристики нейронних мереж не покращуються, а в більшості випадків є навіть гіршими, ніж для зображень, що спотворені туманом. На даному етапі досліджень не є можливим визначення впливу інших методів видалення туману на точність локалізації та класифікації. Можливо, цей метод не є найкращим, а тому вносить додаткові спотворення які знижують точність роботи методів локалізації та класифікації.

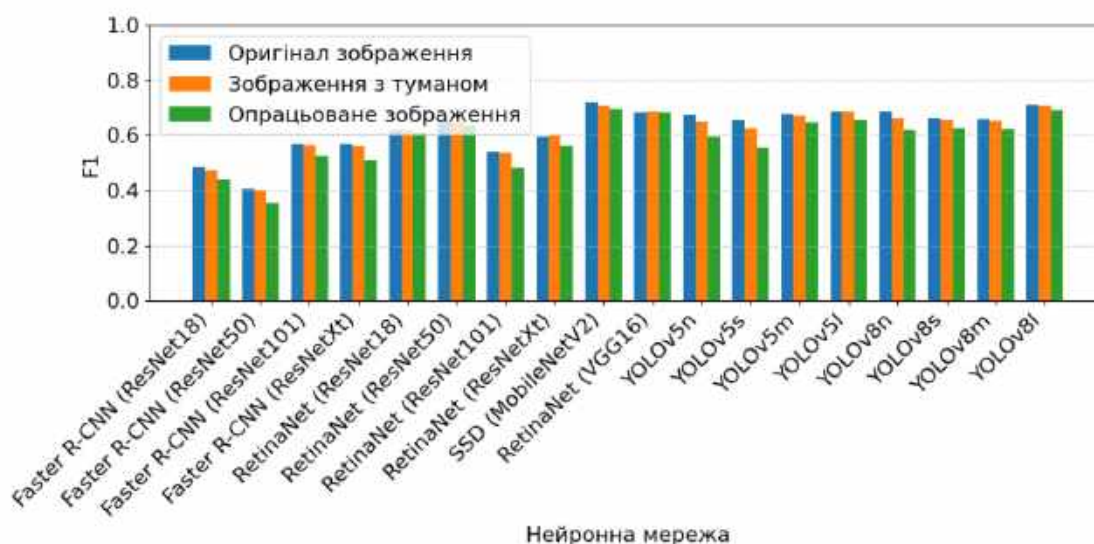


Рисунок 4.26 – Стопчикова діаграма метрики F1 на зображеннях що відновленні після впливу туману

4.3 Висновки до розділу

У розділі було досліджено різні фактори впливу завад та спотворень на зображення. Було розглянуто шум, а також туман та дощ. Для кожної з цих категорій було проведено детальне дослідження впливу на якість зображення та відповідно на точність локалізації та класифікації об'єктів на зображеннях. На основі отриманих результатів можна зробити висновки.

1. Досліджено тип та інтенсивність шуму на зображеннях що отримані за допомогою смартфонів. Отримані результати показують, що шум є просторово корельованим, але з невисоким ступенем кореляції, який залежить від умов зйомки та освітлення. Також визначено що основна частина розподілу СКВ завад лежить в діапазоні 2-7, але враховуючи що алгоритми стиснення можуть вносити додаткові завади варто розглядати більші значення СКВ в дослідженнях.

2. Отримані результати показують, що спад кількості прогнозованих регіонів із збільшенням СКВ завад є досить високим для нейронних мереж YOLO та потребує усунення шуму для підвищення точності локалізації та класифікації. Варто зауважити, що для нейронних мереж Faster R-CNN та RetinaNet спад кількості прогнозованих регіонів є досить низьким, в діапазоні 0,03-0,05. Але при значенні СКВ 25 спад показників для нейронних мереж YOLO досягає 0,25 відносно початкових даних. Цей показник є критичним та унеможливорює використання таких методів в задачах локалізації та класифікації об'єктів при інтенсивних завадах. Результати для точності локалізації показують некритичний рівень спаду для всіх нейронних мереж, в діапазоні 0,01-0,03 одиниці, що є допустимим. Результати метрик класифікації також знижуються при збільшенні інтенсивності шуму. Так для нейронних мереж YOLO показник спаду метрики точності класифікації досягає значення 0,3, що є занадто високим. Для нейронних мереж Faster R-CNN, Retina Net та SSD цей показник є дещо меншим та досягає значення 0,15, але це є також високим значенням.

3. Враховуючи отримані результати було запропоновано модифікацію методу локалізації та класифікації з використанням методу придушення шуму на

зображенні. Це дало досить хороший результат - отримано більш стабільні значення кількості прогнозованих регіонів, що мають значення близькі до зображень без завад для всіх нейронних мереж. Метрики якості локалізації та класифікації також мають результат, який близький до результатів для зображень без завад, на всьому діапазоні досліджуваних СКВ.

4. Визначено, що туман впливає на якість класифікації та кількість прогнозованих регіонів, але точність локалізації залишається на рівні, близькому до оригінальних зображень. Доц має досить малий вплив на якість локалізації та класифікації та дозволяє отримувати значення метрик, що близькі до оригінальних зображень.

5. Запропоновано використання методу видалення туману та оцінено його вплив на точність локалізації та класифікації на зображеннях що спотворені туманом. Показано, що метод *dark channel prior*, який збільшує контраст зображення, для нейронних мереж YOLO навіть знижує точність прогнозувань. Враховуючи це, можна вважати недоцільним використання саме цього методу для підвищення точності в методах локалізації та класифікації.

Отримані результати та запропоновані методи відображені у публікаціях [127, 137] та конференційних матеріалах [126, 138, 139]

ВИСНОВКИ

У ході досліджень, що були виконані в дисертаційній роботі, проведено аналіз точності роботи відомих методів локалізації та класифікації, що використовують нейронні мережі. Нейронні мережі для досліджень були обрані таким чином щоб покривати великий діапазон обчислювального навантаження, а також такі що мають достатню точність локалізації та класифікації. Обрані нейронні мережі було навчено на наборі даних з БПЛА, використовуючи алгоритми донавчання з використанням попередньо натренованих вагів екстракторів ознак. Такий спосіб навчання нейронних мереж дозволяє швидше сходиться та використовувати вибірку, що менша за розміром від розрахованої. Всі навчені нейронні мережі мають достатню точність для використання їх в методах локалізації та класифікації.

Проведено аналіз точності роботи методів локалізації та класифікації при роботі з малорозмірними об'єктами. Визначено, що об'єкти розміром від 200 пікселів прогнозуються з достатньою точністю, для меншого розміру об'єкту необхідно застосовувати модифіковані методи. Використовуючи ці дані вдосконалено метод локалізації та класифікації використовуючи підхід обробки зображення частинами. Це дозволило підняти точність локалізації та класифікації для об'єктів малого розміру, зокрема підвищення відсотку кількості прогнозованих регіонів на 0,15 та якості локалізації на 0,2 в залежності від нейронної мережі.

Запропоновано також метод оцінки відстані до об'єкту маючи інформацію про його фактичні розміри в метрах та характеристики системи формування зображення. Інформацію, що була отримана при оцінці мінімальних характеристик розміру об'єкту було використано в виводі формули, а тому такий спосіб розрахунку має практичне підґрунтя. Використана інформація про дальність до об'єкту може бути використана в методі локалізації та класифікації як умова для використання алгоритму обробки зображення частинами.

Проаналізовано зображення, що отримані за допомогою камер смартфонів та визначено тип та інтенсивність шуму, що може бути присутнім на таких зображеннях. Дослідження проводилось для зображень в різних умовах та визначено що ступінь кореляції шуму є досить не високою та залежить від умов зйомки. За допомогою нейронної мережі NoiseNet було визначено середні значення інтенсивності (СКВ) завад на таких зображеннях, вони мають значення в діапазоні 2-7 залежно від умов зйомки. Враховуючи отримані дані було проведено дослідження точності роботи методів локалізації та класифікації на зображеннях з шумом, визначено, що для деяких нейронних мереж має місце спад відсотку кількості прогнозованих регіонів на 0,25 та метрики якості класифікації на 0,15.

На основі отриманих даних вдосконалено метод локалізації та класифікації шляхом використання методу придушення шуму BM3D та нейронної мережі для оцінки інтенсивності шуму NoiseNet. Використання такого методу дало змогу підвищити точність локалізації та класифікації до рівня зображення, що не має завад. Розроблено програмну реалізацію такого методу з використанням визначення інтенсивності завад та використання цього значення як умови в процесі локалізації та класифікації.

Проаналізовано вплив природніх факторів, зокрема туману та дощу, на точність локалізації та класифікації. Отримані результати показують, що вплив дощу є не суттєвим, з варіацією зміни метрик на 0,01. Вплив туману є дещо вищим, де спад якості класифікації становить в середньому 0,05. На основі отриманих даних вдосконалено та досліджено метод локалізації та класифікації з використанням методу видалення туману на зображенні. В якості методу видалення туману було використано метод Dark Channel Prior та розроблено його програмну реалізацію. Отримані результати не підвищили точність локалізації та класифікації а тому можна вважати використання саме цього методу не доцільним в умовах спотворення зображення туманом.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Bhuyan, M.K., *Computer Vision and Image Processing: Fundamentals and Applications*, CRC Press, 2019, 464 p.
2. Rosenfeld, A., *Multiresolution Image Processing and Analysis*, Springer Berlin, Heidelberg, 1984, 388 p.
3. Sezgin, A., Boyacı, A., “Advancements in Object Detection for Unmanned Aerial Vehicles: Applications, Challenges, and Future Perspectives,” *2024 12th International Symposium on Digital Forensics and Security (ISDFS)*, San Antonio, TX, USA, 2024, P. 1-6, DOI: 10.1109/ISDFS60797.2024.10527339.
4. Wu, Q., Zhou, Y., Real-Time Object Detection Based on Unmanned Aerial Vehicle, *2019 IEEE 8th Data Driven Control and Learning Systems Conference (DDCLS)*, Dali, China, 2019, P. 574-579, DOI: 10.1109/DDCLS.2019.8908984.
5. Cao, Z., Kooistra, L., Wang, W., Guo, L., Valente, J., “Real-Time Object Detection Based on UAV Remote Sensing: A Systematic Literature Review,” *Drones*, vol. 7, no. 10, paper no. 620, 2023. DOI: 10.3390/drones7100620.
6. Sahni, M., Sahni, R., Merigo, J., *Neural Networks, Machine Learning, and Image Processing: Mathematical Modeling and Applications*, CRC Press, 2022, 202 p.
7. Tekin, S., Gok, M., Namdar, M., Başgümüş, A., “Autonomous guidance system for UAVs with image processing techniques,” *Journal of scientific reports-A*, no. 51, P. 149–159, 2022.
8. Basso, M., Pignaton de Freitas, E., “A UAV Guidance System Using Crop Row Detection and Line Follower Algorithms,” *J Intell Robot Syst*, no. 97, P. 605–621, 2020. DOI: 10.1007/s10846-019-01006-0.
9. Surinta, O., Khruahong, S., “Tracking People and Objects with an Autonomous Unmanned Aerial Vehicle using Face and Color Detection,” *2019 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT-NCON)*, Nan, Thailand, 2019, P. 206-210, DOI: 10.1109/ECTI-NCON.2019.8692269.

10. Lo, L.-Y., Yiu, C. H., Tang, Y., Yang, A.-S., Li, B., Wen, C.-Y., “Dynamic Object Tracking on Autonomous UAV System for Surveillance Applications,” *Sensors*, no. 21, article no. 7888, 2021. DOI: 10.3390/s21237888.
11. Raj, E. F. I., Manimla, K., “Environmental Monitoring and Conservation Using Drones,” *Drone Applications for Industry 5.0*, P. 358-380, 2024. DOI: 10.4018/979-8-3693-2093-8.ch019.
12. Chen, H., Gao, X., Li, H., Yang, Z., “A framework for the optimal deployment of police drones based on street-level crime risk,” *Applied Geography*, vol. 162, paper no. 103178, 2024. DOI: 10.1016/j.apgeog.2023.103178.
13. Popov, M., Stankevich, S., Mosov, S., Dugin, S., Golubov, S., Andreiev, A., Lysenko, A., Saprykin, I., “Concept of a geoinformation platform for landmines and other explosive objects detection and mapping with UAV,” *Radioelectronic and Computer Systems*, no. 4, P. 207–216, 2024. DOI: 10.32620/reks.2024.4.17.
14. Yang, Z., Lan, X., Wang, H., “Comparative Analysis of YOLO Series Algorithms for UAV-Based Highway Distress Inspection: Performance and Application Insights,” *Sensors*, no. 25, paper no. 1475, 2025. DOI: 10.3390/s25051475.
15. Rodríguez-Rodríguez, J. A., López-Rubio, E., Ángel-Ruiz, J. A., Molina-Cabello, M. A., “The Impact of Noise and Brightness on Object Detection Methods,” *Sensors*, vol. 24, no. 3. MDPI AG, P. 821, 2024. DOI: 10.3390/s24030821.
16. Fang, W.; Zhang, G.; Zheng, Y.; Chen, Y., “Multi-Task Learning for UAV Aerial Object Detection in Foggy Weather Condition,” *Remote Sens.*, no. 15, paper no. 4617, 2023. DOI: 10.3390/rs15184617.
17. Luo, K., Luo R., Zhou, Y., “UAV detection based on rainy environment,” *2021 IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, Chongqing, China, 2021, P. 1207-1210, DOI: 10.1109/IMCEC51613.2021.9482383.
18. Aposporis, P., “Object Detection Methods for Improving UAV Autonomy and Remote Sensing Applications,” *2020 IEEE/ACM International Conference on*

Advances in Social Networks Analysis and Mining (ASONAM), The Hague, Netherlands, 2020, P. 845-853, DOI: 10.1109/ASONAM49781.2020.9381377.

19. Chen, X., Chen, T., Zhao, Z., Zhang, H., Bennis, M., Ji, Y., “Resource Awareness In Unmanned Aerial Vehicle-Assisted Mobile-Edge Computing Systems,” *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, Antwerp, Belgium, 2020, P. 1-6, DOI: 10.1109/VTC2020-Spring48590.2020.9128981.

20. Briley, A.A., Afghah, F., “Hardware Acceleration for Real-Time Wildfire Detection Onboard Drone Networks,” *IEEE INFOCOM 2024 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Vancouver, BC, Canada, 2024, P. 01-06, DOI: 10.1109/INFOCOMWKSHPS61880.2024.10620877.

21. Swaminathan, T. P., Silver, C., Akilan, T., “Benchmarking Deep Learning Models on NVIDIA Jetson Nano for Real-Time Systems: An Empirical Investigation,” *ArXiv*, P 1-7, 2024. DOI: 10.48550/arXiv.2406.17749.

22. İlhan, H. E., Ozer, S., Kurt, G. K., Ali Cirpan, H., “Offloading Deep Learning Empowered Image Segmentation from UAV to Edge Server,” *2021 44th International Conference on Telecommunications and Signal Processing (TSP)*, Brno, Czech Republic, 2021, P. 296-300, DOI: 10.1109/TSP52935.2021.9522611.

23. Srivastava, A., Rawat, B. S., Singh, G., Bhatnagar, V., Saini, P. K., Dhondiyal, S. A., “A Review of Optimization Algorithms for Training Neural Networks,” *2023 International Conference on Sustainable Emerging Innovations in Engineering and Technology (ICSEIET)*, Ghaziabad, India, 2023, P. 886-890, DOI: 10.1109/ICSEIET58677.2023.10303287.

24. Braunfelds, J., Jakovels, G., Murans, I., Litvinenko, A., Senkans, U., Rumba, R., Onzuls, A., Valters, G., Lidere, E., Plone, E., “Experimental Study on LTE Mobile Network Performance Parameters for Controlled Drone Flights,” *Sensors*, no. 24, paper no. 6615, 2024. DOI: <https://doi.org/10.3390/s24206615>.

25. Lucchesi, L., Endler, M., Souza, B. J. O. de., “Dynamic effects of communication delay, failure rates, and speed on UAV swarm formation,” *Simpósio brasileiro de redes de computadores e sistemas distribuídos (sbrc)*, Porto Alegre: Sociedade Brasileira de Computação, 2024, P. 588-601, DOI: 10.5753/sbrc.2024.1445.

26. Deng, X., Shi, M., Khan, B., “A lightweight CNN model for UAV-based image classification,” *Soft Comput*, no. 29, P. 2363–2378, 2025. DOI: 10.1007/s00500-025-10512-3.
27. Cabrera-Ponce, A.A., Martinez-Carranza, J., “Convolutional Neural Networks for Geo-Localisation with a Single Aerial Image,” *J Real-Time Image Proc*, no. 19, P. 565–575, 2022. DOI: 10.1007/s11554-022-01207-1.
28. Neha, F., Bhati, D., Shukla, D. K., Amiruzzaman, M., “From classical techniques to convolution-based models: A review of object detection algorithms,” arXiv, P. 1-6, 2024. DOI: 10.48550/arXiv.2412.05252.
29. Girshick, R., Donahue, J., Darrell N., Malik, J., “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation,” *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, USA, 2014, P. 580-587, DOI: 10.1109/CVPR.2014.81.
30. Girshick, R., “Fast R-CNN,” *2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, 2015, P. 1440-1448, DOI: 10.1109/ICCV.2015.169.
31. Ren, S., He, K., Girshick, R., Sun, J., “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, P. 1137-1149, 2015. DOI: 10.48550/arXiv.1506.01497.
32. Redmon, J., Divvala, S., Girshick, R., Farhadi, A., “You Only Look Once: Unified, Real-Time Object Detection,” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, P. 779-788, 2016, DOI: 10.1109/CVPR.2016.91.
33. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C., Berg, A. C., “SSD: Single Shot MultiBox Detector,” *Computer Vision - ECCV 2016*, vol. 9905, P. 21-37, 2016. DOI: 10.1007/978-3-319-46448-0_2.
34. Lin, T. Y., Goyal, P., Girshick, R., He, K., Dollár, P., “Focal Loss for Dense Object Detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, P. 318-327, 2020. DOI: 10.1109/TPAMI.2018.2858826.

35. Uijlings, J., van de Sande, K., Gevers, T., Smeulders, A., "Selective Search for Object Recognition," *International Journal of Computer Vision*, vol. 104, P. 154-171, 2013. DOI: 10.1007/s11263-013-0620-5.
36. Felzenszwalb, P., Huttenlocher, D., "Efficient Graph-Based Image Segmentation," *International Journal of Computer Vision*, vol. 59, P. 167-181, 2004. DOI: 10.1023/B:VISI.0000022288.19776.77.
37. Liu, H., Li, I., Liang, Y., Sun, D., Yang, Y., Yang, H., "Research on Deep Learning Model of Feature Extraction Based on Convolutional Neural Network," *2024 IEEE 2nd International Conference on Image Processing and Computer Applications (ICIPCA)*, Shenyang, China, P. 810-816, 2024, DOI: 10.1109/ICIPCA61593.2024.10709168.
38. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., "ImageNet: A Large-Scale Hierarchical Image Database," *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Miami, FL, USA, 2009, P. 248–255, DOI: 10.1109/CVPR.2009.5206848.
39. Krizhevsky, A., Sutskever, I., Hinton, G. E., "ImageNet Classification with Deep Convolutional Neural Networks," *Communications of the ACM*, vol. 60, P. 84–90, 2012. DOI: 10.1145/3065386.
40. He, K., Zhang, X., Ren, S., Sun, J., "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, P. 770–778. DOI: 10.1109/CVPR.2016.90.
41. Howard, A. G., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *arXiv*, 2017, P. 1-9. DOI: 10.48550/arXiv.1704.04861.
42. Liu, S., Deng, W., "Very Deep Convolutional Neural Network Based Image Classification Using Small Training Sample Size," *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, Kuala Lumpur, Malaysia, 2015, P. 730–734. DOI: 10.1109/ACPR.2015.7486599.
43. Elharrouss, O., Akbari, Y., Al-Madeed, N., Al-Maadeed, S., "Backbones-review: Feature Extractor Networks for Deep Learning and Deep Reinforcement

Learning Approaches in Computer Vision,” *Computer Science Review*, vol. 53, article no. 100645, 2024. DOI: 10.1016/j.cosrev.2024.100645.

44. Neubeck, A., Van Gool, L., “Efficient Non-Maximum Suppression,” *18th International Conference on Pattern Recognition (ICPR'06)*, Hong Kong, China, 2006, pp. 850–855. DOI: 10.1109/ICPR.2006.479.

45. Zhong, Y., Wang, J., Peng, J., Zhang, L., “Anchor Box Optimization for Object Detection,” *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Snowmass, CO, USA, 2020, pp. 1275–1283. DOI: 10.1109/WACV45572.2020.9093498.

46. Diwan, T., Anirudh, G., Tembhurne, J.V., “Object detection using YOLO: challenges, architectural successors, datasets and applications,” *Multimed Tools and Applications*, vol. 82, P. 9243–9275, 2023. DOI: 10.1007/s11042-022-13644-y.

47. Kang, C., Kim, S., “Real-time object detection and segmentation technology: an analysis of the YOLO algorithm,” *JMST Advanced*, vol. 5, P. 69-76, 2023. DOI 10.1007/s42791-023-00049-7.

48. Krishna, N. M., Reddy, R. Y., Reddy, M. S. C., Madhav, K. P., Sudham, G., “Object Detection and Tracking Using Yolo,” *2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA)*, Coimbatore, India, P. 1-7, 2021. DOI: 10.1109/ICIRCA51532.2021.9544598.

49. Zhou, D., et al., "IoU Loss for 2D/3D Object Detection," *2019 International Conference on 3D Vision (3DV)*, Quebec City, QC, Canada, P. 85-94, 2019. DOI: 10.1109/3DV.2019.00019.

50. Redmon, J., Farhadi, A., “YOLO9000: Better, Faster, Stronger,” *Computer Vision and Pattern Recognition*, P. 6517-6525, 2016. DOI: 10.1109/CVPR.2017.690

51. Szegedy, C., et al., "Going deeper with convolutions," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, P. 1-9, 2015. DOI: 10.1109/CVPR.2015.7298594.

52. Redmon, J., Farhadi, A., “YOLOv3: An Incremental Improvement,” *arXiv.org*, P. 1-6, 2018. DOI: 10.48550/arXiv.1804.02767.

53. Lin, T. -Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S., “Feature Pyramid Networks for Object Detection,” *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, P. 936-944, 2017. DOI: 10.1109/CVPR.2017.106.
54. Rezatofghi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S., “Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression,” *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, P. 658-666, 2019. DOI: 10.1109/CVPR.2019.00075.
55. Mao, A., Mohri, M., Zhong, Y., “Cross-Entropy Loss Functions: Theoretical Analysis and Applications,” *Proceedings of the 40th International Conference on Machine Learning*, vol. 202, P. 23803-23828, 2023. DOI: 10.48550/arXiv.2304.07288.
56. “YOLOv5: A state-of-the-art real-time object detection system”, Ultralytics, 2021. [Online]. Available: <https://docs.ultralytics.com>. [Accessed: 20- Oct- 2014].
57. Yaseen, M., “What is YOLOv8: An In-Depth Exploration of the Internal Features of the Next-Generation Object Detector,” *arXiv*, P. 1-10, 2024. DOI: 10.48550/arXiv.2408.15857.
58. Lin, T., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollar, P., Zitnick, C., “Microsoft COCO: Common Objects in Context,” *Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science*, vol. 8693, P. 740-755, 2014, DOI: 10.1007/978-3-319-10602-1_48.
59. Lui, T., Fang, S., Zhao, Y., Wang, P., Zhang, J., “Implementation of Training Convolutional Neural Networks,” *arXiv.org*, P. 1-10, 2015. DOI: 10.48550/arXiv.1506.01195.
60. Larochelle, H., Bengio, Y., Louradour, J., Lamblin, P., “Exploring Strategies for Training Deep Neural Networks,” *Journal of machine learning research*, vol. 10, P. 1-40, 2009. DOI: 10.5555/1577069.1577070.

61. Baldi, P., "Gradient descent learning algorithm overview: a general dynamical systems perspective," *IEEE Transactions on Neural Networks*, vol. 6, no. 1, P. 182-195, 1995. DOI: 10.1109/72.363438.
62. Qi, J., Du, J., Siniscalchi, S. M., Ma, X., Lee, C. -H., "On Mean Absolute Error for Deep Neural Network Based Vector-to-Vector Regression," *IEEE Signal Processing Letters*, vol. 27, P. 1485–1489, 2020. DOI: 10.1109/LSP.2020.3016837.
63. Meyer, G. P., "An Alternative Probabilistic Interpretation of the Huber Loss," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, P. 5257-5265, 2021, DOI: 10.1109/CVPR46437.2021.00522.
64. Badithela, A., Wongpiromsarn, T., Murray, R. M., "Evaluation Metrics of Object Detection for Quantitative System-Level Analysis of Safety-Critical Autonomous Systems," *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Detroit, MI, USA, P. 8651-8658, 2023. DOI: 10.1109/IROS55552.2023.10342465.
65. Padilla, R., Passos, W.L., Dias, T.L.B., Netto, S.L., da Silva, E.A.B., "A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit," *Electronics* 2021, vol. 10, no. 279, article no. 279, 2021. DOI: 10.3390/electronics10030279.
66. Ting, K., "Precision and Recall," *Encyclopedia of Machine Learning*, P 781. DOI: 0.1007/978-0-387-30164-8_652.
67. Hand, D.J., Christen, P., Kirielle, N., "F*: an interpretable transformation of the F-measure," *Mach Learn*, vol. 110, P. 451–456, 2021. DOI: 10.1007/s10994-021-05964-1.
68. Zaidi, S. S. A., Ansari, M. S., Aslam, A., Kanwal, N., Asghar, M., Lee, B., "A survey of modern deep learning based object detection models," *Digital Signal Processing*, vol. 126, paper no. 103514, 2022. DOI: 10.1016/j.dsp.2022.103514.
69. Ferrer, L., "Analysis and Comparison of Classification Metrics," *arXiv.org*, P. 1-36, 2022. DOI: 10.48550/arXiv.2209.05355.

70. Atkinson, P., Tatnall, A., "Introduction Neural networks in remote sensing," *International Journal of Remote Sensing*, vol. 18, P. 699-709, 2010. DOI: 10.1080/014311697218700.
71. Miller, D., Kaminsky, E., Rana, S., "Neural network classification of remote-sensing data," *Computers & Geosciences*, vol. 21, P. 377-386, 1995. DOI: 10.1016/0098-3004(94)00082-6.
72. Borovyk, D., Fedoniuk, R., Subbotin, S., Oliinyk, A., Kolpakova, T., "Detection of Vehicles in Aerial Photographs Using Convolutional Neural Networks," *SMARTINDUSTRY-2024: International Conference on Smart Automation & Robotics for Future Industry*, Lviv-Shatsk, Ukraine, paper no. 12, 2024.
73. Каштан, В., Гнатушенко, В., Удовик, І., Шевцова, О., "Нейромережеве розпізнавання об'єктів забудови на аерофотознімках," *Information Technology: Computer Science, Software Engineering and Cyber Security*, no. 1, P 30-39, 2013. DOI: 10.32782/IT/2023-1-5.
74. Schowengerdt, R.A., *Remote Sensing: Models and Methods for Image Processing: Third edition*, Academic Press, San Diego, CA, 2007, 515 p.
75. Wagner, W., "A Better Understanding of Our Earth through Remote Sensing," *Remote Sensing*, vol. 1, no. 1, P. 1-2, 2009, DOI: 10.3390/rs1010001.
76. Saikhom, V., Kalita, M., "UAV for Remote Sensing Applications: An Analytical Review," *Emerging Technology for Sustainable Development. EGTET 2022. Lecture Notes in Electrical Engineering*, vol. 1061, P. 51-59, 2023. DOI: 10.1007/978-981-99-4362-3_6.
77. Zhang, Z., Zhu, L., "A Review on Unmanned Aerial Vehicle Remote Sensing: Platforms, Sensors, Data Processing Methods, and Applications," *Drones*, vol. 7, no. 6, article no. 398, 2023. DOI: 10.3390/drones7060398.
78. Zhang, Y., Wang, Y., Hao, H., Li, Z., Long, Y., Zhang, X., Xia, C., "Unmanned Aerial Vehicle (UAV) Hyperspectral Imagery Mining to Identify New Spectral Indices for Predicting the Field-Scale Yield of Spring Maize," *Sustainability*, vol. 16, no. 24, article no. 10916, 2024. DOI: 10.3390/su162410916.

79. Zhang, X., Cao, X., Yu, A., Yu, W., Li, Z., Quan, Y., "UAVStereo: A Multiple Resolution Dataset for Stereo Matching in UAV Scenarios," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 16, P. 2942-2953, 2023. DOI: 10.1109/JSTARS.2023.3257489.
80. Pecho, P., Škvareková, I., Ažaltovič, V., Bugaj, M., "UAV usage in the process of creating 3D maps by RGB spectrum," *Transportation Research Procedia*, vol. 43, P. 328-333, 2019. DOI: 10.1016/j.trpro.2019.12.048.
81. Каштан В., Казимиренко О., Гнатушенко В., "Порівняльний аналіз ефективності методів розпізнавання автомобілів на аерокосмічних знімках," *Control Systems and Computers*, no. 4, P. 50-60, 2024. DOI: 10.15407/csc.2024.04.050
82. Lu, F., Li, K., Nie, Y., Tao, Y., Yu, Y., Huang, L., Wang, X., "Object Detection of UAV Images from Orthographic Perspective Based on Improved YOLOv5s," *Sustainability*, no. 15, 2023, paper no. 14564. DOI: 10.3390/su151914564.
83. Lukin, V., Ponomarenko, N., Kurekin, A., Lever, K., Pogrebnyak, O., Fernandez, L. P. S., "Approaches to Classification of Multichannel Images," *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, P. 794–803, 2006. DOI: 10.1007/11892755_82.
84. Sendjasni, A., Traparic, D., Larabi, M. -C., "Investigating Normalization Methods for CNN-Based Image Quality Assessment," *2022 IEEE International Conference on Image Processing (ICIP)*, Bordeaux, France, P. 4113-4117, 2022, DOI: 10.1109/ICIP46576.2022.9897268.
85. Koresh, H., "Impact of the Preprocessing Steps in Deep Learning-Based Image Classifications," *National Academy Science Letters*, vol. 47, P. 645-647, 2024. DOI: 10.1007/s40009-023-01372-2.
86. Herwanto, H.W., Handayani, A. N., Wibawa, A. P., Chandrika, K. L. Arai, K., "Comparison of Min-Max, Z-Score and Decimal Scaling Normalization for Zoning Feature Extraction on Javanese Character Recognition," *2021 7th International Conference on Electrical, Electronics and Information Engineering (ICEEIE)*, Malang, Indonesia, P. 1-3, 2021. DOI: 10.1109/ICEEIE52663.2021.9616665.

87. Subasi, A, *Practical Machine Learning for Data Analysis Using Python*, Academic Press, San Diego, CA, 2020, 524 p.
88. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S., “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, article no. 721, P. 8026-8037, 2019. DOI: 10.48550/arXiv.1912.01703.
89. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., Zheng, X., “TensorFlow: A System for Large-Scale Machine Learning,” *OSDI'16: Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, P. 265-283, 2016. DOI: 10.48550/arXiv.1605.08695.
90. Heaton, J., *Applications of Deep Neural Networks with Keras*, Heaton Research, 2022, 577 p.
91. Wu, Y., Liu, L., “Selecting and Composing Learning Rate Policies for Deep Neural Networks,” *ACM Transactions on Intelligent Systems and Technology*, vol. 14, no. 2, article no. 22, P. 1-25, 2023. DOI: 10.1145/3570508.
92. Konar, J., Khandelwal, P., Tripathi, R., “Comparison of Various Learning Rate Scheduling Techniques on Convolutional Neural Network,” *2020 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*, Bhopal, India, 2020, P. 1-5. DOI: 10.1109/SCEECS48394.2020.94.
93. Vapnik, V., *Statistical Learning Theory*, Wiley-Interscience, 1998, 768 p.
94. Du, D., Qi, Y., Yu, H. G., Yang, Y., Duan, K., Li, G., Zhang, W. G., Huang, Q., Tian, Q., “The Unmanned Aerial Vehicle Benchmark: Object Detection and Tracking,” *European Conference on Computer Vision (ECCV)*, vol. 128, P. 1141-1159, 2018. DOI: 10.1007/s11263-019-01266-1.

95. Zhu, P., Wen, L., Bian, X., Ling, H., Hu, Q., "Vision Meets Drones: A Challenge," arXiv, P. 1-11, 2018. DOI: 10.48550/arXiv.1804.07437.
96. Frostig, R., Johnson, M. J., Leary, C., "Compiling machine learning programs via high-level tracing," *SYSML'18*, Stanford, CA USA, February 2018, P. 1-3.
97. Chen, T., Li, M., Li, Y., Lin, M., Wang, N., Wang, M., Xiao, T., Xu, B., Zhang, C., Zhang, Z., "MXNet: A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed Systems," arXiv, P. 1-6, 2015. DOI: 10.48550/arXiv.1512.01274.
98. Al-Rfou, R., Alain, G., Almahairi, A., Angermueller, C., Bahdanau, D., Ballas, N., Bastien, F., Bayer, J., Belikov, A., Belopolsky, A., Bengio, Y., Bergeron, A., Bergstra, J., Bisson, V., Snyder, J. B., Bouchard, N., Boulanger-Lewandowski, N., Bouthillier, X., Zhang, Y., "Theano: A Python framework for fast computation of mathematical expressions (Version 1)," arXiv, P. 1-19, 2016. DOI: 10.48550/ARXIV.1605.02688.
99. Lydia, A. A., Francis, F. S., "Learning Rate Scheduling Policies," *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 1., P. 3641–3644, 2019. DOI: 10.35940/ijitee.a4648.119119.
100. Benoit-Cattin, T., Velasco-Montero, D., Fernández-Berni, J., "Impact of Thermal Throttling on Long-Term Visual Inference in a CPU-Based Edge Device," *Electronics*, vol. 9, no. 12, article no. 2106, 2020. DOI: 10.3390/electronics9122106.
101. Bozcan, I., Kayacan, E., "AU-AIR: A Multi-modal Unmanned Aerial Vehicle Dataset for Low Altitude Traffic Surveillance," *IEEE International Conference on Robotics and Automation*, P. 8504-8510, 2020. DOI: 10.1109/ICRA40945.2020.9196845
102. Tang, G., Ni, J., Zhao, Y., Gu, Y., Cao, W., "A Survey of Object Detection for UAVs Based on Deep Learning," *Remote Sensing*, vol. 16, no. 1. MDPI AG, paper no. 149, 2023. DOI: 10.3390/rs16010149.
103. 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, vol. 10, no. 1, P. 91-124, 2022. DOI: 10.1109/MGRS.2021.3115137.

104. Liu, H., Yu, Y., Liu, S., Wang, W., “A Military Object Detection Model of UAV Reconnaissance Image and Feature Visualization,” *Applied Sciences*, vol. 12, no. 23. MDPI AG, paper no. 12236, 2022. DOI: 10.3390/app122312236.

105. Liu, C., Meng, F., Zhu, Z., Zhou, L., “Object Detection of UAV Aerial Image based on YOLOv8,” *Frontiers in Computing and Intelligent Systems*, vol. 5, no. 3. Darcy & Roy Press Co. Ltd., P. 46–50, 2023. DOI: 10.54097/fcis.v5i3.13852.

106. Tsekhmystro, R., Rubel, O., Lukin, V., “Study of the dependence of accuracy in vehicles search on the size of the object using UAV images,” *Aerospace Technic and Technology*, no. 3, P. 89–98, 2024. DOI: 10.32620/aktt.2024.3.08.

107. Tsekhmystro, R., Rubel, O., Lukin, V., “Investigation of the effect of object size on accuracy of human localisation in images acquired from unmanned aerial vehicles,” *Aerospace Technic and Technology*, no. 2, P. 83–90, 2024. DOI: 10.32620/aktt.2024.2.09.

108. Courtrai, L., Pham, M.-T., Lefèvre, S., “Small Object Detection in Remote Sensing Images Based on Super-Resolution with Auxiliary Generative Adversarial Networks,” *Remote Sensing*, vol. 12, no. 19, paper no. 3152, 2020. DOI: 10.3390/rs12193152.

109. Ünel, F. Ö., Özkalayci, B. O., Çiğla, C., “The Power of Tiling for Small Object Detection,” *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Long Beach, CA, USA, 2019, P. 582-591. DOI: 10.1109/CVPRW.2019.00084.

110. Rukundo, O., “Effects of Image Size on Deep Learning,” *Electronics*, vol. 12, no. 4, paper no. 985, 2023. DOI: 10.3390/electronics12040985.

111. Pope, P., Zhu, C., Abdelkader, A., Goldblum, M., Goldstein, T., “The Intrinsic Dimension of Images and Its Impact on Learning,” *arXiv*, 2021. DOI: 10.48550/ARXIV.2104.08894.

112. Saponara, S., Elhanashi, A., “Impact of Image Resizing on Deep Learning Detectors for Training Time and Model Performance,” *Lecture Notes in Electrical*

Engineering. Springer International Publishing, P. 10–17, 2022. DOI: 10.1007/978-3-030-95498-7_2.

113. Bondžulić, B. P., Stojanović, N. M., Lukin, V., Kryvenko, S., "JPEG and BPG Visually Lossless Image Compression via KonJND-1k Database," *Vojnotehnicki Glasnik*, vol. 72, no. 3, P. 1214–1241, 2024. DOI: 10.5937/vojtehg72-50300.

114. Abramova, V.V., Abramov, S. K., Lukin, V. V., Roenko, A. A., Vozel, B., "Automatic estimation of spatially correlated noise variance in spectral domain for images," *Telecommunications and Radio Engineering*, vol. 73, no. 6, P. 511–527, 2014. DOI: 10.1615/telecomradeng.v73.i6.40.

115. Rubel, A., Rubel, O., Abramova, V., Proskura, G., Lukin, V., "Improved Noisy Image Quality Assessment Using Multilayer Neural Networks," *2019 IEEE 2nd Ukraine Conference on Electrical and Computer Engineering (UKRCON)*, Lviv, Ukraine, 2019, P. 1046-1051. DOI: 10.1109/UKRCON.2019.8879950.

116. Uss, M., Vozel, B., Lukin, V., Chehdi, K., "NoiseNet: Signal-Dependent Noise Variance Estimation with Convolutional Neural Network," *Lecture Notes in Computer Science. Springer International Publishing*, P. 414–425, 2018. DOI: 10.1007/978-3-030-01449-0_35.

117. Rubel, A., Naumenko, A., Lukin, V., "A neural network based predictor of filtering efficiency for image enhancement," *2014 IEEE Microwaves, Radar and Remote Sensing Symposium (MRRS)*, Kiev, Ukraine, 2014, P. 14-17, DOI: 10.1109/MRRS.2014.6956654.

118. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K., "Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, P. 2080-2095, 2007. DOI: 10.1109/TIP.2007.901238.

119. Nayan, A. -A., Saha, J., Raqib Mahmud, K., Kalam, A., Azad, A., Golam Kibria, M., "Detection of Objects from Noisy Images," *2020 2nd International Conference on Sustainable Technologies for Industry 4.0 (STI)*, Dhaka, Bangladesh, 2020, P. 1-6, DOI: 10.1109/STI50764.2020.9350521.

120. Milani, S., Bernardini, R., Rinaldo, R., "Adaptive denoising filtering for object detection applications," *2012 19th IEEE International Conference on Image*

Processing, Orlando, FL, USA, 2012, P. 1013-1016, DOI: 10.1109/ICIP.2012.6467034.

121. Foi, A., Trimeche, M., Katkovnik, V., Egiazarian, K., "Practical Poissonian-Gaussian Noise Modeling and Fitting for Single-Image Raw-Data," *IEEE Transactions on Image Processing*, vol. 17, no. 10, P. 1737-1754, 2008. DOI: 10.1109/TIP.2008.2001399.

122. Azzari, L., Foi, A., "Variance Stabilization for Noisy+Estimate Combination in Iterative Poisson Denoising," *IEEE Signal Processing Letters*, vol. 23, no. 8, P. 1086-1090, 2016. DOI: 10.1109/LSP.2016.2580600.

123. Mäkitalo, M., Foi, A., "Noise Parameter Mismatch in Variance Stabilization, With an Application to Poisson–Gaussian Noise Estimation," *IEEE Transactions on Image Processing*, vol. 23, no. 12, P. 5348-5359, 2014. DOI: 10.1109/TIP.2014.2363735.

124. Abramova, V. V., Abramov, S. K., Lukin, V. V., "Iterative method for blind evaluation of mixed noise characteristics on images," *Information and Telecommunication Sciences*, vol. 0, no. 1, P. 8–14, 2015. DOI: 10.20535/2411-2976.12015.8-14.

125. Lim, S., "Characterization of noise in digital photographs for image processing," *SPIE Proceedings*, vol. 6069, article no. 60690, 2006. DOI: 10.1117/12.655915.

126. Tsekhmystro, R., Rubel, O., Abramova, V., Zriakhov, M., Uss, M., "Mobile Deployment of NoiseNet: Noise Characteristics Assessment in Real-World Images," *2019 IEEE 2nd Ukraine Conference on Electrical and Computer Engineering (UKRCON)*, P. 1112-1117, 2019. DOI: 10.1109/UKRCON.2019.8879797.

127. Цехмистро, Р. В., Абрамова, В. В., Рубель, А. С., Усс, М. Л., Проскура, Г. А., Рубель, О. С., "Оцінка характеристик шуму на реальних зображеннях з використанням згорткової нейронної мережі на мобільній платформі," *Radioelectronic And Computer Systems*, no. 2, P. 60–70, 2019. DOI: 10.32620/reks.2019.2.05.

128. Mäkitalo, M., Foi, A., Fevrale, D., Lukin, V., "Denoising of single-look SAR images based on variance stabilization and nonlocal filters," *2010 International Conference on Mathematical Methods in Electromagnetic Theory*, Kyiv, Ukraine, 2010, P. 1-4. DOI: 10.1109/MMET.2010.5611418.
129. Naumenko, V., Kovalenko, B., Lukin, V., "BPG-based compression analysis of Poisson-noisy medical images," *Radioelectronic and Computer Systems*, no. 3, P. 91–100, 2023. DOI: 10.32620/reks.2023.3.08.
130. Prasath, V. B. S., "Quantum Noise Removal in X-Ray Images with Adaptive Total Variation Regularization," *Informatica*, vol. 28, no. 3, P. 505–515, 2017. DOI: 10.15388/informatica.2017.141.
131. Rubel, A. S., Lukin, V. V., Egiazarian, K. O., "A method for predicting DCT-based denoising efficiency for grayscale images corrupted by AWGN and additive spatially correlated noise," *SPIE Proceedings*, vol. 9399. SPIE, paper no. 93990, 2015. DOI: 10.1117/12.2082533.
132. Rubel, A., Rubel, O., Tsekhmystro, R., Rebrov, V., Lukin, V., "Automatic Decision Undertaking on Expedience of Image Denoising Based on Filter Efficiency Prediction," *Springer Proceedings in Physics. Springer Nature Singapore*, P. 504–524, 2023. DOI: 10.1007/978-981-99-4098-1_44.
133. Tang, M., Zhao, Z., Qiu, J., "A Foggy Weather Simulation Algorithm for Traffic Image Synthesis Based on Monocular Depth Estimation," *Sensors*, vol. 24, no. 6, paper no. 1966, 2024. DOI: 10.3390/s24061966.
134. Yang, L., Kang, B., Huang, Z., Xu, X., Feng, J., Zhao, H., "Depth Anything: Unleashing the Power of Large-Scale Unlabeled Data," *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, P. 10371-10381, 2024. DOI: 10.1109/CVPR52733.2024.00987.
135. Choi, J., Kim, D. H., Lee, S., Lee, S. H., Song, B. C., "Synthesized rain images for deraining algorithms," *Neurocomputing*, vol. 492, P. 421–439, 2022. DOI: 10.1016/j.neucom.2022.04.034.

136. He, K., Sun, J., Tang, X., "Single image haze removal using dark channel prior," *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, USA, 2009, P. 1956-1963. DOI: 10.1109/CVPR.2009.5206515.
137. Tsekhmystro, R., Rubel, O., Prysiazhniuk, O., Lukin, V., "Impact of distortions in UAV images on quality and accuracy of object localization," *Radioelectronic and Computer Systems*, vol. 2024, no. 4, P. 59–67, 2024. DOI 10.32620/reks.2024.4.05.
138. Rubel, O., Tsekhmystro, R., Lukin, V., Egiazarian, K., "Benchmark of Similar Blocks Search under Noisy Conditions," *Electronic Imaging*, vol. 33, no. 10, P. 238-1-238, 2021. DOI: 10.2352/issn.2470-1173.2021.10.ipas-238.
139. Tsekhmystro, R., Lukin, V., "Object localization and recognition in UAV-based noisy images," *Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління*, vol 3, P. 18-19, 2025

ДОДАТОК А

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА

Статті у наукових періодичних виданнях:

1. Tsekhmystro, R., Rubel, O., Lukin, V., “Study of methods for searching and localizing objects in images from aircraft using convolutional neural networks,” *Radioelectronic and Computer Systems*, vol. 2024, no. 1, P. 87–98, 2024. DOI: 10.32620/reks.2024.1.08 (Scopus).

2. Tsekhmystro, R., Rubel, O., Lukin, V., “Study of the dependence of accuracy in vehicles search on the size of the object using UAV images,” *Aerospace Technic and Technology*, no. 3, P. 89–98, 2024. DOI: 10.32620/aktt.2024.3.08 (фаховий).

3. Tsekhmystro, R., Rubel, O., Lukin, V., “Investigation of the effect of object size on accuracy of human localisation in images acquired from unmanned aerial vehicles,” *Aerospace Technic and Technology*, no. 2, P. 83–90, 2024. DOI: 10.32620/aktt.2024.2.09 (фаховий).

4. Tsekhmystro, R., Rubel, O., Prysiashniuk, O., Lukin, V., “Impact of distortions in UAV images on quality and accuracy of object localization,” *Radioelectronic and Computer Systems*, vol. 2024, no. 4, P. 59–67, 2024. DOI: 10.32620/reks.2024.4.05 (Scopus).

5. Цехмистро, Р. В., Абрамова, В. В., Рубель, А. С., Усс, М. Л., Проскура, Г. А., Рубель, О. С., “Оцінка характеристик шуму на реальних зображеннях з використанням згорткової нейронної мережі на мобільній платформі,” *Radioelectronic And Computer Systems*, no. 2, P. 60–70, Jun. 2019. DOI: 10.32620/reks.2019.2.05 (фаховий).

Наукові праці, які засвідчують апробацію матеріалів дисертації:

6. Rubel, O., Tsekhmystro, R., Lukin, V., Egiazarian, K., “Benchmark of Similar Blocks Search under Noisy Conditions,” *Electronic Imaging*, vol. 33, no. 10, P. 238–1–238, 2021. DOI: 10.2352/issn.2470-1173.2021.10.ipas-238. (Scopus).

7. Tsekhmystro, R., Rubel, O., Abramova, V., Zriakhov, M., Uss, M., "Mobile Deployment of NoiseNet: Noise Characteristics Assessment in Real-World Images," *2019 IEEE 2nd Ukraine Conference on Electrical and Computer Engineering (UKRCON)*, Lviv, Ukraine, 2019, P. 1112-1117. DOI: 10.1109/UKRCON.2019.8879797. (Scopus) .
8. Tsekhmystro, R., Oliinyk, V., Proskura, G., Rubel, O., "Web Assembled Benchmark for Image Visual Quality Assessment, Prediction and Improvement," *2020 IEEE 15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET)*, Lviv-Slavske, Ukraine, 2020, P. 791-795. DOI: 10.1109/TCSET49122.2020.235543. (Scopus).
9. Rubel, A., Rubel, O., Tsekhmystro, R., Rebrov, V., Lukin, V., "Automatic Decision Undertaking on Expedience of Image Denoising Based on Filter Efficiency Prediction," *Springer Proceedings in Physics, Springer Nature Singapore*, P. 504–524, 2023. DOI: 10.1007/978-981-99-4098-1_44. (Scopus).
10. Tsekhmystro, R., Lukin, V., "Object localization and recognition in UAV-based noisy images," *Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління*, vol. 3, P. 18-19, 2025.

ДОДАТОК Б

АКТИ ВПРОВАДЖЕННЯ

«23» квітня 2025 р.

«Затверджую»
В.о. директора Інституту радіофізики та
електроніки ім. О.Я. Усикова НАН України
Юрій Логвінов



АКТ

науково-технічної комісії щодо впровадження наукових положень і результатів дисертації на здобуття ступеня доктора філософії аспіранта Національного аерокосмічного університету «ХАІ»

Цехмистра Ростислава Вікторовича

Науково-технічна комісія Інституту радіофізики та електроніки ім. О.Я. Усикова НАН України у складі голови комісії виконувача обов'язків заступника директора з наукової роботи, доктора фіз.-мат. наук Когута Олександра Євгеновича та членів комісії завідувача відділом поширення радіохвиль в природних середовищах, канд. фіз.-мат. наук Роєнка Олександра Миколайовича та завідувача відділом радіофізичної інтроскопії, канд. фіз.-мат. наук Почаніна Геннадія Петровича

склала даний акт про те, що в результаті виконання спільних науково-дослідних робіт, були використані та впроваджені наступні положення та результати, що отримані аспірантом Цехмистром Р.В.:

- результати досліджень типу та інтенсивності завад, що спотворюють зображення, а також їх вплив на ефективність локалізації та класифікації об'єктів;
- результати дослідження граничних розмірів об'єкту в пікселях на зображенні для досягнення достатнього рівня точності їх локалізації та класифікації, а також метод визначення максимальної висоти польоту для досягнення таких умов.

Розроблені в дисертації методи та проведені дослідження дозволили:

- ефективно оцінювати вплив завад на ефективність роботи нейронних мереж та приймати зважені рішення про доцільність проведення фільтрації зображення, враховуючи можливість підвищення метрик ефективності.
- розраховувати можливі параметри пристрою для захоплення зображень, зокрема БПЛА, його висоту польоту, роздільну здатність сенсору та кутову роздільну здатність.

Голова комісії

Члени комісії

Олександр КОГУТ

Олександр РОЄНКО

Геннадій ПОЧАНІН

ДОДАТОК В

ПРОГРАМНИЙ КОД МЕТОДУ ПОТАЙЛОВОЇ ОБРОБКИ

```

import numpy as np
import cv2
import onnxruntime as ort
from ultralytics import YOLO
import logging
import bm3d

def draw_detections(image, detections):
    for x1, y1, x2, y2, cls, conf in detections:
        label = f"{cls}: {conf:.2f}"
        cv2.rectangle(image, (int(x1), int(y1)), (int(x2), int(y2)), (0, 255, 0), 2)
        (w, h), _ = cv2.getTextSize(label, cv2.FONT_HERSHEY_SIMPLEX, 0.5, 1)
        cv2.rectangle(image, (int(x1), int(y1) - 20), (int(x1) + w, int(y1)), (0, 255, 0), -1)
        cv2.putText(image, label, (int(x1), int(y1) - 5),
cv2.FONT_HERSHEY_SIMPLEX,
                    0.5, (0, 0, 0), 1, cv2.LINE_AA)
    return image

def calculate_object_size(image_size, original_w=0.5, original_h=0.5,
camera_properties=None):
    """
    Calculate the size of the object in pixels based on the camera properties and the
    original size of the object.

    :param image_size: size of the image in pixels (height, width)
    :param original_w: size of the object in meters
    :param original_h: size of the object in meters
    :param camera_properties:

```

```

:return: size of the object in pixels
"""

if camera_properties is not None:
    focal_length = camera_properties.get("focal_length", None) # Focal length in
mm
    flying_height = camera_properties.get("flying_height", None) # Flying height
in meters
    sensor_width = camera_properties.get("sensor_width", None) # Sensor width in
mm
    sensor_height = camera_properties.get("sensor_height", None) # Sensor height
in mm
    if focal_length is None or flying_height is None or sensor_width is None or
sensor_height is None:
        raise ValueError("Camera properties must include focal_length, flying_height,
sensor_width, and sensor_height.")
    # Calculate the size of the object in pixels
    pixel_width = (flying_height * sensor_width) / (focal_length * image_size[1])
    pixel_height = (flying_height * sensor_height) / (focal_length * image_size[0])
    # Calculate the size of the object in pixels
    size_in_pixels = ((original_h * 100) / pixel_height) * ((original_w * 100) /
pixel_width)
    return size_in_pixels
else:
    return None

if __name__ == "__main__":
    # Define the detection model
    detector = YOLO('YOLOv8L-visdrone.pt')
    # Load an image
    image = cv2.imread('/mnt/d/data/images/frame_20190905112522_x_0001608.jpg')

```

```

camera_properties = {
    "focal_length": 0.40, # Focal length in centimeters
    "flying_height": 3000, # Flying height in centimeters
    "sensor_width": 0.682, # Sensor width in centimeters
    "sensor_height": 0.384, # Sensor height in centimeters
}

minimal_object_size = calculate_object_size(image.shape[:2], original_w=0.5,
original_h=0.1, camera_properties=camera_properties)
print(f'Minimal object size in pixels: {minimal_object_size}')
detection_results = []
if minimal_object_size is not None and minimal_object_size > 200:
    detection_results_local = detector(image, verbose=False)[0]
    for box, conf, cls_id in zip(detection_results_local.bboxes.xyxy,
detection_results_local.bboxes.conf,
                                detection_results_local.bboxes.cls):
        detection_results.append([*box.cpu().detach().numpy(),
detection_results_local.names[int(cls_id)], float(conf)])
else:
    w_mid, h_mid = image.shape[1] // 2, image.shape[0] // 2
    pad, image_parts = [], []
    for i in range(0, image.shape[0], h_mid):
        for j in range(0, image.shape[1], w_mid):
            pad.append([j, i, j, i])
            image_parts.append(image[i:i + h_mid, j:j + w_mid])
    for tile_no, tile in enumerate(image_parts):
        detection_results_local = detector(tile, verbose=False)[0]
        for box, conf, cls_id in \
            zip(detection_results_local.bboxes.xyxy,
                detection_results_local.bboxes.conf,
                detection_results_local.bboxes.cls):

```

```
detection_results.append([
    *(box.cpu().detach().numpy() + pad[tile_no]),
    detection_results_local.names[int(cls_id)],
    float(conf)])
result_image = draw_detections(image, detection_results)
# Save the result
cv2.imwrite('result.jpg', result_image)
```

ДОДАТОК Г

ПРОГРАМНИЙ КОД МЕТОДУ З ФІЛЬТРАЦІЄЮ ШУМУ

```

import numpy as np
from noise_net import NoiseEstimator, prepare_patch
import cv2
import onnxruntime as ort
from ultralytics import YOLO
import logging
import bm3d

def draw_detections(image, detections):
    classes = detections.names
    for (x1, y1, x2, y2), conf, cls_id in zip(detections.bboxes.xyxy,
detections.bboxes.conf, detections.bboxes.cls):
        cls_id = int(cls_id)
        label = f"{classes[cls_id]}: {conf:.2f}"
        cv2.rectangle(image, (int(x1), int(y1)), (int(x2), int(y2)), (0, 255, 0), 2)
        (w, h), _ = cv2.getTextSize(label, cv2.FONT_HERSHEY_SIMPLEX, 0.5, 1)
        cv2.rectangle(image, (int(x1), int(y1) - 20), (int(x1) + w, int(y1)), (0, 255, 0), -1)
        cv2.putText(image, label, (int(x1), int(y1) - 5),
cv2.FONT_HERSHEY_SIMPLEX,
                    0.5, (0, 0, 0), 1, cv2.LINE_AA)
    return image

if __name__ == "__main__":
    # Define the noise estimator based on NoiseNet
    noise_estimator = NoiseEstimator('NoiseNet.onnx')

    # Define the detection model

```

```

detector = YOLO('YOLOv8L-visdrone.pt')

# Load an image
image =
cv2.imread('/mnt/d/data/images_0_3_0/frame_20190829091111_x_0000156.jpg')

# Predict the noise STD
std = noise_estimator.calculate_noise_level(cv2.cvtColor(image,
cv2.COLOR_RGB2GRAY))

if std is not None and std > 15:
    logging.info(f'Noise STD: {std}. Run the denoise process.')
    image = bm3d.bm3d_rgb(image, std)
else:
    logging.info(f'Noise STD: {std}. Skip the denoise process.')

# Run the detection model
detection_results = detector(image, verbose=False)[0]

# Draw results
result_image = draw_detections(image, detection_results)

# Save the result
cv2.imwrite('result.jpg', result_image)

```

ДОДАТОК Д
ПРОГРАМНИЙ КОД МЕТОДУ DARK CHANNEL PRIOR

```
import cv2
import numpy as np

def min_filter(img, size=15):
    return cv2.erode(img, np.ones((size, size)))

def estimate_dark_channel(img, size=15):
    min_per_pixel = np.min(img, axis=2)
    dark_channel = cv2.erode(min_per_pixel, np.ones((size, size)))
    return dark_channel

def estimate_atmospheric_light(img, dark_channel, top_percent=0.001):
    h, w = dark_channel.shape
    num_pixels = int(max(h * w * top_percent, 1))
    flat_img = img.reshape(-1, 3)
    flat_dark = dark_channel.flatten()
    indices = np.argsort(flat_dark)[-num_pixels:]
    atmo_light = np.mean(flat_img[indices], axis=0)
    return atmo_light

def estimate_transmission(img, atmo_light, omega=0.95, size=15):
    normed = img / atmo_light
    transmission = 1 - omega * estimate_dark_channel(normed, size)
    return transmission

def recover_image(img, transmission, atmo_light, t0=0.5):
    transmission = np.clip(transmission, t0, 1.0)
```

```
recovered = (img - atmo_light) / transmission[:, :, np.newaxis] + atmo_light  
recovered = np.clip(recovered, 0, 1)  
return recovered
```

```
hazy = cv2.imread("frame_20190905142119_x_0000910.jpg").astype(np.float32)  
hazy = hazy / 255.0  
dark = estimate_dark_channel(hazy)  
A = estimate_atmospheric_light(hazy, dark)  
t = estimate_transmission(hazy, A)  
recovered = recover_image(hazy, t, A)  
print("A:", A)  
print("transmission min/max:", t.min(), t.max())  
print("image min/max:", recovered.min(), recovered.max())  
cv2.imwrite("dehazed.jpg", recovered * 255)
```