



## Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ комп'ютерних наук \_\_\_\_\_

Кафедра \_\_\_\_\_ програмної інженерії \_\_\_\_\_

Рівень вищої освіти \_\_\_\_\_ другий (магістерський) \_\_\_\_\_

Спеціальність \_\_\_\_\_ 121 – Інженерія програмного забезпечення \_\_\_\_\_

Тип програми \_\_\_\_\_ освітньо-наукова програма \_\_\_\_\_

Освітня програма \_\_\_\_\_ Інженерія програмного забезпечення \_\_\_\_\_

(шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_

(підпис)

«\_\_\_\_» \_\_\_\_\_ 2024 р.

### ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові \_\_\_\_\_ Супруну Святославу Олександровичу \_\_\_\_\_

(прізвище, ім'я, по батькові)

1. Тема роботи «Дослідження методів розпізнавання технічних об'єктів за допомогою дронів»

Затверджена наказом по університету від 29.03.2024р. № 250 Ст2. Термін подання студентом роботи до екзаменаційної комісії 21.06.20243. Вихідні дані до роботи бібліотеки Python OpenCV, PyTorch, моделі YOLO, середовища розробки Visual Studio Code

4. Перелік питань, що потрібно опрацювати в роботі


Порівняння та аналіз двох методів розпізнавання технічних об'єктів відносно їх швидкості та точності роботи в задачах розпізнавання військової техніки на відеоматеріалах з дронів, тренування моделей, виконання експерименту та аналіз отриманих результатів

## КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної галузі та постановка задачі	30.03.2024	<i>виконано</i>
2	Огляд існуючих методів	01.04.2024	<i>виконано</i>
3	Підготовка проведення експерименту	08.04.2024	<i>виконано</i>
4	Проведення експерименту	15.04.2024	<i>виконано</i>
5	Аналіз результатів отриманих під час експерименту	22.04.2024	<i>виконано</i>
9	Підготовка пояснювальної записки	08.06.2024	<i>виконано</i>
10	Підготовка презентації та доповіді	10.06.2024	<i>виконано</i>
11	Нормоконтроль	11.06.2024	<i>виконано</i>
12	Рецензування	16.06.2024	<i>виконано</i>
13	Занесення диплома в електронний архів	17.06.2024	<i>виконано</i>
14	Попередній захист	19.06.2024	<i>виконано</i>
15	Допуск до захисту у зав. кафедри	20.06.2024	<i>виконано</i>

Дата видачі завдання 29 березня 2024р.

Студент

  
(підпис)

Супрун С.О.

Керівник роботи

(підпис)

проф. Білоус Н.В.

(посада, прізвище, ініціали)

## РЕФЕРАТ / ABSTRACT

Пояснювальна записка містить: 63 с., 43 рис., 2 табл., 16 джерел.

НЕЙРОННА МЕРЕЖА, МАШИННЕ НАВЧАННЯ, КОМП'ЮТЕРНИЙ ЗІР, ТЕХНІЧНИЙ ОБ'ЄКТ, НАБІР ДАНИХ, OPENCV, SSD, YOLO, TENSORFLOW, PYTORCH.

Об'єктом дослідження є аналіз і оцінка методів відстеження та розпізнавання технічних об'єктів з використанням дронів для визначення ефективності та достовірності різних підходів до їх відстеження та розпізнавання.

Мета роботи – проведення комплексного аналізу існуючих методів відстеження та розпізнавання технічних об'єктів за допомогою дронів і визначення їх переваг та недоліків. У процесі дослідження використовуються принципи теорії моделювання та передові підходи до обробки зображень. Основні завдання включають аналіз реалізованих методів відстеження та розпізнавання технічних об'єктів, виявлення найбільш універсального методу оцінки ефективності та точності різних підходів до цього процесу.

Результат роботи – було створено огляд існуючих методів відстеження та розпізнавання технічних об'єктів за допомогою дронів, а також визначено особливі аспекти кожного методу, та були навчені декілька моделей для розпізнавання технічних об'єктів.

NEURAL NETWORK, MACHINE LEARNING, COMPUTER VISION, ENGINEERING OBJECT, DATASET, OPENCV, SSD, YOLO, TENSORFLOW, PYTORCH.

The object of the study is the analysis and assessment of methods of tracking and ranking technical objects using drones to determine the effectiveness and reliability of various approaches to their tracking and ranking.

The purpose of the work is to carry out a comprehensive analysis of existing methods of tracking and classifying technical objects using drones and determining

their advantages and disadvantages. The principles of modeling theory and advanced approaches to image processing are used in the research process. The main tasks include the analysis of implemented methods of tracking and classification of technical objects, identification of the most universal method of evaluating the effectiveness and accuracy of various approaches to this process.

The result of the work - an overview of the existing methods of tracking and classifying technical objects using drones was created, as well as the special aspects of each method were determined, and several models for recognizing technical objects were trained and compared.

## ЗМІСТ

<b>Перелік скорочень</b> .....	8
<b>Вступ</b> .....	9
<b>1 Аналіз предметної галузі</b> .....	11
1.1 Аналіз сфери комп’ютерного зору та машинного навчання.....	11
1.2 Основна інформація про згорткові нейронні мережі.....	12
1.3 Використання згорткових нейронних мереж для відстеження технічних об’єктів за допомогою дронів в різних галузях .....	13
1.4 Проблема що досліджується .....	16
1.5 Постановка технічного завдання кваліфікаційної роботи.....	16
1.6 Засоби що будуть використані для проведення дослідження.....	17
<b>2 Опис методів комп’ютерного зору для розпізнавання технічних об’єктів за допомогою дронів</b> .....	19
2.1 Опис обраних методів.....	19
2.2 YOLO архітектура.....	20
2.3 Архітектура SSD.....	23
2.4 Постановка гіпотез .....	26
<b>3 Огляд реалізації програмного забезпечення</b> .....	27
3.1 Програмна реалізація розрахунку значення точності для моделей.....	27
<b>4 Проведення експериментального дослідження</b> .....	32
4.1 Підготовка набору даних для навчання обраних моделей .....	32
4.2 Навчання обраних моделей. ....	32
4.3 Результати роботи моделей на зображеннях .....	35
<b>Висновки</b> .....	42
<b>Перелік джерел посилання</b> .....	44
Додаток А Перелік джерел посилання за науковими напрямами керівника та науковців кафедри програмної інженерії .....	46
Додаток Б Результат перевірки на унікальність тексту в базі ХНУРЕ.....	47
Додаток В Слайди презентації.....	48
Додаток Г Апробація результатів роботи.....	57
Додаток Д Код для обчислення метрик для об’єктивного оцінювання роботи моделі для розпізнавання технічних об’єктів .....	62

Додаток Е Експертний висновок результатів перевірки кваліфікаційної роботи на відповідність оформлення вимогам ДСТУ 3008:2015 .....	63
-------------------------------------------------------------------------------------------------------------------------------------	----

## ПЕРЕЛІК СКОРОЧЕНЬ

CNN – Convolutional Neural Network

YOLO – You Only Look Once

SSD – Single Shot Detector

mAP – mean Average Precision

GPU – Graphics Processing Unit

TPU – Tensor Processing Unit

VGG – Visual Geometry Group

FAIR – Facebook's AI Research lab

IOU – Intersection Over Union

## ВСТУП

У сучасному світі все більш виражена використання дронів у різноманітних галузях нашого життя, таких як: розвідка, моніторинг ситуацій, забезпечення безпеки, та військова справа. Дрони є ефективним засобом для огляду великих територій, важкодоступних місць, об'єктів інфраструктури забезпечуючи надзвичайну точність та гнучкість їх використання. Одним з основних завдань при використанні дронів є відстежування певних технічних об'єктів.

Дослідження методів розпізнавання технічних об'єктів за допомогою дронів є важливим та актуальним напрямком в покращенні засобів для збору візуальної інформації, оскільки це дає змогу автоматизувати виявлення необхідних об'єктів на відеоматеріалах з дрону, що може пришвидшити та спростити, а з часом автоматизувати аналіз необхідних матеріалів та знаходити в них потрібні об'єкти.

В наш час існує багато методів розпізнавання технічних об'єктів на фото та відеоматеріалах, ці методи відрізняються між собою за швидкістю роботи, точністю, ресурсоемністю. Також робота моделей залежить від якості набору даних на якому вона навчається та від кількості кроків навчання(епох).

Метою цієї роботи є аналіз існуючих методів для розпізнавання технічних об'єктів на відеоматеріалах з дрону різної якості, аналіз отриманих результатів для знаходження найбільш ефективних методів комп'юторного зору, які здатні працювати в умовах обмеженої кількості ресурсів не критично втрачаючи при цьому ефективність пошуку та класифікації необхідних об'єктів.

Для дослідження методів розпізнавання технічних об'єктів за допомогою дронів я обрав такі моделі як YOLO (You Only Look Once) та SSD (Single Shot MultiBox Detector) це дві популярні моделі для відстежування та класифікування об'єктів на фото то відео матеріалах.

Методом цього дослідження є проведення практичного експерименту, навчання моделей на наборі різних зображень військової техніки з дрону. Набір даних являє собою зображення військової техніки з різних ракурсів, в різну пору року, різного освітлення, різної розмірності техніки та видів, на зображеннях високої та низької якості. Для порівняння обраних методів комп'ютерного зору

будуть описані критерії за якими буде визначена більш ефективна модель в тих чи інших умовах.

В результаті виконання магістерської роботи були сформовані рекомендації по використанню досліджених моделей комп'ютерного зору для вирішення тих чи інших проблем в розпізнаванні військових технічних об'єктів, а саме військової техніки.

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

### 1.1 Аналіз сфери комп'ютерного зору та машинного навчання

Комп'ютерний зір та машинне навчання це галузі що стрімко розвиваються в останні роки, тому не дивно, що вони застосовуються у все більшій кількості сфер нашого життя. Ці технології забезпечують комп'ютерам можливість аналізувати та інтерпретувати великі обсяги візуальної інформації, що раніше були доступні тільки для людського ока. Комп'ютерний зір та машинне навчання дозволяє машинам “бачити”, аналізувати, та навчатися за для вдосконалення своїх навичок на основі отриманого досвіду.

Комп'ютерний зір – це галузь науки та технології створення машин, які мають змогу для виявлення, стеження за об'єктами та їх класифікування на візуальних даних. У межах комп'ютерного зору досліджуються алгоритми та методи для вирішення різних завдань. Основні складові комп'ютерного зору включають в себе обробку зображень, розпізнавання образів, класифікування, стеження за об'єктами. Так комп'ютерний зір має широкий спектр застосувань, від медицини до військової справи.

Машинне навчання – це клас методів штучного інтелекту, що включає набір алгоритмів, які застосовують, щоб створити машину, яка вчиться на власному досвіді без явного програмування. Замість того, щоб напругу програмувати комп'ютер для виконання конкретного завдання, ми даємо йому можливість на основі масиву вхідних даних, певних вагових коефіцієнтів та очікуваного результату навчатись самостійно для отримання певних результатів.

Нейронні мережі навчаються шляхом подання великої кількості вхідних даних разом з відповідними вихідними значеннями або мітками. Під час навчання мережі налаштовують свої параметри, відомі як ваги, щоб максимізувати відповідність між прогнозованими результатами та правильними відповідями. Цей процес передбачає обчислення помилок прогнозування, а потім використання алгоритму оптимізації для оновлення ваг моделі з метою підвищення точності прогнозування.

Під час навчання нейронні мережі використовують алгоритми оптимізації, такі як стохастичний градієнтний спуск або його варіації для коригування ваг моделі. Ці алгоритми аналізують градієнти функції втрат, вказуючи напрямок, в якому слід змінити ваги, щоб мінімізувати помилку прогнозування. Після кожної навчальної ітерації ваги моделі оновлюються з урахуванням цих градієнтів, роблячи крок у напрямку, який мінімізує втрати.

Після завершення навчання нейронна мережа оцінюється за допомогою набору тестових даних, які не використовувалися в процесі навчання. Це дозволяє оцінити загальну ефективність та здатність моделі до узагальнення нових даних, що є вирішальним аспектом успішної роботи нейронної мережі в реальних умовах. Далі після навчання модель може працювати на будь-якому масиву даних, та використовувати для цього менше ресурсів ніж для навчання. Для розпізнавання технічних об'єктів на відеоматеріалах частіше за все використовують глибоке машинне навчання а зокрема глибокі згорткові мережі.

Сучасні методи розпізнавання технічних об'єктів зазвичай покладаються лише на технології глибокого навчання. CNN можна використовувати для виділення ознак із вхідного зображення, а наступна повністю підключена нейронна мережа прогнозує клас техніки. CNN зазвичай включає кілька шарів, які можуть навчитися розпізнавати та витягувати різні характеристики зображення, такі як краї, кути та текстури. Повністю пов'язана нейронна мережа приймає ці характеристики як вхідні дані та створює вихідні дані, які представляють передбачуваний технічний об'єкт.[5]

## 1.2 Основна інформація про згорткові нейронні мережі

Згорткові нейронні мережі (CNN) – це один з ключових методів у сфері комп'ютерного бачення, для обробки та аналізу візуальної інформації. Основною ознакою таких мереж є здатність ефективно аналізувати просторову структуру даних, таких як наприклад пікселі на зображеннях.

У згорткових нейронних мережах є декілька наборів шарів, що складаються з набору фільтрів (ядра), що рухаються по масиву вхідних даних та виконують

операцію згортки, вона полягає в обчисленні скалярного добутку між значеннями фільтрів та відповідними частинами вхідних даних. Після шару згортки використовуються агрегувальні шари для зменшення розмірності вхідних даних, при цьому зберігаючи важливі ознаки. Після цих шарів використовується повноз'єднані шари вони зв'язують кожен нейрон одного шару з кожним нейроном іншого шару, та відповідають за виконання завершальної обробки отриманих ознак та за завдання що до класифікації, локалізації, стеження на основі отриманих ознак. Вид згорткової нейронної мережі можна побачити на рисунку 1.1.

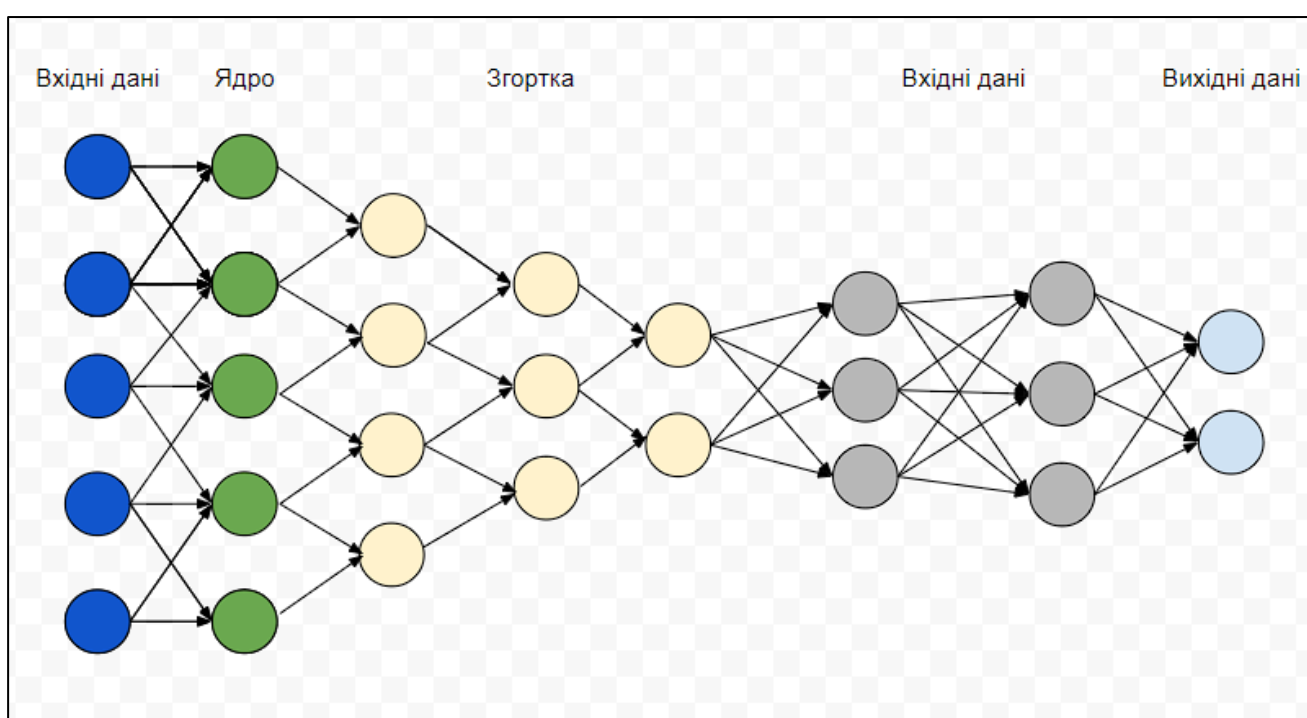


Рисунок 1.1 – Вид згорткової нейронної мережі (виконано самостійно)

На рисунку зображені основні шари згорткової мережі від вхідних даних з ядрами, до вихідних даних.

### 1.3 Використання згорткових нейронних мереж для відстеження технічних об'єктів за допомогою дронів в різних галузях

Використання згорткових нейронних мереж (CNN) для відстеження технічних об'єктів за допомогою дронів має великий потенціал у різних галузях, адже дрони це універсальний безпілотний літальний пристрій, різних розмірів,

форм та функціональних можливостей. Основною функцією дрона це відео- та фотозйомка, а згорткові нейронні мережі можуть використовуватись для аналізу, класифікації та стеження за об'єктами на фото та відео матеріалах з дрону.

Використання дронів та CNN можуть себе проявити в інфраструктурному моніторингу, тобто для огляду різних інфраструктурних об'єктів, таких як: мости, дороги, трубопроводи, будинки, дамби та інші. Таке використання допоможе точно відстежувати потенційні проблеми в інфраструктурних об'єктах та вчасно їх вирішувати.

Іншою галуззю де можна влучно використовувати дронів з комп'ютерним зором є пошуково-рятувальні операції. Такі системи можуть використовуватись для пошуку людей, техніки та інших речей в районах надзвичайних ситуацій на відкритій території.

Наступною важливою галуззю використання дронів з комп'ютерним баченням це транспорт та логістика. Дрони можуть використовуватись для відстежування рухів різних транспортних засобів в умовах де не зручно використовувати звичайне відеоспостереження.

Неоднозначною, але також важливою та актуальною галуззю використання дронів з комп'ютерним зором є військове використання згорткових нейронних мереж для відстежування технічних об'єктів за допомогою дронів. Так у цій сфері дрони можуть використовуватись для розвідки, виявлення ворожих цілей та моніторингу військових об'єктів. Таке використання допоможе автоматично визначати, класифікувати та відстежувати різні об'єкти такі як: військова техніка, військові інфраструктурні об'єкти, інженерні споруди, та інші.

Отже використання згорткових нейронних мереж у поєднанні з дронами виявляє значний потенціал і різних сферах нашого життя, допомагаючи моніторити навколишнє середовище в режимі реального часу, такий підхід дозволяє швидко і точно реагувати на можливі загрози та надзвичайні ситуації.

На приклад для відстежування людей та техніки в критичних ситуаціях, можна використовувати скелетний метод на основі 3D координат суглобів [15]. Для

точного відстеження поз та переміщень людей та техніки в різноманітних ситуаціях та положень техніки з якою вони взаємодіють.

Окремо визначається саме використання дронів у військовій справі, оскільки це є актуальним напрямком їх застосування. Військові безпілотні літальні апарати надають велику кількість переваг під час використання, таких як розвідка, без високого ризику для життя пілотів дрону, змога наносити високоточні удари, вести спостереження за бойовими діями в реальному часі, та реагувати на зміни бойового процесу, що може зберегти особовий склад та техніку під час ведення бою.

У військовому ділі дрони поділяються на розвідувальні, що мають кращу якість зображення, більший час та дальність польоту, але є більшими за розмірами, та дорожчими в ціні, тому використовуються для багаторазового запуску, та на ударні, які несуть з собою бойовий заряд, є дешевшими і це їх основна перевага, з менш якісними камерами, передає аналоговий сигнал, та є одноразовими, адже при використанні несуть підготовлений до бойового застосування заряд, та несуть його до позицій ворога або ворожої техніки. Використовуючи методи комп'ютерного зору на відеоінформації з дрону надає перевагу в класифікації та відстежуванні техніки що дає значну перевагу в швидкості ідентифікації цілі, в майбутніх розробках можливе автоматичне застосування дронів без участі оператора, проте цей процес може зіткнутись з різними труднощами під час своєї роботи, це деякі з них:

- складність в ідентифікації техніки, що відрізняється не значними деталями;
- складність у розрізненні ворожої та союзної техніки, якщо вони мають багато спільних прикмет;
- складність в роботі під час несприятливих атмосферних явищ;
- складність в застосуванні дрона в радіусі враження різними засобами радіоелектронної боротьби;
- складність в розпізнаванні замаскованої техніки, або техніки з специфічним забарвленням;

- складність у роботі в режимі реального часу при обмеженій кількості ресурсів.

Для подолання цих труднощів потрібно використовувати комплексний підхід. Тренувати модель на більшому наборі даних, модернізація дронів за для покращення їх захищеності проти систем радіоелектронної боротьби. Також можна спробувати використати метод видалення низькочастотних складових зображення за допомогою алгоритму швидкого медіанного фільтра, він використовуються для підвищення відношення сигнал / шум, зменшення динамічного діапазону зображень і зниження часу реалізації обчислювальних методів оброблення великих зображень [16].

#### 1.4 Проблема що досліджується

На даний момент існує велика кількість різних моделей щодо об'єктного виявлення на фото та відео даних. Всі вони відрізняються одна від іншої в залежності від швидкості, точності, використання ресурсів, робота в обмеженій кількості ресурсів, стійкість до змін у вхідних даних. Для порівняння були обрані наступні моделі:

- YOLOv7, як основна версія моделі;
- YOLOv7 tiny, як версія що потребує менше ресурсів;
- SSD-MobileNet-v2, як конкуруюча модель.

Порівнюючи ці моделі можна буде визначити яка з них може ефективніше використовуватись для класифікації та відстеження технічних об'єктів за допомогою дронів.

#### 1.5 Постановка технічного завдання кваліфікаційної роботи

Самою основною темою магістерського дослідження в цілому – є перевірка методів для розпізнавання технічних об'єктів за допомогою дрона.

Розіб'ємо задачу на під задачі, котрі необхідно буде розв'язати в ході магістерської кваліфікаційної роботи, якщо не брати до уваги теоретичний матеріал:

- підготувати комплексний набір даних для тренування обраних моделей;
- провести навчання обраних моделей на наборі даних що відповідає темі магістерського дослідження;
- записати дані про процес навчання моделей, задля перевірки швидкості їх навчання;
- порівняти отримані дані з навчання обраних моделей за певними критеріями;
- порівняти роботу обраних та навчених моделей на однаковому наборі даних та однаковій конфігурації системи;
- зробити та обґрунтувати висновки, що до ефективності навчених моделей для виконання поставленої задачі.

Після створення технічного завдання магістерського дослідження, перейдемо до обирання засобів проведення даного дослідження, опису методів які будуть використані під час експерименту та підготовки до проведення самого магістерського дослідження.

#### 1.6 Засоби що будуть використані для проведення дослідження

Виконання мого магістерського дослідження буде зроблено за допомогою мови програмування Python.

В середовищі розробки Visual Studio Code за допомогою окремих розширень.

TensorFlow – це відкрита бібліотека машинного навчання, розроблена компанією Google Brain Team. Вона надає інструменти для побудови та навчання різних моделей машинного навчання, включаючи нейронні мережі [11]. TensorFlow надає гнучкі можливості для створення та навчання моделей на різних рівнях абстракції, що робить її придатною для завдань класифікації зображень , також TensorFlow має обширну документації що допоможе в правильному налаштуванні системи саме для конкретної мети.

PyTorch – це бібліотека машинного навчання з відкритим вихідним кодом, розроблена та підтримувана компанією Facebook's AI Research Lab (FAIR). Особливістю PyTorch є динамічний обчислювальний граф, що робить його більш

інтуїтивно зрозумілим та зручним для експериментування порівняно з деякими іншими бібліотеками, такими як TensorFlow, який використовує статичний обчислювальний граф.

Саме навчання даних систем буде виконуватись у сервісі Google Colab. Google Colab (Colaboratory) – це сервіс від Google, що надає безкоштовне середовище для запуску коду Python у хмарі з Jupyter Notebook. [6] Він дозволяє користувачам створювати та запускати Python-скрипти, аналізувати дані, проводити навчання моделей машинного навчання та багато іншого, прямо у браузері, без необхідності встановлення будь-якого спеціального програмного забезпечення на своєму комп'ютері, він надає доступ до графічних процесорів (GPU) та тензорних процесорів (TPU) Google для прискорення обчислень, а також інтегрований з іншими сервісами Google, такими як Google Drive.

Для підготовки набору даних була використане програмне забезпечення LabelImg – це програмне забезпечення для розмітки зображень, яке часто використовується в галузі машинного навчання, особливо у контексті навчання моделей комп'ютерного зору. Воно надає користувачеві зручний інтерфейс для створення анотацій на зображеннях, що дозволяє розмічати об'єкти на зображеннях та вказувати їхнє розташування, форму та клас.

Roboflow – це платформа для розробки комп'ютерного зору, яка допомагає інженерам та розробникам створювати та розгортати моделі машинного навчання для обробки зображень та відео.

## 2 ОПИС МЕТОДІВ КОМП'ЮТЕРНОГО ЗОРУ ДЛЯ РОЗПІЗНАВАННЯ ТЕХНІЧНИХ ОБ'ЄКТІВ ЗА ДОПОМОГОЮ ДРОНІВ

### 2.1 Опис обраних методів

Дрони стають все більш поширеним інструментом для завдань моніторингу, тому для більш ефективного виконання цього завдання необхідно використовувати методи комп'ютерного зору, які дозволяють дронам розпізнавати та класифікувати об'єкти на зображеннях у реальному часі.

Для швидкої роботи в режимі реального часу потрібно використовувати однопрохідний підхід в обробці зображень, при якому зображення аналізується і інтерпретується тільки один раз, в такому підході алгоритм обробляє кожен піксель зображення лише один раз без зворотних переходів чи додаткових ітерацій.

Такий підхід має низку переваг, включаючи більш ефективне використання обчислювальних ресурсів та більш швидку обробку зображень, ці переваги стали основними критеріями при виборі моделей для використання щоб розпізнавати технічні об'єкти за допомогою дронів в реальному часі і в умовах обмежених ресурсів.

При цьому такий підхід є менш точним та нездатним для виявлення складних, замаскованих або частково прихованих об'єктів на зображеннях, що потребують більш складних ітеративних методів аналізу зображень.

Тому мною були обрані наступні однопрохідні моделі комп'ютерного зору:

- YOLOv7 – це сьома версія моделі YOLO (You Only Lock Once), що є одною з найпопулярніших архітектур для задач виявлення об'єктів у реальному часі;
- YOLOv7-tiny – це зменшена версія архітектури YOLO, розроблена спеціально для задач в умовах обмежених обчислювальних ресурсів;
- SSD-MobileNet-v2 – це комбінація двох архітектур SSD (Single Shot Multibox Detector) та MobileNet-v2. SSD відрізняється тим що має єдину згорткову мережу для передбачення рамок об'єктів;



Функція втрат YOLO використовує суму квадратів втрат та складається з кількох компонентів:

- помилки локалізації рамок об'єкту;
- помилки класифікації об'єктів;
- помилки впевненості (помилки в об'єктності локалізованої рамки).

Загальна функція втрат, що сумує вище наведені в списку помилки (2.1):

$$\begin{aligned}
 & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] \\
 & + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left[ (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right] \\
 & + \lambda_{obj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} (C_i - \hat{C}_i)^2 \\
 & + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{noobj} (C_i - \hat{C}_i)^2 \\
 & + \sum_{i=0}^{S^2} 1_i^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2
 \end{aligned} \tag{2.1}$$

де перші два доданка відповідають за локалізацію  $(x, y)$  та  $(w, h)$  об'єкта, при умові, якщо  $1_{ij}^{obj} \neq 0$ , тобто коли на фото є об'єкт;

третій та четвертий доданки відповідають за помилки впевненості, при умові, якщо  $1_{ij}^{obj} \neq 0$ ;

останній доданок відповідає за класифікацію об'єкта в клітинці  $i$ , при умові, якщо  $1_{ij}^{obj} \neq 0$ .

Після обчислення функції втрат у процесі навчання нейронної мережі YOLO використовується метод зворотного розповсюдження помилки корекції параметрів моделі. Цей метод дозволяє визначити, як потрібно змінити ваги нейронів у кожному прошарку мережі, щоб мінімізувати функцію втрат.

Цей процес оновлення ваг повторюється на кожній ітерації навчання для кожного міні-батчу даних, поки не буде досягнуто умови зупинки, таке як досягнення певної кількості епох навчання або збіжність моделі. Після завершення

навчання, модель можна використовувати для виявлення об'єктів у нових зображеннях.

YOLOv7 є значним розвитком стандартної архітектури YOLO. Воно включає ряд поліпшень, які спрямовані на підвищення швидкості і точності виявлення об'єктів. Так, якщо порівнювати з попередньою версією, то максимальна точність YOLOv7-E6E (56,8% AP) моделі реального часу на +13,7% AP вище, ніж поточна найточніша модель YOLOv6-s (43,1% AP) на наборі даних COCO [7]. Так сьома версія пропонує модульну архітектуру, яка дозволяє легко налаштовувати та адаптувати модель під конкретні завдання та вимоги. Це включає можливість зміни шарів, налаштування параметрів і використання різних методів навчання і аргументації даних.

YOLOv7-tiny – полегшена модель YOLOv7, зазвичай має меншу кількість шарів і параметрів в порівнянні з повною версією. Це досягається за рахунок зменшення глибини, шляхом використання більш легких згорткових фільтрів і оптимізації архітектури для підвищення продуктивності на пристроях з обмеженими обчислюваними ресурсами. Відмінно від попередніх версій YOLO, YOLOv7-tiny пропонує ефективну архітектуру мережі ELAN і пропонує метод допоміжної голови для збільшення витрат на тренування та покращення точності без впливу на час висновків [8]. Схема потоку алгоритму наведена на рисунку 2.2.



Рисунок 2.2 – Схема потоку алгоритму YOLOv7-tiny (за даними [8])

Однак, за рахунок зменшення розміру і складності моделі YOLOv7-tiny може спостерігатися деяке зниження точності виявлення об'єктів у порівнянні з повною версією YOLOv7. Тем не менше, YOLOv7-tiny залишається ефективним інструментом для задач виявлення об'єктів у реальному часі на пристроях з обмеженими ресурсами, де потрібен баланс між продуктивністю та точністю.

## 2.3 Архітектура SSD

Архітектура SSD (Single Shot Multibox Detector) являє собою інноваційний метод виявлення об'єктів на зображеннях, який включає кілька ключових компонентів для досягнення високої точності та ефективності. Архітектура SSD моделі знаходиться на рисунку 2.3.

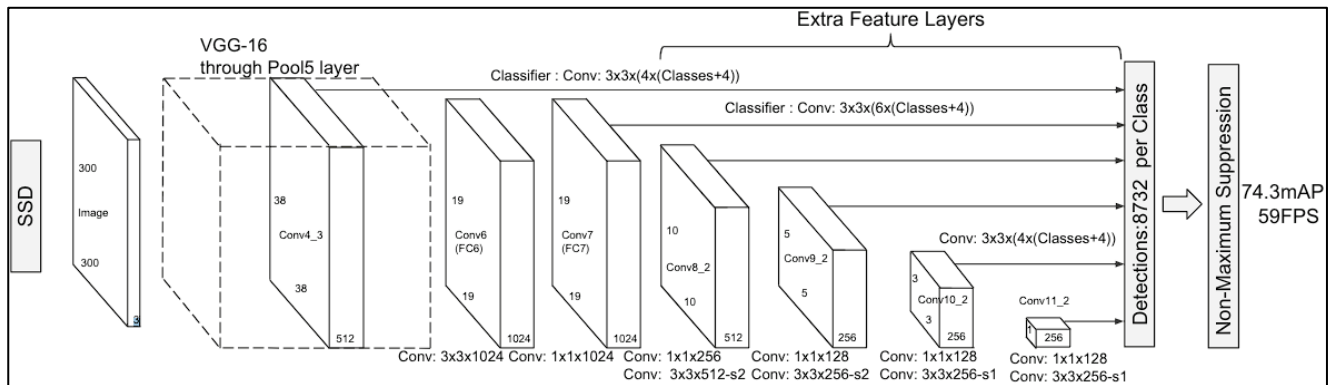


Рисунок 2.3 – Архітектура SSD моделі з архітектурою VGG16 (за даними [13])

Перша частина архітектури SSD - це згорткова нейронна мережа, що обробляє вхідне зображення і витягує ознаки з нього. Ці ознаки являють собою інформацію про текстуру, форму та контекст об'єктів на зображенні. Як згорткову мережу зазвичай використовують відомі архітектури, такі як VGG, ResNet або MobileNet, які попередньо навчені на великих наборах даних, на рисунку 2.3 показано приклад з використанням архітектурою VGG16, яку модель SSD модифікує, змінюючи повністю з'єднані шари на шари згортки, ці шари поступово зменшуються в розмірах і дозволяють прогнозувати виявлення об'єктів в різних масштабах.

Кожен доданий функціональний рівень (або, за бажанням, існуючий функціональний рівень з базової мережі) може створити фіксований набір передбачень виявлення за допомогою набору згорткових фільтрів [1]. Якщо подивитись на рисунок, то ці шари можна побачити у верхній частині архітектури SSD моделі, так для шару ознак розміром  $m \times n$  з  $p$  каналами, базовим елементом для прогнозування параметрів потенційного виявлення об'єкту є невелике ядро

$3 \times 3 \times p$ , яке генерує оцінку для класифікації чи зміщення форми відносно координат стандартної рамки. Таким чином в кожній з  $m \times n$  позицій де застосовується ядро, генерується вихідне значення. Всі вихідні значення зміщення обмежувальної рамки вимірюються відносно позиції стандартної рамки, що була розрахована на початкових функціональних рівнях, для кожної позиції на карті ознак.

В кінці знаходяться класифікаційні шари, що відповідають за виявлення заданих класів у об'єктах на зображенні, та регресійного, що відповідає за відображення прямокутників навколо об'єктів, та вирівнює їх по реальним розмірам об'єкта, також на цьому етапі розраховується загально цільова функція втрат, що впливає на зміну вагових коефіцієнтів при кожній ітерації роботи алгоритму.

Загальна функція втрат є зваженою сумою втрат локалізації та втрат впевненості (2.2)

$$L(x, c, l, g) = \frac{1}{N} (L_{conf}(x, c) + \alpha L_{loc}(x, l, g)) \quad (2.2)$$

де  $N$  – кількість зіставлених стандартних рамок;

$L_{conf}(x, c)$  – втрата впевненості, представляє собою softmax функцією для оцінок впевненості ( $c$ ) для кожного класу;

$L_{loc}(x, l, g)$  – втрата локалізації між параметрами передбаченої рамки ( $l$ ) та еталонної ( $g$ );

$\alpha$  – гіперпараметр для балансування двох складових втрат.

Таким чином функція втрат для моделі SSD (2.2) складається з двох основних частин, де перша відповідає за правильну класифікацію об'єкту, а друга за передбачення її точного місцезнаходження, та побудови точної рамки навколо об'єкту.

Як згорткову мережу для SSD в ході роботи буде використано MobileNet. Архітектура цієї згорткової мережі спочатку була створена для виконання задач по виявленню об'єктів у мобільних пристроях та для вбудованих програм комп'ютерного зору. Своєї ефективності CNN MobileNet досягає завдяки реалізації глибинних роздільних звивин, об'єднуючи глибинні згортки та поточкові (1x1) згортки[10]. На рисунку 2.4 показано базове представлення архітектури моделі MobileNet.

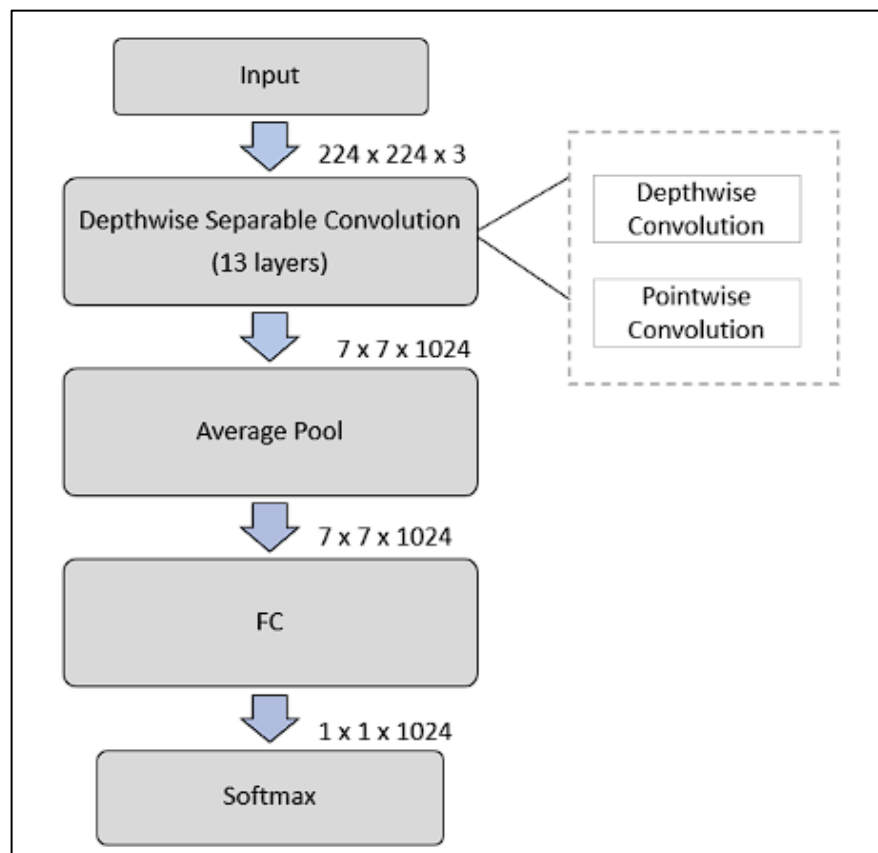


Рисунок 2.4 – Базова архітектура моделі MobileNet (за даними [10])

Тому така унікальна комбінація зменшує кількість параметрів в порівнянні з CNN, що використовують звичайні згортки такої самої глибини, дозволяючи MobileNet створювати легкі глибинні нейронні мережі, що пропонують баланс між оптимальним часом затримки та зменшеним розміром моделі. Керувати цим балансом можна завдяки двом гіперпараметрам  $\alpha$  (множник ширини) та  $\rho$  (множник глибини або роздільної здатності)[10]. Таким чином гіперпараметр  $\alpha$

відповідає за кількість каналів на кожному рівні, а  $\rho$  за просторові розміри вхідних тензорів, змінюються вони від 1 до 0, зменшуючи параметри  $\alpha$  та  $\rho$  зменшується точність розпізнавання об'єктів, але в ту ж чергу збільшується швидкість роботи та зменшується кількість ресурсів.

#### 2.4 Постановка гіпотез

Після огляду досліджуваних алгоритмів, можна сформулювати наступні гіпотези:

- модель SSD має більшу точність ніж версії YOLO, оскільки в SSD використовується багато конвекційних шарів згортки, що покращують точність;
- YOLOv7 працює швидше ніж модель SSD, оскільки використання великої кількості конвекційних шарів сповільнює роботу моделі;
- YOLOv7-tiny працює швидше ніж YOLOv7 та SSD, оскільки це спрощена версія YOLOv7 вона має працювати швидше та використовувати менше ресурсів;
- YOLOv7-tiny найменш точна модель серед розглянутих, оскільки має спрощену архітектуру заради більшої швидкості роботи;
- YOLOv7 використовує більше ресурсів ніж SSD-MobileNetv2, через свою велику кількість параметрів та складнішу архітектуру.

Описані вище гіпотези будуть перевірятися в ході проведення експерименту, що буде описаний в наступній частині кваліфікаційної роботи.

### 3 ОГЛЯД РЕАЛІЗАЦІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

#### 3.1 Програмна реалізація розрахунку значення точності для моделей

Для того, щоб порівняти обрані моделі комп'ютерного зору, потрібно використовувати певні метрики. Найкращою метрикою для такого порівняння це mAP, вона використовує IOU (Intersection Over Union).

IOU – це метрика, яка визначає ступіть перекриття двох областей, щоб її визначити потрібна наявність двох обмежувальних коробок об'єкту, одна розрахована за допомогою методу комп'ютерного зору, інша має бути достовірною, тому береться із введених людиною даних. Застосовуючи цю метрику можна об'єктивно оцінити наскільки вірно модель передбачає об'єкти та його межі, значення можуть бути вірними TP (True Positive) чи хибними FP (False Positive).

Розраховується ця метрика як відношення площі перетину TP та FP обмежувальних боксів до загальної площі цих боксів, за формулою (3.1).

$$IOU = \frac{\text{area of overlap}}{\text{area of union}} \quad (3.1)$$

де *area of overlap* – це площа перетину TP та FP;

*area of union* – загальна площа TP та FP.

Візуалізацію цієї формули можна побачити на рисунку 3.1

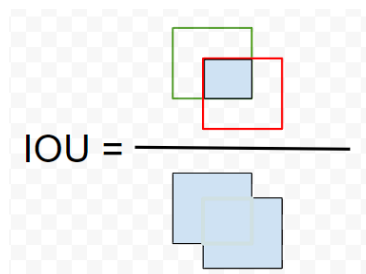


Рисунок 3.1 – Візуалізація формули IOU. (виконано самостійно)

Програмна реалізація по знаходженню цієї метрики реалізована на мові програмуванні Python, вона зчитує дані про реальні та передбачені об'єкти з

текстових файлів, розраховує метрику IOU та зберігає ці дані в текстовий документ, або може візуалізувати їх на графіку.

Таким чином на рисунку 3.2 можна побачити реалізації функцію, що завантажує з вказаної директорії реальні дані по знаходженню об'єкту. Вона отримує файли формату .txt з введеними обмежувальними рамками під час створення дата сету, тому вони вважаються TP.

```

for txt_file in dr_files_list:
    file_id = os.path.basename(os.path.normpath(txt_file)).split(".txt", 1)[0]
    lines = file_lines_to_list(txt_file)
    for line in lines:
        class_name, confidence, left, top, right, bottom = line.split()
        if class_name not in dr_classes:
            dr_classes.append(class_name)
        bbox = f"{left} {top} {right} {bottom}"
        detection_data = {"confidence": float(confidence), "file_id": file_id, "bbox": bbox}
        # Додавання до списку передбачень для відповідного класу
        if class_name in bbox:
            bbox[class_name].append(detection_data)
        else:
            bbox[class_name] = [detection_data]

```

Рисунок 3.2 – Функція для зчитування TP даних (виконано самостійно)

А на рисунку 3.3 функцію для обробки та завантаженню файлів з передбаченими моделлю результатами.

```

dr_files_list = glob.glob(DR_PATH + '/*.txt')
dr_files_list.sort()
dr_classes = list(gt_counter_per_class.keys())

for txt_file in dr_files_list:
    file_id = os.path.basename(os.path.normpath(txt_file)).split(".txt", 1)[0]
    lines = file_lines_to_list(txt_file)
    for line in lines:
        class_name, confidence, left, top, right, bottom = line.split()
        if class_name not in dr_classes:
            dr_classes.append(class_name)
        bbox = f"{left} {top} {right} {bottom}"
        detection_data = {"confidence": float(confidence), "file_id": file_id, "bbox": bbox}
        # Додавання до списку передбачень для відповідного класу
        if class_name in bbox:
            bbox[class_name].append(detection_data)
        else:
            bbox[class_name] = [detection_data]

```

Рисунок 3.3 – Функція для зчитування FP даних (виконано самостійно)

Вона приймає в себе файли в форматі .txt, з обмежувальними рамками які передбачила модель комп'ютерного зору, вони вважаються за FP.

Головною функцією цього програмного забезпечення є функція підрахунку метрики IOU, зображений він на рисунку 3.4

```

sum_AP = 0.0
ap_dictionary = {}
lamr_dictionary = {}
# open file to store the output
with open(output_files_path + "/output.txt", 'w') as output_file:
    output_file.write("# AP and precision/recall per class\n")
    count_true_positives = {}
    for class_index, class_name in enumerate(gt_classes):
        count_true_positives[class_name] = 0
        # Load detection-results of that class
        dr_file = TEMP_FILES_PATH + "/" + class_name + "_dr.json"
        dr_data = json.load(open(dr_file))
        # Assign detection-results to ground-truth objects
        nd = len(dr_data)
        tp = [0] * nd # creates an array of zeros of size nd
        fp = [0] * nd
        for idx, detection in enumerate(dr_data):
            file_id = detection["file_id"]
            # open ground-truth with that file_id
            gt_file = TEMP_FILES_PATH + "/" + file_id + "_ground_truth.json"
            ground_truth_data = json.load(open(gt_file))
            ovmax = -1
            gt_match = -1
            # load detected object bounding-box
            bb = [ float(x) for x in detection["bbox"].split() ]
            for obj in ground_truth_data:
                # look for a class_name match
                if obj["class_name"] == class_name:
                    bbgt = [ float(x) for x in obj["bbox"].split() ]
                    bi = [max(bb[0],bbgt[0]), max(bb[1],bbgt[1]), min(bb[2],bbgt[2]), min(bb[3],bbgt[3])]
                    iw = bi[2] - bi[0] + 1
                    ih = bi[3] - bi[1] + 1
                    if iw > 0 and ih > 0:
                        # compute overlap (IoU) = area of intersection / area of union
                        ua = (bb[2] - bb[0] + 1) * (bb[3] - bb[1] + 1) + (bbgt[2] - bbgt[0]
                            + 1) * (bbgt[3] - bbgt[1] + 1) - iw * ih
                        ov = iw * ih / ua
                        if ov > ovmax:
                            ovmax = ov
                            gt_match = obj
            if not bool(gt_match["used"]):
                # true positive
                tp[idx] = 1
                gt_match["used"] = True
                count_true_positives[class_name] += 1
                # update the ".json" file
                with open(gt_file, 'w') as f:
                    f.write(json.dumps(ground_truth_data))

```

Рисунок 3.4 – Функція обчислення метрики IOU (виконано самостійно)

Функція починається з ініціалізації змінних, а саме sum\_AP для зберігання сумарної середньої точності для всіх класів (mAP), а також змінні для зберігання середньої точності та логарифмічного рангу для кожного класу.

Наступним кроком створюється та відкривається текстовий документ з назвою `output.txt` та ще одна змінна для збереження кількості TP для кожного класу.

Потім отримується список всіх класів об'єктів які відмічені в моделі і завантажуються до файлу результати виявлення об'єктів для кожного класу FP.

Далі відбувається ініціалізації змінних `nd`(кількість об'єктів), `tr` та `fr` списки розміром `nd` заповнені нулями, які потім використовуються для відстеження того чи було правильне співставлення знайденого об'єкта з об'єктом у реальних даних якщо мова йде про `tr`, та для помилково знайдених об'єктах якщо мова йде про `fr`.

Далі починається цикл який, який створений для обробки кожної детекції об'єкту зі списку всіх об'єктів `dr_data`, які містять результати розпізнавання для певного класу техніки `class_name`. Зчитування `file_id` що є унікальним ідентифікатором файлу до якого відноситься поточний розрахунок, далі відкривається шлях до файлу з реальними даними про розпізнавання об'єкту для відповідного `file_id`, та зчитування з відти даних про об'єкт. Потім ініціалізуються змінні `ovmax` та `gt_match`, де перша буде відповідати за максимальне значення перекриття яке було знайдено між `tr` та `fr`, а друга за зберігання посилання на об'єкт для якого проводяться розрахунки, а також отримання координат в змінну `bb` (`xmin`, `ymin`, `xmax`, `ymax`) для області (`bounding box`) для поточного розпізнавання.

Після ініціалізації змінних та посилань відкривається цикл з розрахунком IOU, в якому для кожного об'єкта в файлі `.json` з реальними даними про об'єкт, які мають спільний клас (`classname`) виконуються наступні дії:

- зчитуються координати `bbgt` (реальної обмежувальної рамки);
- обчислюється область перетину (`ih, iw`) між розпізнаваним об'єктом та реальною обмежувальною рамкою;
- обчислюється проща об'єднання (`ua`) так коефіцієнт перекриття `IOU(ov)`;
- у випадку якщо (`ov`) більше ніж `ovmax`, то оновлюється значення `ovmax`, тобто відбувається вибір об'єкта для поточного розпізнавання на основі значення його перекриття (`IOU`);
- якщо `gt_match` не використовувався, тобто об'єкт ще не використовувався, то індекс `tr` встановлюється як 1, що значить що це правдиве співпадіння,

збільшується лічильник `count_true_positives` для кожного класу об'єктів і оновлюється файл з новими даними, і такий об'єкт позначається як використаний.

Після цього відбувається запис та збереження розрахованих даних до файлу `output.txt`, дані які містить цей файл що можна побачити на рисунку 3.5.

```
# mAP of all classes
mAP = 53.15%

# Number of ground-truth objects per class
BMP-BMD: 2
BMPBMD: 51
BTR: 28
KAMAZ: 2
MTLB: 9
RLS: 4
RSZO: 12
SAU: 13
TANK: 95
URAL: 8
ZRK: 11

# Number of detected objects per class
BMPBMD: 65 (tp:29, fp:36)
BTR: 35 (tp:19, fp:16)
KAMAZ: 3 (tp:2, fp:1)
MTLB: 9 (tp:6, fp:3)
RLS: 3 (tp:3, fp:0)
RSZO: 8 (tp:7, fp:1)
SAU: 11 (tp:6, fp:5)
TANK: 135 (tp:53, fp:82)
URAL: 9 (tp:5, fp:4)
ZRK: 9 (tp:7, fp:2)
```

Рисунок 3.5 – Результат роботи програмного забезпечення для моделі SSDMobileNet-v2 (виконано самостійно)

Так в результаті роботи в файлі `output` можна побачити метрику `mAP`, кількість всіх об'єктів та кількість `TP` та `FP` для кожного класу.

## 4 ПРОВЕДЕННЯ ЕКСПЕРИМЕНТАЛЬНОГО ДОСЛІДЖЕННЯ

### 4.1 Підготовка набору даних для навчання обраних моделей

Для підготовки 2000 зображень військової техніки знятих з дронів з такими класами як: BMP-BMD, BTR, KAMAZ, MTLB, RLS, RSZO, SAU, URAL, ZRK. Я використав програму LabelImg, завантаживши до неї ці зображення і виділяв об'єкти на фото рамками та помічав їх одним з вище перерахованих класом, дані зберігав в двох форматах .txt для YOLO та .xml для SSD. Для YOLO був створений файл data.yaml, котрий містить в собі шляхи до тренувального набору, набору валідації та до тестового набору, а також вище названі класи технічних засобів для виявлення, його вміст можна побачити на рисунку 4.1.

```
1 train: ../train/images
2 val: ../valid/images
3 test: ../test/images
4
5 nc: 10
6 names: ['BMP-BMD', 'BTR', 'KAMAZ', 'MTLB', 'RLS', 'RSZO', 'SAU', 'TANK', 'URAL', 'ZRK']
7
```

Рисунок 4.1 – Вміст файлу data.yaml (виконано самостійно)

В свою чергу для SSD який працює на бібліотеці TensorFlow потрібно перевести дані з формату .xml до формату .csv в зв'язку з особливостями роботи бібліотеки TensorFlow, тому був написаний скрипт на мові програмування Python.

Далі цей набір даних завантажив до Google диску, щоб підключити його до сервісу Google Colab, де буде проходити навчання обраних моделей.

### 4.2 Навчання обраних моделей.

Навчання відбувалось на сервісі Google Colab, з типом середовища виконання Python, графічним процесором NVIDIA L4 Tensor Core з 22.5 гігабайт відео пам'яті, та 53 гігабайти оперативної пам'яті. З використанням тензорних ядер, що значно пришвидшують процес навчання. Була придбана підписка Colab Pro, що надала доступ до 100 обчислювальних блоків.

Для навчання моделей YOLOv7 та YOLOv7-tiny використовувалась бібліотека PyTorch, а для SSD була використана бібліотека TensorFlow. Вони були встановлені в середовище виконання Colab з репозиторіїв на GitHub. Для навчання YOLOv7 та YOLOv7-tiny були використані наступні інструкції: batch 16, epochs 50. Це означає, що в один момент часу буде оброблятися одразу 16 зображень, та буде 50 проходів через весь набір даних під час навчання. Для моделі SSD-MobileNet-v2 були використані наступні інструкції: batch 16, num\_steps 40000. Параметр num\_steps відповідає за кількість кроків, що буде виконано перед завершенням навчання.

Час навчання, параметри запуску, розмір зображень моделей зображено в таблиці 4.1.

Таблиця 4.1 – Параметри моделей, та час їх навчання.

Модель що навчалась	Час навчання (хв.)	Зображень за крок	Кількість	Розмір зображень
SSD-MobileNet-v2	146.5	16	40000 кроків	320x320
YOLOv7	135.6	16	55 епох	640x640
YOLOv7-tiny	68.3	16	45 епох	640x640

З таблиці 4.1 зрозуміло, що довше за все навчалась SSD-MobileNet-v2, швидше за все пройшла навчання полегшена версія YOLOv7, YOLOv7-tiny.

В результаті було отримано три моделі, які були навчені розрізняти військову техніку за наступними класами: BMP-BMD, BTR, KAMAZ, MTLB, RLS, RSZO, SAU, URAL, ZRK.

Для перевірки успішності навчання можна подивитись на графіки функцій загальних витрат для кожної моделі, що була натренована.

Так для моделі SSD-MobileNet-v2 цей графік функції можна побачити на рисунку 4.2.

З цього рисунка видно що втрати на протязі навчання зменшувались, що свідчить про те, що модель вдало навчалась, змінюючи ваги для зменшення кривої втрат з кожним кроком.



Рисунок 4.2 – Графік функції загальних втрат для моделі SSD-MobileNet-v2 (виконано самостійно)

Для моделі YOLOv7 графік функції загальних витрат можна побачити на рисунку 4.3.

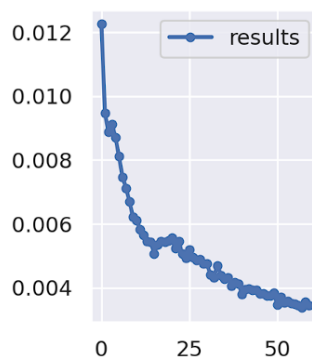


Рисунок 4.3 – Графік функції загальних втрат для моделі YOLOv7 (виконано самостійно)

Схожий з минулим виявився і графік функції загальних втрат для моделі YOLOv7, ми бачимо стрімкий спад спочатку, і потім з кожною наступною епохою він зменшується.

Для моделі YOLOv7-tiny графік функції загальних втрат можна побачити на рисунку 4.4.

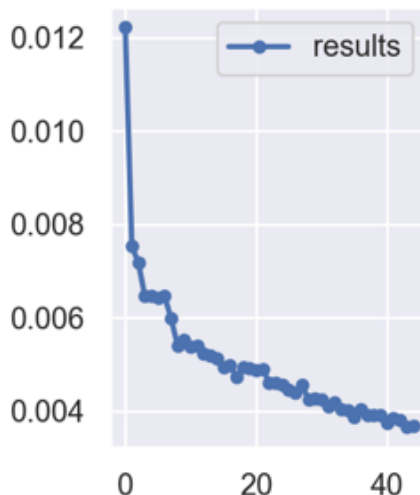


Рисунок 4.4 – Графік функції загальних втрат для моделі YOLOv7-tiny (виконано самостійно)

Графік функції втрат для моделі YOLOv7-tiny показує дуже схожі результати з попередніми графіками.

З графіків функцій загальних втрат для обраних моделей можна зрозуміти, що вони успішно пройшли навчання, відносно кількості епох(кроків) навчання та вмісту набору даних для їх навчання, тому можна перевірити їх роботу в розпізнаванні об'єктів.

### 4.3 Результати роботи моделей на зображеннях

Тепер можна запустити їх на наборі тестових даних, що відмінні від тих даних на яких ці моделі тренувались. І подивитись на точність моделей в роботі.

Так на рисунку 3.5 зображений результат роботи моделі SSD-MobileNet-v2, на зображенні з декількома класами техніки, а саме клас "TANK" з точністю 92%, та клас техніки "BTR" з точністю 94%, що є високим показником точності класифікації об'єкта, при цьому взявши об'єкти які знаходяться попарно в один

прямокутник, що зв'язано з особливостями архітектури SSD моделі, проте модель не ви ділила окремо кожну одиницю техніки, а взяла їх в дві спільні рамки.



Рисунок 4.5 – Результат роботи моделі SSD (виконано самостійно)

На наступному рисунку 3.6 зображений результат роботи YOLOv7 на тому самому фотоматеріалі.



Рисунок 4.6 – Результат роботи моделі YOLOv7 (виконано самостійно)

Подивившись на рисунок, можна зрозуміти, що модель YOLOv7 знайшов два класи об'єктів на зображенні.

При цьому для класу “TANK” точність склала 87% та 77%, а точність для класу BTR склала 0.56%. Що є непоганим результатом моделі відносно об'єму набору даних та кількості епох навчання.

На наступному рисунку 4.7 зображений результат роботи моделі YOLOv7-tiny на відповідному фотоматеріалі.



Рисунок 4.7 – Результат роботи моделі YOLOv7-tiny (виконано самостійно)

Тут ми бачимо, що версія YOLOv7-tiny знайшла тільки один клас техніки під назвою “TANK” з точністю в 43%, а клас “BMP” не був знайдений, що й не дивно, при тому, що ця модель при тому самому наборі навчальних даних, пройшла їх майже в двічі швидше.

Якщо протестувати моделі на фото меншої роздільної здатності, то можна отримати результат що зображений на рисунку 3.8, на якому можна побачити, що SSD з точністю 92% виділив клас техніки “KAMAZ”, але не точно відобразив його обмежувальною рамкою, модель YOLOv7 краще впоралась зі створенням обмежувальної рамки для техніки класу “URAL” з точністю 41%, хоча й помітила розкидані навколо речі як клас техніки “RSZO”, модель YOLOv7-tiny в свою чергу

знайшла також знайшла на фото техніку класу “URAL” з точністю 40%, проте обмежувальна рамка не точно відображає габарити техніки на зображенні.



Рисунок 4.7 – Результат роботи моделей. (виконано самостійно)

Можна зробити проміжний висновок, що моделі працюють, виконують поставлені їм задачі по розпізнаванню військової техніки, і кожна справляється в міру її можливостей

#### 4.4 Аналіз роботи моделей на зображеннях

Для початку аналізу потрібно перш за все необхідно визначитись з критеріями для оцінювання роботи моделей SSD, YOLOv7 та YOLOv7-tiny. Ними можуть бути такі критерії як:

- точність виявлення об’єктів (mAP);
- точність (precision);
- повнота (recall);
- час на виявлення об’єктів;
- використання ресурсів;

Розібравшись з критеріями на які буде спиратись аналіз моделей, треба з’ясувати шлях отримання тих чи інших даних для критерію з роботи навчених моделей, точність (precision) – це вміння моделі розпізнавати тільки актуальні

об'єкти, та дорівнює відношенню вірно позитивних до всіх прогнозів, формула для знаходження точності (4.1) записана нижче.

$$Precision = \frac{TP}{TP + FP} \quad (4.1)$$

де TP – вірно позитивний прогноз точності (True Positive, TP);

FP – хибно позитивний прогноз точності (False Positive, FP).

Повнота (recall) – це вміння моделі знаходити всі об'єкти (будувати правильні обмежувальні рамки), та дорівнює відношенню вірно позитивних прогнозів до всіх об'єктів, що зазначені обмежувальною рамкою. Формула для повноти (4.2) зображена нижче.

$$Recall = \frac{TP}{TP + FN} \quad (4.2)$$

де TP – вірно позитивний прогноз рамки об'єкта (True Positive, TP);

FN – істинно хибний прогноз рамки об'єкта (False Positive, FP).

Вирахувавши ці два параметра можна оцінювати ефективність розпізнавання об'єктів за допомогою моделей.

При використанні тієї чи іншої моделі можна з'ясувати кількість пам'яті графічного процесора що вона використовує під час своєї роботи, це і буде метрикою використання ресурсів.

Час на виявлення об'єктів також зазначається при виконанні моделлю розпізнавання об'єктів на зображеннях

Точність виявлення об'єктів (mAP) – це середнє значення точності для всіх класів об'єктів, вираховується він як середнє арифметичне для всіх класів середньої точності для кожного класу [12]. Вона базується на середній інтеграції кривої Precision-Recall для кожного класу.

Дізнавшись про всі необхідні метрики можна перейти до складання таблиці порівняння моделей, в якій по вертикалі будуть обрані моделі, а по горизонталі наступні параметри: Назва моделі, значення точності (mAP), обсяг пам'яті GPU(ГБ), та час виявлення(мс). Побудуємо таблицю 3.2

Таблиця 4.2 – Результати роботи моделей.

Модель	значення точності (mAP)	обсяг пам'яті GPU(ГБ),	час виявлення(мс)
SSD-MobileNet-v2	53.15%	9.6	31
YOLOv7	51.3%	12.7	23
YOLOv7-tiny	35.5%	2.75	16

З отриманих результатів можна зробити певні висновки що до ефективності роботи навчених моделей, якщо повертатись до гіпотез, що були сформовані після огляду теоретичного матеріалу, переглянемо сформовані гіпотези:

- модель SSD дійсно має невелику перевагу в точності перед версіями YOLO, через свою архітектуру;
- модель YOLOv7 дійсно працює швидше ніж SSD;
- YOLOv7-tiny працює найшвидше серед обраних моделей;
- YOLOv7-tiny використовує менше всіх ресурсів;
- YOLOv7-tiny має найменшу точність;
- YOLOv7 використовує найбільше ресурсів серед обраних моделей.

Якщо говорити про практичне використання в умовах обмежених ресурсів, то найкращим варіантом буде модель YOLOv7-tiny, адже використовує майже в 5 раз менше ресурсів та працює швидше ніж інші моделі. Якщо головним аспектом практичного використання є точність, то ефективніше буде SSD-MobileNet-v2, через менше використання GPU, та більшу точність.

#### 4.5 Фактори що могли вплинути на результати досліджень

При навчанні моделей та їх використанні, існує багато факторів, що негативно впливають на їх роботу. На результати цього дослідження могли вплинути наступні фактори:

- недостатня якість чи кількість навчального масиву даних, 2000 зображень може бути не оптимальним значенням, і при розширенні набору даних якісними зображеннями ефективність роботи моделей може бути кращою;
- конфігурація програмного забезпечення також впливає на ефективність роботи, на деяких системах це може не працювати, через відсутність тензорних чи CUDA ядра в графічному процесорі;
- версія програмного забезпечення також має значення, якщо використовувати бібліотеки різних версій, як PyTorch чи TensorFlow можна отримати різні результати ефективності роботи;
- параметри навчання моделей також мають велике значення, а саме кількість епох та значення буферу прямо впливають на ефективність роботи моделей;

Найбільш сильно на результати цього дослідження вплинули саме параметри навчання моделей, а саме можливо не достатня кількість епох, так якщо орієнтуватись не на кількість епох а на час роботи, то результати дослідження були б більш об'єктивними, також недостатня кількість зображень може бути фактором що прямо впливає на точність та ефективність роботи моделей.

## ВИСНОВКИ

В результаті виконання магістерської дослідницької роботи були виконані всі поставлені задачі, метою яких було дослідити методи розпізнавання технічних засобів на відеоматеріалах з дрону. Для виконання цієї мети було обрано на порівняння три моделі комп'ютерного зору, а саме: YOLOv7, YOLOv7-tiny та SSD-MobileNetv2.

По вищеназваним моделям був розглянутий теоретичний матеріал, були розглянуті їх особливості архітектури, на основі цього матеріалу були сформувані тези, які перевірялись під час практичного експерименту, під час якої, ці моделі були навчені для виконання проблем по розпізнаванню військової техніки на матеріалах з дрону.

Було описане програмне забезпечення для обчислення метрики заради порівняння описаних моделей.

В результаті експериментального дослідження були отримані вичерпні результати з яких можна робити висновки щодо ефективного використання моделей у тих чи інших обставинах.

Модель YOLOv7-tiny потрібно використовувати в задачах, де найважливішими факторами є обмеженість в ресурсах та часі, адже ця модель показала як найшвидша та найменш вибаглива до апаратної складової пристрою, хоч і компромісним фактором в цій моделі виступає точність, проте при збільшенні часу та масиву даних для навчання ця модель зможе не критично відставати від більш точних та менш енергоефективних моделей.

SSD-MobileNetv2 слід використовувати в задачах, де основним критерієм є точність, адже завдяки своїй архітектурі, ця модель показує кращі результати в точності прогнозування, але гірші в швидкості роботи.

Модель YOLOv7 в ході дослідження була обрана як компромісний варіант між точністю та швидкістю роботи, вона всього на 2% менш точна за модель SSD та працює швидше, проте споживає найбільшу кількість ресурсів під час роботи.

Отже якщо говорити про практичне застосування результатів експерименту, то найбільш перспективною є модель YOLOv7-tiny через свою швидкість роботи

та використанню ресурсів, точність є змога покращити якщо витратити більше часу на навчання та використати більший об'єм навчальних даних для тренування цієї моделі.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. SSD: Single Shot MultiBox Detector / W. Liu et al. Computer Vision – ECCV 2016. Cham, 2016. P. 21–37. URL: [https://doi.org/10.1007/978-3-319-46448-0\\_2](https://doi.org/10.1007/978-3-319-46448-0_2) (date of access: 8.06.2024).
2. You Only Look Once: Unified, Real-Time Object Detection / J. Redmon et al. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016. 2016. URL: <https://doi.org/10.1109/cvpr.2016.91> (date of access: 8.06.2024).
3. Koonce B. VGG Network. Convolutional Neural Networks with Swift for Tensorflow. Berkeley, CA, 2021. P. 35–50. URL: [https://doi.org/10.1007/978-1-4842-6168-2\\_4](https://doi.org/10.1007/978-1-4842-6168-2_4) (date of access: 8.06.2024).
4. Classification of retinal images based on convolutional neural network / N. A. El-Hag et al. Microscopy Research and Technique. 2020. URL: <https://doi.org/10.1002/jemt.23596> (date of access: 9.06.2024)
5. N. V. Bilous, I. A. Ahejian, and V. V. Kaluhin, “DETERMINATION AND COMPARISON METHODS OF BODY POSITIONS ON STREAM VIDEO,” RIC, no. 2, p. 52, Jun. 2023, doi: 10.15588/1607-3274-2023-2-6.
6. Bisong E. Google Colaboratory. Building Machine Learning and Deep Learning Models on Google Cloud Platform. Berkeley, CA, 2019. P. 59–64. URL: [https://doi.org/10.1007/978-1-4842-4470-8\\_7](https://doi.org/10.1007/978-1-4842-4470-8_7) (date of access: 10.06.2024).
7. Wang C.-Y., Bochkovskiy A., Liao H.-Y. M. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023. 2023. URL: <https://doi.org/10.1109/cvpr52729.2023.00721> (date of access: 18.06.2024).
8. Wang W., Shi B., Wang X. A Car Target Detection Method based on YOLOv7-tiny. International Journal of Science and Engineering Applications. 2023. P. 40–42. URL: <https://doi.org/10.7753/ijsea1203.1015> (date of access: 18.06.2024).
9. Deep Learning Classification of Systemic Sclerosis Skin Using the MobileNetV2 Model / M. Akay et al. IEEE Open Journal of Engineering in Medicine and

Biology. 2021.Vol.2.P.104–110.URL: <https://doi.org/10.1109/ojemb.2021.3066097> (date of access: 18.06.2024).

10. Koonce B. MobileNet v1. Convolutional Neural Networks with Swift for Tensorflow. Berkeley, CA, 2021. P. 87–97. URL: [https://doi.org/10.1007/978-1-4842-6168-2\\_8](https://doi.org/10.1007/978-1-4842-6168-2_8) (date of access: 19.06.2024).

11. Karim M. R. TensorFlow: Powerful Predictive Analytics with TensorFlow: Predict valuable insights of your data with TensorFlow. Packt Publishing, 2018. 164 p.

12. Mean Average Precision. Encyclopedia of Database Systems. Boston, MA, 2009. P. 1703. URL: [https://doi.org/10.1007/978-0-387-39940-9\\_3032](https://doi.org/10.1007/978-0-387-39940-9_3032) (date of access: 20.06.2024).

13. Arab Marcomini L., Cunha A. L. Truck Axle Detection Using Convolutional Neural Networks. SSRN Electronic Journal. 2023. URL: <https://doi.org/10.2139/ssrn.4383043> (date of access: 22.06.2024).

14. Nenchoo, Tantrairatn. Real-Time 3D UAV Pose Estimation by Visualization. Proceedings. 2020. Vol. 39, no. 1. P. 18. (date of access: 22.06.2024). URL: <https://doi.org/10.3390/proceedings2019039018>.

15. N. Bilous, O. Svidin, I. Ahejian, and V. Malko, “A skeleton-based method for exercise recognition based on 3D coordinates of human joints,” IJ-AI, vol. 13, no. 2, p. 1805, Jun. 2024, doi: 10.11591/ijai.v13.i2.pp1805-1816.

16. Bilous N. V. Deletion method of image low-frequency components using fast median filter algorithm / N. V. Bilous, A. I. Krasov, V. P. Vlasenko // Journal of Engineering Sciences – 2016. – P. 7-14. DOI: 10.21272/jes.