

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій**

«На правах рукопису»
УДК 004.8

До захисту допущено:
Завідувач кафедри
_____ Олександр РОЛІК
« » _____ 2024 р.

**Магістерська дисертація
на здобуття ступеня магістра
за освітньо-професійною програмою «Інформаційне забезпечення
робототехнічних систем»
зі спеціальності 126 «Інформаційні системи та технології»
на тему: «Автоматизована система виявлення гвинтокрилих БПЛА в
небі за допомогою штучного інтелекту»**

Виконав:
студент 2 курсу, групи ІК-21мп
Грисюк Михайло Олександрович _____

Керівник:
доц. кафедри ІСТ, к.ф.-м.н.,
Пономаренко Роман Миколайович _____

Рецензент:
доц. кафедри ІСТ
КНУ імені Тараса Шевченка, к.т.н.,
Кравченко Ольга Віталіївна _____

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

Київ – 2024 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Рівень вищої освіти – другий (магістерський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційне забезпечення робототехнічних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 2024 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Грисюку Михайлу Олександровичу

1. Тема дисертації «Автоматизована система виявлення гвинтокрилих БПЛА в небі за допомогою штучного інтелекту», науковий керівник дисертації Пономаренко Роман Миколайович, доцент кафедри ІСТ, к.ф.-м.н., затверджені наказом по університету від «07» 11 2023 р. № 5168-с
2. Термін подання студентом дисертації «08» 01 2024 р.
3. Об'єкт дослідження: розробці автоматизованої системи, яка використовує штучний інтелект для ефективного виявлення та ідентифікації гвинтокрилих безпілотних літальних апаратів в повітрі.
4. Вихідні дані: швидке виявлення дрона, зображення з виявленим БПЛА позначеним рамкою
5. Перелік завдань, які потрібно розробити: огляд існуючих методів виявлення БПЛА, формування вимог до системи, підготовка датасету для навчання, навчання та тестування моделі, розробка веб-додатку та тестування
6. Орієнтовний перелік графічного (ілюстративного) матеріалу: Архітектура нейронної мережі для виявлення БПЛА

7. Дата видачі завдання 01.09.2023 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Огляд існуючих методів виявлення БПЛА	01.09.2023-07.09.2023	
2	Формування вимог до системи	08.09.2023-14.09.2023	
3	Підготовка датасету для навчання	15.09.2023-01.10.2023	
4	Навчання та тестування моделі	02.10.2023-01.11.2023	
5	Розробка веб-додатку та тестування	02.11.2023-17.12.2023	
6	Подання дисертації на передзахист	18.12.2023	
7	Розробка стартап-проєкту	19.12.2023-31.12.2023	
8	Подання дисертації на захист	08.01.2024	

Студент

Михайло ГРИСЮК

Науковий керівник

Роман ПОНОМАРЕНКО

РЕФЕРАТ

Автоматизована система виявлення гвинтокрилих БПЛА в небі за допомогою штучного інтелекту: 107 с., 23 табл., 41 рис., 9 дод., 36 джерел.

ВИЯВЛЕННЯ ОБ'ЄКТІВ, КОМП'ЮТЕРНИЙ ЗІР, ШТУЧНИЙ ІНТЕЛЕКТ, БПЛА, БЕЗПЕКА

Актуальність теми "Автоматизована система виявлення гвинтокрилих БПЛА в небі за допомогою штучного інтелекту" обумовлена зростанням використання безпілотних літальних апаратів у цивільних та військових сферах, що вимагає розробки ефективних та надійних засобів їх виявлення та відслідковування для забезпечення безпеки повітряного простору, конфіденційності та протидії можливим загрозам безпілотних технологій.

Мета дослідження полягає в розробці та оптимізації автоматизованої системи, використовуючи штучний інтелект, для надійного та ефективного виявлення гвинтокрилих безпілотних літальних апаратів в повітрі, сприяючи підвищенню безпеки повітряного простору та забезпеченню відповідного реагування на потенційні загрози.

Об'єктом дослідження є розробка автоматизованої системи, яка використовує штучний інтелект для ефективного виявлення та ідентифікації гвинтокрилих безпілотних літальних апаратів в повітрі.

Предметом дослідження є розробка, впровадження та оптимізація системи, яка використовує штучний інтелект для ефективного виявлення та ідентифікації гвинтокрилих безпілотних літальних апаратів в повітрі з метою забезпечення безпеки та контролю над повітряним простором.

ABSTRACT

Automated detection system of rotary wing UAVs in the sky using artificial intelligence: 107 p., 23 tab., 41 draw., 9 app., 36 sources.

OBJECT DETECTION, COMPUTER VISION, ARTIFICIAL INTELLIGENCE, UAV, SECURITY

The relevance of the topic "Automated system for detecting rotorcraft UAVs in the sky with the help of artificial intelligence" is due to the growing use of unmanned aerial vehicles in the civil and military spheres, which requires the development of effective and reliable means of their detection and tracking to ensure airspace safety, privacy and countering possible threats of unmanned aerial vehicles technologies.

The purpose of the research is to develop and optimize an automated system using artificial intelligence for reliable and effective detection of rotorcraft unmanned aerial vehicles in the air, contributing to the improvement of airspace security and ensuring an appropriate response to potential threats.

The object of the research is the development of an automated system that uses artificial intelligence to effectively detect and identify rotorcraft unmanned aerial vehicles in the air.

The subject of the study is the development, implementation and optimization of a system that uses artificial intelligence for effective detection and identification of rotorcraft unmanned aerial vehicles in the air in order to ensure safety and control over airspace.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ.....	9
ВСТУП.....	10
1 ОГЛЯД ІСНУЮЧИХ МЕТОДІВ ВИЯВЛЕННЯ БПЛА.....	12
1.1 Аудіо виявлення	12
1.2 Теплове виявлення	13
1.3 Виявлення за допомогою радіо частот.....	14
1.4 Виявлення за допомогою Wi-Fi	15
1.5 Радіолокаційне виявлення.....	16
1.6 Візуальне виявлення	17
Висновки до розділу 1	18
2 МАШИННЕ НАВЧАННЯ ТА ГЛИБОКЕ НАВЧАННЯ.....	19
2.1 Поняття штучного інтелекту.....	19
2.2 Метод машинного навчання.....	20
2.3 Глибоке навчання (DL).....	21
2.4 Різниця між машинним і глибоким навчанням.....	23
2.5 Огляд концепцій машинного та глибокого навчання.....	23
2.6 Ключові відмінності між машинним навчанням і глибоким навчанням	24
2.7 Обмеження машинного навчання.....	26
Висновки до розділу 2	26
3 ВИЯВЛЕННЯ ОБ'ЄКТІВ ТА ЙОГО МЕТОДИ	28
3.1 Виявлення об'єктів	28
3.2 Останні технологічні досягнення комп'ютерного зору	28
3.3 Недоліки та переваги виявлення об'єктів.....	29
3.4 Віха в сучасному виявленні об'єктів.....	30
3.5 Одноступінчаті та двоступеневі детектори об'єктів глибокого навчання.....	31
3.6 Порівняння алгоритмів виявлення об'єктів	33
3.6.1 Найкращий алгоритм виявлення об'єктів у реальному часі за точністю	33
3.6.2 Найшвидший алгоритм виявлення об'єктів у реальному часі.....	34

3.7 Найпопулярніші алгоритми виявлення об'єктів.....	36
3.7.1 YOLO.....	37
3.7.2 SSD.....	37
3.7.3 R-CNN	38
3.7.4 Маска R-CNN.....	39
3.7.5 SqueezeDet.....	40
3.7.6 MobileNet.....	40
3.7.7 YOLOR	40
Висновки до розділу 3	41
4 РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ	42
4.1 Засоби розробки.....	42
4.1.1 Python.....	42
4.1.2 Roboflow	45
4.1.3 Visual Studio Code	46
4.2 Підготовка даних для навчання моделі.....	47
4.3 Навчання моделі	54
4.4 Аналіз результатів показників моделі.....	61
4.5 Розробка клієнтської частини	74
4.5.1 Опис структури додатку.....	78
4.5.2 Тестування роботи додатку.....	80
Висновки до розділу 4	83
5 СТАРТАП-ПРОЄКТ	85
5.1 Опис ідеї проекту	85
5.2 Технологічний аудит ідеї проекту	87
5.3 Аналіз ринкових можливостей запуску стартап-проекту.....	88
5.4 Розроблення ринкової стратегії проекту.....	97
Висновки до розділу 5	102
ВИСНОВКИ.....	103
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	105
ДОДАТОК А.....	108

ДОДАТОК Б	109
ДОДАТОК В	110
ДОДАТОК Г	111
ДОДАТОК Д	112
ДОДАТОК Е	113
ДОДАТОК Ж	114
ДОДАТОК И	115
ДОДАТОК К	116

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

БПЛА - безпілотний літальний апарат;
ШІ (AI) – штучний інтелект (artificial intelligence);
МН (ML) – машинне навчання (machine learning);
ГН (DL) – глибоке навчання (deep learning);
TPU – Tensor Processing Unit;
VPU – Vision Processing Unit;
GPU – Graphics Processing Unit
CNN – Convolutional Neural Network;
R-CNN – Regional Convolutional Neural Network;
SSD – Single-Shot Detector;
YOLO – You Only Look Once;
COCO – Common Objects in Context;
ONNX – Open Neural Network Exchange.

ВСТУП

За останні десятиліття безпілотні літаючі апарати (БПЛА) стали важливою складовою повітряного простору і знайшли застосування в різних сферах, включаючи військовий сектор, цивільну авіацію, аграрний сектор, та розвідку. Зростання кількості та видів БПЛА породжує нові виклики в галузі безпеки та контролю в повітряному просторі. Проблема полягає в тому, що з розвитком технологій виробництва БПЛА вони стають все доступнішими та менш помітними на радарях та візуальних системах спостереження. Це створює потенційно небезпечні ситуації, коли БПЛА можуть незаконно здійснювати різні види діяльності, включаючи розвідку, злочини та порушення повітряних просторів.

Актуальність дослідження полягає в необхідності розробки ефективних методів для виявлення та класифікації БПЛА в реальному часі. Такі методи дозволили б забезпечити безпеку повітряного простору та вчасно реагувати на потенційні загрози. Штучний інтелект, зокрема методи машинного навчання та нейронні мережі, можуть виявити свою ефективність у цій галузі, оскільки вони здатні аналізувати великі обсяги даних та розпізнавати складні моделі поведінки БПЛА.

Проблема виявлення БПЛА стає настільки актуальною, що її вирішення впливає на безпеку та прозорість повітряних просторів, важливу для громадянської авіації, оборони, правопорядку та національної безпеки. Дослідження з використанням штучного інтелекту для виявлення гвинтокрилих БПЛА має великий потенціал і може внести суттєвий внесок у вирішення цієї актуальної проблеми.

У цьому контексті, магістерська дисертація присвячена розробці та дослідженню автоматизованої системи, що використовує штучний інтелект для виявлення гвинтокрилих БПЛА в небі. Дана робота розглядає важливі аспекти виявлення та класифікації БПЛА, а також допоможе зміцнити безпеку та

ефективність контролю в повітряному просторі в умовах швидкого розвитку цієї технології.

1 ОГЛЯД ІСНУЮЧИХ МЕТОДІВ ВИЯВЛЕННЯ БПЛА

В епоху швидкого розвитку технологій дрони стали звичним явищем у нашому небі. Вони використовуються для всього: від аерофотозйомки та доставки посилок до спостереження за дикою природою та допомоги у пошуково-рятувальних місіях.

Однак із зростанням використання БПЛА зростає потреба в виявленні дронів.

Виявлення дрона у небі передбачає використання різних методів і технологій, таких як візуальне спостереження, радіочастотний аналіз, радари, акустичні датчики та теплові детектори. Поєднання кількох методів виявлення може підвищити загальну ефективність ідентифікації дронів у небі [1-3].

На сучасному етапі існують різні методи виявлення БПЛА. Є такі методи виявлення:

- аудіо;
- теплове;
- радіочастотне;
- Wi-Fi;
- радіолокаційне;
- візуальне.

1.1 Аудіо виявлення

Акустичні датчики дозволяють використовувати унікальний звуковий сигнал БПЛА та визначати його місцезнаходження з надзвичайно високою точністю.

Ці датчики виконують розширений аналіз звукового спектру навколишнього середовища, виявляючи активність, яка відповідає унікальним акустичним ознакам різних типів дронів.

Коли ці датчики виявляють присутність дрона, вони автоматично пробують звук і порівнюють його з інтегрованою базою даних. Коли збіг знайдено, система записує ідентифікаційну інформацію та надсилає автоматичне сповіщення.

Ця передова сенсорна технологія настільки надійна, що жоден дрон неможливо зламати, щоб вимкнути його.

Однак ці датчики не позбавлені недоліків, і розуміння цих обмежень має вирішальне значення для їх ефективного використання. Калібрування датчика відіграє важливу роль у точності виявлення дронів. Без належного калібрування датчик може не виявляти звуки дрона або давати хибнопозитивні результати.

Ще одна проблема – шум. Датчики повинні розрізняти шум дрона від іншого шуму навколишнього середовища, що може бути важко в шумному середовищі.

Незважаючи на ці обмеження, акустичне виявлення залишається дуже надійним методом інспекції за допомогою дронів. З розвитком технологій ці проблеми вирішуються все більш складними рішеннями акустичного виявлення.

1.2 Теплове виявлення

Теплові детектори являють собою справжню революцію у виявленні літальних апаратів, оскільки вони вловлюють тепло, що виділяється двигунами та іншими компонентами.

Завдяки своїм інфрачервоним можливостям ці детектори можуть вловлювати інфрачервоне випромінювання, що випромінюється будь-яким об'єктом з температурою вище абсолютного нуля, і перетворювати його у видиме зображення. Цей процес, який називається тепловізором, дозволяє виявити наявність дронів, невидимих неозброєним оком.

Коли справа доходить до виявлення дронів вночі, в гру вступають теплові детектори. Замість того, щоб покладатися на видиме світло, вони зчитують теплові сигнатури, що робить їхні здібності нічного бачення безцінними.

Однак важливо пам'ятати про обмеження виявлення цих пристроїв.

Несприятливі погодні умови, такі як туман або сильний дощ, можуть серйозно вплинути на його роботу. Крім того, дрони, виготовлені з матеріалів з низьким рівнем випромінювання, наприклад полірованих або відбиваючих поверхонь, може бути складніше виявити. Незважаючи на ці обмеження, теплові детектори в поєднанні з іншими технологіями можуть значно покращити ваші можливості виявлення дронів.

1.3 Виявлення за допомогою радіо частот.

Використовуючи потужність радіочастотного аналізу, можна вловлювати сигнали на відстані кілометрів і потенційно виявляти присутність літаків у місцевості. Також відомий як аналіз радіочастотного спектру, цей метод дозволяє контролювати широкий діапазон частот від 70 МГц до 6 ГГц і виявляти радіочастотні перешкоди, які випромінює БПЛА під час спілкування з оператором.

Декодування радіочастотних сигналів є важливою частиною цього процесу, надаючи вам інформацію, необхідну для відстеження та ідентифікації вашого дрона.

Він також використовує технологію радіочастотної фільтрації для фільтрації шумів і зосередження на важливих сигналах, тим самим підвищуючи точність виявлення.

Основною перевагою цього методу є дальність виявлення: системи радіо частот можуть виявляти БПЛА на великих відстанях, що робить їх ефективними для виявлення далеких об'єктів. В деяких випадках, системи радіо частот можуть ідентифікувати конкретний тип БПЛА на основі їх радіосигналів або транспондерів.

Можливість виявлення за обмеженою видимістю: системи радіо частот працюють незалежно від видимості та погодних умов, що робить їх ефективними навіть у ситуаціях з обмеженою видимістю. Також радіосигнали рухаються зі

швидкістю світла, що дозволяє системам радіо частот швидко реагувати на зміни у місцезнаходженні БПЛА.

Недоліками виявлення за допомогою радіо частот є:

- простота перехоплення сигналу: радіосигнали можуть бути легко перехоплені та відключені зловмисниками, що ускладнює виявлення та відстеження дронів;

- потрібне спеціальне обладнання: використання радіочастот для виявлення дронів вимагає спеціального обладнання та знань для ефективного використання;

- ризик перешкод і перешкод: радіочастоти можуть зазнавати перешкод і перешкод, що може вплинути на точність виявлення та призвести до помилок;

- проблеми конфіденційності та нормативні вимоги: використання радіочастот для виявлення дронів може порушувати конфіденційність і на нього поширюються нормативні обмеження в деяких юрисдикціях.

1.4 Виявлення за допомогою Wi-Fi

Перевага використання Wi-Fi для виявлення дронів полягає в тому, що Wi-Fi є широко поширеним і доступним засобом зв'язку, який використовується в різноманітних пристроях і системах. Багато дронів використовують Wi-Fi для зв'язку зі своїми контролерами, завдяки чому їх легко помітити. Система виявлення також може відстежувати сигнал Wi-Fi, який випромінює дрон, і визначати його місцезнаходження, особливо якщо сигнал досить потужний. Використання Wi-Fi для виявлення дронів є дуже економічним порівняно з іншими технологіями, оскільки не потребує дорогого спеціалізованого обладнання. Недоліком є обмежена дальність зв'язку. Мережі Wi-Fi мають обмежений діапазон зв'язку, зазвичай кілька сотень метрів. Це робить Wi-Fi менш ефективним для виявлення дронів на великих відстанях. Потужність сигналу Wi-Fi може змінюватися залежно від багатьох факторів, таких як перешкоди, погодні умови тощо. Це може ускладнити стабільне виявлення дронів. Крім того, сигнали Wi-Fi можуть бути легко перехоплені іншими користувачами або потенційними

зловмисниками, що порушує безпеку та конфіденційність даних. Деякі дрони можуть використовувати для зв'язку інші технології, як-от радіо, GSM або інші технології, через що їх неможливо виявити через Wi-Fi.

Багато в чому ефективність використання Wi-Fi для виявлення дронів залежить від конкретних умов і середовища. Цей підхід може бути дуже ефективним в одних ситуаціях і менш прийнятним в інших. Тому важливо розглядати його як один із можливих методів виявлення та добре адаптувати його до конкретних вимог та обмежень системи безпеки.

1.5 Радіолокаційне виявлення

Радар – ефективно виявляє присутність БПЛА на дуже великих відстаннях. Включаючи оптику у свою функціональність, радіолокаційна технологія може ефективно розширити свої можливості, щоб забезпечити повне покриття, коли це необхідно. Радарні системи складаються з передавача, який випромінює радіохвилі, відомі як радіолокаційні сигнали, які взаємодіють з об'єктами шляхом відображення або розсіювання. Потім ці змінені хвилі приймаються радарним приймачем, де складні алгоритми перетворюють їх у візуальний формат, пропонуючи цінну інформацію про розміри, структуру та склад об'єкта.

Об'єкти, розташовані ближче чи далі від передавача, створюють «ефект Доплера», тобто спотворення або вигин радіохвиль. Імпульсні доплерівські радарні системи, які використовуються для виявлення дронів, періодично випромінюють радіохвилі та вимірюють кривизну радіолокаційного сигналу, що повертається, щоб оцінити відстань, швидкість і характеристики виявленого об'єкта. Однак дрони здебільшого виготовлені з невидимого для радарів пластику, лише їхні металеві камери, батареї та двигуни забезпечують відбивну платформу для сигналів радарів. Тут на допомогу приходить мікроімпульсний доплер, більш точна система, яка запускає серію дуже близьких імпульсів, щоб отримати точніше зображення цілі спостереження, що є спробою ідентифікувати невеликі об'єкти, такі як камери дронів або двигуни.

Безпілотні літальні апарати менші за пілотовані літальні апарати і, як правило, літають близько до землі, що ускладнює їх виявлення для всіх, крім найбільш спеціалізованих радарів.

1.6 Візуальне виявлення

В цьому методі камера використовується для визначення місцезнаходження рухомого об'єкта в повітрі та розрізнення між БПЛА та птахами на основі їх розміру, траєкторії польоту та способу руху. Важливо відзначити, що дрони мають здатність швидко змінювати свій напрямок. Тому, якщо об'єкт різко змінить свій курс, навряд чи це буде птах або літак, а скоріше дрон.

Дрони також можуть мати унікальні моделі польоту, які виділяють їх - незвичайні кружляння або нерегулярні рухи, які відрізняються від типових візерунків птахів чи інших літаків.

Великим недостатком цього метода, це неможливість виявлення БПЛА при поганому освітленості (наприклад вночі).

Переваги та обмеження теплових детекторів наведено в таблиці 1.1.

Таблиця 1.1 – Переваги та обмеження теплових детекторів

Переваги теплових детекторів	Обмеження теплових детекторів
Інфрачервоний функціонал забезпечує чітку візуалізацію	На продуктивність можуть вплинути суворі погодні умови
Чудово підходить для нічного бачення завдяки опорі на теплові сигнатури	Труднощі з виявленням дронів із поверхнями з низьким рівнем випромінювання
Можна встановлювати на дрони або використовувати як автономну систему	Не ідеально підходить для виявлення безпілотників із полірованими або відбиваючими поверхнями

Висновки до розділу 1

На сучасному етапі існують різні методи виявлення БПЛА. Виявлення дрона в небі передбачає використання різних методів і технологій, таких як візуальне спостереження, радіочастотний аналіз, радар, акустичні датчики та теплові детектори.

Із зростанням використання безпілотних літальних апаратів зростає й потреба в їх виявленні. Поєднання кількох методів виявлення може підвищити загальну ефективність ідентифікації дронів у небі.

У цій роботі буде використовуватись візуальний метод для виявлення БПЛА, враховуючи його легкість у виявленні, дешевизні використання. Також за допомогою цього метода буде можливість використовувати штучний інтелект для навчання моделі.

2 МАШИННЕ НАВЧАННЯ ТА ГЛИБОКЕ НАВЧАННЯ

2.1 Поняття штучного інтелекту

Понад 200 років технологічні інновації були головною рушійною силою економічного зростання. Найважливішими з них є так звані технології загального призначення, такі як парова машина, електрика та двигун внутрішнього згорання. Кожна інновація викликає хвилю інновацій і можливостей у різних галузях. Найважливішою загальною технологією нашого часу є штучний інтелект.

Штучний інтелект або штучний інтелект є однією з найдавніших галузей інформатики, і вона дуже широка [5]. Він включає в себе аспекти імітації когнітивних функцій для вирішення проблем реального світу та побудови комп'ютерних систем, які навчаються та мислять як люди. Тому штучний інтелект часто називають машинним інтелектом, щоб протиставити його людському інтелекту.

Сфера штучного інтелекту обертається навколо інтерфейсу між інформатикою та когнітивною наукою. Штучний інтелект може означати що завгодно: від комп'ютерних програм, які грають у шахи, до самокерованих автомобілів і систем комп'ютерного зору.

Зараз штучний інтелект викликає великий інтерес завдяки успіху машинного навчання (ML). Штучний інтелект, зокрема машинне навчання, стосується здатності машини постійно покращувати свою продуктивність без потреби людини пояснювати, як саме виконувати всі покладені на неї завдання. В останні роки машинне навчання стало більш ефективним і широко використовуваним. Тепер можна створювати системи, які навчаються виконувати завдання самостійно [6].

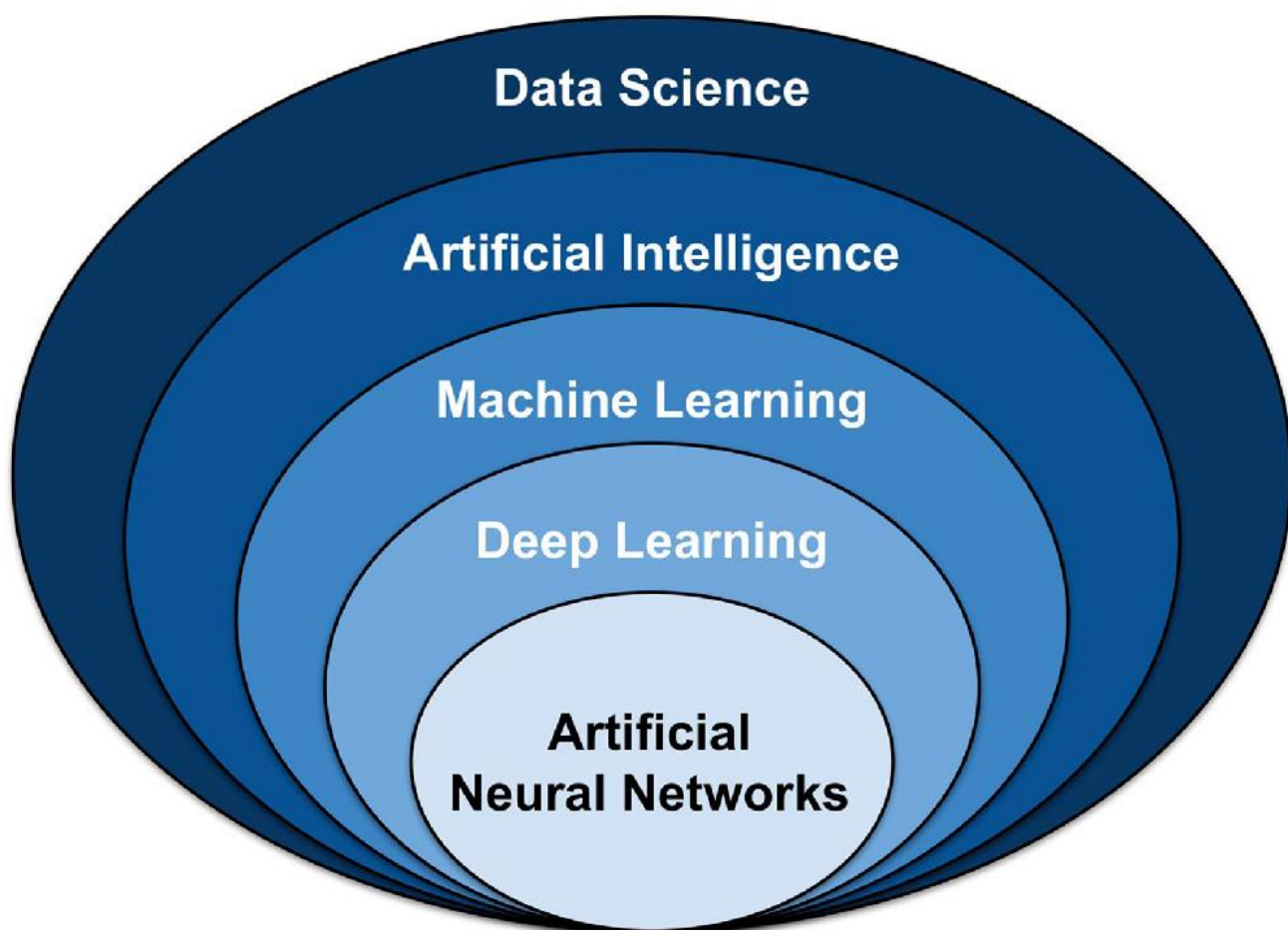


Рисунок 2.1 – Графік взаємозв'язку між дисциплінами ШІ [6]

2.2 Метод машинного навчання

Машинне навчання є розділом штучного інтелекту. Графік взаємозв'язку між дисциплінами ШІ зображено на рисунку 2.1. Основний принцип машинного навчання полягає в тому, що машини використовують дані для «навчання» [7]. У результаті системи машинного навчання можуть швидко застосовувати знання та навчання з великих наборів даних, щоб досягти успіху в розпізнаванні людей, розпізнаванні мови, розпізнаванні об'єктів, перекладі та багатьох інших завданнях [8].

Замість того, щоб розробляти та програмувати програмне забезпечення з конкретними інструкціями для виконання завдань, машинне навчання дозволяє системам навчатися, розпізнавати шаблони та робити прогнози самостійно.

Машинне навчання — це дуже практична сфера штучного інтелекту, яка спрямована на розробку програмного забезпечення, яке може автоматично навчатися на основі попередніх даних, отримувати знання з досвіду та поступово покращувати свою поведінку навчання, щоб робити прогнози на основі нових даних.

Хоча машинне навчання підпадає як підрозділ штучного інтелекту, терміни штучний інтелект і машинне навчання часто використовуються як синоніми. Машинне навчання можна розглядати як основу штучного інтелекту, оскільки воно передбачає застосування методів, керованих даними.

Машинне навчання використовує серію вхідних даних, а потім вивчає вхідні дані. Тому методи машинного навчання використовують дані для розуміння контексту, чуттєвого сприйняття та прийняття рішень в умовах невизначеності.

Алгоритми машинного навчання часто використовуються як частина систем штучного інтелекту для виявлення тенденцій і закономірностей у даних.

Застосування машинного навчання охоплює науку, інженерію та бізнес, що веде до більшої кількості рішень, заснованих на фактах.

Машинне навчання використовується для створення різних автоматизованих систем рекомендацій штучного інтелекту. Прикладом машинного навчання є персоналізовані рекомендації фільмів для Netflix або музичні рекомендації для сервісів потокової передачі музики на вимогу.

Розробка нових статистичних алгоритмів навчання та доступність великих даних (великих наборів даних) і недорогих обчислень сприяли величезному прогресу в машинному навчанні.

2.3 Глибоке навчання (DL)

Сьогодні дуже популярним методом машинного навчання є глибоке навчання (DL). Глибоке навчання — це модель машинного навчання, яка базується на глибоких нейронних мережах і має довгу історію [9].

Глибоке навчання є підмножиною машинного навчання. Він використовує деякі методи машинного навчання для вирішення реальних проблем за допомогою нейронних мереж, які імітують прийняття рішень людиною. Тому глибоке навчання вчить машини робити те, що природно робить людський мозок [10].

Найбільшою особливістю глибокого навчання є його ієрархічна структура, яка є основою штучних нейронних мереж. Кожен рівень розширює знання попереднього рівня.

Завдання глибокого навчання можуть бути дуже дорогими, покладатися на значні обчислювальні ресурси та вимагати великих наборів даних для навчання моделі. При глибокому навчанні алгоритм навчання повинен знати різні параметри, які спочатку можуть призвести до багатьох помилкових спрацьовувань.

Алгоритму глибокого навчання можна дати вказівку «вивчити», як виглядає собака. Щоб зрозуміти найдрібніші деталі, які відрізняють собак від інших тварин, таких як лисиці чи пантери, потрібні масивні набори даних зображень.

Загалом, глибоке навчання підтримує штучний інтелект, який найбільше схожий на людину, особливо в області комп'ютерного зору. Іншим комерційним прикладом глибокого навчання є візуальне розпізнавання обличчя, яке використовується для захисту та розблокування мобільних телефонів.

Глибоке навчання також має бізнес-додатки для обробки великих обсягів даних, наприклад мільйонів зображень. і розпізнавати певні характеристики. Пошук на основі тексту, виявлення шахрайства, виявлення кадрів, розпізнавання рукописного тексту та шаблонів, пошук зображень, розпізнавання обличчя – це всі завдання, які можна виконувати за допомогою глибокого навчання. Такі великі компанії штучного інтелекту, як Meta/Facebook, IBM і Google, використовують мережі глибокого навчання, щоб замінити ручні системи. Список тих, хто використовує комп'ютерний зір, швидко зростає, оскільки реалізується все більше варіантів використання.

2.4 Різниця між машинним і глибоким навчанням

Глибоке навчання та машинне навчання — обидві гілки штучного інтелекту, при цьому глибоке навчання є підмножиною машинного навчання. Таким чином, хоча глибоке навчання є компонентом машинного навчання, воно відрізняється від звичайних методів машинного навчання.

Глибоке навчання виділяється з-поміж інших типів машинного навчання завдяки своїм відмінним перевагам, які привели до його широкої популярності як переважаючої алгоритмічної технології в сучасному світі.

Продуктивність алгоритмів, які використовуються в машинному навчанні, покращується зі збільшенням обсягу даних. Глибоке навчання, з іншого боку, покладається на шари, тоді як машинне навчання покладається на введення даних для самостійного навчання.

2.5 Огляд концепцій машинного та глибокого навчання

Процес навчання машин на основі даних є спільною рисою машинного навчання (ML) і глибокого навчання (DL). Однак важливо зазначити, що методи навчання, які використовуються цими двома технологіями, суттєво відрізняються, що призводить до різних результатів. Крім того, глибоке навчання пропонує численні переваги, такі як масштабованість, здатність полегшувати контрольоване та неконтрольоване навчання та сприяти обміну знаннями. Ці атрибути роблять глибоке навчання дуже впливовою дисципліною в царині машинного навчання, яку часто називають однією з найбільш грізних «наук моделювання».

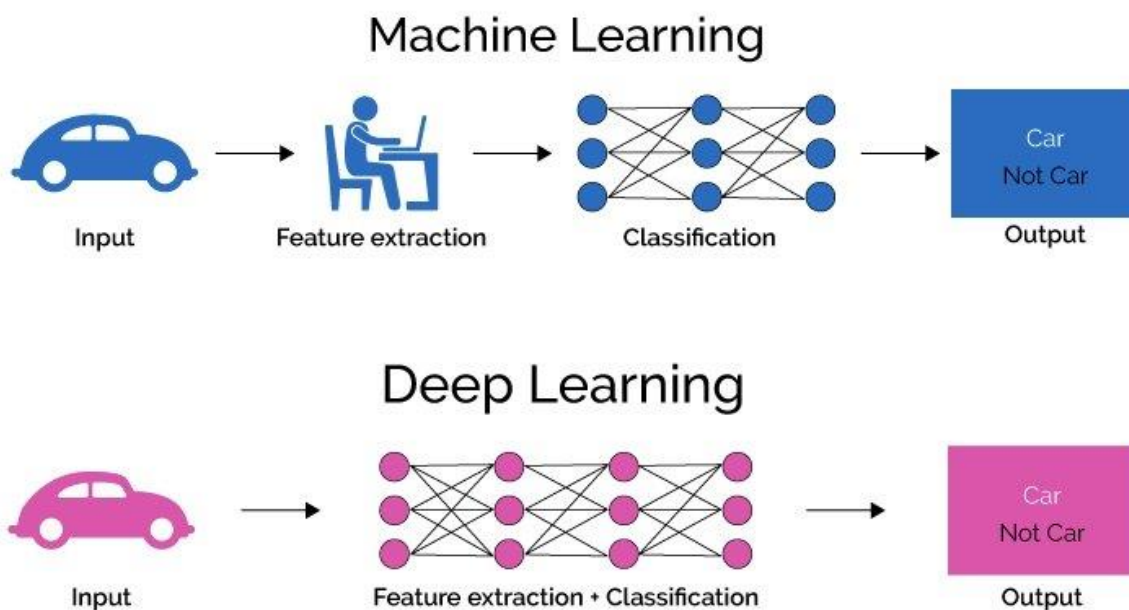


Рисунок 2.2 – Машинне навчання та глибоке навчання [6]

2.6 Ключові відмінності між машинним навчанням і глибоким навчанням

Розвиток глибокого навчання був прискорений завдяки використанню нейронних мереж і появи високошвидкісних комп'ютерів. І навпаки, звичайні методи машинного навчання досягли стану стагнації з точки зору продуктивності [12].

Процес навчання моделей машинного навчання можна прискорити завдяки використанню машинного навчання, що дозволяє пришвидшити час навчання та покращити результати з більшими наборами даних. З іншого боку, глибоке навчання потребує значної обчислювальної потужності для ефективного навчання нейронних мереж, що складаються з кількох шарів.

Розвиток глибокого навчання значно прискорився завдяки використанню нейронних мереж і доступності високошвидкісних комп'ютерів. З іншого боку, альтернативні методи машинного навчання досягли стану стагнації з точки зору продуктивності. Людське втручання: щоразу, коли машинне навчання передбачає нове навчання, люди-розробники повинні втрутитися та налаштувати алгоритм для досягнення навчання. Навпаки, нейронні мережі в глибокому навчанні можна

навчати на кількох рівнях, а інтелектуальні алгоритми можуть навчити машину використовувати знання, отримані від одного рівня до іншого, для подальшого навчання без втручання людини.

У традиційному машинному навчанні розробники інструктують машину, які типи функцій шукати. У глибокому навчанні процес вилучення функцій повністю автоматизований, це показано на рисунку 2.2. Це робить виділення ознак у глибокому навчанні більш точним і орієнтованим на результат. Методи машинного навчання вимагають постановки проблеми, яка розбиває проблему на різні частини, які потім вирішуються, а потім об'єднує результати на завершальному етапі. Методи глибокого навчання, як правило, вирішують проблеми "від кінця до кінця", роблячи процес навчання швидшим і надійнішим.

Оскільки нейронні мережі глибокого навчання базуються на кількох рівнях знань і не потребують втручання людини, для навчання їм потрібні великі обсяги даних. Навпаки, машинне навчання базується на керованому дослідженні вибірок даних, які все ще великі, але відносно невеликі.

Можливості самонавчання DL забезпечують швидші та точніші результати порівняно з ML. У традиційному машинному навчанні помилки розробника можуть призвести до неправильних рішень і низької точності, в результаті чого машинне навчання буде менш гнучким, ніж глибоке навчання.

На відміну від традиційних алгоритмів машинного навчання, для глибокого навчання потрібні машини високого класу. Графічний процесор, або графічний процесор, — це міні-версія цілого комп'ютера, але призначена лише для одного конкретного завдання — це відносно простий, але дуже паралельний комп'ютер, який може виконувати кілька завдань одночасно. Запуск нейронних мереж, як навчання, так і застосування мережі, найкраще досягається за допомогою графічних процесорів. Нове апаратне забезпечення AI включає прискорювачі TPU і VPU для програм глибокого навчання.

2.7 Обмеження машинного навчання

Машинне навчання, як правило, не є ідеальним рішенням для дуже складних проблем, таких як завдання комп'ютерного зору, які імітують людський «зір» і інтерпретують зображення на основі особливостей. Глибоке навчання забезпечує комп'ютерний зір завдяки надзвичайно точним архітектурам нейронної мережі, що неможливо з традиційними машинне навчання.

У той час як машинне навчання потребує сотень або навіть тисяч доповнених або необроблених вхідних даних для досягнення ефективної точності, глибоке навчання вимагає меншої кількості анотованих зображень для навчання. Без глибокого навчання комп'ютерний зір не був би таким точним, як сьогодні

Висновки до розділу 2

У цьому конкретному розділі було представлено всебічне дослідження фундаментальних принципів, що лежать в основі штучного інтелекту, машинного навчання і глибокого навчання, а також заглиблюючись у складні взаємозв'язки та відмінності, які існують між їх.

Починаючи з точного визначення, цей розділ заглиблюється в сферу штучного інтелекту, яка класифікується як наукова дисципліна, яка прагне створити системи зі здатністю виконувати завдання, зазвичай пов'язані з людським інтелектом. У ширшій галузі штучного інтелекту з'являються дві основні гілки: машинне навчання та глибоке навчання, причому останнє є підмножиною першого.

Наведено детальний аналіз концепцій, пов'язаних із машинним і глибоким навчанням. Машинне навчання стосується процесу створення алгоритмів, які отримують знання з даних без необхідності явного програмування. І навпаки, глибоке навчання — це конкретна гілка машинного навчання, яка використовує глибокі нейронні мережі для вирішення різноманітних завдань із вирішення проблем.

Машинне навчання та глибоке навчання, хоча й відрізняються, можна відрізнити за кількома ключовими факторами. Машинне навчання здатне використовувати різні моделі, такі як дерева рішень або опорні векторні методи, тоді як глибоке навчання покладається на використання багаторівневих глибоких нейронних мереж з метою автономного отримання уявлень із даних.

Зрештою було визнано недоліки кожного підходу. Машинне навчання стикається з перешкодами, такими як необхідність у великій кількості розроблених вручну функцій для вилучення функцій. З іншого боку, глибоке навчання вимагає великої кількості даних і обчислювальної потужності для ефективного навчання складних моделей.

3 ВИЯВЛЕННЯ ОБ'ЄКТІВ ТА ЙОГО МЕТОДИ

3.1 Виявлення об'єктів

Виявлення об'єктів — це важливе завдання комп'ютерного зору, яке виявляє екземпляри візуальних об'єктів певних категорій (таких як люди, тварини, автомобілі чи будівлі) у цифрових зображеннях (таких як фотографії чи відеозображення) [13]. Метою виявлення об'єктів є розробка обчислювальних моделей, які надають найосновнішу інформацію, необхідну для програм комп'ютерного зору.

В останні роки швидкий прогрес технології глибокого навчання значно прискорив розвиток технології розпізнавання об'єктів. За допомогою мереж глибокого навчання та обчислювальної потужності графічних процесорів було значно покращено продуктивність детекторів об'єктів і трекерів, а також досягнуто серйозних проривів у виявленні об'єктів [14].

3.2 Останні технологічні досягнення комп'ютерного зору

Технологія створення зображень ШІ досягла значного прогресу за останні роки. Доступні різноманітні камери, у тому числі комерційні камери безпеки та камери відеоспостереження. Це забезпечує більшу гнучкість додатків, які більше не залежать від спеціальних датчиків, дорогого встановлення та вбудованих апаратних систем, які потрібно замінювати кожні три-п'ять років.

Зараз обчислювальна потужність значно збільшена, а ефективність значно покращена. В останні роки обчислювальні платформи еволюціонували в напрямку розпаралелювання за допомогою прискорювачів ШІ, таких як багатоядерна обробка, графічні процесори (GPU) і тензорні процесори (TPU).

Таке апаратне забезпечення дозволяє виявляти об'єкти та відстежувати їх за допомогою комп'ютерного зору майже в реальному часі. Таким чином, швидкий розвиток глибоких згорткових нейронних мереж (CNN) і вдосконалення

обчислювальних можливостей GPU є головними рушійними силами величезного прогресу в області виявлення цілей на основі комп'ютерного зору [15].

3.3 Недоліки та переваги виявлення об'єктів

Детектори об'єктів є гнучкими, їх можна навчити виконувати різноманітні завдання та налаштувати для професійних застосувань. Автоматична ідентифікація об'єктів, людей і сцен може надати корисну інформацію для автоматизованих завдань (підрахунок, перевірка, перевірка тощо) у всьому ланцюжку створення вартості підприємства.

Однак основним недоліком детекторів об'єктів є те, що вони дуже інтенсивні з точки зору обчислень і потребують великої обчислювальної потужності. Особливо коли моделі розпізнавання об'єктів розгортаються в масштабі, операційні витрати можуть швидко зрости та загрожувати економіці бізнес-випадків використання [16].

Ось як працює розпізнавання об'єктів

Виявлення об'єктів можна виконувати за допомогою традиційних методів обробки зображень або сучасних мереж глибокого навчання:

– технологія обробки зображень, як правило, не потребує історичних даних для навчання і практично не контролюється. OpenCV є популярним інструментом для обробки зображень. Перевагою є тим, що ці завдання не вимагають від людей вручну позначати анотовані зображення даних (для навчання під наглядом). А недоліком – ці методи обмежені такими факторами, як: складні сцени (без суцільного фону), оклюзія (частково закриті об'єкти), освітлення та тіні та ефекти безладу;

– методи глибокого навчання зазвичай базуються на навчанні під наглядом або без нагляду, при цьому методи під наглядом є стандартним методом у задач комп'ютерного зору. Продуктивність обмежена обчислювальною потужністю GPU, яка з кожним роком стрімко зростає. Перевагою є тим, що виявлення об'єктів із глибоким навчанням значно надійніше до оклюзій, складних сцен і

складного освітлення. А недоліком – потрібна велика кількість навчальних даних. Процес анотування зображення трудомісткий і дорогий. Наприклад, позначення 500 000 зображень для навчання спеціального алгоритму виявлення об'єктів DL вважається невеликим набором даних. Однак багато контрольних наборів даних (MS COCO, Caltech, KITTI, PASCAL VOC, V5) забезпечують доступність позначених даних.

Сьогодні виявлення об'єктів глибокого навчання широко прийнято дослідниками та прийнято компаніями комп'ютерного зору для розробки комерційних продуктів.

3.4 Віха в сучасному виявленні об'єктів

Сфера розпізнавання об'єктів не така нова, як здається. Насправді розпізнавання об'єктів продовжувало розвиватися протягом останніх 20 років. Прогрес у виявленні об'єктів зазвичай поділяють на два окремі історичні періоди (до та після впровадження глибокого навчання):

До 2014 року – період традиційного виявлення об'єктів:

- Детектор Віоли-Джонса (2001), піонерська робота, що поклала початок розвитку традиційних методів виявлення об'єктів;
- HOG Detector (2006), популярний дескриптор функцій для виявлення об'єктів у комп'ютерному зорі та обробці зображень;
- DPM (2008) вперше представив регресію обмежувальної рамки.

Після 2014 року – період виявлення глибокого навчання.

Основний двоетапний алгоритм виявлення цілі:

- RCNN і SPPNet (2014);
- Fast RCNN і Faster RCNN (2015);
- Маска R-CNN (2017);
- Pyramid Network/FPN (2017);
- G-RCNN (2021).

Основний одноетапний алгоритм виявлення цілей YOLO (2016):

- SSD (2016);
- RetinaNet (2017);
- YOLOv3 (2018);
- YOLOv4 (2020);
- Au Lac (2021);
- YOLOv7 (2022);
- YOLOv8 (2023).

Щоб зрозуміти, який алгоритм найкраще підходить для певного випадку використання, важливо зрозуміти основні характеристики. Спочатку буде розглянуто основні відмінності між пов'язаними алгоритмами розпізнавання зображень для виявлення об'єктів, перш ніж обговорювати окремі алгоритми.

3.5 Одноступінчаті та двоступеневі детектори об'єктів глибокого навчання

Як можна побачити з попереднього підрозділу, сучасні методи виявлення об'єктів можна розділити на два основних типи: одноступеневі детектори об'єктів і двоступеневі детектори об'єктів.

Як правило, детектори об'єктів на основі глибокого навчання витягують ознаки з вхідних зображень або відеозображень. Детектор об'єктів вирішує дві наступні задачі:

- знайти будь-яку кількість об'єктів (можна навіть нуль);
- класифікація кожного окремого об'єкту і оцінка його розміру за допомогою обмежувальних рамок.

Щоб спростити процес, потрібно розділити ці завдання на два етапи. Інші методи поєднують ці два завдання в один крок (однокаскадний детектор), досягаючи вищої продуктивності за рахунок точності.

У двоступінчастому детекторі об'єктів функції глибини використовуються для пропозиції приблизних областей об'єкта перед їх використанням для класифікації зображень і регресії обмежувальної рамки об'єктів-кандидатів.

Двоступенева архітектура складається з пропозиції регіону об'єкта з використанням традиційних методів комп'ютерного бачення або глибоких мереж, після чого класифікації об'єкта на основі ознак, витягнутих із запропонованих регіонів за допомогою регресії обмежувальної рамки.

Двоступеневі методи досягають найвищої точності виявлення, але часто повільніші. Оскільки для кожного зображення існує багато кроків висновку, продуктивність (кількість зображень за секунду) не така висока, як у одноступінчастого детектора.

Різні двоступінчасті детектори включають регіональні згорткові нейронні мережі (RCNN), розроблені за допомогою Faster R-CNN або Mask R-CNN. Остання розробка — гранульований RCNN (G-RCNN).

Двоступеневий детектор об'єктів спочатку знаходить цікаву область і використовує цю обрізану область для класифікації. Однак такі багатоступеневі детектори, як правило, не піддаються наскрізному навчанню, оскільки відсікання є недиференційованим процесом.

Одноступінчастий детектор: одноступінчастий детектор передбачає обмежувальні рамки на зображеннях без кроку пропозиції області. Процес вимагає менше часу, тому його можна використовувати для додатків у режимі реального часу.

Одноступінчасті детектори об'єктів надають перевагу швидкості висновку та надшвидкі, але погано виявляють об'єкти неправильної форми або групи дрібних об'єктів.

До найпопулярніших однокаскадних детекторів відносяться YOLO, SSD і RetinaNet. Останніми детекторами реального часу є YOLOv7 (2022), YOLOR (2021) і YOLOv4-Scaled (2020). Ознайомтеся з порівнянням бенчмарків нижче.

Основні переваги використання одноетапних алгоритмів для виявлення об'єктів включають загалом вищу швидкість виявлення та більшу структурну простоту й ефективність порівняно з багатоступеневими детекторами.

3.6 Порівняння алгоритмів виявлення об'єктів

Коли справа доходить до оцінки різних моделей для виявлення об'єктів у реальному часі, набір даних Microsoft COCO широко вважається найпопулярнішим тестом. Ефективність різних алгоритмів зазвичай вимірюється за допомогою метрики MAP (середня точність).

3.6.1 Найкращий алгоритм виявлення об'єктів у реальному часі за точністю

Виходячи із середньої точності (AP), найефективнішим алгоритмом виявлення об'єктів у реальному часі в наборі даних MS COCO є YOLOv7. Слідом за ними йдуть Vision Transformers (ViT), такі як Swin і DualSwin, а також PP-YOLOE, YOLOR, YOLOv4 і EfficientDet. Ця інформація проілюстрована на рисунку 3.1.

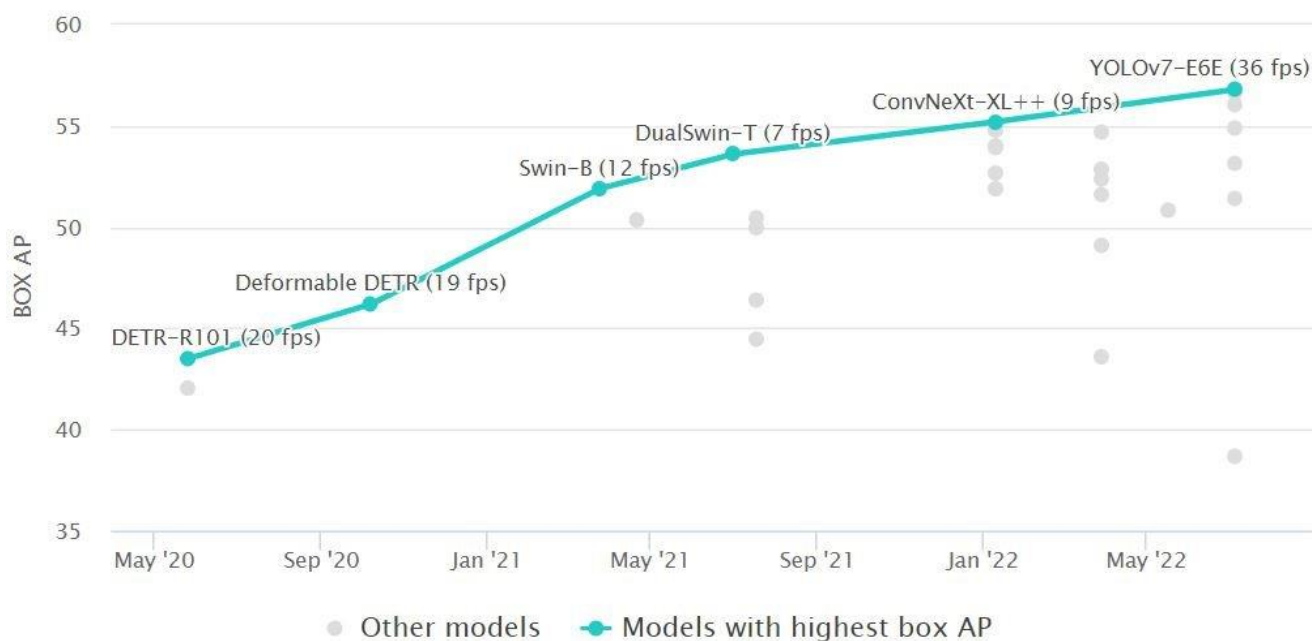


Рисунок 3.1 – Виявлення об'єктів у режимі реального часу на COCO Benchmark: сучасний рівень середньої точності (AP) [15]

3.6.2 Найшвидший алгоритм виявлення об'єктів у реальному часі

Крім того, у наборі даних MS COCO важливим еталонним показником є час висновку (мс/кадр, менше — краще) або кадрів за секунду (FPS, вище — краще). Швидкий розвиток технології комп'ютерного зору є очевидним, коли розглядати порівняння часу висновку.

Виходячи з поточного часу висновку, що зображено на рисунку 3.2, YOLOv7 досягає 3,5 мілісекунд на кадр у порівнянні з 12 мілісекундами для YOLOv4 і 29 мілісекундами для популярного YOLOv3. Але впровадження YOLO (одноступінчатого детектора) значно скорочує час висновку порівняно з усіма раніше створеними методами, такими як двоступінчастий метод Mask R-CNN (333 мс).

На технічному рівні значуще порівняння різних архітектур і версій моделей досить складне.



Рисунок 3.2 – Порівняльний графік різних моделей по швидкості обробки відео (кадрів за секунду) [15]

Порівнюючи останню версію YOLO - YOLOv8 з YOLOv7 і YOLOv6 - остання версія (YOLOv8) показує найкращу продуктивність у тестах реального часу, це можна побачити на рисунках 3.3 та 3.4.

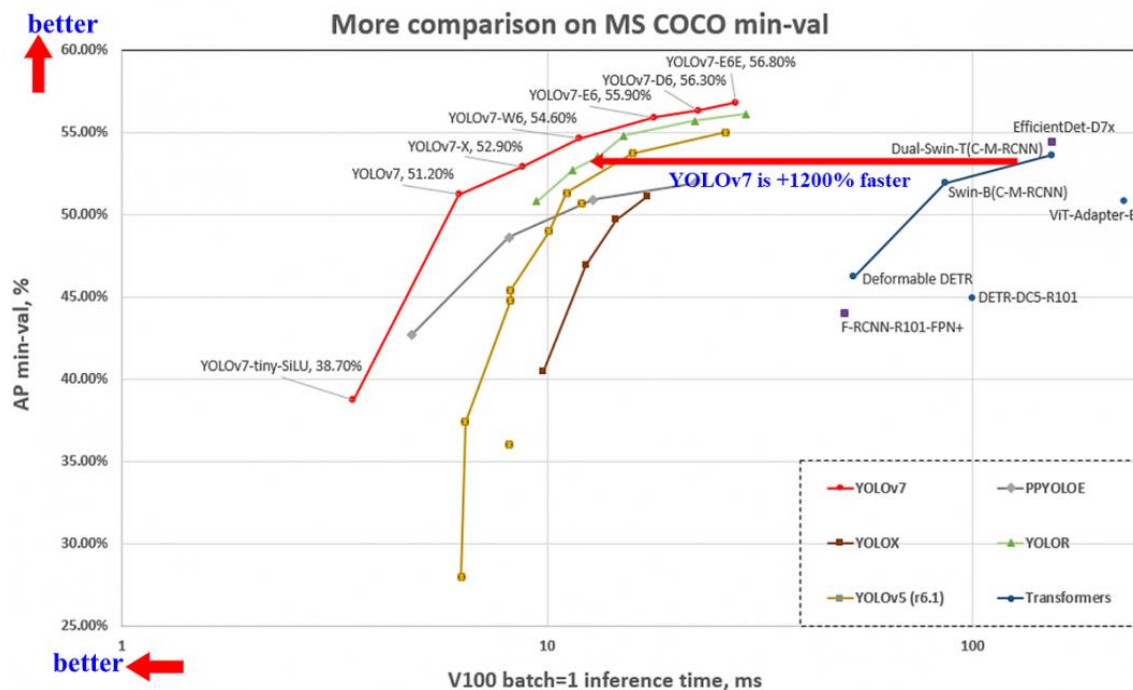


Рисунок 3.3 – Порівняння продуктивності YOLOv7, YOLOv5, YOLOR і Vit Transformers [15]

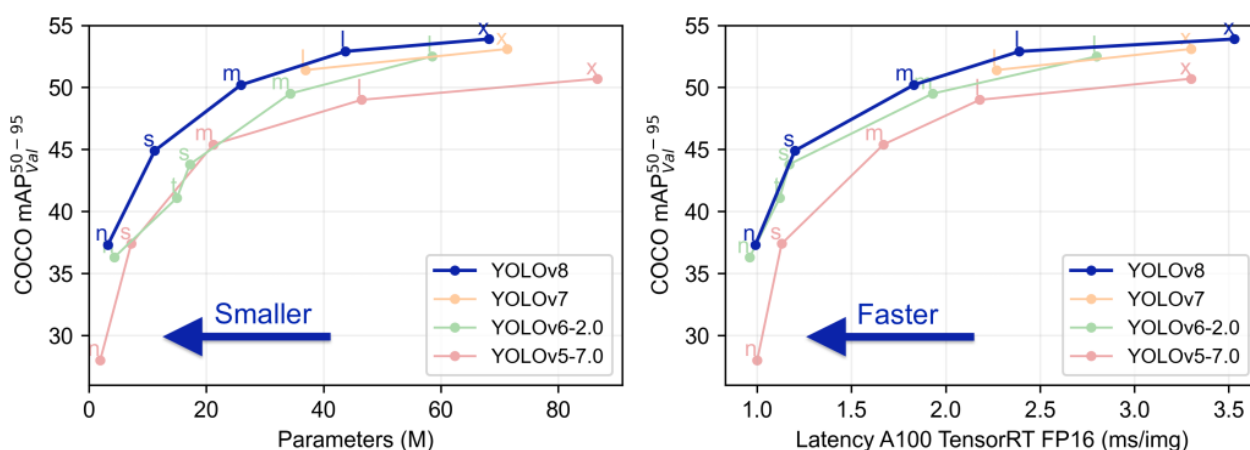


Рисунок 3.4 – Порівняння алгоритмів YOLO YOLOv7 і YOLOv8 [15]

Випадки використання для виявлення об'єктів дуже різноманітні; існує майже нескінченна кількість можливостей зробити комп'ютери схожими на

людей, автоматизувати ручні завдання або створити нові продукти та служби штучного інтелекту. Він був реалізований у програмах комп'ютерного бачення для різноманітних застосувань від спортивного виробництва до аналізу продуктивності.

3.7 Найпопулярніші алгоритми виявлення об'єктів

Поширені алгоритми виявлення цілей включають згорткову нейронну мережу (R-CNN, регіональну згорткову нейронну мережу), швидкий R-CNN і YOLO (You Only Look Once). R-CNN належить до сімейства R-CNN, тоді як YOLO належить до сімейства одноразових детекторів. Нижче ми надаємо огляд і відмінності поширених алгоритмів виявлення об'єктів, що зображені на рисунку 3.5.

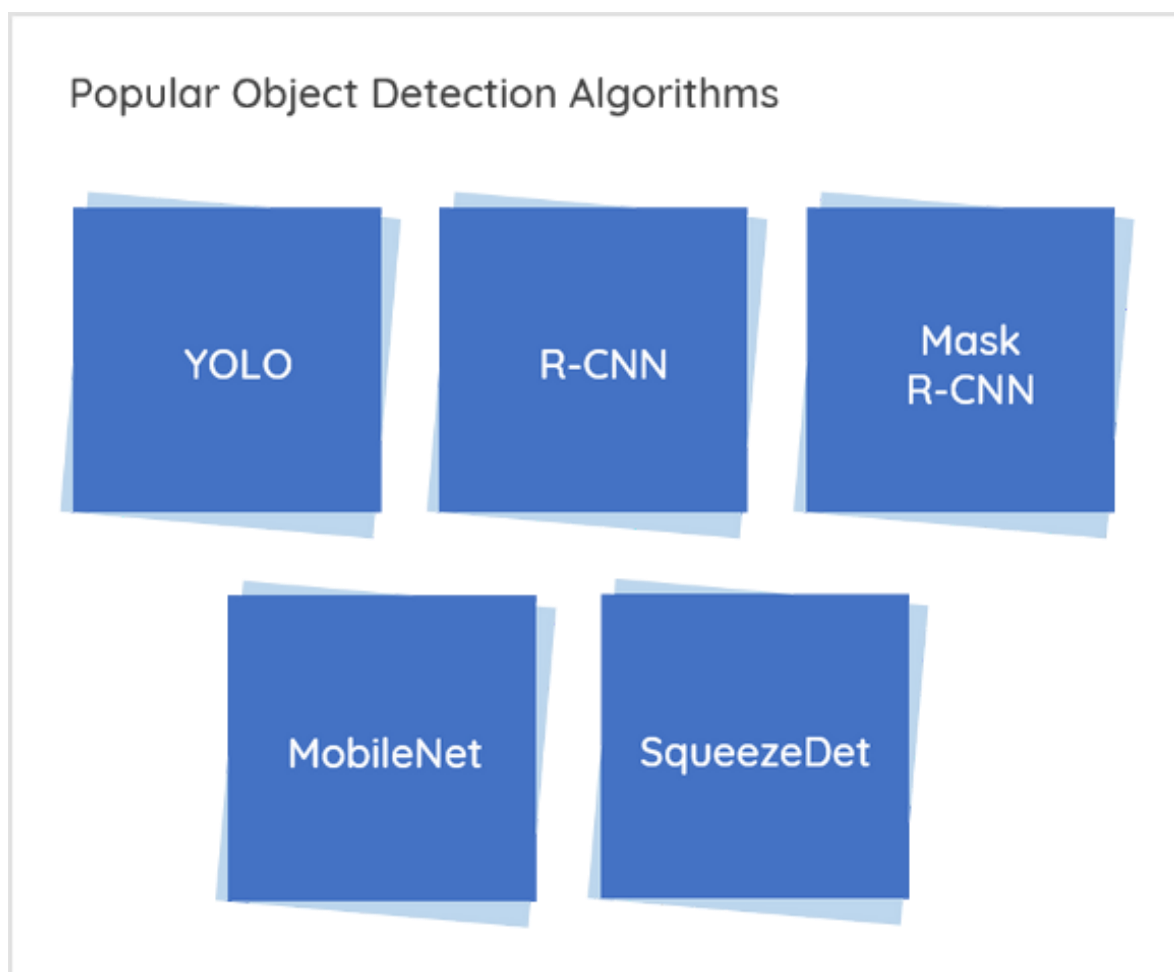


Рисунок 3.5 – Популярні алгоритми виявлення об'єктів [15]

3.7.1 YOLO

Абревіатура YOLO, скорочення від «You Only Look Once», набула широкої популярності як алгоритм виявлення об'єктів у реальному часі, який використовують відомі компанії комп'ютерного зору в різних комерційних продуктах. Спочатку представлений у 2016 році оригінальний детектор об'єктів YOLO з тих пір зазнав архітектурних удосконалень, які зробили його значно швидшим за аналоги [17].

З тих пір було випущено кілька версій і варіантів YOLO, кожна зі значними покращеннями продуктивності та ефективності. YOLOv4 — це вдосконала версія офіційного YOLOv3. Дослідницькі групи випустили власні версії YOLO, такі як YOLOv5, YOLOv7 або YOLOv8. Основними нововведеннями є покращення мозаїчних даних, самозмагання та нормалізація між міні-серіями.

Відомий розробник Ultralytics створив ще одну широко визнану модель під назвою YOLOv8. Ця модель надає пріоритет швидкості, точності та зручній функціональності, що робить її ідеальним вибором для ряду завдань, таких як виявлення та відстеження об'єктів, сегментація екземплярів, класифікація зображень і оцінка пози.

3.7.2 SSD

SSD – популярний одноразовий детектор, який може передбачати кілька класів. Цей метод використовує єдину глибоку нейронну мережу для виявлення об'єктів на зображеннях шляхом дискретизації вихідного простору обмежувальних рамок у набір стандартних рамок, що охоплюють різні пропорції та масштаби в кожному місці карти функцій [18].

Детектор об'єктів зображення генерує бали для наявності кожної категорії об'єктів у кожному стандартному полі та налаштовує поля, щоб краще відповідати формі об'єкта. Крім того, мережа об'єднує прогнози з кількох карт функцій з різною роздільною здатністю для обробки об'єктів різного розміру.

Детектори SSD легко навчити та інтегрувати з програмними системами, які потребують компонентів виявлення об'єктів. Порівняно з іншими одноступеневими методами, SSD досягає вищої точності навіть із меншим розміром вхідного зображення.

3.7.3 R-CNN

Згорткові нейронні мережі на основі регіону або функції регіону з CNN (R-CNN) є проривними методами застосування глибоких моделей для виявлення об'єктів. Модель R-CNN спочатку вибирає кілька областей пропозицій із зображення (наприклад, блоки прив'язки є одним із методів вибору), а потім позначає їхні категорії та обмежувальні рамки (наприклад, зміщення). Ці теги створюються на основі попередньо визначених класів, переданих програмі. Потім вони використовували згорткову нейронну мережу (CNN) для виконання прямих обчислень і вилучення ознак із кожної запропонованої області [19].

У R-CNN вхідне зображення спочатку ділиться на майже дві тисячі регіональних частин, а потім до кожної області застосовується згорткова нейронна мережа. Обчисліть розмір області та вставте правильну область у нейронну мережу. Можна зробити висновок, що такий детальний підхід може призвести до обмежень у часі. Порівняно з YOLO час навчання значно довший, оскільки рамки класифікуються та створюються окремо, а нейронна мережа застосовується до однієї області за раз.

У 2015 році Fast R-CNN був розроблений з метою значно скоротити час навчання. Original R-CNN обчислює функції нейронної мережі незалежно для кожної з 2000 цікавих областей, тоді як Fast R-CNN запускає нейронну мережу один раз для всього зображення. Це дуже схоже на архітектуру YOLO, але через простоту коду YOLO все ще є швидшою альтернативою Fast R-CNN.

Наприкінці мережі є новий метод, званий об'єднанням регіонів інтересу (ROI), який розрізає, змінює форму та класифікує (класифікація зображень) кожну область інтересу з вихідного тензора мережі. Це робить Fast R-CNN більш

точним, ніж оригінальний R-CNN. Однак завдяки цій техніці виявлення навчання Fast R-CNN і детектора R-CNN вимагає менше введення даних.

3.7.4 Маска R-CNN

R-CNN є подальшим розвитком Fast R-CNN. Різниця між ними полягає в тому, що Mask R-CNN додає гілку передбачення маски об'єкта паралельно до існуючої гілки виявлення обмежувальної рамки. Маску R-CNN легко навчити та додає невеликі накладні витрати на Faster R-CNN; він може працювати зі швидкістю 5 кадрів в секунду [20]. Приклад маски R-CNN із сегментацією зображення та виявленням об'єкта зображення зображено на рисунку 3.6.



Рисунок 3.6 – Приклад маски R-CNN із сегментацією зображення та виявленням об'єкта зображення [15]

3.7.5 SqueezeDet

SqueezeDet — це назва глибокої нейронної мережі для комп'ютерного зору, випущеної в 2016 році. Розроблена спеціально для автономного водіння, SqueezeDet використовує технологію комп'ютерного зору для виявлення об'єктів [21]. Як і YOLO, це алгоритм одноразового детектора.

У SqueezeDet згорткові шари використовуються лише для вилучення карт функцій, а також служать вихідними шарами для обчислення обмежувальних рамок і ймовірностей класів. Конвеєр виявлення моделі SqueezeDet складається лише з одного прогону нейронної мережі, що робить його надзвичайно швидким.

3.7.6 MobileNet

MobileNet — це одноразова мережа виявлення кількох ящиків, розроблена для виконання завдань виявлення об'єктів [22]. Модель реалізована за допомогою фреймворку Caffe. Як згадувалося раніше, вихід моделі є типовим вектором, що містить дані відстежуваного об'єкта.

3.7.7 YOLOR

YOLOR – це новий детектор об'єктів, запущений у 2021 році. Алгоритм застосовує як неявні, так і явні знання для навчання моделі. Тому YOLOR може вивчати універсальне представлення та виконувати кілька завдань за допомогою цього універсального представлення [23].

Інтеграція неявних знань у явні через вирівнювання простору ядра, уточнення прогнозів і багатозадачне навчання. Завдяки такому підходу YOLOR досягає значного покращення продуктивності розпізнавання об'єктів.

Порівняно з іншими методами виявлення об'єктів у тесті набору даних COCO, MAP YOLOR на 3,8% вищий, ніж PP-YOLOv2 за тієї самої швидкості висновку. Швидкість логічного висновку збільшена на 88% порівняно з Scaled-

YOLOv4, що робить його найшвидшим детектором об'єктів у реальному часі, доступним на даний момент.

Висновки до розділу 3

У третій частині нашої роботи був проведений поглиблений аналіз теми виявлення об'єктів, розкриваючи ключові аспекти цього напрямку в області комп'ютерного зору. Перш за все, концепція виявлення об'єктів чітко визначена як процес комп'ютерних систем, які визначають місцезнаходження та класифікують об'єкти на зображеннях або відео.

Далі було розглянуто останні технологічні досягнення в області комп'ютерного зору, зосереджуючись на перевагах і недоліках різних підходів. Виділено основні переваги виявлення об'єктів, включаючи потенціал для автоматизації та вдосконалення в різних сферах, таких як безпека, медицина та автономні транспортні засоби.

Порівняно сучасні методи виявлення об'єктів, відзначено їх переваги та недоліки. Вони особливо підкреслюють важливість швидкості та точності виявлення об'єктів, оскільки це ключові характеристики для вирішення реальних завдань.

За результатами бенчмарків ми вирішили обрати алгоритм YOLOv8 для завдання виявлення безпілотних літальних апаратів (БПЛА) у небі. Цей вибір виправданий високою ефективністю та швидкістю алгоритму, що робить його ідеальним для виявлення об'єктів у режимі реального часу, наприклад дронів.

4 РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ

4.1 Засоби розробки

Для навчання моделі було обрано мову програмування Python, щоб реалізувати послуги виявлення дронів. Python має різноманітні бібліотеки для машинного навчання, статистичного аналізу, спрощення навчання нейронних мереж, побудови графіків, збору даних тощо.

Для заданих цілей буде використовуватись Ultralytics YOLOv8, розширену модель, яка базується на досягненнях попередніх ітерацій YOLO. У цій оновленій версії представлено нові елементи та вдосконалення для підвищення продуктивності та адаптивності. YOLOv8 спеціально розроблений, щоб віддати пріоритет швидкості, точності та зручності користувача, що робить його винятковим варіантом для широкого спектру завдань, включаючи виявлення та відстеження об'єктів, сегментацію екземплярів, класифікацію зображень та оцінку пози.

Щоб створити набір даних для позначення дронів на зображеннях, буде використатись Roboflow, інструмент, який полегшує завдання комп'ютерного зору у сфері глибокого навчання. Це дозволяє розробникам створювати програми комп'ютерного зору незалежно від їхніх навичок чи досвіду.

Розробка веб-застосунку здійснюється в додатку Visual Studio Code з використанням бібліотеки React.

4.1.1 Python

Використання Python, мови програмування на підвищеному рівні, широко поширене у створенні веб-додатків, різноманітного програмного забезпечення та моделей машинного навчання. Ця мова також служить основою для програм комп'ютерного зору [24]. З інтеграцією штучного інтелекту та машинного навчання в різні сектори та платформи відомі корпорації почали інвестувати

значні кошти в ці галузі. Як наслідок, зріс попит на експертів у галузі машинного навчання та штучного інтелекту.

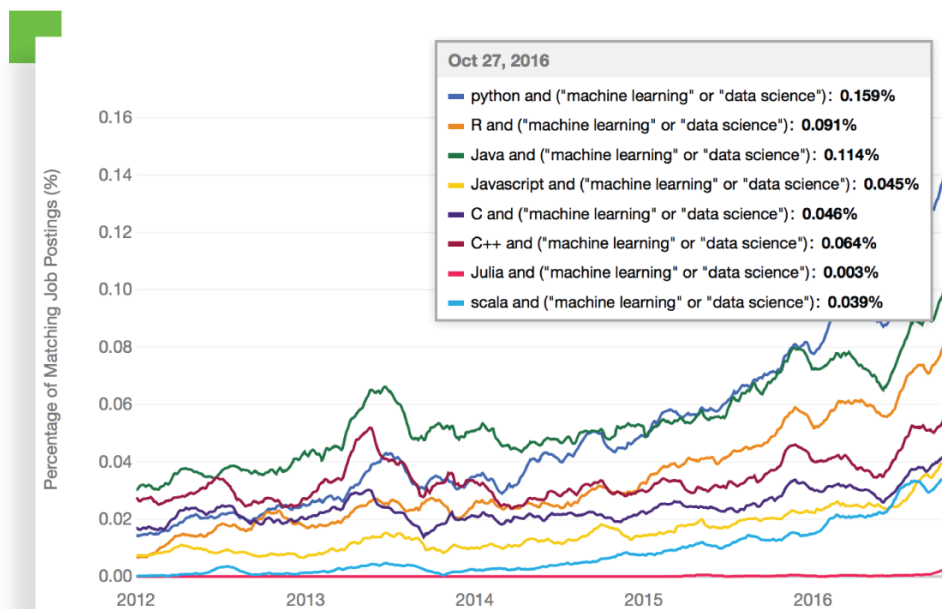


Рисунок 4.1 – Графік пошуку тенденцій мов програмування для штучного інтелекту та машинного навчання на indeed.com [24]

Згідно з графіком на рисунку 4.1, Python для штучного інтелекту та машинного навчання є основною мовою програмування.

Використання Python для штучного інтелекту та машинного навчання має численні переваги.

Домінування Python у сфері штучного інтелекту можна пояснити його чудовою бібліотечною екосистемою. Широкий спектр доступних бібліотек є ключовим фактором, який робить Python кращою мовою програмування для ШІ. Ці бібліотеки, які по суті є колекціями попередньо написаних модулів коду, отриманих із різних платформ, таких як PyPi, надають розробникам можливість легко впроваджувати певні функції та виконувати різноманітні операції. Використовуючи ці бібліотеки, розробники можуть уникнути необхідності

починати програмування з нуля для кожного проекту. Це особливо вигідно в машинному навчанні, де необхідна безперервна обробка даних. Бібліотеки Python забезпечують плавний доступ, обробку та трансформацію даних, сприяючи ефективним робочим процесам машинного навчання.

Індустрія машинного навчання та штучного інтелекту потребує зручної та ефективної обробки величезних обсягів даних. Однією з переваг роботи в цій галузі є низький вступний бар'єр, що дає змогу більшій кількості науковців швидко прийняти Python для розробки ШІ без додаткового тягаря вивчення нової мови. Python, з його схожістю на повсякденну англійську, сприяє плавному процесу навчання. Його простий синтаксис спрощує роботу зі складними системами та встановлює чіткі зв'язки між різними елементами. Тому логічно використовувати Python для ШІ через його акцент на простоті.

Python є чудовим варіантом для машинного навчання завдяки своїй винятковій гнучкості. Ця мова програмування пропонує вибір між об'єктно-орієнтованим програмуванням або сценаріями, дозволяючи розробникам адаптуватися до свого бажаного стилю. Одна з переваг Python полягає в тому, що немає необхідності перекомпілювати вихідний код, що дозволяє програмістам вносити зміни та оперативно спостерігати за результатами. Крім того, Python можна легко інтегрувати з іншими мовами для досягнення певних цілей. Гнучкість Python дає змогу розробникам використовувати їхні бажані стилі програмування та навіть комбінувати різні підходи для ефективного вирішення різноманітних проблем.

Python пропонує значні переваги для розвитку ШІ та машинного навчання завдяки своїй незалежності від платформи. Ця універсальна мова не тільки зручна та легка для сприйняття, але й бездоганно працює на різних платформах, таких як Windows, MacOS, Linux, Unix тощо. Перенесення процесу з однієї платформи на іншу просто вимагає від розробників незначних коригувань і зміни кількох рядків коду для створення виконуваної версії, яка підходить для потрібної платформи.

Читабельність Python є винятковою, тому будь-який розробник Python легко може зрозуміти та внести зміни до чужого коду, а також копіювати чи

поширювати його. Ця відсутність плутанини, помилок і суперечливих парадигм сприяє безперервному обміну алгоритмами, ідеями та інструментами між професіоналами в галузі штучного інтелекту та машинного навчання, зрештою підвищуючи ефективність.

Розробникам штучного інтелекту слід віддавати перевагу своїм навичкам візуалізації, оскільки Python пропонує низку виняткових бібліотек, які служать потужними інструментами для представлення даних. Важливо відзначити, що штучний інтелект, глибоке навчання та машинне навчання мають можливість представляти дані в різних форматах, забезпечуючи читабельність і доступність.

4.1.2 Roboflow

Roboflow — це фреймворк розробника комп'ютерного бачення для кращого збору даних для попередньої обробки та навчання моделі, доповнення даних. Roboflow має публічні набори даних, легко доступні для користувачів, а також має доступ для завантаження власних даних. Roboflow підтримує різні формати анотацій [25]. У попередній обробці даних є такі кроки, як орієнтація зображення, зміна розміру, контрастування та інші.

Увесь робочий процес можна координувати з командами всередині структури. Для навчання моделі вже існує купа бібліотек моделей, таких як EfficientNet, MobileNet, Yolo, TensorFlow, PyTorch тощо. Після цього, моделі та параметри візуалізації також доступні, отже, охоплюючи весь стан техніки.

Roboflow використовується в різних галузях комп'ютерного зору для таких випадків, як виявлення витоку газу, виявлення рослин проти бур'янів, технічне обслуговування літаків, оцінка пошкоджень даху, супутникові зображення, лічильник трафіку, прибирання сміття та багато іншого.

Переваги Roboflow:

- простий у використанні завдяки інтуїтивно зрозумілому інтерфейсу;
- відмінна технічна підтримка з швидким реагуванням;
- добре розроблена програма;

- Пропонує широкий спектр функцій, не перевантажуючи користувачів;
- Спрощує робочий процес в одному робочому просторі;
- Надає нестандартний навчальний зошит;
- Покращує точність моделі класифікації зображень за допомогою служби доповнення зображень;
- Потенціал для допомоги початківцям у повторних питаннях – програмування;
- Спеціально розроблені інструменти для допомоги в розробці ігор AR, які використовують комп'ютерне бачення;
- Ліцензія спільноти корисна для навчання та початку роботи.

Недоліки:

- Очікується експорт класифікації до можливостей Snap AR;
- Бракує функціональності для масового редагування міток класів;
- Для класифікацій Yolo потрібна додаткова документація;
- Дехто вважає ціни високими та потенційно негнучкими для користувачів, яким не потрібні всі функції.

4.1.3 Visual Studio Code

Visual Studio Code (VS Code), доступний для операційних систем Windows, macOS і Linux, — це редактор вихідного коду, який поєднує в собі простоту з потужними можливостями. Він працює безпосередньо на вашому комп'ютері та пропонує підтримку різних мов програмування, включаючи JavaScript, Node.js і TypeScript. Крім того, VS Code надає розширення для багатьох інших мов, таких як C++, C#, Python, Go та Java.

По суті, VS Code — це стандартний текстовий редактор, який пропонує розширені можливості за допомогою розширень. Ці розширення розширюють існуючі можливості та функції редактора. Візьмемо, наприклад, розширення редактора, яке дозволяє розробникам використовувати автоматичне заповнення. Ця функція дозволяє автоматично пропонувати та завершувати блоки коду під час

написання. Це загальна функція, яку можна знайти в сучасних IDE, і вона значно спрощує процес розробки. Крім того, у вас є можливість налаштувати функцію динамічного попереднього перегляду. Це дозволяє переглядати веб-сторінки в режимі реального часу, коли вносяться зміни.

VS Code — це чудова програма для веб-розробки, яка пропонує текстовий інтерфейс, для роботи якого потрібні мінімальні ресурси. З інтегрованими всіма основними інструментами редактор забезпечує безперебійну розробку, додатково розширену широким набором розширень, які значно пришвидшують процес.

4.2 Підготовка даних для навчання моделі

Для успішного навчання моделі на виявлення дронів в небі було проведено попередню підготовку та збір даних. Завантажено близько 10 000 зображень із різних джерел Інтернету, які представляють різні об'єкти, що можуть з'явитися на фотографіях або відео. На цих зображеннях було зображені такі речі:

- дрони;
- літаки;
- гвинтокрили;
- птахи;
- ракети.

Цей набір даних був ретельно зібраний для ідентифікації різних об'єктів в атмосфері, які можуть викликати плутанину при ідентифікації дронів. Щоб забезпечити високу точність моделі, важливо включити різноманітні об'єкти, щоб уникнути помилкових виявлень і забезпечити надійну роботу системи.

Крім того, враховуючи необхідність підвищення точності виявлення, особливу увагу було приділено додаванню зображень об'єктів, які можуть нагадувати дрони, наприклад літаки, гірокоптери, птахів і ракети. Це допомагає моделі відрізнити дрон від інших об'єктів у небі та зменшує ймовірність помилкової тривоги.

Зібравши ці зображення, було завантажено їх на сервіс Roboflow, попередньо там зареєструвавшись, це зображено на рисунку 4.2. Функціональність цього сервісу стала важливим інструментом для подальших досліджень, оскільки дозволяє ефективно позначати місцезнаходження дронів і збільшувати кількість дронів шляхом попередньої обробки.

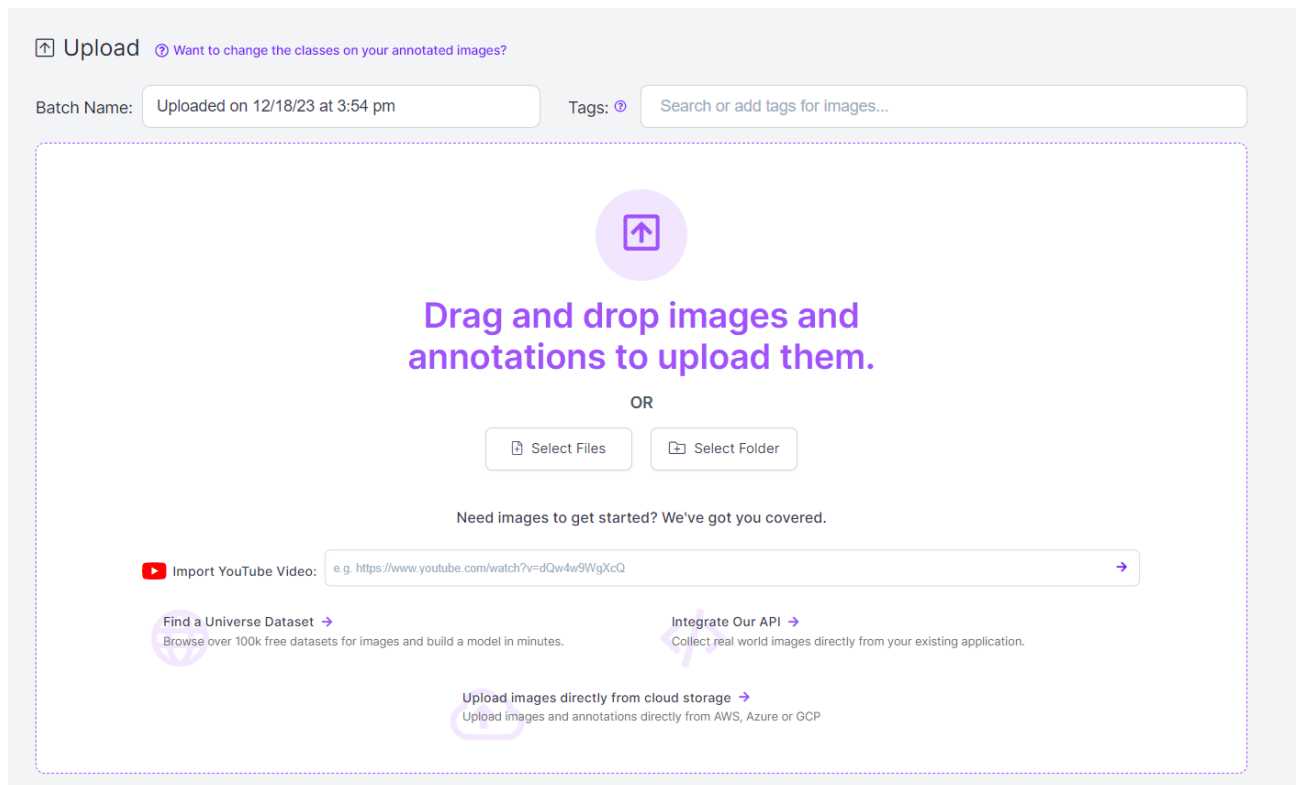


Рисунок 4.2 – Сторінка Roboflow для завантаження зображень

Після завантаження було використано функцію анотації Roboflow, щоб позначити зображення, які містили дрони. Це зображено на рисунку 4.3. Для інших зображень було позначано їх нульовими анотаціями та пропущено через процес, оскільки вони не містять дронів. Мета цього стратегічного рішення — дати нейронній мережі більш точне розуміння конкретного місця розташування дрона, що підвищить точність моделі під час навчання. Завдяки такому підходу ми не лише готуємо дані для подальшої обробки, але й надаємо моделі контекст, необхідний для ефективного навчання й оцінювання.

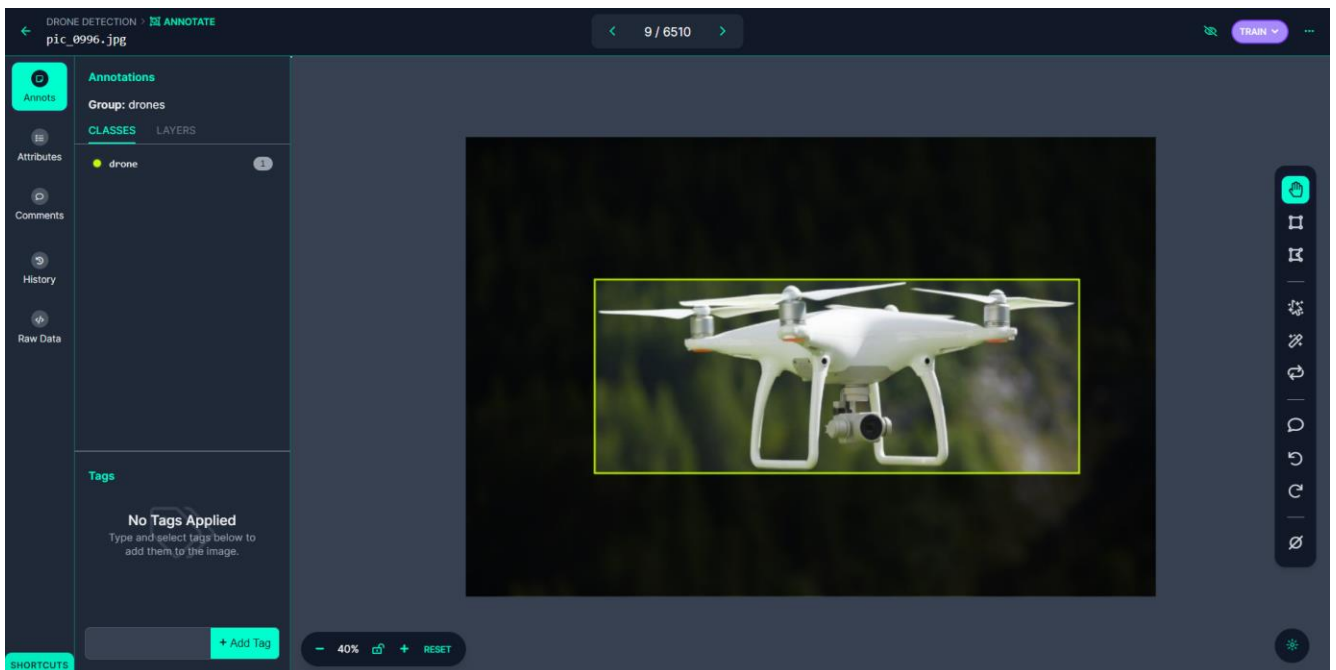


Рисунок 4.3 – Додавання анотації до зображень

Після додавання анотацій зображення попередньо обробляються, а кількість збільшується.

Попередня обробка зображень — це етап форматування зображень перед їх використанням для навчання моделі та висновків. Це включає, але не обмежується зміною розміру, орієнтації та корекцією кольору [27].

Попередня обробка зображення також може скоротити час навчання моделі та збільшити швидкість розпізнавання моделі. Якщо вхідні зображення особливо великі, зменшення розміру цих зображень значно покращить час навчання моделі без істотного зниження продуктивності моделі.

Попередня обробка є важливим етапом очищення даних зображення перед його підготовкою до використання в моделях комп'ютерного зору. Попередня обробка важлива як з технічних причин, так і з причин продуктивності.

Повністю пов'язані шари в згорткових нейронних мережах, загальній архітектурі комп'ютерного зору, вимагають, щоб усі зображення були масивами однакового розміру. Якщо змінити розмір зображення, модель може не працювати належним чином.

Збільшення зображень — це операція, яка застосовується до зображень для створення різних версій схожого вмісту, щоб надати моделі ширший діапазон навчальних прикладів. Наприклад, довільна зміна повороту, яскравості або масштабу вхідного зображення вимагає від моделі враховувати, як об'єкт зображення виглядатиме за різних обставин.

Операції масштабування зображення є формою попередньої обробки зображення, але є ключова відмінність: етап попередньої обробки зображення застосовується як до навчального, так і до тестового наборів, тоді як розширення зображення застосовується лише до навчальних даних. Тому те, що в деяких випадках може бути додатковим перетворенням, в інших випадках може бути кращим як етап попередньої обробки.

Неможливо по-справжньому відобразити кожен сценарій реального світу, який може охоплювати модель. Ось де аугментація може допомогти. Доповнюючи зображення, можна збільшити розмір вибірки навчальних даних і додати нові випадки, які може бути важко знайти в реальному світі.

Це особливо важливо, коли набір зібраних даних може бути невеликим. Модель глибокого навчання відповідатиме прикладам, наведеним у навчанні, тому створення варіацій у вхідних зображеннях може створити нові корисні навчальні приклади.

Для наших зображень ми будемо використовувати такі методи доповнення даних (рисунок 4.4):

– змінити колір. Зміна кольору є прикладом трансформації зображення, яке можна застосовувати до всіх зображень (навчання та тестування) або випадково змінювати під час навчання лише як доповнення. Колір менш актуальний для нашої моделі. Якщо використовувати градації сірого, не доведеться турбуватися про збір кожного зображення. Модель може вивчати більш загальні характеристики об'єктів, які не залежать від кольору. Кольорові зображення зберігаються як червоні, зелені та сині значення, тоді як зображення у градаціях сірого зберігаються лише як діапазон від чорного до білого. Це означає, що для

CNN наша модель повинна використовувати лише одну матрицю на зображення замість трьох;

- віддзеркалення. Віддзеркалення зображення навколо осі X або Y змушує нашу модель визнати, що об'єкти не завжди потрібно читати зліва направо чи зверху вниз. Для контекстів, які покладаються на порядок (наприклад, текстова інтерпретація), перегортання може бути не логічним;

- випадкове обертання. Поворот зображення особливо важливий, коли модель можна використовувати в нестационарних місцях, наприклад у мобільних додатках. Обертання може бути складним, оскільки воно також створює «мертві пікселі» на краях зображення, а для обмежувальних рамок потрібна тригонометрія для оновлення будь-яких обмежувальних рамок;

- випадкова яскравість і експозиція. Регулювання яскравості доволіно освітлених і затемнених зображень найбільш застосовне, якщо модель може працювати при різних налаштуваннях освітлення. Важливо враховувати максимальне і мінімальне освітлення в кімнаті;

- випадковий шум. Додавання шуму до зображення може приймати різні форми. Поширеною технікою є «шум солі та перцю», коли пікселі зображення випадковим чином перетворюються на повністю чорні або повністю білі. Хоча навмисне додавання шуму до зображень може знизити ефективність навчання, це може стати метою, якщо модель переполює неправильні функції.

Після попередньої обробки та доповнення зображень набір даних класифікується на три окремі групи: навчання, перевірка та тестування [28].

Навчальний набір даних відноситься до набору зразків даних, які використовуються для навчання моделі. Цей набір даних є фактичними даними, які використовуються для визначення ваг і зміщень нейронної мережі. Модель вивчає ці дані та отримує з них знання.

Набір даних перевірки служить неупередженим показником того, наскільки добре модель відповідає навчальному набору даних під час коригування гіперпараметрів моделі. Оскільки конфігурація моделі включає навички перевірки набору даних, оцінки стають все більш упередженими.

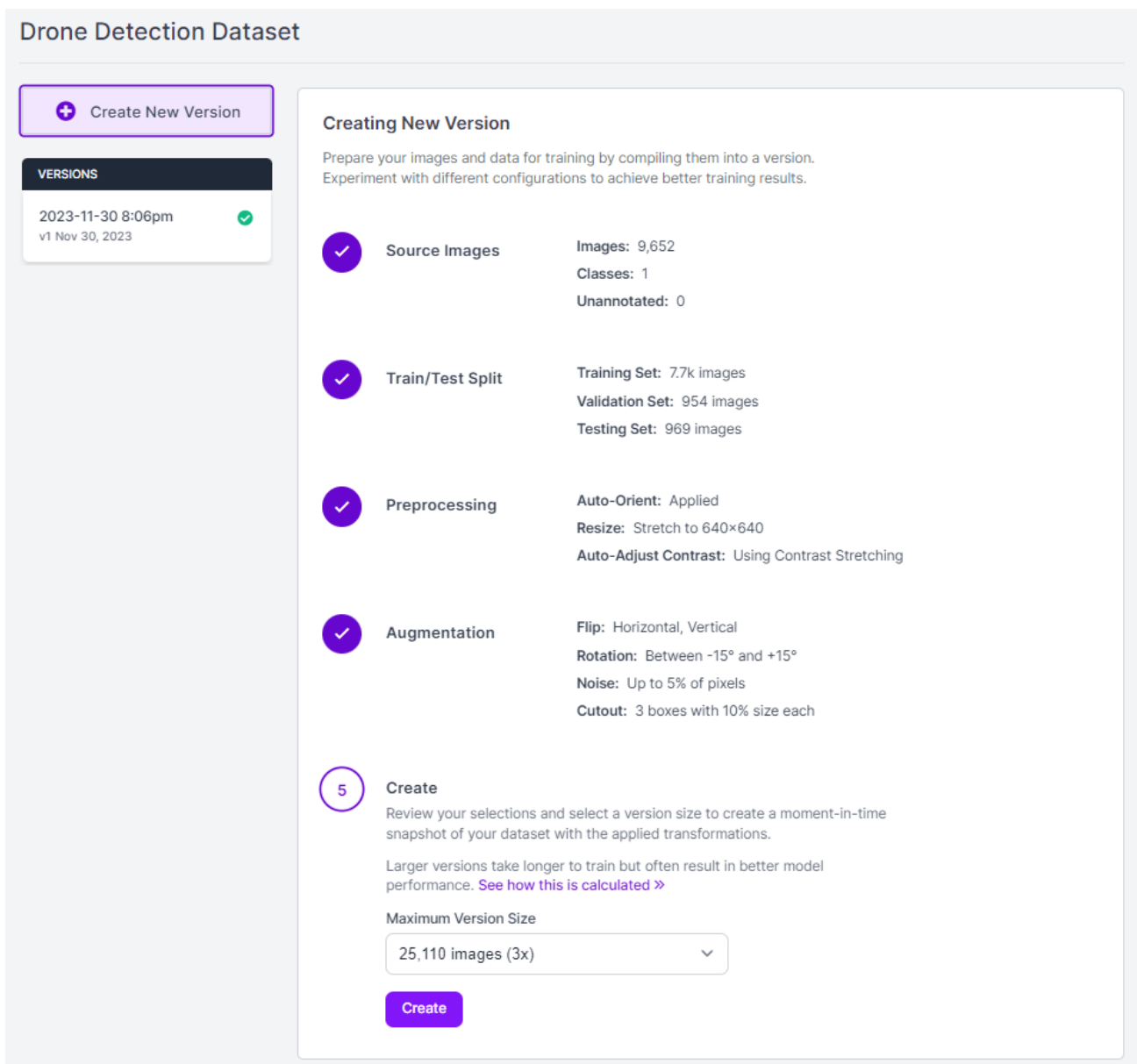


Рисунок 4.4 – Препроцесинг зображень для майбутнього навчання

Тестовий набір даних — це репрезентативний зразок даних, який використовується для створення неупередженої оцінки остаточної моделі, яка підганяється до набору навчальних даних.

Золотим стандартом для оцінки продуктивності моделі є тестовий набір даних, який слід використовувати лише після того, як модель пройшла ретельне навчання з використанням як навчального, так і перевірного наборів. Набори тестів зазвичай використовуються для порівняння та оцінки різних моделей. Однак не рекомендується використовувати перевірочний набір як заміну належного тестового набору, оскільки така практика вважається неоптимальною.

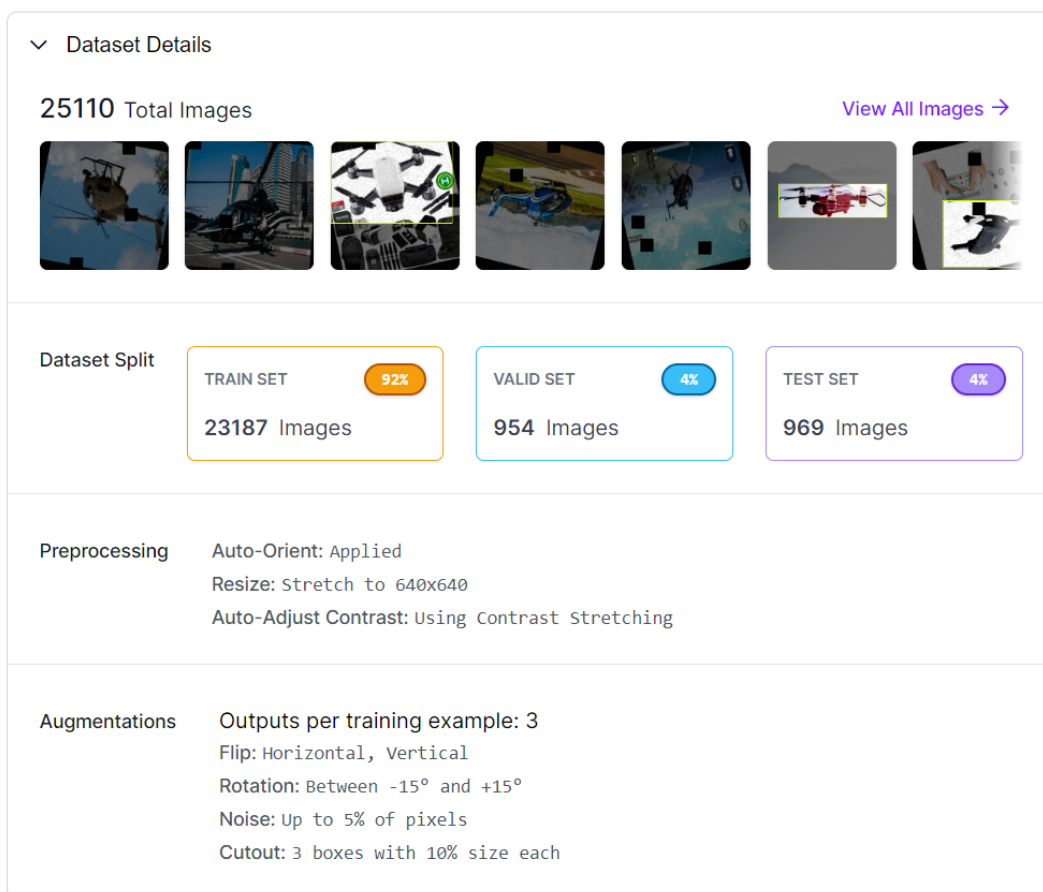


Рисунок 4.5 – Детальна інформація про створений датасет

Було отримано загалом 25110 зображень, з яких 23187 (92%) використовувалися для навчання, 354 (4%) для перевірки та 969 (4%) для тестування. Це можна побачити на рисунку 4.5.

Коли все це буде зроблено, було завантажено цей навчальний набір на комп'ютер даних у форматі YOLOv8 для подальшого використання, приклад зображено на рисунку 4.6.

QR-код з посиланням на датасет зображено на рисунку 4.7.

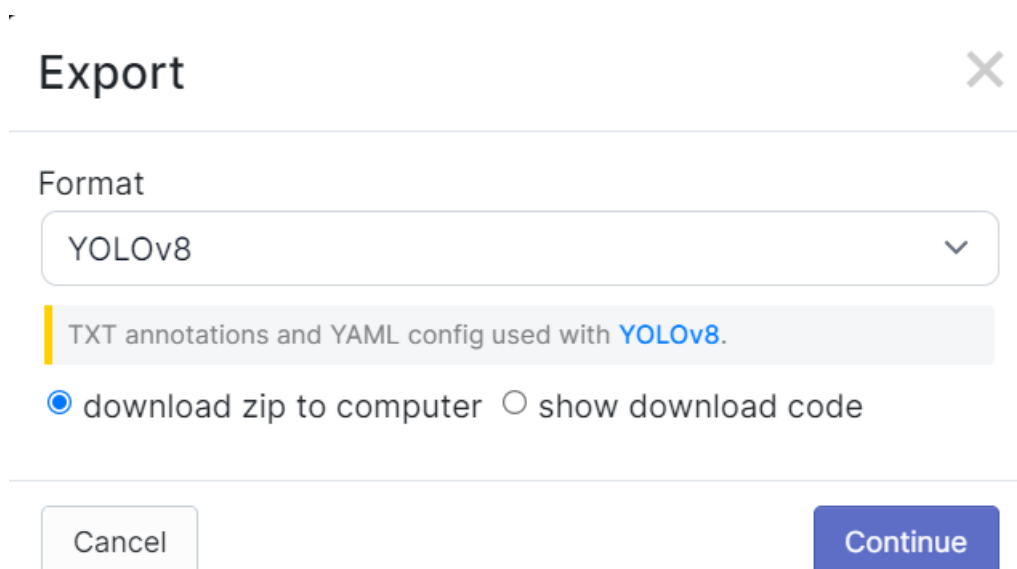


Рисунок 4.6 – Вікно для завантаження датасету



Рисунок 4.7 – QR-код із посиланням на датасет

4.3 Навчання моделі

Після завантаження та розкриття архіву, було отримано такі папки і файли які зображені на рисунку 4.8:

- images – папка в якій знаходяться підпапки train, val, test, в саме яких знаходяться наші зображення;

– labels – папка в якій знаходяться підпапки train, val, test, в саме яких знаходяться txt файли з координатами рамок для дронів. Приклад txt файлу з координатами рамки для дрону зображено на рисунку 4.9;

– data.yaml – файл, в якому записано основна інформація про датасет (його знаходження та інше). Його вміст можна побачити на рисунку 4.10.

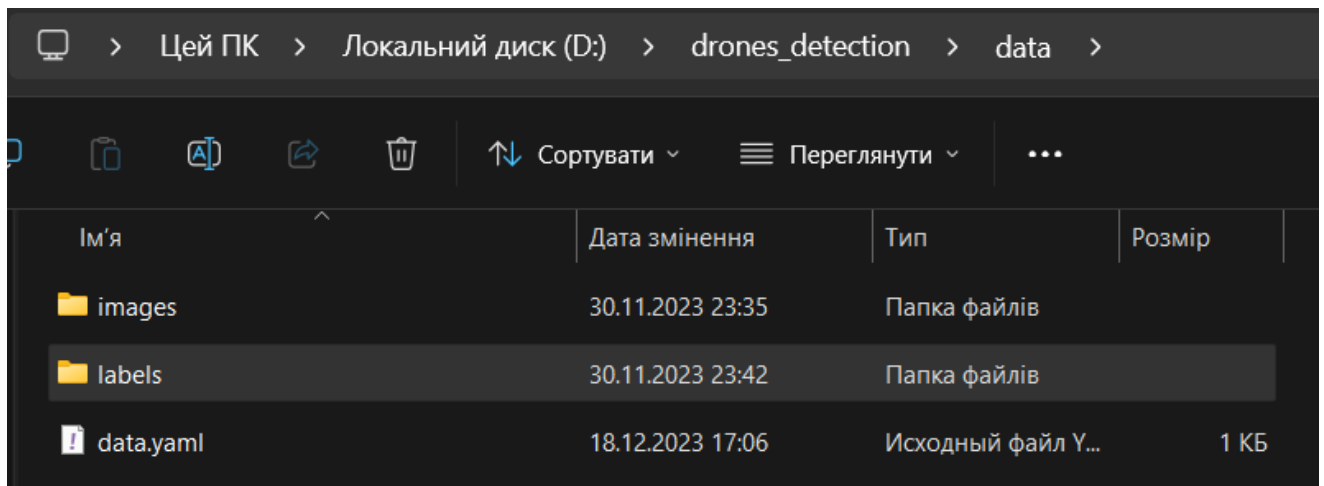


Рисунок 4.8 – Структура датасету

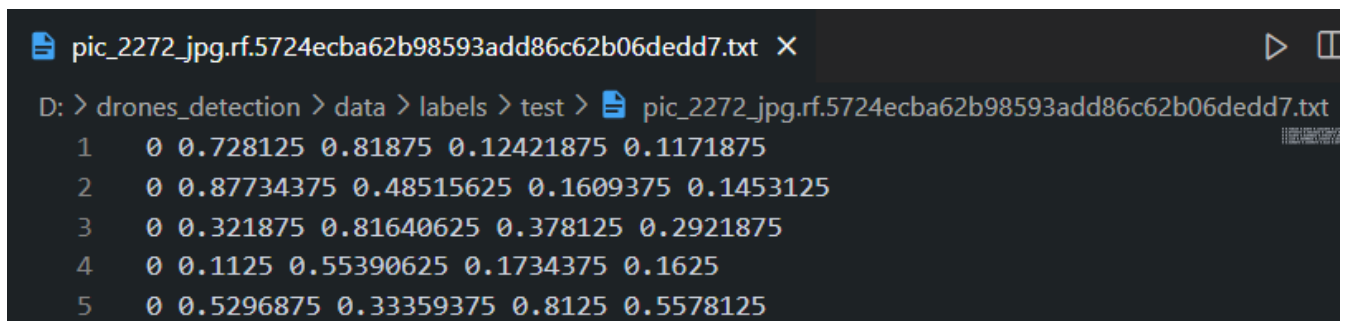
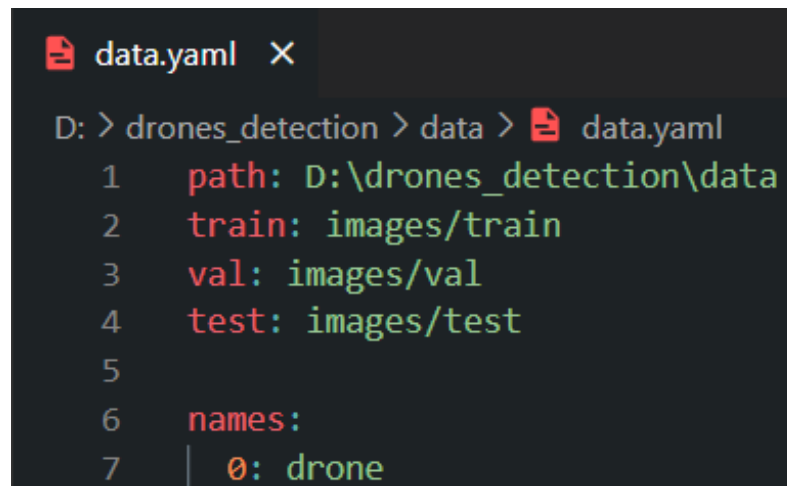


Рисунок 4.9 – Приклад txt файлу з координатами рамки для дрону

A screenshot of a code editor window titled 'data.yaml'. The editor shows a file path 'D: > drones_detection > data > data.yaml' and the following YAML content:

```
1 path: D:\drones_detection\data
2 train: images/train
3 val: images/val
4 test: images/test
5
6 names:
7   0: drone
```

Рисунок 4.10 – data.yaml

Модель було навчено на комп'ютері з такими характеристиками:

- оперативна пам'ять – 16 ГБ;
- процесор - 12th Gen Intel(R) Core(TM) i7-12650H 2.30 GHz;
- відеокарта – NVIDIA GeForce RTX 3050 Laptop GPU.

Перед навчанням потрібно завантажити певні бібліотеки та встановити правильні версії для належного навчання.

По-перше, потрібно встановити CUDA, апаратну та програмну архітектуру для паралельних обчислень, яка дозволяє значно підвищити продуктивність обчислень завдяки використанню графічних процесорів Nvidia [29].

Є кілька переваг використання CUDA для машинного навчання. По-перше, він забезпечує паралельну обробку на графічних процесорах NVIDIA, значно покращуючи швидкість і продуктивність завдань навчання та логічного висновку. Це означає, що складні моделі машинного навчання можна навчати та виконувати швидше, скорочуючи час і витрати.

Крім того, CUDA пропонує ефективне керування пам'яттю та безперебійну передачу даних між процесором і графічним процесором, максимізуючи використання наявних ресурсів. Ця функція виявляється особливо корисною при роботі з великими наборами даних і завданнями, які потребують значної обчислювальної потужності.

CUDA пропонує явну перевагу завдяки широкій підтримці спільноти машинного навчання. Численні відомі фреймворки глибокого навчання, включаючи TensorFlow і PyTorch, пропонують бездоганну інтеграцію з прив'язками CUDA, полегшуючи легке включення прискорення GPU.

Використовуючи CUDA, експерти в галузі машинного навчання можуть використовувати величезні обчислювальні можливості графічних процесорів, дозволяючи їм ефективно навчати та впроваджувати складні моделі. Ця технологія має потенціал для підвищення точності та ефективності в широкому діапазоні програм, таких як розпізнавання зображень і мови, обробка природної мови та системи рекомендацій.

CUDA забезпечує значні переваги для машинного навчання, включаючи прискорене навчання та логічні висновки, ефективне використання ресурсів і широку підтримку.

Крім того, за допомогою Conda було створено віртуальне середовище для Python і встановлено ці пакети, як показано на рисунку 4.11.

Після успішного встановлення необхідних пакетів і CUDA наступним кроком є створення файлу під назвою `main.py` для запуску нашої моделі. У цьому файлі було визначено та налаштовано різні параметри процесу навчання моделі.

Для початку було встановлено фундаментальні змінні, включаючи розташування навчальних і тестових наборів даних, а також структуру моделі. Важливо точно налаштувати ці змінні, щоб досягти найвищого рівня продуктивності та точності під час тренувального процесу.

Варто зазначити, що за основу нашого навчального процесу ми обрали модель YOLOv8 Nano. YOLOv8 — це потужна та швидка архітектура виявлення об'єктів, а її варіант «Nano» може використовувати оптимізовану версію для ресурсомістких платформ, таких як вбудовані системи або пристрої з обмеженими обчислювальними ресурсами [30].

Тому, створивши та налаштувавши файл `main.py`, можна ефективно запускати процес навчання моделі YOLOv8 Nano, забезпечуючи визначені параметри та архітектуру для досягнення оптимальних результатів.

```

requirements.txt X
D: > drones_detection > requirements.txt
1 # Ultralytics requirements
2 # Example: pip install -r requirements.txt
3
4 # Base -----
5 matplotlib>=3.3.0
6 numpy>=1.22.2 # pinned by Snyk to avoid a vulnerability
7 opencv-python>=4.6.0
8 pillow>=7.1.2
9 pyyaml>=5.3.1
10 requests>=2.23.0
11 scipy>=1.4.1
12 torch>=1.8.0
13 torchvision>=0.9.0
14 tqdm>=4.64.0
15
16 # Logging -----
17 # tensorboard>=2.13.0
18 # dvclive>=2.12.0
19 # clearml
20 # comet
21
22 # Plotting -----
23 pandas>=1.1.4
24 seaborn>=0.11.0
25
26 # Export -----
27 # onnx>=1.12.0 # ONNX export
28 # onnxsim>=0.4.1 # ONNX simplifier
29
30 # Extras -----
31 psutil # system utilization
32 py-cpuinfo # display CPU info
33 thop>=0.1.1 # FLOPs computation
34 # ipython # interactive notebook
35 # albumentations>=1.0.3 # training augmentations
36 # pycocotools>=2.0.6 # COCO mAP
37 # roboflow

```

Рисунок 4.11 – requirements.txt

Модель було навчено протягом 100 епох. По-перше, як концептуальний проект, було спробовано отримати найкращі результати за допомогою обмеженого навчання. Оскільки маємо майже 25000 зображень, навіть навчання для 100 епох займе багато часу і повинно дати пристойні результати.

Щоб мати справедливе порівняння між моделями, у всіх експериментах було встановлено розмір пакета 4. Також встановлено цей розмір, оскільки комп'ютер не може навчити моделі з більшими значеннями.

Оскільки дрон на деяких зображеннях може бути дуже маленьким, встановлено розмір зображення на роздільну здатність 640 під час навчання, хоча це збільшить час навчання, можна очікувати кращих результатів порівняно з результатом навчання з роздільною здатністю зображення. На рисунку 4.13 зображено приклад навчання нашої моделі.

```

main.py  X
D: > drones_detection > main.py
1  from ultralytics import YOLO
2
3  if __name__ == '__main__':
4
5      model = YOLO("yolov8n.yaml")
6
7      results = model.train(data="data/data.yaml", epochs=100, batch=4, imgsz=640)

```

Рисунок 4.12 – main.py

```

Administrator: Windows Pow...
Class      Images  Instances  Box(P)    R      mAP50  mAP50-95): 100% | 120/120 [00:08<00:00, 14.83it/s]
all        954      648        0.984     0.932  0.974    0.791

Epoch 96/100 GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
0.801G  0.7944    0.4538    1.198     2          640: 100% | 5797/5797 [07:08<00:00, 13.54it/s]
Class   Images  Instances  Box(P)    R      mAP50  mAP50-95): 100% | 120/120 [00:08<00:00, 14.24it/s]
all     954      648        0.982     0.932  0.974    0.79

Epoch 97/100 GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
0.801G  0.7931    0.4553    1.202     2          640: 100% | 5797/5797 [07:12<00:00, 13.40it/s]
Class   Images  Instances  Box(P)    R      mAP50  mAP50-95): 100% | 120/120 [00:08<00:00, 14.79it/s]
all     954      648        0.981     0.933  0.973    0.792

Epoch 98/100 GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
0.801G  0.7811    0.4513    1.189     1          640: 100% | 5797/5797 [07:10<00:00, 13.46it/s]
Class   Images  Instances  Box(P)    R      mAP50  mAP50-95): 100% | 120/120 [00:08<00:00, 14.93it/s]
all     954      648        0.98      0.934  0.973    0.792

Epoch 99/100 GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
0.801G  0.7689    0.4437    1.177     2          640: 100% | 5797/5797 [07:13<00:00, 13.38it/s]
Class   Images  Instances  Box(P)    R      mAP50  mAP50-95): 100% | 120/120 [00:07<00:00, 15.13it/s]
all     954      648        0.977     0.934  0.972    0.792

Epoch 100/100 GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
0.801G  0.7704    0.4427    1.183     3          640: 100% | 5797/5797 [07:13<00:00, 13.36it/s]
Class   Images  Instances  Box(P)    R      mAP50  mAP50-95): 100% | 120/120 [00:07<00:00, 15.08it/s]
all     954      648        0.977     0.934  0.972    0.793

100 epochs completed in 12.400 hours.
Optimizer stripped from runs\detect\train2\weights\last.pt, 6.3MB
Optimizer stripped from runs\detect\train2\weights\best.pt, 6.3MB

Validating runs\detect\train2\weights\best.pt...
Ultralytics YOLOv8.0.220 Python-3.9.18 torch-2.1.1+cu121 CUDA:0 (NVIDIA GeForce RTX 3050 Laptop GPU, 4096MiB)
YOLOv8n summary (fused): 168 layers, 3005843 parameters, 0 gradients, 8.1 GFLOPs
Class   Images  Instances  Box(P)    R      mAP50  mAP50-95): 100% | 120/120 [00:07<00:00, 17.01it/s]
all     954      648        0.977     0.934  0.972    0.792

Speed: 0.2ms preprocess, 3.3ms inference, 0.0ms loss, 0.7ms postprocess per image
Results saved to runs\detect\train2
(yolo-env) PS D:\drones_detection>

```

Рисунок 4.13 – Навчання моделі

Архітектуру моделі зображено в Додатках А-К.

Після завершення процесу навчання було вивчено та аналізовано різні показники ефективності та метрики, щоб повністю зрозуміти ефективність нашої

моделі навчання. Цей аналіз включатиме ретельний перегляд графіків, таблиць збитків, діаграм та інших наочних посібників.

Спочатку було зосереджено на графіках, які можуть містити такі важливі показники, як точність, відновлення та середнє значення F1. Графіки такого характеру допоможуть нам визначити здатність моделі розпізнавати об'єкти та виконувати важливі завдання на тестових даних.

Далі проаналізовано карту втрат, яка надає інформацію про регіони або особливості вхідного зображення, які модель вважає важливими. Це дає точну інформацію про те, де модель зосереджує свою увагу під час розпізнавання об'єктів.

Крім того, діаграми та інші візуальні елементи допоможуть зрозуміти зміни під час навчання та визначити можливі проблеми або вдосконалення, які можна внести в модель.

Цей всебічний аналітичний підхід дозволить отримати вичерпний і просвітлений звіт про продуктивність моделі та визначити шляхи подальшого вдосконалення моделі.

QR-код з посиланням на репозиторій з моделлю та кодом для навчання зображено на рисунку 4.14.



Рисунок 4.14 – QR-код із посиланням на репозиторій з моделлю та кодом для навчання

4.4 Аналіз результатів показників моделі

Показники продуктивності є ключовими інструментами для оцінки точності та ефективності моделей виявлення об'єктів. Вони показують, наскільки ефективно модель розпізнає та локалізує об'єкти на зображеннях. Крім того, вони допомагають зрозуміти, як модель обробляє помилкові позитивні та помилкові негативні результати. Ця інформація є критичною для оцінки та покращення продуктивності моделі. У цьому підрозділі було розглянуто різні показники продуктивності, пов'язані з YOLOv8, що вони означають і як їх інтерпретувати [31].

Показники виявлення об'єктів:

- intersection over Union (IoU): IoU – це метрика, яка кількісно визначає перекриття між прогнозованими і наземними обмежувальними рамками. Він відіграє фундаментальну роль в оцінці точності націлювання;

- середня точність (AP): AP обчислює площу під кривими точності та запам'ятовування, забезпечуючи єдине значення, яке підсумовує точність і продуктивність запам'ятовування моделі;

- середня точність (mAP): mAP розширює концепцію AP шляхом обчислення середнього значення AP для кількох класів об'єктів. Це корисно в сценаріях виявлення функцій із кількома класами та може забезпечити комплексну оцінку продуктивності моделі;

- точність і відкликання: точність кількісно визначає частку справжніх позитивних результатів серед усіх позитивних прогнозів і оцінює здатність моделі уникати помилкових позитивних результатів. Recall, з іншого боку, обчислює частку справжніх позитивних результатів серед усіх справжніх позитивних результатів і вимірює здатність моделі виявляти всі екземпляри класу;

- оцінка F1: Оцінка F1 є гармонійним середнім значенням точності та запам'ятовування, що забезпечує збалансовану оцінку продуктивності моделі з урахуванням хибнопозитивних і хибнонегативних результатів.

Частина результатів – це розбивка показників ефективності за категоріями. Ці деталі дуже корисні, коли ми намагаємося зрозуміти, наскільки добре модель працює для кожного конкретного класу, особливо в наборах даних з різними класами об'єктів.

Результати індикатора і їх оцінка:

- class: представляє назву класу об'єкта, у нашому випадку один клас «drone»;
- зображення: цей показник повідомляє про кількість зображень, які містять класи функцій у наборі перевірки. У нас є 954 зображення для перевірки;
- примірники: це кількість разів, коли клас з'являється на всіх зображеннях у наборі перевірки. Маємо 648 разів;
- P (точність) = 0,977: точність виявлених об'єктів, що вказує на кількість правильних виявлень;
- R (recall) = 0,934: здатність моделі розпізнавати всі екземпляри об'єкта на зображенні;
- mAP50 = 0,972: середня точність, розрахована з пороговим значенням перетину над U (IoU) 0,50. Це міра точності моделі, що враховує лише «легкі» виявлення;
- mAP50-95 = 0,791: середнє значення середньої точності, розраховане при різних порогових значеннях IoU в діапазоні від 0,50 до 0,95. Він забезпечує повне уявлення про продуктивність моделі на різних рівнях складності виявлення;
- швидкість роботи: попередня обробка = 0,2 мс, висновок = 3,3 мс, втрата = 0 мс, постобробка = 0,7 мс на зображення.

Окрім створення числових показників, метод також дає візуальні результати, які можуть забезпечити більш інтуїтивне розуміння продуктивності моделі. Ось розбивка візуальних результатів, які очікували:

Крива показників F1, що зображено на рисунку 4.15, відображає значення показників F1 для різних порогових значень. Ця крива слугує інструментом для аналізу та оцінки балансу моделі між виявленням хибно позитивних та хибно негативних результатів при різних значеннях порогу. Інтерпретація даної кривої

дозволяє здійснити узагальнене уявлення про ефективність моделі в залежності від обраного порогу, а також надає змогу визначити оптимальне значення порогу для досягнення бажаних результатів в дослідженні.

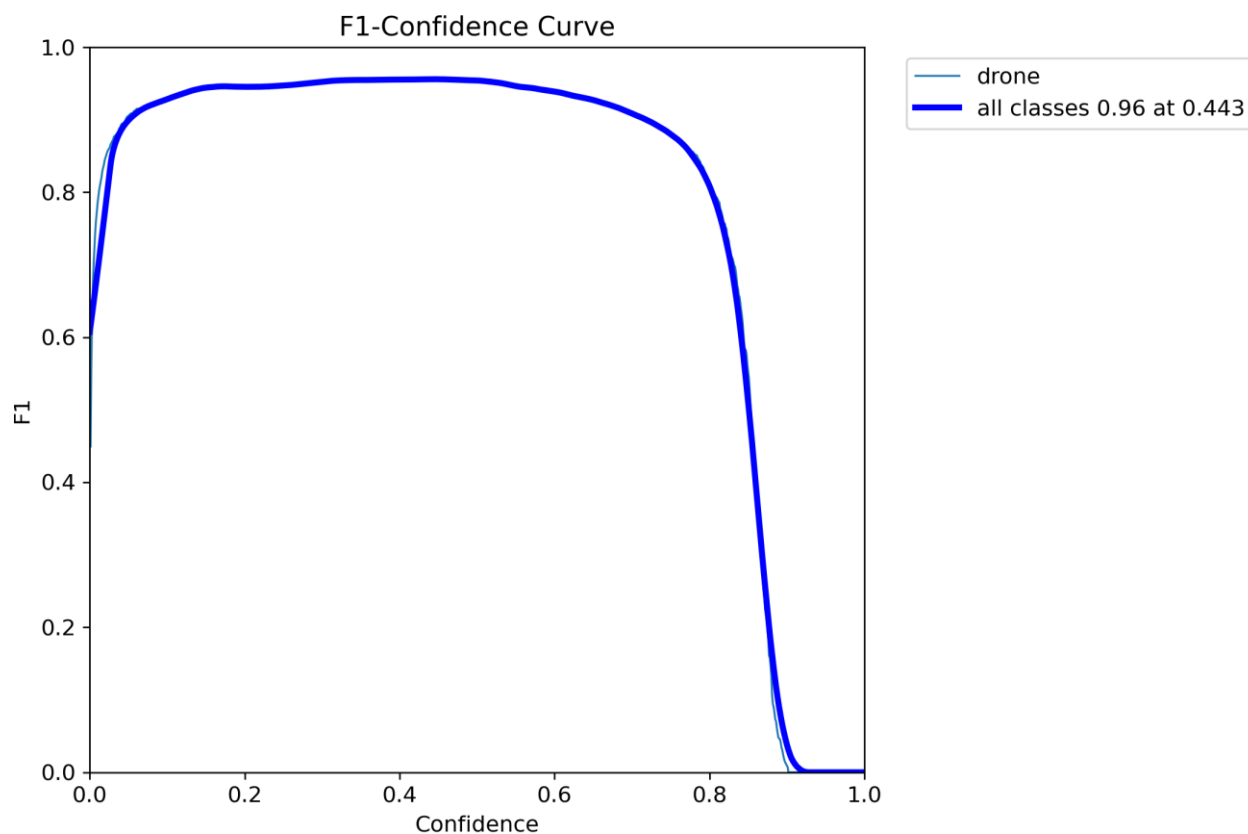


Рисунок 4.15 – Крива показників F1

Крива точності-запам'ятовування, що зображена на рисунку 4.16, є всебічною візуалізацією у вирішенні завдань класифікації, оскільки вона ілюструє компроміси між точністю та запам'ятовуванням при різних порогових значеннях. Особливо важливою стає ця крива при роботі з незбалансованими класами, де важливо знаходити оптимальний баланс між вірно позитивними та вірно негативними результатами. Аналізуючи цю криву, можна здійснити обґрунтовані вибори порогових значень для досягнення найбільш ефективної роботи класифікаційної моделі в конкретному контексті.

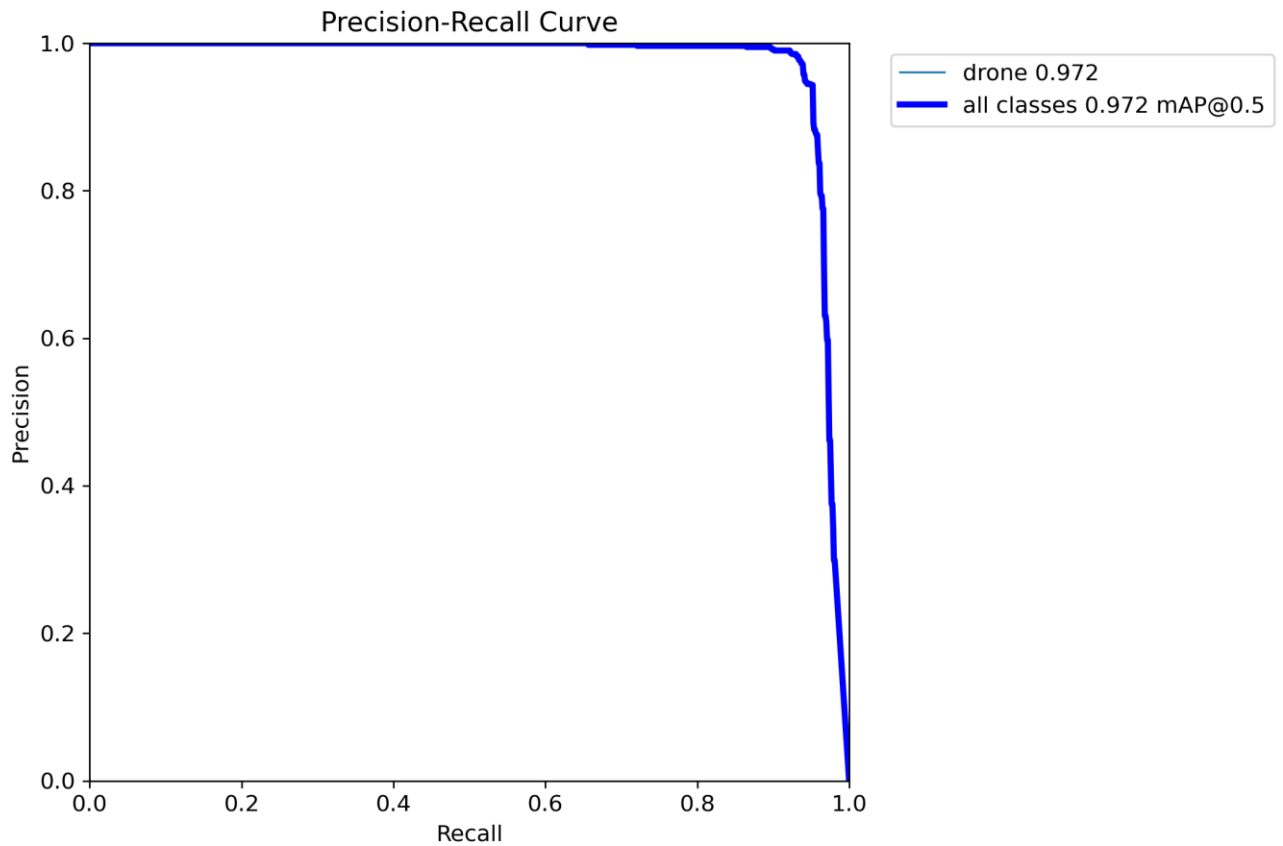


Рисунок 4.16 – Крива точності-запам'ятовування

Крива точності, що зображена на рисунку 4.17, є графічним відображенням значень точності в залежності від різних порогових значень. Це візуальне представлення дозволяє аналізувати, як точність моделі змінюється при зміні порогових значень. Дана крива надає інформацію про те, як ефективно модель працює при різних рівнях чутливості та специфічності, допомагаючи визначити оптимальні порогові значення для досягнення найкращих результатів в конкретному випадку.

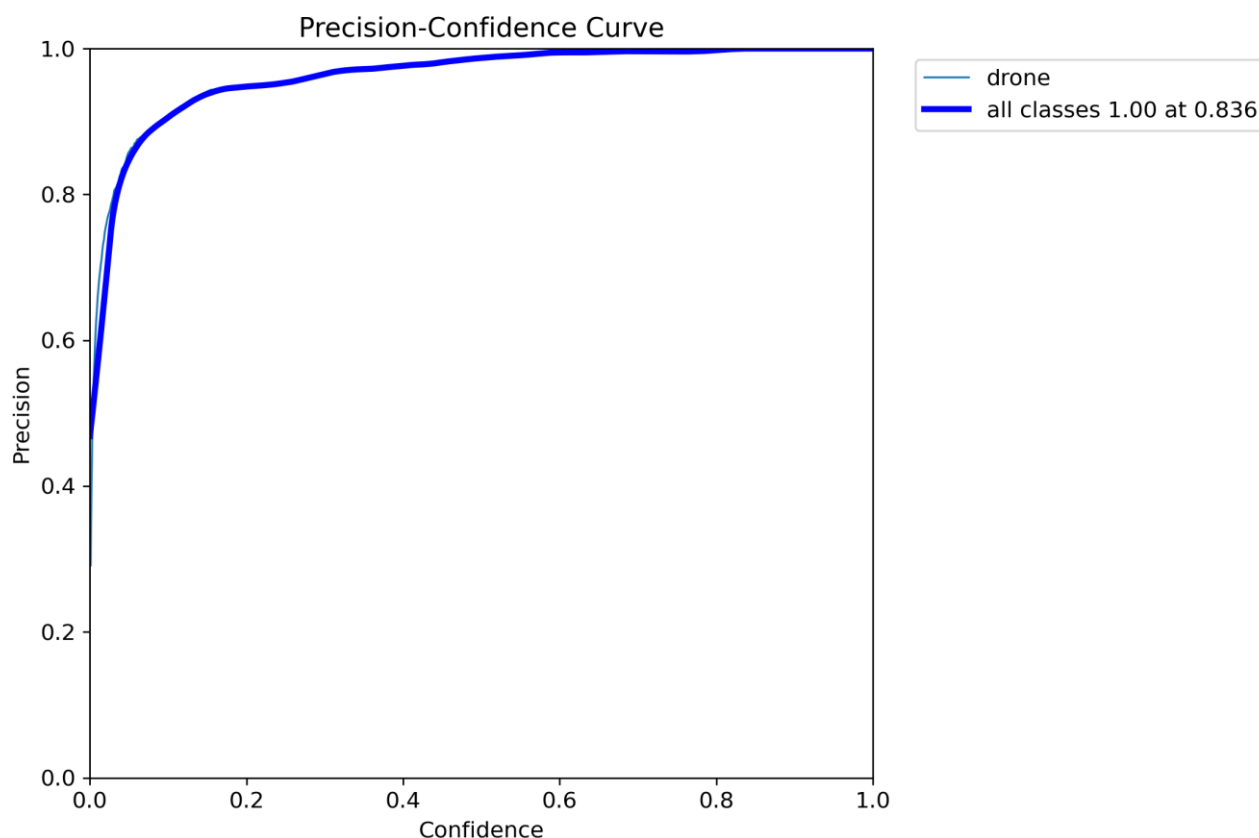


Рисунок 4.17 – Крива точності

Крива запам'ятовування, що зображена на рисунку 4.18, є графічним зображенням змін значень запам'ятовування в залежності від різних порогових значень. Це візуальне представлення надає інформацію про те, як ефективність запам'ятовування моделі еволюціонує при зміні порогових значень. Аналізуючи цю криву, можна здійснювати обґрунтовані вибори порогових значень для досягнення оптимального балансу між вірно позитивними та вірно негативними результатами в конкретному контексті використання моделі класифікації.

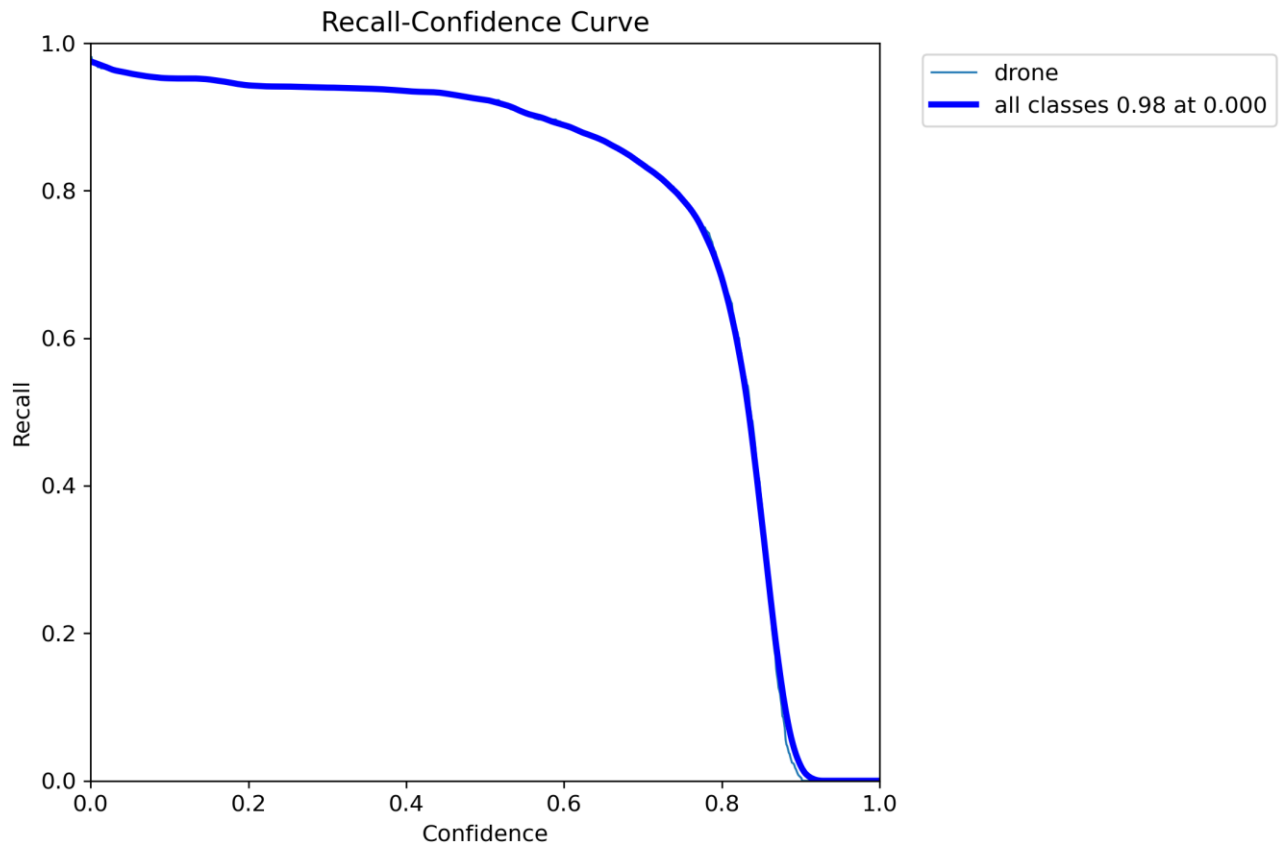


Рисунок 4.18 – Крива запам'ятовування

Матриця плутанини, що зображена на рисунку 4.19, є інструментом, який забезпечує детальний огляд результатів класифікації, представляючи кількість істинно позитивних, істинно негативних, хибних позитивних та помилкових негативних класифікацій для кожного класу. Ця матриця дозволяє визначити ефективність моделі в розпізнаванні конкретних класів та виявити можливі плутанини між ними, що допомагає вдосконалити модель та виправити її недоліки в процесі навчання та оптимізації.

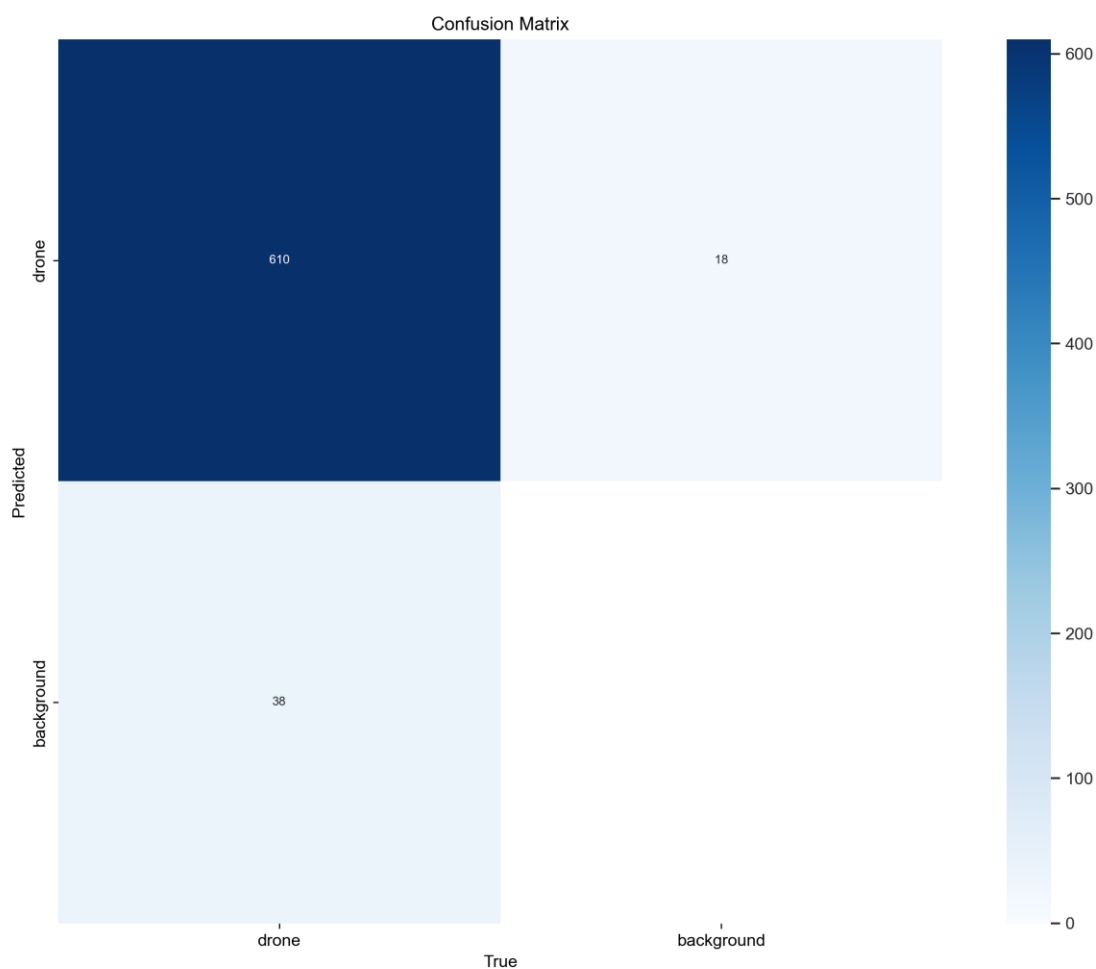


Рисунок 4.19 – Матриця плутанини

Нормалізована матриця плутанини, що зображена на рисунку 4.20, є візуалізацією, яка представляє собою нормалізовану версію звичайної матриці плутанини. У цьому форматі дані відображаються у відсотковому відношенні, що полегшує порівняння ефективності між класами, оскільки вони представлені в пропорційних значеннях, а не в необроблених числових показниках. Ця візуалізація дозволяє швидко та зрозуміло оцінити, як кожен клас веде себе в контексті загальної класифікації, сприяючи аналізу та вдосконаленню моделі.

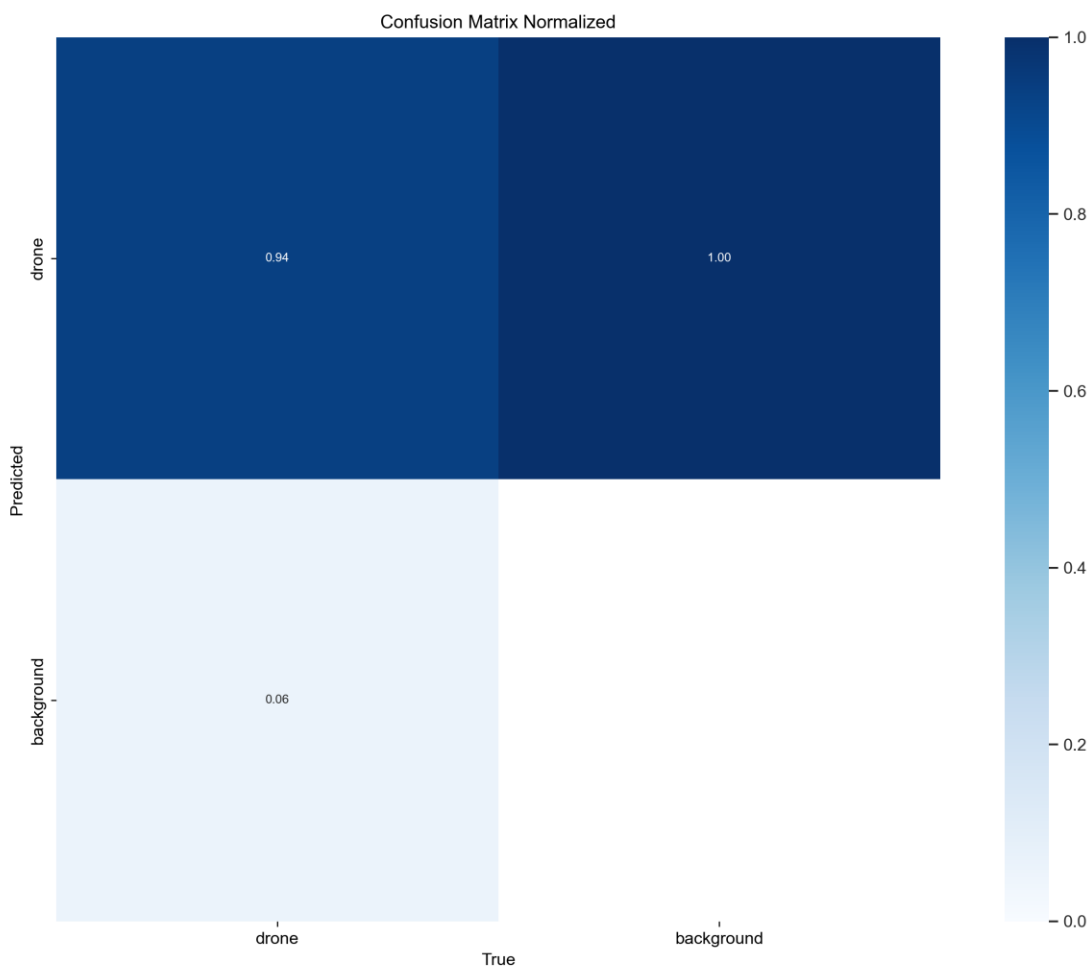


Рисунок 4.20 – Нормалізована матриця плутанини

Мітки пакетів перевірки, що зображені на рисунку 4.21, є зображеннями, які ілюструють основні мітки правдивості для окремих пакетів у наборі даних для перевірки. Ці зображення надають чітке уявлення про те, які об'єкти знаходяться на зображеннях та їх розташування в межах набору даних. Аналіз цих міток допомагає оцінити відповідність прогнозів моделі реальним об'єктам у валідаційному наборі та виявити можливі розбіжності або точності моделі в порівнянні з даними перевірки.

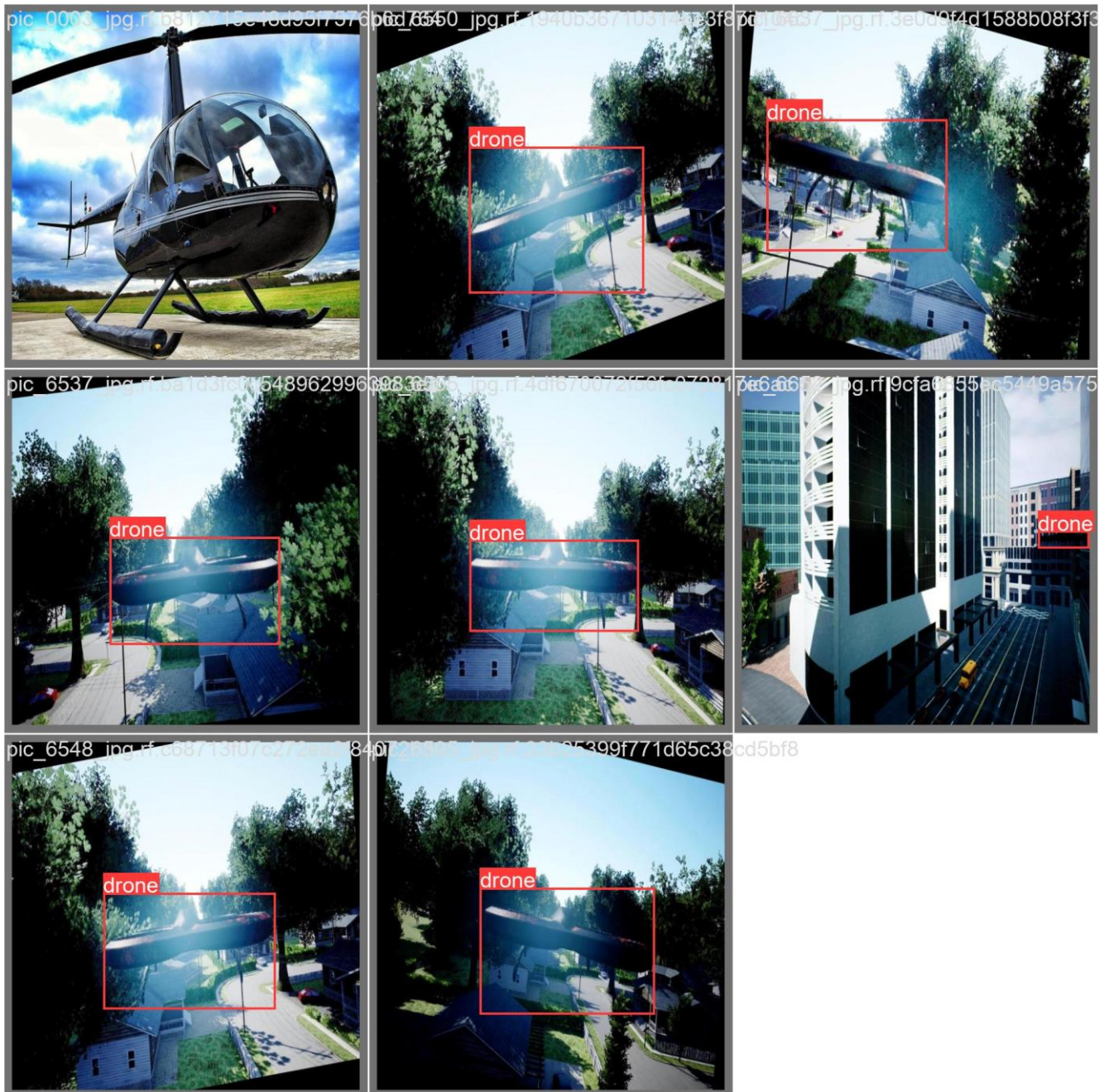


Рисунок 4.21 – Мітки пакетів перевірки

Передбачення пакетів перевірки зображені на рисунку 4.22. На відміну від зображень міток, ці візуальні елементи відображають прогнози, зроблені моделлю YOLOv8 для відповідних пакетів. Порівнюючи їх із зображеннями міток, можна легко оцінити, наскільки добре модель візуально виявляє та класифікує об'єкти.

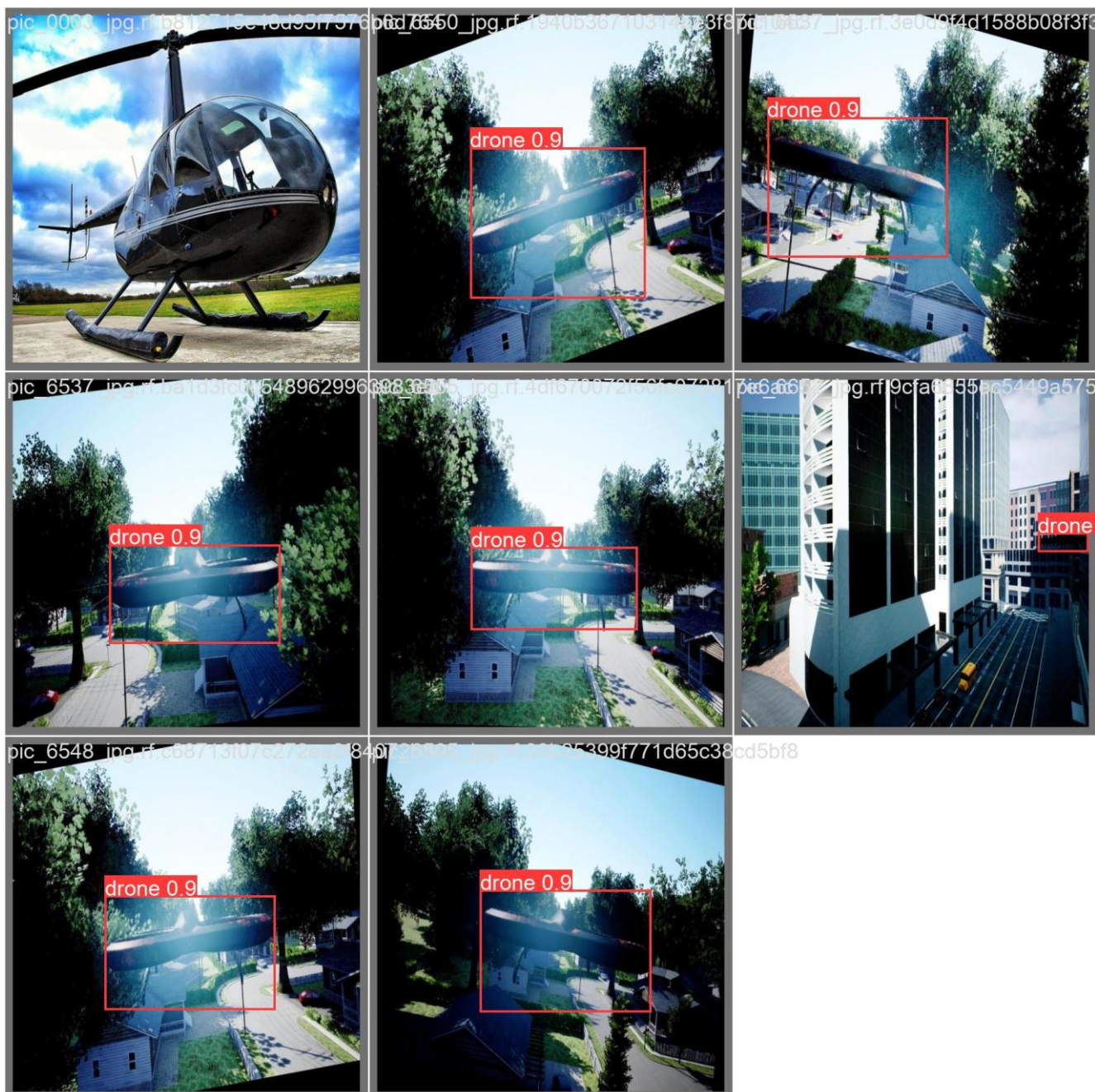


Рисунок 4.22 - Передбачення пакетів перевірки

Аналізуючи отримані результати, можна визначити, що показники моделі на валідаційних даних виявилися вражаюче високими. Можна стверджувати, що модель проявила себе дуже ефективно та швидко. Це особливо важливо у контексті виявлення дронів у реальному часі.

Термін "доволі непогані" в даному висловлюванні підкреслює, що результати перевищують очікувані та можуть бути оцінені як високі або задовільні. Модель продемонструвала високий рівень якості роботи, де кожна

класифікація та виявлення здійснюються якісно, без значних затримок чи втрат точності.

Особливий акцент робиться на швидкості роботи моделі, що важливо для виявлення дронів у реальному часі. Швидкість реакції моделі може визначати її придатність до застосування в ситуаціях, де час є критичним фактором, наприклад, в області безпеки або моніторингу. Такий висновок свідчить про те, що модель не лише надійно впоралася із завданням виявлення дронів, але й робить це дуже ефективно та оперативно.

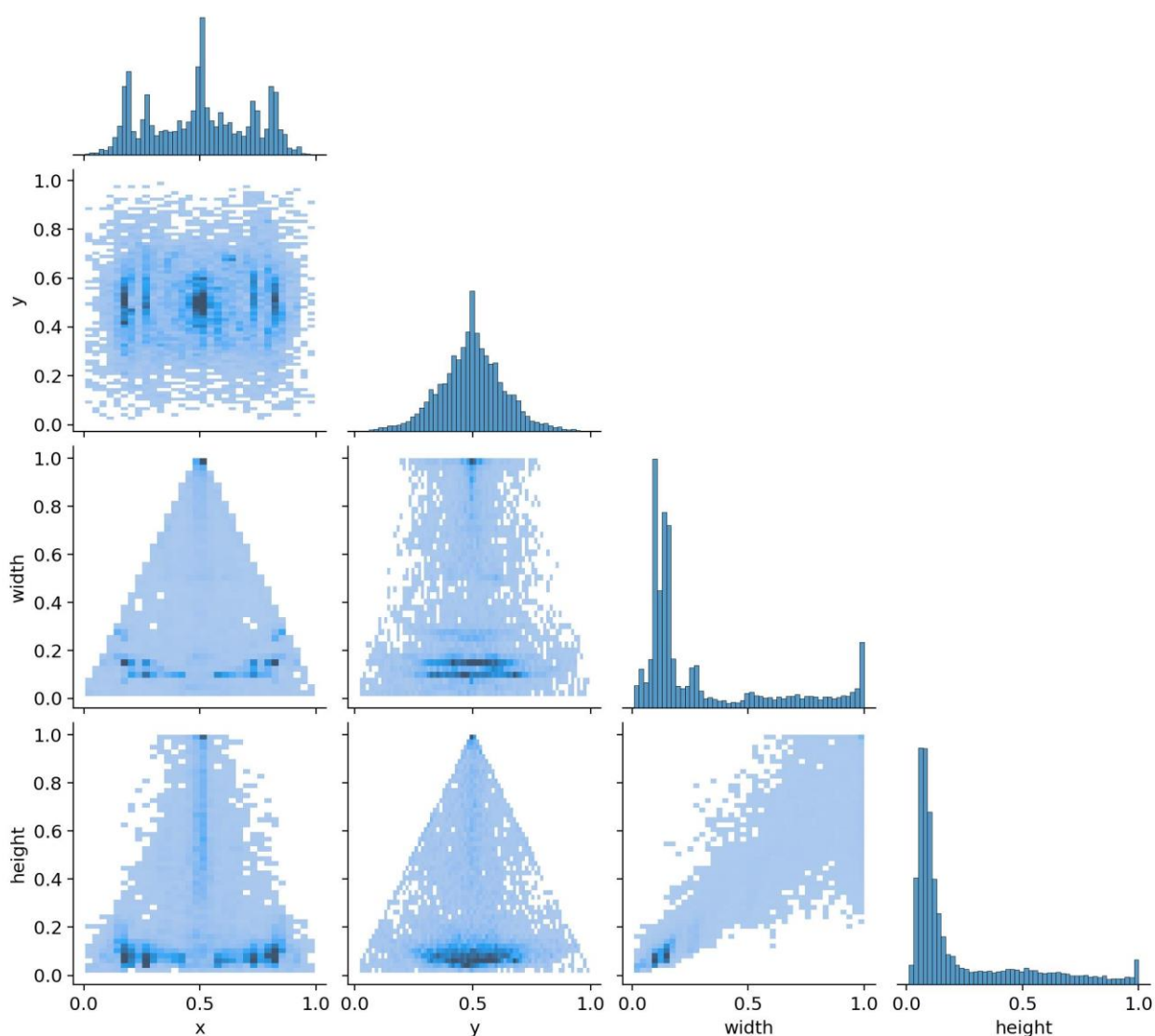


Рисунок 4.23 – Корелограма міток

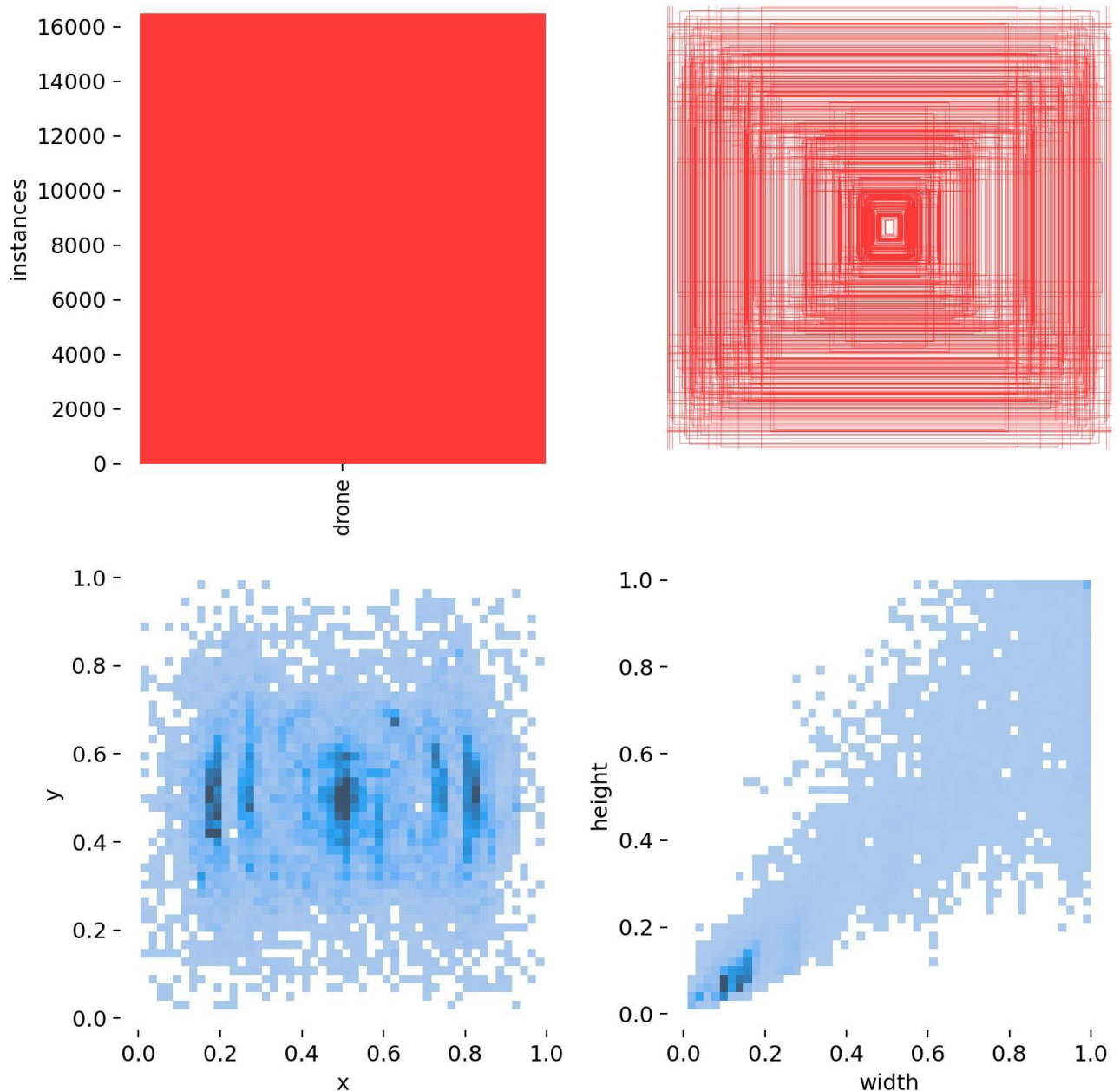


Рисунок 4.24 – Деталі розташування міток на зображеннях

На рисунках 4.23 та 4.24 можна знайти вичерпну інформацію щодо точного розташування міток, що вказують на дрони, на визначених зображеннях. Ці рисунки надають деталізоване представлення просторового розподілу та розташування дронів на зображеннях.

Аналізуючи рисунки, можна візуально визначити координати міток, які вказують на місцезнаходження дронів на кожному конкретному зображенні. Це дозволяє детально досліджувати просторові характеристики, такі як розташування

та взаємне розміщення дронів, а також їхній відносний розмір на кожному зображенні.

Розглядаючи цю детальну інформацію, надається можливість краще розуміти результати виявлення дронів, а також оцінити точність та надійність моделі. Дані рисунки виступають важливим інструментом для валідації та перевірки ефективності алгоритмів виявлення дронів на зображеннях, а їхній ретельний аналіз сприяє вдосконаленню та оптимізації алгоритмів обробки зображень.

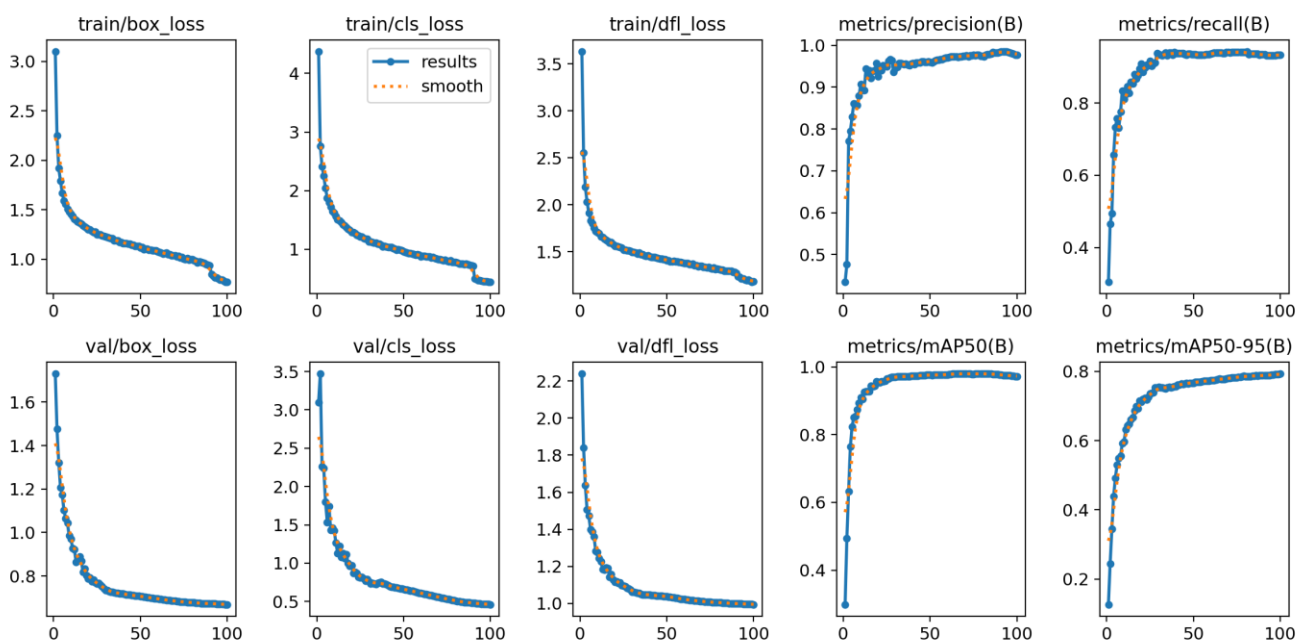


Рисунок 4.25 – Карта і графіки втрат після навчання

Із графіків втрат, що зображені на рисунку 4.25, можна здійснити детальний аналіз, що дозволяє зробити висновок про те, що процес навчання нашої моделі протікає плавно і стабільно. Ця інформація виявляється ключовою, оскільки вказує на ефективність та надійність самого процесу навчання.

Графіки втрат представляють собою динаміку зменшення величини втрат (loss) протягом кожної епохи або ітерації навчання. Стабільний та плавний спад графіка втрат свідчить про те, що модель ефективно пристосовується до поданих даних та вдосконалюється з кожним кроком навчання.

Важливо зауважити, що плавний процес навчання свідчить про відсутність великих коливань чи різких змін у втратах, що може вказувати на стабільність моделі. Це важливий аспект, оскільки демонструє, що модель не тільки ефективно вчиться, але і реагує на дані без значних варіацій чи впливів, що можуть виникнути в процесі навчання.

Такий позитивний висновок графіків втрат підкреслює надійність та оптимальність обраного методу навчання, а також свідчить про успішне досягнення моделлю оптимальних вагових коефіцієнтів для ефективного виявлення та класифікації об'єктів.

4.5 Розробка клієнтської частини

Після того, як успішно створили нейромережу для ефективного виявлення дронів і вивчили характеристики системи, наступним етапом є розробка спеціального веб-додатку, який забезпечить клієнтам комфортну взаємодію. Цей вбудований у браузер додаток не тільки вирішує завдання зручності для користувача, але й надає можливість тестувати моделі за допомогою веб-камер і локальних відео.

Веб-додатки будуть розроблені з урахуванням основних потреб користувачів, орієнтуючись на зручність та інтуїтивність. Планується вбудувати в програму функції, які спростять взаємодію з нейронними моделями, як-от завантаження зображень для аналізу.

Крім того, було надано особливу увагу можливості тестування моделі. Користувачі зможуть проводити експерименти з виявлення дронів за допомогою веб-камери або завантажуючи локальні відеофайли. Це дасть можливість перевірити валідність і точність моделі в різних умовах.

Такий підхід до розробки веб-додатків забезпечить повне використання та перевірку нейронних моделей у реальних умовах, що зробить взаємодію із системою ефективною та зручною для користувача.

Для цього було використано мову JavaScript та фреймворк React для створення додатку. Також для реалізації було використано такі додаткові бібліотеки які зображені на рисунку 4.26:

– Material UI (MUI) — це бібліотека компонентів React з відкритим кодом. Він містить повну колекцію готових компонентів, які можна витягнути та підготувати до виробничого використання [32]. Інтерфейс Material UI має гарний дизайн і містить набір параметрів налаштування, які спрощують реалізацію власної системи дизайну на основі наших компонентів.

– Craco розшифровується як Create React App Configuration Override і дозволяє отримати всі переваги Create React App без використання функції «eject» [33]. Це потрібно для забезпечення коректної роботи модуля OpenCV.

– OpenCV (Бібліотека комп'ютерного зору з відкритим вихідним кодом) — це бібліотека з відкритим кодом, яка містить сотні алгоритмів комп'ютерного зору [34]. Ця бібліотека потрібна головним чином, щоб підготувати кадри з відео для нейронної обробки, щоб мати можливість обробляти їх у браузері.

– ONNX Runtime Web — це бібліотека Javascript для запуску моделей ONNX у браузері та Node.js [36]. ONNX Runtime Web використовує технології WebAssembly і WebGL, щоб забезпечити оптимізоване середовище виконання моделі ONNX для центральних і графічних процесорів.

– copy-webpack-plugin — це плагін, який копіює наявні окремі файли або цілі каталоги до каталогу збірки. В цьому проєкті використаємо плагін копіювання, щоб скопіювати *.wasm у вихідну папку.

```

"dependencies": {
  "@emotion/react": "^11.11.1",
  "@emotion/styled": "^11.11.0",
  "@mui/icons-material": "^5.15.0",
  "@mui/material": "^5.15.0",
  "@techstark/opencv-js": "^4.8.0-release.10",
  "@testing-library/jest-dom": "^5.17.0",
  "@testing-library/react": "^13.4.0",
  "@testing-library/user-event": "^13.5.0",
  "onnxruntime-web": "^1.16.3",
  "react": "^18.2.0",
  "react-dom": "^18.2.0",
  "react-scripts": "5.0.1",
  "web-vitals": "^2.1.4"
},
"devDependencies": {
  "@craco/craco": "^7.1.0",
  "copy-webpack-plugin": "^11.0.0"
},

```

Рисунок 4.26 – package.json

Щоб забезпечити сумісність із веб-браузерами, модель має пройти процес перетворення в певний формат під назвою ONNX. Відкритий обмін нейронними мережами (ONNX) є широко прийнятим стандартом для представлення моделей машинного навчання [36]. Однією з ключових переваг ONNX є його здатність сприяти взаємодії між різними фреймворками штучного інтелекту з відкритим кодом, підвищуючи таким чином гнучкість впровадження штучного інтелекту. Використовуючи спільний формат зберігання для нейронних мереж, ми можемо заощадити час, усунути залежності від конкретних фреймворків і дослідити різні варіанти інфраструктури. Фрагмент коду на рисунку 4.27 ілюструє перетворення нашої моделі на ONNX.

```

1 from ultralytics import YOLO
2 model = YOLO("./runs/detect/train2/weights/best.pt")
3 model.export(format="onnx")

```

Рисунок 4.27 – export_to_ONNX.py

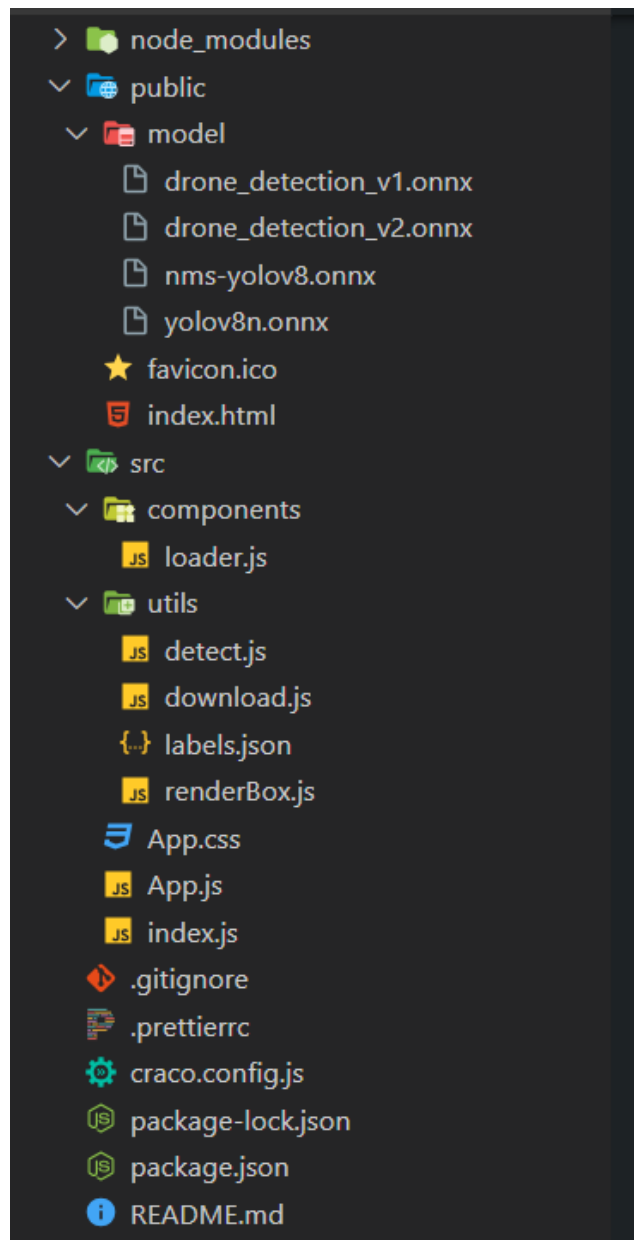


Рисунок 4.28 – Структура файлів додатку

Після успішного перетворення нашої моделі у формат ONNX, було розпочато процес інтеграції у порожній проект, який включав в себе додаткові бібліотеки, що були попередньо описані. Створений порожній проект служив основою для подальшої роботи з нашою моделлю та її інтеграцією у визначені вимоги середовища.

Далі, з використанням цього проекту, було розроблено файли з конкретною архітектурою, яка була проілюстрована на рисунку 4.28. Кожен створений файл

мав відповідальність за конкретну частину функціоналу, що був необхідний для оптимальної роботи моделі у новому середовищі.

Потім, надано опис та пояснення для кожного створеного файлу чи папки в рамках нашого проекту. Це включало визначення завдань та функціональності кожного компонента, який сприяв зрозумілому розподілу відповідальностей та легшому управлінню кодовою базою.

Цей процес інтеграції та створення файлів з визначеною архітектурою є важливим етапом для забезпечення правильної та ефективної роботи нашої моделі в новому середовищі, а опис кожного елемента сприяє зрозумінню структури проекту для подальших розширень та обслуговування.

4.5.1 Опис структури додатку

В папці `node_modules` містяться залежності проекту. Він автоматично створився при ініціалізації проекту і обновляється при встановленні нових бібліотек.

Папка `public` використовується для розміщення статичних ресурсів, таких як HTML-файли (`index.html`), зображення (`favicon.ico`), які не потребують обробки React. Також там знаходяться папка `model` в якій знаходяться наші створені моделі які будуть використовуватись для виявлення дронів в небі на відео.

А в папка `src` є основною папкою для розміщення вихідного коду нашого React-додатку. В цій папці зазвичай знаходиться весь код, який пишеться для розробки. В цій папці також знаходяться такі підпапки та файли:

- `components`;
- `utils`;
- `App.css`;
- `App.js`;
- `index.js`.

Папка `components` відіграє роль зберігання наших компонентів для додатку. У нашому конкретному випадку в цій папці буде розташований лише один компонент, а саме кастомний компонент завантаження, який має назву `loader.js`.

В `utils` створені файли які записані функції для обробки, загрузки зображень та виведення результату на екран:

- `detect.js` – у цьому файлі розміщені дві ключові функції. Перша функція відповідає за виявлення об'єктів на зображенні, що, ймовірно, використовується для реалізації функціоналу розпізнавання об'єктів у додатку. Друга функція відповідає за попередню обробку зображення, підготовляючи його для подальшого аналізу та виявлення;

- `downloads.js` – функція, яка відповідає за коректне завантаження моделі, яка, використовується в додатку для подальшої обробки та розпізнавання об'єктів.

- `labels.json` – тут записано список позначень для моделі. В нашому випадку там буде лише один елемент – `drone`;

- `renderBox.js` – функція яка відповідає за створення візуальної рамки. Ця функція може використовуватися для візуального виділення об'єктів на зображенні, надаючи інформацію про їхнє місцезнаходження.

`Index.js` в `React` є одним з ключових файлів, який використовується для початкового завантаження та рендерингу додатку. У цьому файлі визначається головний вузол (кореневий елемент) додатку, куди буде вставлятися весь `React`-компонент, що відповідає за відображення інтерфейсу користувача. Також може відбуватися ініціалізація додаткових сервісів, підключення глобальних стилів чи налаштувань.

Файл `App.css` містить класи, які визначають власні стилі для відображення та оформлення компонентів та елементів, які використовуються у веб-додатку. Ці стилі можуть включати параметри, такі як кольори, розміри, відступи, тіні та інші стилізаційні атрибути, надаючи додатку певний зовнішній вигляд та користувацький інтерфейс.

`App.js` – це основний компонент для нашого додатку. У цьому файлі визначено основну структуру та вигляд додатку. Тут записано основний макет та

викликали функції, що були раніше визначені, такі як ініціалізація `orepvc.js` та інші. `App.js` може бути визначено як вузол, в якому об'єднуються всі компоненти та функціонал додатку, формуючи центральну частину логіки та інтерфейсу веб-додатку.

Файл `.gitignore` використовується для визначення списку файлів та папок, які Git повинен ігнорувати при відстеженні змін у репозиторії. Цей файл допомагає управляти тим, які файли та папки повинні бути включені або виключені під час використання системи контролю версій.

`Crao.config.js` використовується для налаштування і кастомізації параметрів в середовищі CRA (Create React App).

`Package.json` та `package-lock.json` - це два файли, які визначають та управляють залежностями проекту та їх версіями.

QR-код з посиланням на репозиторій з кодом зображено на рисунку 4.29.



Рисунок 4.29 – QR-код із посиланням на репозиторій з кодом для веб-додатку

4.5.2 Тестування роботи додатку

У даному підрозділі розглядатиметься функціонал, що реалізований у нашому веб-додатку, а також проведення тестування роботи моделі безпосередньо у браузері. Під час запуску додатку у веб-браузері користувача вітає екран загрузки, де відображається інформація про готовність додатку до

роботи, виражена у відсотках завершення ініціалізації. Приклад зображений на рисунку 4.30. Цей етап є важливим моментом, оскільки він вказує на те, що веб-додаток готовий до використання та правильного відображення результатів роботи моделі.

Після завантаження програмного додатку вступаємо в контекст стративного екрану, приклад зображений на рисунку 4.31, де надається можливість вибрати джерело відео для подальшого відтворення. На цій сторінці користувач може обрати одне з двох джерел: відеопотік безпосередньо з камери, яка підключена до комп'ютера, або завантажити відео з локального середовища зберігання.

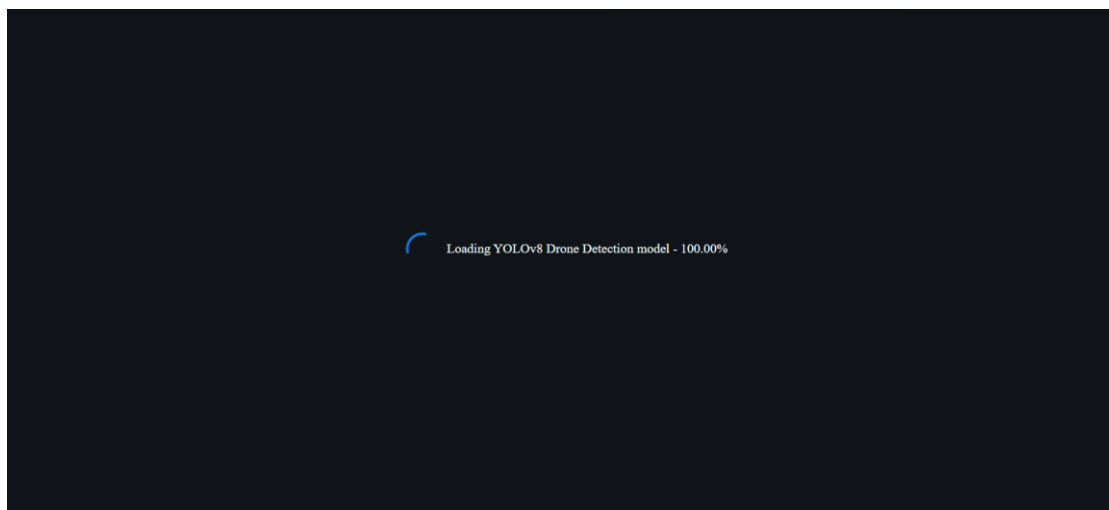


Рисунок 4.30 – Екран завантаження

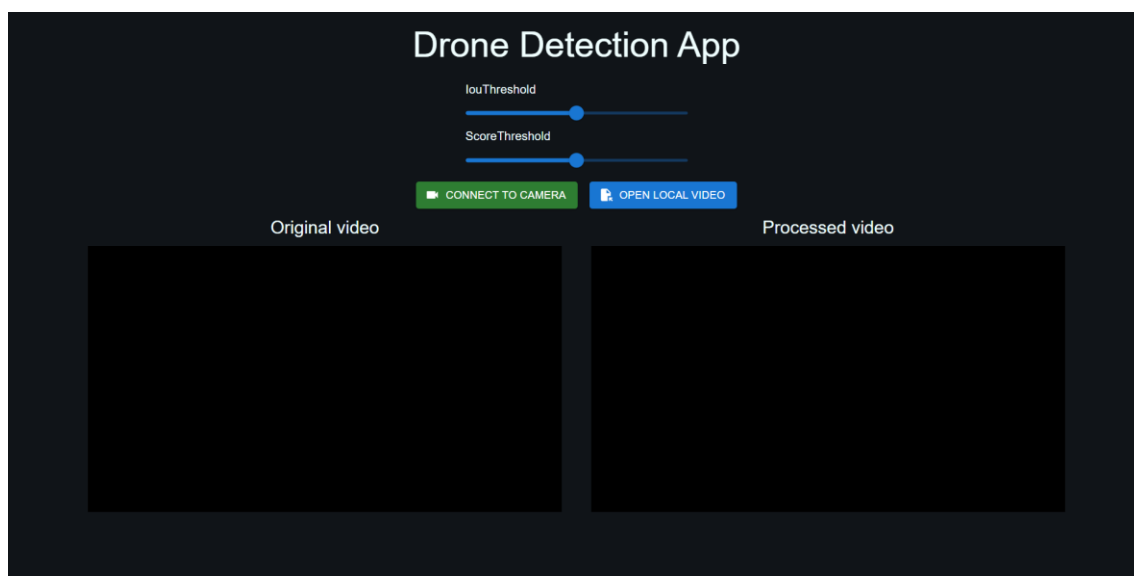


Рисунок 4.31 – Стартовий екран

При цьому створено два важливих інструменти управління параметрами обробки відео:

– `IoUThreshold` – цей параметр дозволяє визначити ступінь перекриття між областями, що передбачаються нейронною мережею, та фактичними обмежувальними рамками на відео. Змінюючи це значення, користувач впливає на точність прогнозування та визначення меж об'єктів;

– `ScoreThreshold` – цей параметр дозволяє встановити поріг, який визначає, який високий має бути рівень впевненості, щоб область була відзначена рамкою на відео. Змінюючи це значення, користувач має змогу контролювати, які області вважатимуться значущими та будуть відображатися на відео.

Крім того, на інтерфейсі веб-додатку присутні блоки, в межах яких буде відображатися оригінальне відео, а також відео після обробки, супроводжуване рамками, що вказують на місцезнаходження дрона. Ці блоки створюють можливість користувачеві в режимі реального часу спостерігати за вхідним та обробленим відеопотоком, визначаючи точне місцезнаходження об'єкта на зображенні.

Після ініціалізації додатку та обрання відео, користувач отримує можливість спостерігати за роботою нашої моделі, яка відзначає місцезнаходження дрону на відеозаписі. Цей процес ілюструється на рисунках 4.32 та 4.33, де демонструється ефективність та точність виявлення дрона за допомогою нашої моделі.

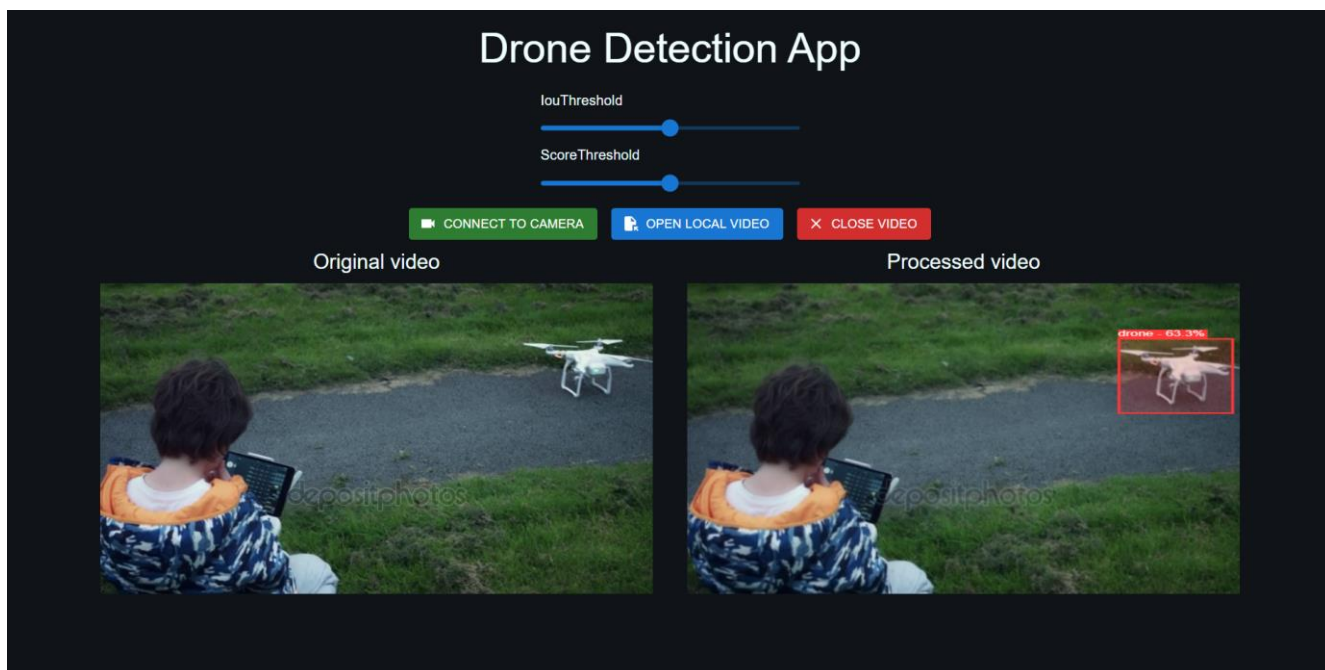


Рисунок 4.32 – Тестування локального відео №1

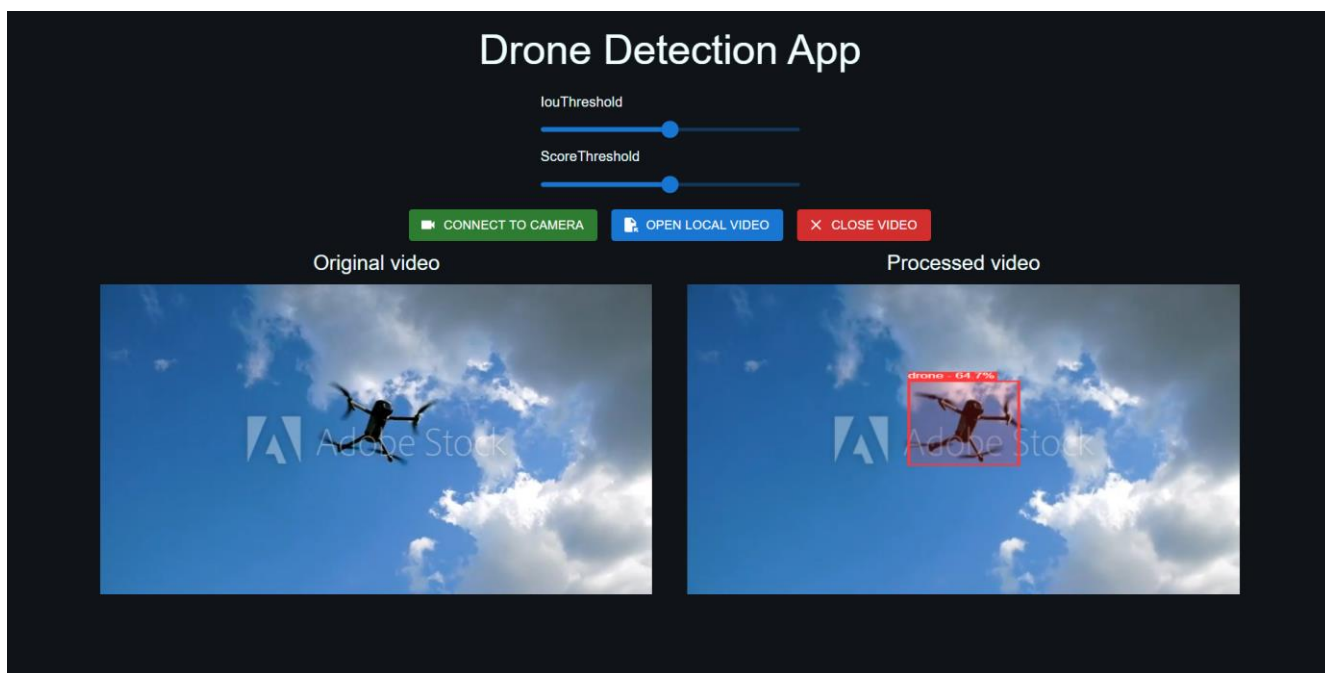


Рисунок 4.33 - Тестування локального відео №2

Висновки до розділу 4

У четвертому розділі ми визначили середовища, в яких відбувалась розробка нашої моделі та веб-додатку, і детально описали їхні переваги, пояснивши вибір саме цих середовищ.

Зазначені середовища були обрані з урахуванням їхньої спрямованості на ефективний розвиток та тестування нейронних мереж, а також для забезпечення оптимальних умов для веб-додатку. Розглянуто властивості кожного середовища, їхні можливості та те, як вони сприяли успішній реалізації нашого проекту.

В подальшому нами було детально розглянуто етап за етапом процес підготовки зображень для створення датасету, який передбачав їх систематичну обробку та підготовку для створення датасету з метою подальшого використання в навчанні нашої моделі.

Після цього ми детально розглянули етапи навчання нашої моделі, обираючи за основу модель YOLOv8. Розповіли про кроки, які включали в себе налаштування параметрів для оптимального навчання та вибір відповідних алгоритмів для досягнення високої точності та ефективності у виявленні дронів на зображеннях.

Після завершення процесу навчання моделі ми перейшли до розробки веб-додатку, що надавав нам можливість тестувати нашу модель в режимі реального часу. У межах цього додатку можна експериментувати з різними параметрами та вибирати різні джерела відео, щоб систематично перевіряти ефективність та точність моделі при різних умовах використання.

Після проведення тестування моделі в веб-додатку можна стверджувати, що модель ефективно працює при виявленні дронів великих або середніх розмірів. Проте виникають певні труднощі при розпізнаванні дронів малих та дрібних розмірів, оскільки моделі важко обробляти такі зображення та правильно ідентифікувати їх як дрони. Тому вставивши відео, наприклад, з гвинтокрилом малих розмірів модель може зрозуміти, що це дрон.. Для покращення точності моделі можна розглянути можливість збору більше зображень з дронами малих розмірів або налаштування параметрів навчання моделі, але це також може вплинути на її швидкодію.

5 СТАРТАП-ПРОЄКТ

В цьому розділі буде описано майбутній стартап-проект [36].

5.1 Опис ідеї проекту

В таблиці 5.1 було проаналізовано ідею проекту, напрямки застосування та потенційні вигоди для користувача.

Таблиця 5.1 – Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигода для користувача
Створення автоматизованої системи для виявлення БПЛА в небі, використовуючи штучний інтелект.	Військова сфера.	За допомогою системи користувачі можуть її використовувати встановлюючи її на власні БПЛА, для виявлення ворожих з їхнім майбутньою ліквідацією та ін.
	Для домашнього користування	Користувачі можуть її використовувати для встановлення їх вдома для власної безпеки та конфіденційності.

В наступній таблиці 5.2 було виконано порівняльний аналіз за технікоекономічними характеристиками з іншими системами.

Таблиця 5.2 – Визначення сильних, нейтральних та слабких характеристик ідеї проекту

№	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів				W	N	S
		Мій проєкт	AARTOS	Aerialarmor	AirSentinel			
1	Вартість використання сервісу	+	-	-	-			+
2	Використання штучного інтелекту	+	+	+	+		+	
3	Радіочастотне виявлення	-	+	+	-	+		
4	Оптичне виявлення	+	+	+	-			+
5	Виявлення за допомогою радара	-	+	-	+	+		
6	Теплове виявлення	-	+	+	+	+		
7	Використання додаткових пристроїв	+	-	-	-			+

З цієї таблиці можна зробити, що наш сервіс є бюджетний варіант, який має використовуватися більш для захисту власного будинку від дронів, і йому потрібно лише доступ до камер нагляду в будинку. В інших сервісах є можливість використовувати інші методи виявлення (радар, тепловий, радіочастотний та

інші), і вони краще підійдуть для захисту великих підприємств, або інших важливо інших будівель.

5.2 Технологічний аудит ідеї проєкту

Далі було виконано аналіз можливості реалізації автоматизованої системи для виявлення БПЛА в небі в таблиці 5.3.

Таблиця 5.3 - Технологічна здійсненність ідеї проєкту

№	Ідея проєкту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Мова програмування	Python	Наявна	Доступна
		C++	Наявна	Доступна
Обрана технологія реалізації проєкту: Python				
2	Нейромережний підхід	YOLO	Наявна	Доступна
		R-CNN	Наявна	Доступна
		Retina-Net	Наявна	Доступна
		SSD	Наявна	Доступна
Обрана технологія реалізації проєкту: YOLO				
3	Модель для виявлення дрону	Власна реалізація	Власна реалізація	Власна реалізація
Обрана технологія реалізації проєкту: Власна реалізація				
4	Створення веб-інтерфейсу	React	Наявна	Доступна
		Vue	Наявна	Доступна
		Angular	Наявна	Доступна
Обрана технологія реалізації проєкту: React				

Виходячи з цієї таблиці можна зробити висновок, що ідея проєкту є технологічно здійсненою, всі технології для реалізації є наявні та вони є легкодоступними.

5.3 Аналіз ринкових можливостей запуску стартап-проєкту

Далі потрібно визначити основні загрози та можливості, що можуть бути при реалізації проєкту. Це дасть правильне прогнозування майбутніх напрямів розвитку проєкту.

В наступній таблиці 5.4 було проведено аналіз попиту на проєкт.

Таблиця 5.4 – Попередня характеристика потенційного ринку стартап-проєкту

№	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	7
2	Загальний обсяг продаж, грн/ум.од	Потрібно провести додаткові дослідження
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Велика конкуренція, високі витрати на дослідження та розробку
5	Специфічні вимоги до стандартизації та сертифікації	Вимоги до стандартів безпеки та сумісності з іншими системами.
6	Середня норма рентабельності в галузі (або по ринку), %	Від 20% до 35%, залежно від сегменту та ефективності продукту

Загалом на ринку є достатня кількість великих гравців, які разом генерують достатню велику суму доходу через системи рекомендацій. Враховуючи цей дохід і рівень прибутковості близько 30%, можна зробити висновок, що цей ринок

досить привабливий для інвестування, що підтверджує постійне створення нових продуктів у цій галузі. Однак, оскільки йдеться про особистий захист та безпеку важливо враховувати стандарти та сертифікаційні вимоги для забезпечення відповідності та довіри з боку клієнтів.

Далі потрібно зробити огляд потенційних клієнтів у таблиці 5.5.

Таблиця 5.5 – Характеристика потенційних клієнтів

№	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Безпека конфіденційності та власної безпеки	Зацікавлені особи що бажають захистити свій будинок від дронів.	Особи даної групи використовують сервіс для власної безпеки	Виявлення має працювати бисрто та точно, щоб не було ніяких затримок та помилкових виявлень

Тут можна зробити висновок що основними користувачами автоматизованої системи для виявлення БПЛА є особи які зацікавленні в безпеці власного будинку від БПЛА, які хочуть щоб їх сервіс працював бисрто і точно.

Після того, як було визначено потенційних клієнтів, необхідно провести аналіз ринкового середовища, який включає в себе фактори ризиків, що зображено в таблиці 5.6, та можливостей в таблиці 5.7.

Таблиця 5.6 – Фактори загроз

№	Фактор	Зміст загрози	Можлива реакція компанії
1	Конкуренція	Поява конкурентів або імітація продукту іншими компаніями.	Постійне вдосконалення продукту, інвестування в маркетинг, швидка реакція на зміни у конкурентному середовищі.
2	Технологічні ризики	Технічні проблеми, витоки даних, невдачі в системі виявлення БПЛА	Розробка та вдосконалення технічних рішень, ретельна тестування перед випуском продукту, використання заходів кібербезпеки..
3	Зміни в законодавстві	Зміни в правових нормах, які можуть вплинути на функціонування стартапу	Постійне слідкування за змінами в законодавстві, співпраця з юридичними експертами, адаптація бізнес-процесів до нових вимог

Таблиця 5.7 – Фактори можливостей

№	Фактор	Зміст можливості	Можлива реакція компанії
1	Ріст ринку БПЛА	Збільшення використання безпілотних літальних апаратів в різних галузях, що призводить до зростання попиту на системи виявлення БПЛА	Збільшення виробництва, розширення ринків збуту, активна маркетингова кампанія для залучення нових клієнтів.
2	Технологічні	Введення нових технологій	Інвестування в

№	Фактор	Зміст можливості	Можлива реакція компанії
	інновації	та інновацій у систему виявлення БПЛА, які покращують ефективність та надійність продукту.	дослідження і розробку, швидке впровадження нових технологій, патентування унікальних рішень.
3	Потреба у безпеці та обороні	Зростання попиту на безпілотні системи для виявлення у військовій, оборонній та безпеці великих об'єктів.	Розширення спеціалізованих функцій для задоволення потреб безпеки та оборони, залучення контрактів в цих сегментах.

Далі було описано загальні риси конкуренції на ринку і представлено їх в таблиці 5.8.

Таблиця 5.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
Тип конкуренції: Олігополія	На ринку існує кілька великих гравців, які контролюють більшість частки ринку для систем виявлення БПЛА.	Конкуренція обмежена декількома сильними гравцями. Компанії можуть взаємодіяти та конкурувати за впровадження нових технологій та інновацій.
Рівень конкурентної боротьби: Національний	Змагання відбувається на національному рівні, з	Потребує широкого масштабу маркетингу,

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
	великим обсягом ринку та конкурентами з різних регіонів країни.	національних стратегій поширення та взаємодії з різними галузями.
Галузева ознака: Внутрішньогалузева	Конкуренція зосереджена серед компаній, що спеціалізуються в технологіях виявлення БПЛА в небі.	Заохочує спеціалізацію та розвиток власних технічних рішень для конкретної галузі.
Конкуренція за видами товарів: Товарно-родова	Системи для виявлення БПЛА різних виробників виконують схожі функції та мають однаковий базовий споживчий попит.	Стимулює розробку унікальних функцій та покращення якості для вирішення від конкурентів.
Характер конкурентних переваг: Нецінова	Конкуренція базується на якості, інноваціях, підтримці клієнтів та технічних рішеннях.	Заохочує до вдосконалення технічних характеристик та обслуговування клієнтів, а не тільки зниження цін.
Інтенсивність конкуренції: Марочна	Компанії активно розвивають свої бренди, використовуючи маркетингові стратегії та рекламні кампанії.	Важливість створення та утримання сильного бренду, що привертає увагу та довіру клієнтів.

Наступним кроком є аналіз галузевої конкуренції за моделлю 5 сил М. Портера. Аналіз вивчатиме основні компоненти: прямі конкуренти, потенційні конкуренти, постачальники, клієнти та загрози заміників у галузі. На основі отриманих даних будуть зроблені висновки про інтенсивність конкуренції, наявності можливості виходу на ринок, як постачальники та клієнти впливають на ринок, і з якими обмеженнями доведеться зіткнутися через наявність заміників. Результати наведені в таблиці. 5.9.

Таблиця 5.9 – Аналіз конкуренції в галузі за М. Портером

	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
Складові аналіз	AARTOS, Aerialarmor, AirSentinel	Ви можете створювати програмні продукти, які використовують в відкриті технології.	-	Прибутки клієнта	Краще співвідношення ціни та якості, лояльність користувачів до певного товару, зменшити витрати на обслуговування виробу.
Висновки	Інтенсивна конкуренція	Можливі інші конкуренти, але можливість вийти на ринок	-	Різноманітність цін має сприяти відвідуванню	Існує ризик того, що альтернативи можуть

Складові аналіз	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
		є цілком реальною.		клієнтів з різним економічним рівнем.	бути більш привабливи ми для деяких клієнтів.

Після аналізу було встановлено, що в галузі існує сильна конкуренція та товари-замінники.

Далі розглянуто фактори конкурентноспроможності у таблиці 5.10

Таблиця 5.10 – Обґрунтування факторів конкурентноспроможності

№	Фактор конкурентноспроможності	Обґрунтування
1	Технологічна інноваційність	Впровадження передових технологій у системи виявлення БПЛА може забезпечити перевагу у точності, швидкості та функціональності порівняно з конкурентами. Інновації можуть включати в себе вдосконалені алгоритми виявлення, використання штучного інтелекту та машинного навчання.
2	Якість та ефективність	Висока якість виявлення та ефективність роботи системи створюють позитивний враження від продукту. Клієнти шукають надійні та ефективні рішення для виявлення безпілотних літальних апаратів.

3	Цінова конкурентоспроможність	В умовах конкуренції цінова політика може визначати привабливість продукту. Якщо конкуренти пропонують схожі технології, але за більш низьку ціну, це може вплинути на вибір клієнтів.
4	Бренд та репутація	Створення сильного бренду та доброї репутації може впливати на вибір клієнтів. Посилення бренду може бути досягнуто через успішні проекти, рекомендації клієнтів та активний маркетинг.

Після визначення факторів конкурентоспроможності варто проаналізувати сильні та слабкі сторони проекту, визначивши рейтинги продуктів конкурентів та порівнявши їх із власними проектами, результати наведено в таблиці. 5.11.

Таблиця 5.11 – Порівняльний аналіз сильних та слабких сторін проекту

№	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з розроблюваною системою						
			-3	-2	-1	0	+1	+2	+3
1	Технологічна інноваційність	14				+			
2	Якість та ефективність	12			+				
3	Цінова конкурентоспроможність	19							+
4	Бренд та репутація	11		+					

Далі на основі вже існуючої інформації треба зробити SWOT-аналіз. Тому в наступній таблиці 5.12 буде це описано.

Таблиця 5.12 – SWOT – аналіз стартап-проекту

<p>Сильні сторони:</p> <ul style="list-style-type: none"> - легкість використання - ціноутворення 	<p>Слабкі сторони:</p> <ul style="list-style-type: none"> - можлива неточність результатів - відсутня рекламна компанія
<p>Можливості:</p> <ul style="list-style-type: none"> - розвиток функціоналу та технологій; - зростання попиту з часом. 	<p>Загрози:</p> <ul style="list-style-type: none"> - велика конкуренція; - відсутність великої кількості споживачів на початку

Також варто визначити можливі альтернативи ринкової реалізації пускового проекту, які могли б бути використані для залучення інвестицій чи клієнтів. Можливі методи реалізації наведені в таблиці. 5.13.

Таблиця 5.13 – Альтернативи ринкового впровадження стартап-проекту

№	Альтернатива ринкової поведінки	Ймовірність отримання ресурсів	Термін реалізації
1	Швидке ринкове впровадження з розробкою основних функцій	Середня	6 місяців
2	Презентація проекту на конференції	Висока	3 місяця
3	Планове впровадження в ринок з додаванням додаткового функціоналу	Низька	1 рік
4	Впровадження рекламної компанії	Низька	5 місяців

Найкращий варіант то це презентація нашого стартап-проекту на якійсь тематичній конференції. Це допоможе на початку не використовувати багато власних коштів, і якщо презентація пройде успішно і зацікавить інвесторів за допомогою яких можна заручитись їхніми коштами для подальшого розвитку.

5.4 Розроблення ринкової стратегії проекту

В цьому розділі було визначено стратегії для захоплення ринку. У таблиці 5.13 наведено альтернативні версії ринкового впровадження стартап-проекту.

Таблиця 5.14 – Вибір цільових груп потенційних споживачів

№	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтований попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Сучасні споживачі	Середня	Середня	Середня	Середня
Які цільові групи обрано: сучасні споживачі.					

Далі в таблиці 5.15, сформовано базову стратегію розвитку.

Таблиця 5.15. – Базова стратегія розвитку стартап-проекту

№	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Розробити бета-версію продукту та розгорнути її для обмеженої кількості користувачів	Презентувати готового продукту на конференціях	Залучайте вибрану групу потенційних користувачів за допомогою мінімальних налаштувань	Стратегія спеціалізації

Тому стратегія розвитку проекту була обрана на основі стратегії спеціалізації, а тестова версія проекту була розгорнута для обмеженої кількості

користувачів, щоб можна було детально вивчити фактичні потреби реальних користувачів.

Визначення конкурентних стратегій поведінки наведено в таблиці. 5.16

Таблиця 5.16. – Визначення базової стратегії конкурентної поведінки.

№	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
1	Проект не є «першопрохідцем»	Компанія буде шукати нових споживачів та намагатися забирати існуючих у конкурентів	Будуть копіюватись деякі характеристики	Стратегія зайняття конкурентної ніші

Далі потрібно визначити стратегію позиціонування, яка визначатиме набір атрибутів, на основі яких користувачі зможуть класифікувати продукти. Стратегія позиціонування наведена в таблиці 5.17.

Таблиця 5.17 – Визначення стратегії позиціонування

№	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1	Зручність використання, Безкоштовна можливість використання базового продукту, наявність описаного функціоналу	Стратегія спеціалізації	Можливість використання функціоналу різними цільовими групами споживачів.	Зручність використання, функціональність, якість

5.5 Розроблення маркетингової програми стартап-проекту

У наступній таблиці 5.18 буде наведено ключові переваги концепції потенційного товару. А у таблиці 5.19 - опис трьох рівнів моделі товару.

Таблиця 5.18 – Ключові переваги концепцій потенційного товару

№	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Потреба у легкості	Пропонує простий інтерфейс для виявлення БПЛА	Легкість

	використанні		
2	Потреба у надійному наданні сервісу	Можливість автоматично повідомляти про виявлення БПЛА втручання клієнта	Надійність

Таблиця 5.19 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові
I. Товар за задумом	Автоматизована система виявлення гвинтокрилих БПЛА в небі
II. Товар у реальному виконанні	Сервіс повинен працювати швидко та без помилок та коректно розпізнавати БПЛА в небі
	Марка: “DroneDetect”
III. Товар із підкріпленням	До продажу: основні функції сервісу
	Після продажу: всі функції сервісу, технічна підтримка
За рахунок чого потенційний товар буде захищено від копіювання: за рахунок захисту інтелектуальної власності.	

Наступним кроком є визначення меж встановлення ціни, виконаний опис наведено у таблиці 5.20.

Таблиця 5.20 – Визначення меж встановлення ціни

Рівень цін на товаризамінники	Рівень цін на товари аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
Відсутні	Від 50\$ в місяць	Середній та високий	Нижня: 10\$ Верхня: 100\$

Щоб далі зрозуміти реалізацію концепції, необхідно визначитися системою продажів, як показано в таблиці 5.21.

Таблиця 5.21 – Формування системи збуту

Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
Купівля товару на певний період (4 тижня)	Технічна підтримка	Клієнт купує товар у виробника	Комбінована система збуту

Також важливо визначити концепції маркетингових комунікацій, що наведено в таблиці 5.22.

Таблиця 5.22 – Концепція маркетингових комунікацій

Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
Фокус на покращення сервісу	Соціальні мережі, інтернет, живе спілкування	Зручність використання, функціональність, якість	Надати цільовим клієнтам інформацію про основні переваги продукту	Практичний показ роботи продукту.

Висновки до розділу 5

У цьому розділі розглядається стартап-проект зі створення автоматизованої система виявлення гвинтокрилих БПЛА в небі.

Цільовими клієнтами визначено зацікавлені особи що бажають захистити свій будинок від дронів.

Під час опису технічної здійсненності видно, що всі необхідні технології доступні та доступні розробникам, а деякі частини потрібно реалізувати самостійно, наприклад, саму модель для виявлення дронів.

Було визначено основні характеристики ринку, і на ринку є 7 основних. Аналітики вважають, що інвестиції в цю галузь є привабливими, оскільки можливі прибутки перевищують прибуток, заснований на прибутку.

Визначено основу для факторів ризику, можливостей і основних стратегій розвитку. Також окреслено поняття трьох рівнів моделей продукту, цінових обмежень і маркетингових комунікацій. Як правило, факторів можливостей більше, ніж факторів ризику, що надає цій концепції потенціал. Факторами можливостей є збільшення розміру клієнтів, поява нових технологій на ринку, поява іноземних інвесторів тощо.

Дехто вважає, що конкуренція жорстка, але в міру додавання в проект інноваційних технологій можна буде залучити клієнтів, тому що в цій сфері найбільшу роль при виборі відіграє функціональність, а не бренд розробника. Для впровадження інноваційних підходів важливо постійно аналізувати ринок: нові технології та методи, конкурентів і вимоги гравців галузі.

ВИСНОВКИ

У цій магістерській дисертації було розроблено автоматизовану систему для виявлення гвинтокрилих БПЛА в небі за допомогою штучного інтелекту.

Перш за все, було розглянуто існуючі методи для виявлення БПЛА. Розглянули переваги та недостатки кожного методу. Для системи було вирішено взяти метод візуального виявлення, враховуючи його легкість у виявленні, дешевизні використання. Також за допомогою цього методу була можливість використовувати штучний інтелект для навчання моделі.

У другому розділі було розглянуто основні концепції штучного інтелекту, машинного навчання і глибокого навчання, а також аналіз зв'язків та відмінності між ними. Було наведено детальний огляд концепцій машинного та глибокого навчання. Машинне навчання визначається як метод розробки алгоритмів, які «навчаються» з даних без явного програмування. З іншого боку, глибоке навчання — це підмножина машинного навчання, яке використовує глибокі нейронні мережі для вирішення проблем.

Вирішивши взяти метод візуального виявлення у третій частині роботи було проведено аналіз теми виявлення об'єктів, розкриваючи ключові аспекти цього напрямку в області комп'ютерного зору. Розглянуто останні технологічні досягнення в області комп'ютерного зору, зосереджуючись на перевагах і недоліках різних підходів. Порівняно сучасні методи виявлення об'єктів, відзначено їх переваги та недоліки. Вони особливо підкреслюють важливість швидкості та точності виявлення об'єктів, оскільки це ключові характеристики для вирішення реальних завдань. За результатами бенчмарків було вирішено обрати алгоритм YOLOv8 для завдання виявлення безпілотних літальних апаратів у небі. Цей вибір виправданий високою ефективністю та швидкістю алгоритму, що робить його ідеальним для виявлення об'єктів у режимі реального часу, наприклад дронів.

У четвертому розділі визначено середовища, в яких відбувалась розробка нашої моделі та веб-додатку, і детально описано їхні переваги, пояснивши вибір саме цих середовищ. Було детально розглянуто етап за етапом процес підготовки зображень для створення датасету, з метою подальшого використання в навчанні моделі та розглянуто етапи навчання нашої моделі.

Після завершення процесу навчання моделі перейшли до розробки веб-додатку, що надавав можливість тестувати модель в режимі реального часу.

Після проведення тестування моделі в веб-додатку можна стверджувати, що модель ефективно працює при виявленні дронів великих або середніх розмірів.

На основі впровадженої системи розроблено стартап-проект. Для її створення було проаналізовано систему, її сильні, слабкі сторони та можливості для вдосконалення, порівняно з конкурентами, цінність і вартість для потенційних користувачів, визначено можливі варіанти поширення. Розроблена система рекомендацій відповідає встановленим вимогам, виконує необхідні функції.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. How to Detect a Drone in the Sky. URL: <https://www.droneblog.com/detect-a-drone/>
2. How to Detect a Drone in the Sky. URL: <https://dronesurveyservices.com/how-to-detect-a-drone-in-the-sky/>
3. 10 технологій виявлення та протидії дронам на сьогоднішній день. URL: <https://bezpeka.club/10-technologies-for-detecting-countermeasures-drones/>
4. Радіолокація. URL: <https://uk.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B4%D1%96%D0%BE%D0%BB%D0%BE%D0%BA%D0%B0%D1%86%D1%96%D1%8F>
5. Artificial intelligence (AI). URL: <https://www.techtarget.com/searchenterpriseai/definition/AI-Artificial-Intelligence>
6. What's the difference between Machine Learning and Deep Learning? URL: <https://viso.ai/deep-learning/deep-learning-vs-machine-learning/>
7. What is machine learning and how does it work? In-depth guide. URL: <https://www.techtarget.com/searchenterpriseai/definition/machine-learning-ML>
8. Machine learning. URL: https://en.wikipedia.org/wiki/Machine_learning
9. Deep learning. URL: https://en.wikipedia.org/wiki/Deep_learning
10. What is deep learning? URL: <https://www.ibm.com/topics/deep-learning>
11. Feature Engineering или стероиды для ML моделей. URL: <https://habr.com/ru/companies/ruvds/articles/680498/>
12. Major Differences between Machine Learning and Deep Learning. URL: <https://skill-lync.com/blogs/technical-blogs/cse-major-differences-between-machine-learning-and-deep-learning>
13. Object detection. URL: https://en.wikipedia.org/wiki/Object_detection
14. What is Object Detection? The Ultimate Guide. URL: <https://blog.roboflow.com/object-detection/>
15. Object Detection in 2024: The Definitive Guide. URL: <https://viso.ai/deep-learning/object-detection/>

16. Як працює класифікація зображень? URL: <https://www.unite.ai/uk/how-does-image-classification-work/>
17. YOLO. URL: <https://medium.com/@khwabkalra1/yolo-you-only-look-onc-523b01ec4f4d>
18. How single-shot detector (SSD) works? URL: <https://developers.arcgis.com/python/guide/how-ssd-works/>
19. What is R-CNN? URL: <https://blog.roboflow.com/what-is-r-cnn/>
20. Mask R-CNN: архитектура современной нейронной сети для сегментации объектов на изображениях. URL: <https://habr.com/ru/articles/421299/>
21. Fast object detection with SqueezeDet on Keras. URL: <https://medium.com/omnius/fast-object-detection-with-squeezedet-on-keras-5cdd124b46ce>
22. MobileNet: меньше, быстрее, точнее. URL: <https://habr.com/ru/articles/352804/>
23. YOLOR — Объяснение статьи и выводы – Углубленный анализ . URL: <https://habr.com/ru/articles/707098/>
24. Reasons Why Python is Good for AI and ML. URL <https://djangostars.com/blog/why-python-is-good-for-artificial-intelligence-and-machine-learning/>
25. Roboflow. URL: <https://medium.com/red-buffer/roboflow-d4e8c4b52515>
26. Visual Studio Code. URL: https://uk.wikipedia.org/wiki/Visual_Studio_Code
27. What is Image Preprocessing and Augmentation? URL: <https://blog.roboflow.com/why-preprocess-augment/>
28. About Train, Validation and Test Sets in Machine Learning URL: <https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7>
29. What are the benefits of using CUDA for machine learning? URL: <https://blog.roboflow.com/why-preprocess-augment/>
30. Ultralytics YOLOv8. URL: <https://habr.com/ru/articles/710016/>
31. YOLO Performance Metrics. URL: <https://docs.ultralytics.com/guides/yolo-performance-metrics/>

32. Material UI – Overview. URL: <https://mui.com/material-ui/getting-started/>
33. CRACO. URL: <https://craco.js.org/docs/>
34. OpenCV. URL: <https://opencv.org/about/>
35. Почему ONNX так популярен в ML: конвертации, утилиты и инференс.
URL: <https://habr.com/ru/articles/757080/>
36. Розроблення стартап-проекту : Методичні рекомендації до виконання розділу магістерських дисертацій для студентів інженерних спеціальностей / За заг. ред. О.А. Гавриша. – Київ : НТУУ «КПІ», 2016. – 28 с.