

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління

ТУРЧЕНКО Віктор Володимирович

**Прикладний програмний інтерфейс для ідентифікації
літальних апаратів за допомогою глибоких нейронних мереж /
Application programming interface for aircraft identification using deep
neural networks**

Спеціальність 122 – Комп'ютерні науки
Освітньо-професійна програма – Комп'ютерні науки

Дипломний проект

Виконав студент групи КН-41
В. В. Турченко

Науковий керівник:
к.т.н., доцент Н. М. Васильків

Дипломний проект допущено до
захисту
« ___ » _____ 2023 р.

Завідувач кафедри
_____ М. П. Комар

Тернопіль – 2023

Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Освітній ступінь «бакалавр»

Спеціальність 122 – Комп'ютерні науки
Освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ М. П. Комар
« ____ » _____ 2022р.

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЕКТ СТУДЕНТУ

Турченку Віктору Володимировичу

(прізвище, ім'я, по батькові)

1. Тема проекту: Прикладний програмний інтерфейс для ідентифікації літальних апаратів за допомогою глибоких нейронних мереж / Application programming interface for aircraft identification using deep neural networks
керівник проекту к.т.н., доцент Н.М. Васильків
затверджені наказом по університету від 08 грудня 2022 р. № 491.

2. Строк подання студентом закінченого проекту 01 червня 2023 р.

3. Вихідні дані до проекту: технічне завдання.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

- провести аналіз предметної області;
- проаналізувати існуючі аналоги ідентифікації об'єктів;
- зробити постановку задач дослідження;
- розробити структуру прикладного програмного інтерфейсу;
- провести аналіз створених та доступних у відкритому доступі архітектур глибоких НМ для вирішення задач виявлення ОІ-ХВ та ОІ-БН і задачі класифікації БН;
- розробити алгоритмічне забезпечення ППІ;
- створити набір даних для навчання НМ, що має складатися з великої кількості зображень літальних апаратів;

- здійснити серію експериментальних досліджень з оцінки похибок детекції та класифікації;
- реалізувати програмні модулі системи;
- розробити та протестувати ППІ.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

- структурна схема моделі згорткової нейронної мережі;
- схема алгоритму прикладного програмного інтерфейсу.

6. Консультанти розділів проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
Н. контроль	к.т.н., доцент Н. М. Васильків		

7. Дата видачі завдання 08 грудня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту	Строк виконання етапів проекту	Примітка
1	Аналіз предметної області	30.12.2022	
2	Моделювання ППІ для ідентифікації на основі глибоких нейронних мереж	24.03.2023	
3	Реалізація прикладного програмного інтерфейсу	12.05.2023	
4	Повне завершення та оформлення дипломного проекту	01.06.2023	

Студент _____ В. В. Турченко

Керівник проекту _____ Н. М. Васильків

РЕФЕРАТ

Пояснювальна записка до дипломного проекту: 77 с., 30 рис., 1 табл., 4 додатки, 35 джерел.

Метою роботи є створення прикладного програмного інтерфейсу для ідентифікації літальних апаратів за допомогою глибоких нейронних мереж.

Об'єктом дослідження є процес ідентифікації літального апарату.

Предмет дослідження – програмний інтерфейс для ідентифікації літальних апаратів на основі глибоких нейронних мереж.

Для проведення досліджень використано чисельний та статистичний аналіз, методи теорії машинного навчання і глибоких нейронних мереж.

Розроблено прикладний програмний інтерфейс для ідентифікації літальних апаратів за допомогою глибоких нейронних мереж.

Прикладний програмний інтерфейс може використовуватись для ідентифікації літальних апаратів на спеціалізованих контрольних пунктах аеропортів та прикордонних об'єктів, для забезпечення моніторингу повітряного простору у разі відмови інших систем ідентифікації літальних апаратів.

ЛІТАЛЬНИЙ АПАРАТ, БОРТОВИЙ НОМЕР, ІДЕНТИФІКАЦІЯ,
ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ, ПРИКЛАДНИЙ ПРОГРАМНИЙ ІНТЕРФЕЙС

ABSTRACT

The bachelor's thesis explanatory note: 77 pages, 30 figures, 1 table, 4 appendices, 35 sources.

The aim of the bachelor's thesis is to create an application programming interface for aircraft identification using deep neural networks.

The object of research is the process of aircraft identification.

The subject of research is a programming interface for identification of aircraft based on deep neural networks.

Numerical and statistical analysis, methods of machine learning theory, and deep neural networks were used for the research.

An application programming interface for aircraft identification using deep neural networks has been developed.

The application programming interface can be used to identify aircraft at specialized checkpoints at airports and border facilities, to ensure airspace monitoring in case of failure of other aircraft identification systems.

AIRCRAFT, BOARD NUMBER, IDENTIFICATION, CONVOLUTIONAL NEURAL NETWORKS, APPLICATION PROGRAMMING INTERFACE

ТЕХНІЧНЕ ЗАВДАННЯ

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

1.1 Прикладний програмний інтерфейс для ідентифікації літальних апаратів за допомогою глибоких нейронних мереж.

1.2 Область застосування – ідентифікація літальних апаратів.

2. ОСНОВА ДЛЯ РОЗРОБЛЕННЯ

Основою для розроблення є завдання на дипломний проект, затверджене кафедрою інформаційно-обчислювальних систем і управління факультету комп'ютерних інформаційних технологій Західноукраїнського національного університету.

3. ПРИЗНАЧЕННЯ РОЗРОБЛЕНОГО КОМПЛЕКСУ

Метою роботи є створення прикладного програмного інтерфейсу для ідентифікації літальних апаратів за допомогою глибоких нейронних мереж.

4. ДЖЕРЕЛА РОЗРОБЛЕННЯ

Джерелами даної розробки є матеріали навчальної і реферативної літератури, технічна документація, науково-дослідні статті, журнали, Інтернет.

5. ТЕХНІЧНІ ВИМОГИ

5.1 Основні функціональні вимоги до програмного інтерфейсу:

- приймання зображення від клієнта;
- здійснення детекції хвостової частини;
- здійснення детекції бортового номеру;
- здійснення класифікації літерно-цифрового позначення бортового номера;
- зберігання в сховищах даних отриманих та опрацьованих зображень;
- надсилання повідомлення про результат ідентифікації.

5.2 Вимоги до апаратних засобів:

– прикладний програмний інтерфейс запускається на персональному комп'ютері Lenovo Legion S7 з 32 гігабайт оперативної пам'яті та графічним процесором Nvidia GeForce RTX 3060.

5.3 Вимоги до програмних засобів:

- для розробки – програмне забезпечення Python 3.10;
- для реалізації моделей глибокого навчання – бібліотека машинного навчання Pytorch;
- для створення прикладного програмного інтерфейсу – FastAPI.

6. ПОРЯДОК КОНТРОЛЮ

6.1 Представлення дипломного проекту на попередній захист.

6.2 Представлення дипломного проекту на захист.

Завдання прийняв до виконання _____ В. В. Турченко

Керівник дипломного проекту _____ Н. М. Васильків

ЗМІСТ

Перелік умовних позначень	9
Вступ.....	10
1 Аналіз предметної області.....	11
1.1 Коротка характеристика літального апарату.....	11
1.2 Процес ідентифікації літального апарату	13
1.3 Огляд і аналіз існуючих аналогів	16
1.4 Постановка задачі проектування	21
2 Моделювання ППІ для ідентифікації на основі глибоких нейронних мереж	24
2.1 Структура ППІ для ідентифікації літальних апаратів за допомогою глибоких нейронних мереж.....	24
2.2 Модель згорткової нейронної мережі YOLO	29
2.3 Алгоритми ідентифікації на основі глибоких нейронних мереж.....	33
2.4 Експериментальні дослідження модулів детекції.....	37
3 Реалізація прикладного програмного інтерфейсу.....	46
3.1 Програмна реалізація	46
3.2 Інтерфейс користувача.....	50
3.3 Тестування прикладного програмного інтерфейсу	56
Висновки	62
Список використаних джерел	64
Додаток А Модель згорткової нейронної мережі. Схема структурна... Помилка!	
Закладку не визначено.	
Додаток Б Схема алгоритму прикладного програмного інтерфейсу..... Помилка!	
Закладку не визначено.	
Додаток В Код програми	Помилка! Закладку не визначено.
Додаток Г Довідка про використання	Помилка! Закладку не визначено.

					ДП.КН.8091602.077.ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розробив		Гурченко В. В.			Прикладний програмний інтерфейс для ідентифікації літальних апаратів за допомогою глибоких нейронних мереж	Літ.	Арк.	Аркушів
Перевір.		Васильків Н.М.					8	77
Н. Контр.		Васильків Н.М.			ЗУНУ.ФКІТ.КН-41			
Затвердив		Комар М. П.						

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ОІ – область інтересу

ОРС – оптичне розпізнавання символів

НМ – нейронні мережі

ЗГНМ – згорткова нейронна мережа

ОІ-ХВ – область інтересу - хвостова частина

ОІ-БН – область інтересу - бортовий номер

ППІ – прикладний програмний інтерфейс

БН – бортовий номер

ПК – персональний комп'ютер

					ДП.КН.8091602.077.ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

З розвитком сучасних технологій збільшилась кількість різних видів літальних апаратів (літаки, гелікоптери, планери та дрони), які зазвичай використовуються для перевезення пасажирів і вантажів, а також у військових, рекреаційних і наукових цілях. Кожен із цих літальних апаратів повинен бути певним чином позначений. Тому з метою безпеки здійснення санкціонованих та виявлення несанкціонованих повітряних переміщень необхідно їх ідентифікувати.

Сучасні системи ідентифікації літальних апаратів базуються на використанні систем навігації та радарів, які дозволяють виявляти та ідентифікувати літальні апарати на основі їхніх внутрішніх ідентифікаторів або технічних параметрів. Однак, у разі виникнення проблем з цими методами ідентифікації, такими як нестача чіткої інформації, обмежене радарне покриття або наявність перешкод для передачі сигналів, візуальна ідентифікація залишається єдиним методом ідентифікації літальних апаратів.

Використовуючи унікальні візуальні ознаки літального апарату, зокрема його бортовий номер, можна провести ідентифікацію літального апарату за допомогою камер спостереження. Таким чином, задача ідентифікації літальних апаратів на зображенні є актуальною.

Метою дипломного проекту є створення прикладного програмного інтерфейсу для ідентифікації літальних апаратів за допомогою глибоких нейронних мереж.

Об'єктом дослідження є процес ідентифікації літального апарату.

Предмет дослідження – програмний інтерфейс для ідентифікації літальних апаратів на основі глибоких нейронних мереж.

Для проведення досліджень доцільно використати чисельний та статистичний аналіз, методи теорії машинного навчання і глибоких нейронних мереж.

					ДП.КН.8091602.077.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Коротка характеристика літального апарату

Літальні апарати – це транспортні засоби, що призначені для польотів в атмосфері Землі. Вони зазвичай використовуються для перевезення пасажирів і вантажів, а також у військових, рекреаційних і наукових цілях. Існують різні типи літальних апаратів, включаючи літаки, гелікоптери, планери та дрони. Дослідження у цій роботі будуть стосуватися літаків [1].

Літак зазвичай складається з різних компонентів, які працюють разом, щоб полегшити політ. Ці компоненти включають фюзеляж (основний корпус), крила, оперення (хвостове оперення), шасі, двигуни та поверхні керування (елерони, рулі висоти та кермо). Літальні апарати мають системи керування польотом, які дозволяють пілотам маневрувати ними. Основними органами керування польотом є елерони (керування креном), рулі висоти (керування тангажом) і кермо напряду (керування поворотом). Цими органами управління керує пілот за допомогою колонок керування, педалей або інших механізмів [1].

Експлуатація повітряних суден регулюється національними та міжнародними авіаційними органами для забезпечення безпеки та стандартизації. Ці правила охоплюють такі сфери, як ліцензування пілотів, технічне обслуговування та перевірки повітряних суден, процедури управління повітряним рухом та управління повітряним простором.

Комерційна авіація передбачає використання повітряних суден для перевезення пасажирів і вантажів. Вона охоплює регулярні авіалінії, чартерні компанії та вантажні перевізники. Комерційні літаки підлягають сертифікаціям для забезпечення безпеки та комфорту пасажирів [2].

Бортовий номер повітряного судна – це унікальний літерно-цифровий код, який присвоюється йому з метою ідентифікації. Він також відомий як реєстраційний або N-номер (у США). Формат і правила, пов'язані з номерами бортів, можуть дещо відрізнятись в різних країнах, але загалом вони мають одну мету: ідентифікувати літальний апарат.

					ДП.КН.8091602.077.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

Формуванням і призначенням хвостових номерів зазвичай керує уповноважений орган цивільної авіації або регулюючий орган кожної країни. Ці органи влади ведуть реєстр повітряних суден і призначають бортові номери окремим літакам на основі конкретних інструкцій. Бортові номери зазвичай розміщують на вертикальному стабілізаторі або хвостовому кілі літака. Також у деяких країнах авіакомпанії можуть дублювати бортовий номер під крилами літака. На рисунку 1.1 представлено зображення літального апарату з бортовим номером на хвостовому кілі [3, 4].



Рисунок 1.1 – Зображення бортового номера на хвостовій частині літального апарату

Значення бортового номера може відрізнитися залежно від країни та конкретного формату, який використовується. У деяких випадках бортовий номер може надати інформацію про країну реєстрації літака або організацію, якій він належить. Наприклад, усі номери літальних апаратів в Сполучених Штатах Америки починаються з «N», у Канаді – з «C», у Німеччині – з «D», в Україні – з «UR». Решта цифр і літер, як правило, визначаються на розсуд власника літального апарату, але в більшості випадків складатимуться з трьох цифр, за якими слідують дві літери.

Іноді літери можуть позначати географічний регіон або організацію, пов'язану з літаком.

					ДП.КН.8091602.077.ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

В інших випадках бортові номери можуть бути призначені випадковим чином або послідовно без будь-якого конкретного значення. Основне призначення хвостових номерів – надати унікальність для кожного повітряного судна, що дозволяє легко ідентифікувати та відстежувати авіаційними органами, органами управління повітряним рухом та іншими відповідними сторонами.

Бортовий номер, що присвоюється літальному апарату, ніколи не змінюється впродовж усього часу використання останнього. Тому за допомогою бортового номера можливо ідентифікувати повітряне судно та дізнатися інформацію критичної важливості (модель, рік випуску, власники літака, дати технічного обслуговування та інше) [3].

1.2 Процес ідентифікації літального апарату

Основний компонент керування сучасного повітряного судна – система керування польотом (Flight Management System). Це спеціалізована комп’ютерна система, яка автоматизує широкий спектр завдань у польоті, зменшуючи робоче навантаження на екіпаж. Основною функцією такої системи є керування планом польоту [5].

Використовуючи різні датчики (наприклад, датчики глобального позиціонування GPS та інерційні датчики INS, що часто підтримуються радіонавігацією) для визначення положення літака, система керування польотом може направляти літак за необхідною траєкторією. Коли система керування повітряним рухом, що знаходиться на землі в центрі керування польотами, ідентифікує літак за його унікальним внутрішнім номером, літак переходить до «переклички». Літак надає дані з GPS для визначення його локації центром керування польотами.

Однак, якщо літак не відкликається та не містить модулів ідентифікації, виявити його можливо лише за допомогою радару.

Радіолокаційна станція або радар (від англійського radar – скорочення від radio detection and ranging, радіохвильове виявлення та визначення відстані) –

					ДП.КН.8091602.077.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

система для виявлення повітряних, морських і наземних об'єктів, а також для визначення їхньої дальності та геометричних параметрів, яка використовує метод, заснований на випромінюванні радіохвиль і реєстрації їх віддзеркалень від об'єктів [6].

Отже, робота радару полягає у тому, що радіолокатор надсилає електромагнітну хвилю та приймає її відбиття від об'єктів навколишнього середовища, що дозволяє визначити відстань до об'єкту та його фізичні параметри. Таким чином, об'єкт з'являється на радарі та його геолокацію можна відслідковувати. Проте ніякої ідентифікаційної інформації від повітряного судна за допомогою такого підходу отримати неможливо.

Інколи виявити літальний апарат з використанням радару неможливо через низьку висоту польоту або через особливість рельєфу земної поверхні. Прикладом цього є польоти малих літальних апаратів, які використовують зловмисники для транспортування заборонених речовин або наркотичних препаратів через кордони країн у Південній Америці [7].

Проблеми із надсиланням сигналів можуть виникнути випадково, або вони можуть бути створені навмисно в деяких випадках. Наприклад, військові винищувачі США F-22 і F-35 використовують технологію «Стелс», що захищає літак від виявлення високочастотними радаром [6].

Єдиним методом ідентифікації літальних апаратів, окрім вище перелічених, залишається візуальна ідентифікація. Сучасні камери дозволяють зближувати та виявляти літальні апарати на далекій відстані. Наприклад, камера Nikon, дозволяє побачити літак у задовільній якості на висоті польоту 10 км (рисунок 1.2) [5].

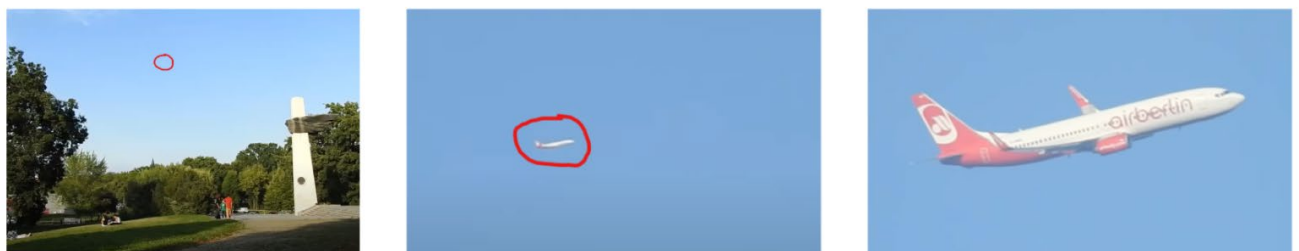


Рисунок 1.2 – Зближення повітряного судна за допомогою камери Nikon [5]

									ДП.КН.8091602.077.ПЗ	Арк.
										14
Змн.	Арк.	№ докум.	Підпис	Дата						

Застосовуючи більш професійне обладнання, можна отримати зображення у відмінній якості. Розмістивши професійні камери спостереження на спеціалізованих контрольних пунктах, можна отримати зображення літального апарата у відмінній якості та візуально його ідентифікувати за бортовим номером. Таким чином, задача ідентифікації літального апарата за його зображенням є актуальною.

На рисунку 1.3 зображена узагальнена схема ідентифікації літального апарату. Для початку вхідне зображення або кадр з відео подається на вхід блоку 1. На цьому етапі виявляється область інтересу (ОІ), що необхідна для подальшої ідентифікації.

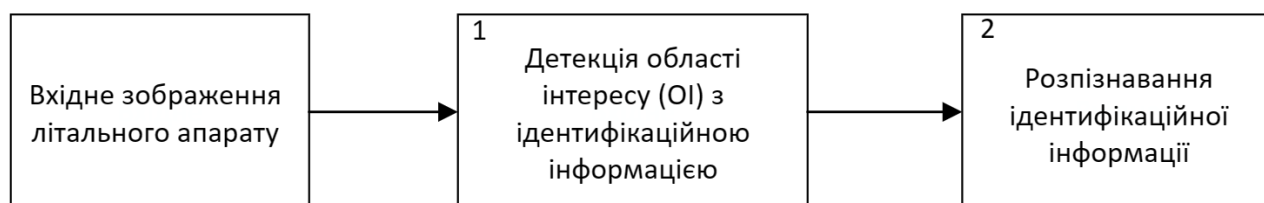


Рисунок 1.3 – Узагальнена схема ідентифікації літального апарату

Область інтересу – це ділянка зображення, яка є важливою для вирішення певної задачі. Після виявлення областей інтересу, вони позначаються обмежувальними рамками та класифікуються. Як було визначено вище, для ідентифікації літального апарату необхідно використати його бортовий номер, а отже ОІ в цьому випадку – це область зображення, яка містить унікальний літерно-цифровий номер літака.

На рисунку 1.4 зображено детекцію ОІ на прикладі літального апарата компанії міжнародних авіаліній України. Для детекції ОІ необхідно спочатку наблизити хвостову частину літального апарату.

Наступним етапом є переведення зображення бортового номера у літерно-цифровий код, наприклад, у форматі ASCII, і в подальшому використати як цифри або текст (див. блок 2 рисунку 1.3). Для цього необхідно застосувати технологію

ідентифікації:

- транспортних засобів за допомогою їх номерних знаків;
- адреси та/або місця доставки при автономній доставці товарів та кореспонденції за допомогою дронів і наземних роботів;
- об'єктів навколишнього середовища при автономному керуванні транспортними засобами.

Ідентифікація транспортних засобів шляхом розпізнавання їх номерних знаків переважно використовується для оплати штрафів та проїзду по автомагістралях. Оплата за використання автомагістралей є одним із важливих засобів формування доходів для відомств, відповідальних за їх експлуатацію та утримання [8].

Процес збору мита автоматизовано за допомогою датчиків і камер у багатьох країнах, де номерні знаки та тип транспортного засобу автоматично визначаються для розрахунку та оплати за проїзд. Головною перевагою такої автоматизації є уникнення довгих черг у місцях збору оплати. Однак такі системи покладаються на попередні знання про конкретний формат номерних знаків та їх встановлене положення на передній частині автомобіля. З цієї причини їх продуктивність погіршиться для транспортних засобів, що не використовують визначених форматів та положення номерних знаків.

Існуючим рішенням цієї проблеми є використання електронних міток на основі радіочастотної ідентифікації (Radio-frequency identification – RFID), встановлених на транспортних засобах. Однак, це супроводжується додатковими витратами. Крім того, електронні мітки встановлюють лише деякі учасники руху на автомагістралях.

У [8] автори пропонують автоматичну систему для збору плати за проїзд, яка складається з трьох кроків:

- розпізнавання типу транспортного засобу (легковий, вантажний, автобус);
- детекція ОІ – локалізація номерного знаку;
- ідентифікація номерного знаку за допомогою ОРС.

Для розпізнавання типу та ідентифікації номерного знаку транспортного

					ДП.КН.8091602.077.ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

засобу використано згорткові нейронні мережі (ЗГНМ). Однак, процес ідентифікації є складним через варіативність зображень. Існує багато факторів, які можуть перешкоджати виявленню номерного знаку:

- безлад на фоні, що створює проблеми для алгоритмів детекції;
- нерівномірне розташування номерних знаків на зображенні;
- кілька номерних знаків на зображенні;
- забруднення на номерних знаках;
- пошкодження на номерних знаках;
- відсутність номерних знаків.

Розглянемо методи ідентифікації, що використовують відомі компанії: Japan Post Co., Амазон, Тесла.

Japan Post Co., японський постачальник поштових, логістичних і кур'єрських послуг, тестує в Токіо автономного робота, поштового кур'єра, який здійснює автоматичну ідентифікацію адрес отримувачів, може маневрувати навколо перешкод і перетинати перехрестя зі світлофорами. Розроблений токійською робототехнічною компанією ZMP Inc., DeliRo має невеликий розмір, вбудовані камери та датчики і може перевозити пакунки вагою до 30 кілограмів, рухаючись зі швидкістю шість кілометрів на годину. Робот за допомогою алгоритмів розпізнавання об'єктів і детекції області інтересу оминає перешкоди та автономно перевозить пакунки і кореспонденцію [9].

Компанія Амазон доставляє товари клієнтам швидко, безпечно та з невеликими витратами за допомогою автономних дронів, які використовують системи ідентифікації областей інтересу та місць доставки у реальному часі [10]. Проблема полягає в тому, що замовлень є дуже багато і потрібно масштабувати процес доставки до великої кількості клієнтів, водночас доставляючи не тільки до рідко населених місцевостей, де відсутні перешкоди, а і у складних ситуаціях, де на шляху дрона можливі різні перешкоди, такі як інші літальні апарати, дерева, птахи, лінії електропередач та будівлі.

Галузь логістики переповнена безпілотними літальними апаратами, але не всі системи дронів однакові. Наприклад, більшість дронів не мають можливості

					ДП.КН.8091602.077.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

відчувати й уникати інших літаків і перешкод, що може створювати проблеми. Використання таких дронів вимагає візуальних спостерігачів (людини) уздовж маршруту кожного польоту, щоб допомогти безпілотникам уникнути небезпек. Цей тип безпілотника можливо розгорнути відносно швидко, але він обмежує операції доставки невеликим радіусом. Такий дрон не несе великої користі, адже він не є автономним та переносить невелику кількість вантажу, вимагаючи присутності оператора.

Компанія Амазон створила складну та передову систему «відчуй і уникни», яка дозволить дрону працювати без візуальних спостерігачів на більших відстанях безпечно та надійно. Система «відчуй і уникни» забезпечує безпеку дрона та гарантує, що він не завдасть шкоди навколишньому середовищу при його переміщенні та приземленні. Під час польоту до місця доставки дрони повинні мати можливість ідентифікувати статичні та рухомі перешкоди. Алгоритми ідентифікації використовують різноманітний набір технологій для виявлення об'єктів, зокрема архітектури машинного навчання у поєднанні з іншими датчиками. Такі автономні дрони не потребують керування оператором та можуть доставляти товари зі складів швидко і економно [10].

Іншою відомою розробкою для автоматичної ідентифікації об'єктів навколишнього середовища є автономне керування транспортними засобами. Автопілот компанії Тесла виявляє знаки, світлофори, перешкоди попереду та дозволяє безпечно пересуватися вулицями міст без втручання людини у процес керування [11].

Якщо водій не зобов'язаний виконувати всі завдання керування, то такий рівень автономності називається SAE 5.

Автономні транспортні засоби на цьому рівні не обмежені геозонами та не залежать від погодних умов і можуть безпечно та комфортно перевозити людей. Підхід Тесла для досягнення рівня автономності SAE 5 базується на навчанні НМ на даних поведінки сотень тисяч водіїв Тесла в режимі звичайного водіння без автопілоту. НМ при навчанні також використовувала інформацію з інших датчиків автомобіля, таких як камери видимого світла, карти навігації, ультразвукові

					ДП.КН.8091602.077.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

датчики паркування тощо.

Тесла прийняла свідоме рішення не використовувати лідар, тобто метод визначення дальності шляхом наведення лазера на об'єкт та вимірювання часу повернення відбитого променя до приймача.

Отже, не використовуючи лідар, система автопілоту може забезпечити повне автономне керування автомобілем, базуючись на навченій НМ та зображеннях з його камер.

Як було згадано в 1.1, підзадачею процесу ідентифікації літального апарату з метою визначення його бортового номера є класифікація літер та цифр на зображенні, отриманому з ОІ.

Відома технологія розпізнавання цифр та літер OPC використовується багатьма компаніями у своїх програмних продуктах. Adobe використовує технологію для перетворення документів та файлів з формату PDF у текстовий формат, який можливо редагувати. За допомогою OPC Microsoft та Google у своїх програмних продуктах OneNote та Google Drive дозволяють користувачам копіювати текст, що розташований на зображеннях та редагувати їх [12].

Технологія OPC використовує інтелектуальне розпізнавання символів (Intelligent character recognition – ICR), щоб опрацьовувати текст так само, як це роблять люди. Так як OPC працює тільки з друкowanими символами, необхідно було створити інтелектуальну систему, що розпізнавала б і рукописні символи.

Інтелектуальне розпізнавання символів застосовує передові методи машинного навчання, зокрема НМ, які аналізують текст на багатьох рівнях, багаторазово обробляючи зображення. НМ шукає різні атрибути зображення, такі як криві, лінії, перетини та петлі, і поєднує результати всіх цих різних рівнів, щоб отримати остаточний результат. Незважаючи на те, що інтелектуальне розпізнавання символів зазвичай обробляє зображення по одному символу, процес є швидким, а результати отримуються за секунди та з малою похибкою [12].

База зображень рукописних цифр MNIST була створена у 1998 році для розробки алгоритмів класифікації цифрової інформації [13].

На рисунку 1.5 представлено типові зображення, які формують базу даних

									ДП.КН.8091602.077.ПЗ	Арк.
										20
Змн.	Арк.	№ докум.	Підпис	Дата						

точок прямокутника ОІ-ХВ та координати чотирьох точок прямокутника ОІ-БН, для отриманого набору даних визначити розміри вибірок навчання, перевірки та тестування НМ.

Потрібно провести аналіз створених та доступних у відкритому доступі архітектур глибоких НМ, що придатні для вирішення задачі виявлення ОІ-ХВ. Для цього застосувати вибрану архітектуру глибокої НМ в режимі передачі знань (transfer learning) та здійснити серію експериментальних досліджень з оцінки похибки детекції.

Потрібно провести аналіз створених та доступних у відкритому доступі архітектур глибоких НМ, що придатні для вирішення задачі виявлення ОІ-БН, і застосувати вибрану архітектуру глибокої НМ в режимі передачі знань, здійснити серію експериментальних досліджень з оцінки похибки детекції.

Потрібно провести аналіз створених та доступних у відкритому доступі архітектур глибоких НМ, що придатні для вирішення задачі класифікації алфавітно-цифрової інформації бортового номера літального апарату, застосувати вибрану архітектуру глибокої НМ в режимі передачі знань, здійснити серію експериментальних досліджень з оцінки похибки класифікації.

Необхідно створити архітектуру програмної системи та визначити кількість програмних модулів системи та взаємозв'язки між модулями.

Потрібно здійснити проектування та реалізацію усіх програмних модулів системи.

Необхідно створити прикладний програмний інтерфейс (ППІ), який дозволить надсилати зображення літального апарату та отримувати його ідентифікаційну інформацію. Він повинен містити у собі елементи виявлення помилок у ході роботи та можливість повторної спроби виконання поставлених задач. У разі виникнення невідвортної помилки прикладний програмний інтерфейс повинен відображати відповідні сповіщення.

Також необхідно провести тестування коректної роботи прикладного програмного інтерфейсу.

Таким чином, так як метою роботи є створення прикладного програмного

					ДП.КН.8091602.077.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		22

інтерфейсу для ідентифікації літальних апаратів за допомогою глибоких нейронних мереж, то для її досягнення необхідно:

- провести аналіз предметної області;
- проаналізувати існуючі аналоги ідентифікації об'єктів;
- розробити структуру прикладного програмного інтерфейсу;
- провести аналіз створених та доступних у відкритому доступі архітектур глибоких НМ для вирішення задач виявлення ОІ-ХВ та ОІ-БН і задачі класифікації бортового номеру (БН);
- розробити алгоритмічне забезпечення ППІ;
- створити набір даних для навчання НМ, що має складатися з великої кількості зображень літальних апаратів;
- здійснити експериментальні дослідження для оцінювання похибок детекції та класифікації;
- реалізувати програмні модулі системи;
- розробити та протестувати ППІ.

					ДП.КН.8091602.077.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

вирішення індивідуальних задач на кожному з етапів, у результаті чого літальний апарат ідентифікується.

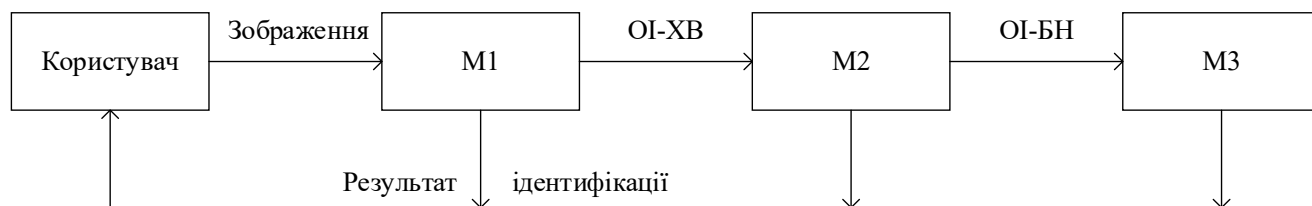


Рисунок 2.1 – Структура ППІ для ідентифікації літальних апаратів

Набір вхідних даних та набір даних ОІ-ХВ використовуються як для навчання НМ у модулях М1 та М2, так і для тестування системи ідентифікації літальних апаратів. Модуль М3 не проходить процес навчання, адже він містить технологію ОРС, яка перетворює зображення у текст з малою похибкою. Технологія ОРС представлена багатьма розробниками у формі готових програмних модулів, зокрема популярним загальнодоступним рішенням EasyOCR [14].

Набір вхідних даних – це вибірка зображень або кадрів з відео, що застосовуються для навчання НМ у модулі М1. Зображення літальних апаратів на цьому етапі є довільного розміру. Для підготовки такого набору даних необхідно зібрати велику кількість зображень та провести анотацію ОІ-ХВ.

Для кожного зображення формується файл (анотація) формату xml, у якому міститься інформація про координати чотирьох точок прямокутника, що формує обмежувальну рамку ОІ-ХВ. Коли програмна система ідентифікації літальних апаратів працює в режимі детекції, вхідними даними виступають такі ж зображення літальних апаратів, проте в цьому режимі анотації не створюються.

Модуль М1 – це детектор ОІ-ХВ. Для детекції необхідних елементів на зображенні доцільно використати глибокі ЗГНМ.

У загальному випадку існує два способи використання ЗГНМ:

- з нуля, коли вагові коефіцієнти такої НМ ініціалізуються випадковим чином;
- в режимі передачі знань (transfer learning), коли така НМ була попередньо

навчена виконувати подібне завдання на іншому, як правило, великому наборі даних.

Популярним підходом є попереднє навчання таких архітектур на задачі класифікації тисячі різних об'єктів (Imagenet1K), тому можна допустити, що вони вже мають певні знання про об'єкти зовнішнього світу, на яких вони були навчені. Як правило, застосування вибраних архітектур ЗГНМ в режимі передачі знань пришвидшує процес їх навчання та може значно зменшити похибку детекції і класифікації.

Існують архітектури ЗГНМ, що вирішують задачі детекції елементів на зображенні з малою похибкою. Такі архітектури знаходяться у відкритому доступі та їх використання є дозволеним і безкоштовним. Їх доцільно вибирати з урахуванням таких характеристик, як розмір вхідного зображення, швидкість роботи архітектури та похибка виявлення елементів. Так як ОІ-ХВ зазвичай займає 30%-40% розміру літального апарату, для виявлення такого об'єкту доцільно застосувати архітектуру ЗГНМ з невеликим розміром вхідного зображення, наприклад 128x128 або 256x256 пікселів. Стиснення зображення до вищезгаданої розмірності не вплине на збільшення похибки, адже ОІ-ХВ є великим об'єктом (див. рисунок 1.4). Детекція на таких розмірах вхідного зображення здійснюється швидко, адже НМ необхідно опрацювати меншу кількість пікселів. Модель буде перенавчатися для виявлення ОІ-ХВ шляхом визначення координати чотирьох точок прямокутника.

Модуль М1 повертає координати, за якими створюється набір даних ОІ-ХВ. ОІ-ХВ вирізається із зображення вхідного набору даних по координатах, отриманих в результаті виконання модуля М1. Тоді для кожного зображення ОІ-ХВ доцільно знову формувати файл (анотацію) формату xml, у якому міститься інформація про координати прямокутника з обмежувальною рамкою ОІ-БН, що буде використано модулем М2.

Модуль М2 – це детектор ОІ-БН. Процес детекції ОІ-БН подібний до процесу, описаного для модуля М1. Однак, для модуля М2 необхідно використати інші характеристики архітектури ЗГНМ, адже розмір ОІ-БН є невеликим відносно ОІ-

										ДП.КН.8091602.077.ПЗ	Арк.
											26
Змн.	Арк.	№ докум.	Підпис	Дата							

ХВ (див. рисунок 1.4). Стиснене зображення у розмір 128x128 пікселів втрачає деталізацію об'єктів, тому архітектури ЗГНМ можуть не розпізнати ОІ-БН на такому зображенні. Тому для детекції ОІ-БН необхідно обрати таку архітектуру ЗГНМ, розмір вхідного зображення якої є великим, наприклад 512x512 або 1024x1024 пікселів. Потрібно зауважити, що через велику кількість пікселів на зображенні процес детекції за допомогою модуля М2 триватиме довше, аніж за допомогою модуля М1, проте це забезпечить вищу точність детекції такої невеликої ОІ-БН. Для модуля М2 також доцільно застосувати режим передачі знань.

Модуль М2 повертає координати, за якими створюється набір даних ОІ-БН. ОІ-БН вирізається із зображення набору даних ОІ-ХВ по координатах, отриманих в результаті виконання модуля М2. Набір даних ОІ-БН – це зображення максимального наближення БН на корпусі літального апарату.

Модуль М3, як було описано раніше, це готовий програмний модуль, задачею якого є представити бортовий номер літального апарату у форматі кодування ASCII (у формі звичайної цифрової інформації).

Отже, після проходження процесу навчання, доцільно об'єднати модулі М1, М2 та М3 в режимі детекції, згідно з послідовністю, яка зображена на діаграмі потоків даних на рисунку 2.2.

На діаграмі користувач розглядається як зовнішня сутність, що взаємодіє з програмною системою ідентифікації літальних апаратів шляхом ініціалізації запиту та отримання відповіді, що позначає початок та завершення роботи системи. Запит користувача – це зображення літального апарату, який необхідно розпізнати.

Вхідне зображення зберігається у сховищі зображень літального апарату D1, після чого також зберігається у сховищі опрацьованих зображень D5 та передається на вхід модуля М1 – детектора ОІ-ХВ. Сховище зображень D5 використовується для зберігання всіх опрацьованих зображень, в подальшому для проведення додаткових навчань модулів М1 та М2. Модуль М1 виконує детекцію на вхідному зображенні. Результатом детекції є зображення ОІ-ХВ, яке зберігається у сховищі D2 для подальшої обробки. Наступним кроком є передача даних, а саме зображення

					ДП.КН.8091602.077.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		27

Для реалізації модулів M1 та M2 доцільно використати згорткову нейронну мережу YOLOv7.

2.2 Модель згорткової нейронної мережі YOLO

Модель YOLO (англійською You Only Look Once – дивишся лише раз) версії v7 є останнім розробленим варіантом моделі у наборі моделей YOLO, які є одноетапними детекторами об'єктів.

На початку існування моделі YOLO вона набула широкої популярності в спільнотах комп'ютерного зору та машинного навчання як надійне рішення для завдань виявлення об'єктів. Компактний розмір, гнучкість і можливість навчання на одному графічному процесорі відрізняють її від масивних архітектур, створених у відомих технічних лабораторіях. Хоча останні демонструють надзвичайну ефективність, вони часто створюють проблеми, коли справа доходить до їх роботи на споживчому обладнанні. Швидкість детекції в режимі реального часу моделей YOLO пропонує перевагу оптимального розподілу апаратних ресурсів, що дозволяє на практиці досягати відмінної продуктивності з мінімальними вимогами до обчислювальної потужності.

У рамках алгоритму YOLO, кадри вхідних зображень проходять процес знаходження основних ознак (features) зображення за допомогою потужної НМ [15].

Ознаки зображення – це певні характеристики або форми окремих частин елементів зображення, які фіксують важливу інформацію про його вміст. Ознаки зображення можна знайти на різних рівнях деталізації, починаючи від низькорівневих ознак, які фіксують основні візуальні атрибути, до високорівневих ознак, які представляють більш складні та абстрактні поняття.

У додатку А представлено узагальнену структуру моделі згорткової нейронної мережі YOLO, що складається з трьох ключових компонентів, які забезпечують його функціональність та ефективність:

- сенсорний компонент;

					ДП.КН.8091602.077.ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

- комунікаційний компонент;
- керуючий компонент.

Сенсорний компонент детектора YOLO являє собою ЗГНМ, відповідальну за об'єднання пікселів зображення для створення ознак на різних рівнях деталізації.

Зазвичай він складається з кількох згорткових шарів і попередньо навчений на великому наборі даних (наприклад, ImageNet1K). Згортковий рівень є основним блоком ЗГНМ. Він виконує операцію згортки, тобто математичну операцію, яка застосовує набір фільтрів до вхідного зображення. Метою згорткового рівня є виділення релевантних ознак і фіксація просторових зв'язків у даних. Згортка, в контексті ЗГНМ, відноситься до поелементного множення та підсумовування вхідних даних і набору навчальних фільтрів (також відомих як вагові коефіцієнти). Фільтри – це невеликі (3x3) матриці, які «ковзають» або «згортаються» по вхідному зображенню, обчислюючи скалярний добуток у кожному пікселі вхідного зображення. На рисунку 2.3 зображено приклад процесу згортки. Значення x_0 до x_{15} – це цифрові значення кожного з пікселів зображення розміром 4x4, значення w_0 до w_8 – це вагові коефіцієнти фільтру розміром 3x3, значення y_0 до y_3 – це матриця-результат згортки розміром 2x2. Таким чином, вхідне зображення розміром 4x4 зменшилось у розмір 2x2 та містить у собі важливі ознаки зображення [15].

Процес згортки ефективно сканує вхідне зображення та створює карту ознак, підсвічуючи конкретні шаблони або ознаки, наявні у вхідних даних. Операція згортки включає три основні кроки:

- множення – фільтр застосовується до невеликої кількості пікселів ділянки вхідного зображення шляхом поелементного множення відповідних елементів фільтра та цифрового значення пікселя;
- підсумовування – результати поелементного множення підсумовуються, утворюючи одне значення. Це значення представляє активацію певної ознаки в цьому конкретному місці;
- крок і відступ – фільтр «ковзає» по вхідному зображенню з певним кроком, переміщуючи певну кількість пікселів на кожному кроці. Крім того, до

					ДП.КН.8091602.077.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		30

Поєднання ознак з різних шарів сенсорного компоненту дозволяє багатомасштабний аналіз і дає змогу мережі захоплювати об'єкти різного розміру та складності.

Похибка навчання ЗГНМ YOLO обчислюється згідно з виразом [15]:

$$L = -[y * \log(p) + (1 - y) * \log(1 - p)], \quad (2.1)$$

де L – похибка перехресної ентропії;

y – справжня мітка класу (0 або 1);

p – прогнозована ймовірність позитивного класу.

Для багатокласової класифікації використовується інша формула обчислення похибки перехресної ентропії. Під час фази навчання моделі метою є мінімізація втрати перехресної ентропії шляхом коригування параметрів моделі, за допомогою методів оптимізації таких як градієнтне сходження.

Похибка детекції (похибка $L1$) використовується для вимірювання розбіжності між прогнозованими та реальними координатами ОІ. Похибка $L1$, також відома як середня абсолютна помилка (mean absolute error, MAE), обчислює середню абсолютну різницю між прогнозованими та реальними значеннями. Похибка $L1$ у YOLO обчислюються за наступною формулою [15]:

$$L1 = \alpha_k * \sum [1_{об.} * (|x - x_p| + |y - y_p| + |w - w_p| + |h - h_p|)], \quad (2.2)$$

де $L1$ – похибка детекції;

α_k – це ваговий коефіцієнт, який врівноважує вплив регресійної похибки детекції з іншими компонентами загальної функції похибки;

$1_{об.}$ – функція-індикатор, яка дорівнює 1, якщо об'єкт присутній в ОІ, і 0 в іншому випадку;

x, y, w, h – представляють передбачувані координати ОІ (координати центру, ширина та висота);

x_p, y_p, w_p, h_p – представляють реальні координати ОІ на зображенні.

										ДП.КН.8091602.077.ПЗ	Арк.
											32
Змн.	Арк.	№ докум.	Підпис	Дата							

Похибка $L1$ заохочує модель мінімізувати абсолютні відмінності між передбачуваними координатами ОІ та реальними координатами для об'єктів, присутніх на зображенні. Завдяки мінімізації похибки $L1$ модель вчиться точніше прогнозувати координати ОІ та покращує локалізацію об'єктів.

У YOLO похибка присутності об'єкта в ОІ обчислюється на основі передбаченої ймовірності його присутності в ОІ, а також реальної інформації про присутність цього об'єкта в цій же ОІ. Похибка присутності об'єкта в ОІ обчислюється за допомогою бінарної похибки перехресної ентропії, яка порівнює прогнозовану ймовірність присутності з істинною, що вказує, присутній об'єкт у області зображення чи ні.

Разом три компоненти: сенсорний, комунікаційний та керуючий, утворюють інтегровану архітектуру, що дозволяє детектору YOLO ефективно виявляти та класифікувати об'єкти на зображеннях із точним прогнозуванням ОІ та надійними можливостями розпізнавання об'єктів.

2.3 Алгоритми ідентифікації на основі глибоких нейронних мереж

2.3.1 Алгоритм ППІ для ідентифікації літальних апаратів за допомогою глибоких нейронних мереж

Схема алгоритму ППІ для ідентифікації літальних апаратів за допомогою глибоких НМ представлена у додатку Б. Як зазначено у підрозділі 2.1, ініціювання ідентифікації літальних апаратів виконується користувачем-клієнтом шляхом надсилання запиту до системи.

Важливою особливістю структури прикладних програмних інтерфейсів є можливість формування запитів до системи з різних джерел; користувачем може бути будь-який програмний модуль, який має доступ до прикладного програмного інтерфейсу. Цим програмним модулем може бути веб-додаток, додаток, створений для операційної системи комп'ютера, або додаток для мобільного телефону. Таким чином, прикладний програмний інтерфейс забезпечує незалежність та мобільність, з можливістю надсилання запитів з будь-якої системи незалежно від її структури.

					ДП.КН.8091602.077.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

Початком роботи серверного компоненту ППІ для ідентифікації літальних апаратів у режимі детекції є отримання запиту від користувача-клієнта, який містить вхідне зображення літального апарату. Сервер зберігає вхідне зображення у сховищах D1 та D5 (див. рисунок 2.2) та викликає модуль M1 для здійснення детекції. Якщо детекція не була успішною, модуль M1 генерує повідомлення про помилку, яка завадила успішному виконанню, та передає це повідомлення процесу формування відповіді користувачу. Якщо модуль M1 успішно виконує детекцію, то зображення ОІ-ХВ переміщається до сховища D2. Таким ж чином працюють модулі M2 та M3, переміщаючи результати виконання у сховища D3 та D4 відповідно, або генеруючи відповідні повідомлення про помилку. Завершальним процесом роботи серверного компоненту ППІ для ідентифікації літальних апаратів є процес формування та надсилання відповіді користувачу.

2.3.2 Алгоритм навчання глибоких нейронних мереж модулів M1 та M2

Для навчання та оцінки ефективності моделі НМ використовуються навчальна, перевірна та тестова вибірки даних, що є підмножинами набору даних [17].

Навчальна вибірка – це частина всього набору даних, яка використовується для навчання моделі НМ. Вона містить велику кількість позначених прикладів, які модель використовує для вивчення закономірностей, зв'язків і функцій у даних.

Вибірка перевірки використовується для налаштування параметрів моделі під час процесу навчання та оцінки її продуктивності. Вона відокремлена від навчальної вибірки та допомагає налаштовувати гіперпараметри, приймати рішення щодо архітектури моделі, методів регуляризації або вибору функцій. Вибірка перевірки забезпечує неупереджену оцінку продуктивності моделі під час навчання.

Тестова вибірка використовується для оцінки кінцевої продуктивності навченої моделі. Це окрема та незалежна підмножина, яку модель не бачила під час навчання чи перевірки. Тестова вибірка забезпечує неупереджену оцінку здатності моделі прогнозувати нові дані. Це допомагає оцінити, наскільки добре модель

					ДП.КН.8091602.077.ПЗ	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

працюватиме в реальних умовах.

Процес створення цих вибірок зазвичай передбачає поділ вхідного набору даних. Типовий розподіл становить 70-80% для навчання, 10-15% для перевірки та 10-15% для тестування. Поділ повинен забезпечити репрезентативний розподіл даних у всіх наборах, гарантуючи, що модель навчається на різноманітних прикладах і може добре узагальнювати нові дані. Слід подбати про те, щоб вибірки були взаємовиключними, тобто жодна точка даних не з'являлася в більш ніж одній вибірці, щоб запобігти будь-якій формі витоку даних [17].

На рисунку 2.4 зображено алгоритм навчання детекторів ОІ-ХВ та ОІ-БН протягом однієї епохи. Через невелику кількість пам'яті у графічному процесорі, навчальна вибірка зображень зазвичай розділяється на частини, що називаються пакетами даних. Пакети даних можуть містити різну кількість зображень, адже у різних графічних процесорів різна кількість пам'яті. У випадку навчання YOLO кожен пакет даних містить вісім зображень. Епоха навчання завершується тоді, коли модель пройшла всі зображення у навчальній вибірці, а отже опрацювала всі пакети даних.

Алгоритм розпочинається з ініціації змінних, які контролюють кількість ітерацій у циклі. Змінна «i» представляє поточну ітерацію, «M» відображає кількість зображень у навчальній вибірці, а «N» вказує на кількість пакетів даних у навчальній вибірці.

Кожна ітерація циклу включає завантаження i-того пакету даних, нормалізацію пакету, обчислення YOLO у режимі прямого поширення інформації та обчислення похибки навчання. Потім відбувається модифікація вагових коефіцієнтів та порогів YOLO згідно з алгоритмом зворотнього поширення помилки.

Алгоритм зворотнього поширення помилки змінює вагові коефіцієнти НМ на основі обчислених градієнтів, щоб мінімізувати похибку між прогнозованими та фактичними результатами. Він відіграє вирішальну роль у процесі навчання НМ і дозволяє вивчати складні закономірності та робити точні прогнози.

					ДП.КН.8091602.077.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

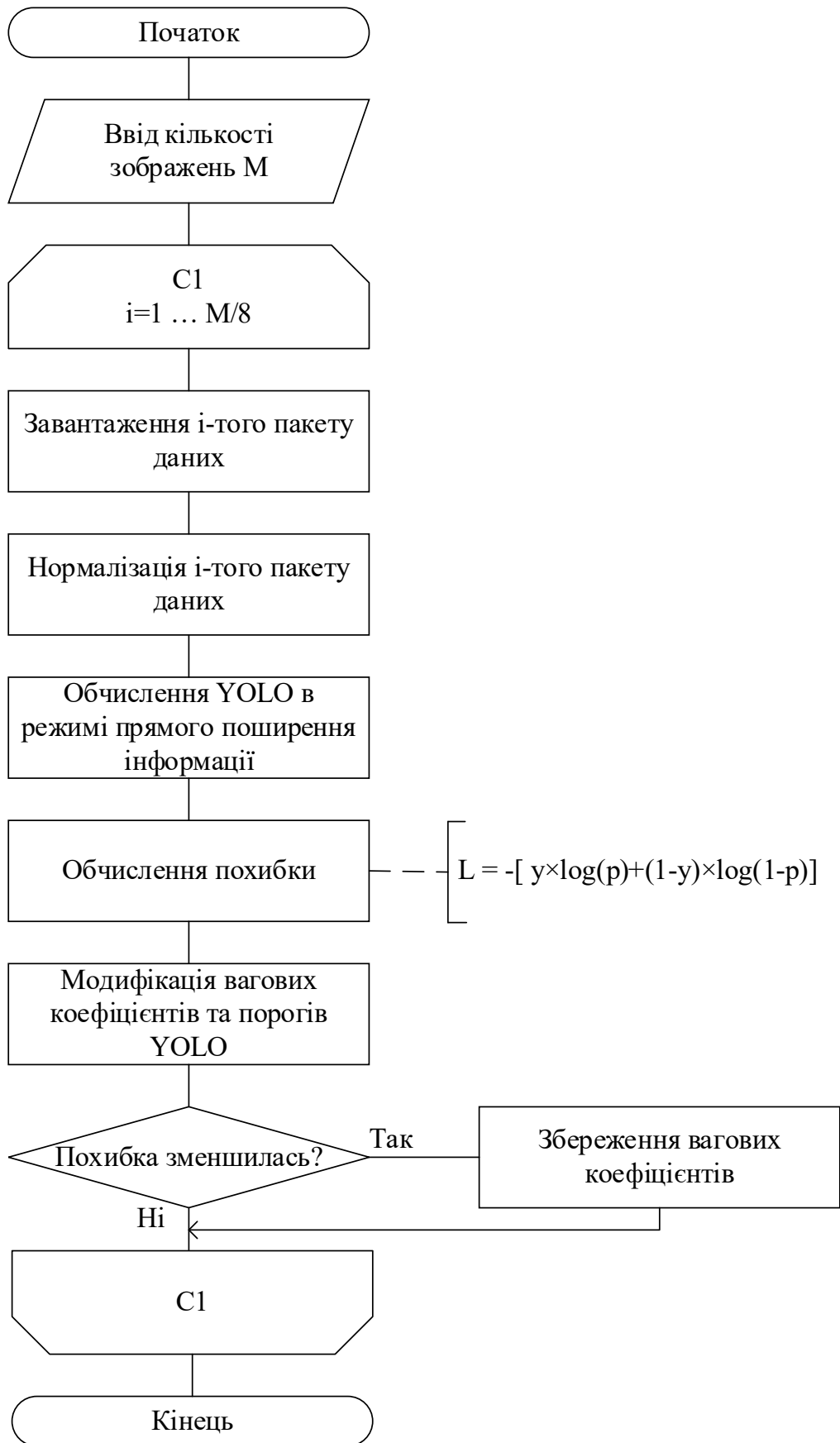


Рисунок 2.4 – Схема алгоритму навчання НМ модулів детекції протягом однієї епохи

На цьому завершується дія циклу та розпочинається нова ітерація до тих пір, поки всі пакети даних не будуть оброблені. Після завершення всіх ітерацій обчислюється похибка на вибірці перевірки, і вона порівнюється з похибкою, що була обчислена на попередній епосі. Якщо модель показала покращення та зменшила похибку, то вагові коефіцієнти моделі зберігаються, і модель переходить до наступної епохи.

Процес оцінки похибки на вибірці перевірки та порівняння її з попереднім результатом є важливим етапом в процесі навчання моделі. Він дозволяє визначити, чи вдається моделі покращувати свої прогнози та зменшувати похибку з кожною епохою.

2.4 Експериментальні дослідження модулів детекції

Для ефективного навчання модулів M1 та M2 необхідно зібрати значну кількість зображень літальних апаратів. Для цього використані доступні вибірки з набору даних з назвою «Точна візуальна класифікація літаків» (Fine-Grained Visual Classification of Aircraft, FGVC-Aircraft), який є набором для точної візуальної категоризації літаків та містить 10200 зображень літаків, по 100 зображень для кожного з 102 різних моделей літаків, більшість з яких є пасажирськими літаками [18]. З огляду на необхідність детекції ОІ-ХВ та ОІ-БН важливо зібрати зображення різних типів літаків та авіакомпаній, тому згаданий набір даних є відповідним для навчання модулів M1 та M2.

Відповідно до 2.1 для кожного зображення необхідно створити анотацію. Формат анотації відповідає стандарту YOLO, якщо використовує текстовий файл для збереження координат центру прямокутника на зображенні (x та y), а також ширини (w) та висоти (h) прямокутника. Для створення таких анотацій існують декілька спеціалізованих програм, таких як LabelImg, RectLabel, Labelbox, а також багато веб-орієнтованих рішень, таких як VGG Image Annotator (VIA), Roboflow та інші.

Для анотації було обрано 3143 зображення з вибірки FGVC-Aircraft та

					ДП.КН.8091602.077.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		37

розподілено:

- 2309 для вибірки навчання;
- 426 для вибірки перевірки;
- 408 для вибірки тестування.

Зображення у вибірці FGVC-Aircraft є довільного розміру, проте безпосередньо перед тренуванням вони будуть перетворені у розмір 320x320 пікселів для модуля M1 та 640x640 пікселів для модуля M2.

Зображення були проанотовані вручну за допомогою програмного інструменту LabelImg [19], інтерфейс користувача якого зображений на рисунку 2.5.



Рисунок 2.5 – Інтерфейс користувача додатка LabelImg

Згідно з 2.3.2 важливо забезпечити взаємовиключення між вибірками, щоб жодне зображення літального апарату не з'являлося в більш, ніж одній вибірці, тим самим запобігаючи будь-якому витоку даних. Варто відмітити, що також має значення розподіл літальних апаратів однакової моделі та авіакомпанії для того, щоб уникнути перенавчання (overfitting) моделі на один тип літака або авіакомпанію.

Основні характеристики перенавчання моделі включають надмірно низьку похибку навчання (похибка перехресної ентропії або похибка $L1$), але високу

									Арк.
									38
Змн.	Арк.	№ докум.	Підпис	Дата					

похибку верифікації або тестування. Модель «запам'ятовує» навчальні дані замість того, щоб вивчати шаблони, які можна узагальнити. Запобігання перенавчання допомагає зберегти універсальність моделі та уникнути недооцінки інших типів літаків та авіакомпаній.

Навчання модулів M1 та M2 було здійснено на персональному комп'ютері, який містить процесор AMD Ryzen 7 5800H з 32ГБ оперативної пам'яті. Головним фактором для навчання НМ є присутність графічного процесору у системі. ПК містить Nvidia GeForce RTX 3060 laptop GPU з 6 ГБ пам'яті. Таких характеристик ПК достатньо для відносно швидкого навчання модулів та в подальшому обслуговування серверного компоненту ППІ для ідентифікації літальних апаратів за допомогою глибоких НМ у режимі детекції.

2.4.1 Навчання та тестування модуля M1

Для навчання моделі глибокої ЗГНМ для модуля M1 було використано 2309 анотованих зображень. Кожна анотація містила інформацію про прямокутник OI-XB на зображенні. Процес навчання моделі глибокої ЗГНМ для модуля M1 тривав 26 епох, що тривало дві години на вищезгаданому ПК. Після кожної епохи вираховувалась похибка $L1$ та похибка присутності об'єкта в OI. Ці похибки вираховувалися для вибірки тренування та перевірки. Графік зміни похибки $L1$ для вибірки тренування та перевірки зображено на рисунку 2.6.

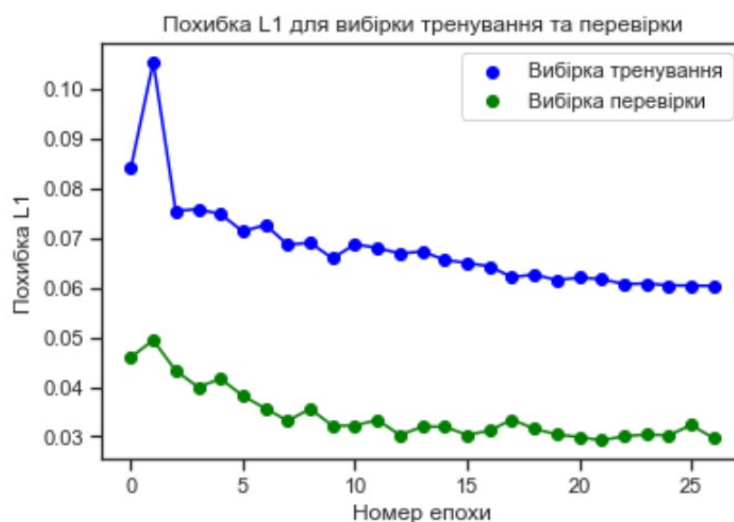


Рисунок 2.6 – Графік зміни похибки $L1$ для вибірки тренування та перевірки

					ДП.КН.8091602.077.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		39

Показники похибки $L1$ знижуються, що означає покращення моделі глибокої ЗГНМ для модуля M1.

На рисунку 2.7 зображено графік зміни похибки присутності об'єкта в ОІ для вибірки тренування та перевірки. Як видно, обидві похибки зменшуються впродовж тренування як для вибірки тренування, так і для вибірки перевірки. В результаті модель глибокої ЗГНМ для модуля M1 покращується з кожною епохою навчання. Зменшення похибок для вибірки перевірки означає, що модель не є перенавчена до даних в тренувальній вибірці, та що модель буде адекватно працювати на нових даних з реального світу.

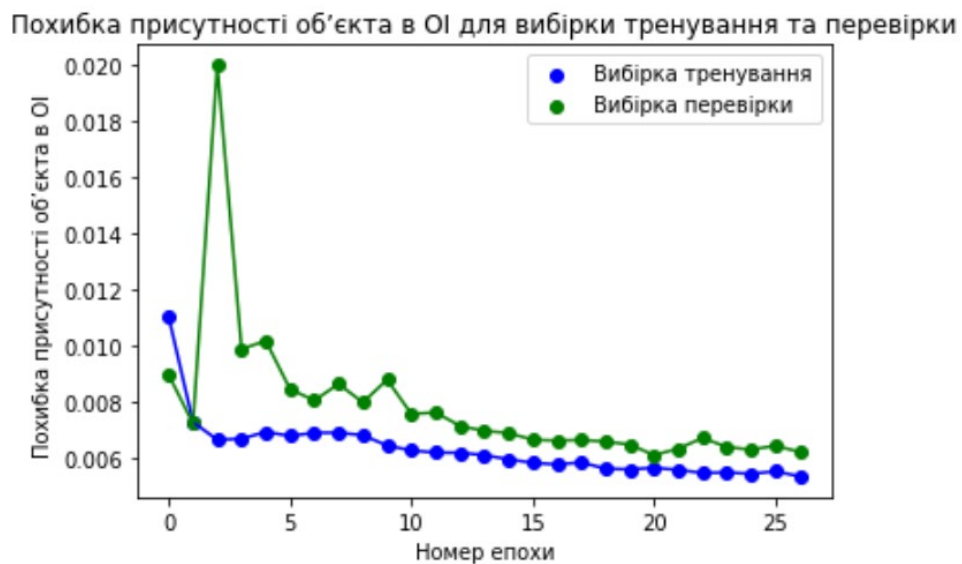


Рисунок 2.7 – Графік зміни похибки присутності об'єкта в ОІ для вибірки тренування та перевірки

Після кожної епохи навчання моделі глибокої ЗГНМ для модуля M1 також обчислюється похибка середньої точності (mean Average Precision, mAP) на вибірці перевірки. Похибка mAP використовується для вимірювання точності та ефективності моделі при виявленні об'єктів на зображенні. Вона кількісно визначає, наскільки добре модель локалізує та ідентифікує об'єкти на основі точності та запам'ятовування виявлених обмежувальних рамок. Вища mAP вказує на кращу продуктивність виявлення об'єктів із значеннями, як правило, в діапазоні від 0 до 1. Це забезпечує показник здатності моделі точно виявляти об'єкти в різних

тренування, так і для вибірки перевірки, тому модель глибокої ЗГНМ для модуля М2 покращується з кожною епохою навчання.

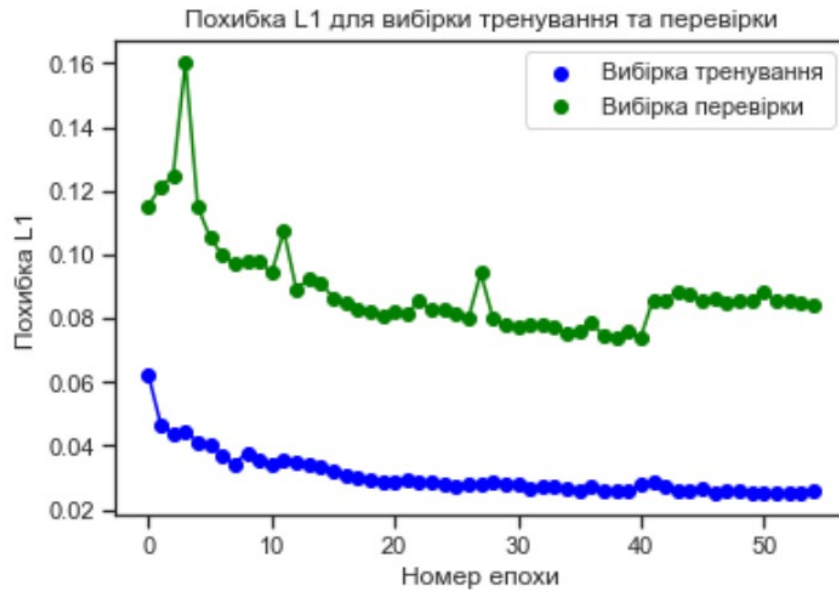


Рисунок 2.9 – Графік зміни похибки $L1$ для вибірки тренування та перевірки

Так само, як і для моделі з модуля М1, зменшення похибок для вибірки перевірки означає, що модель глибокої ЗГНМ для модуля М2 не є перенавчена до даних в тренувальній вибірці, та що модель буде працювати на нових даних з реального світу.

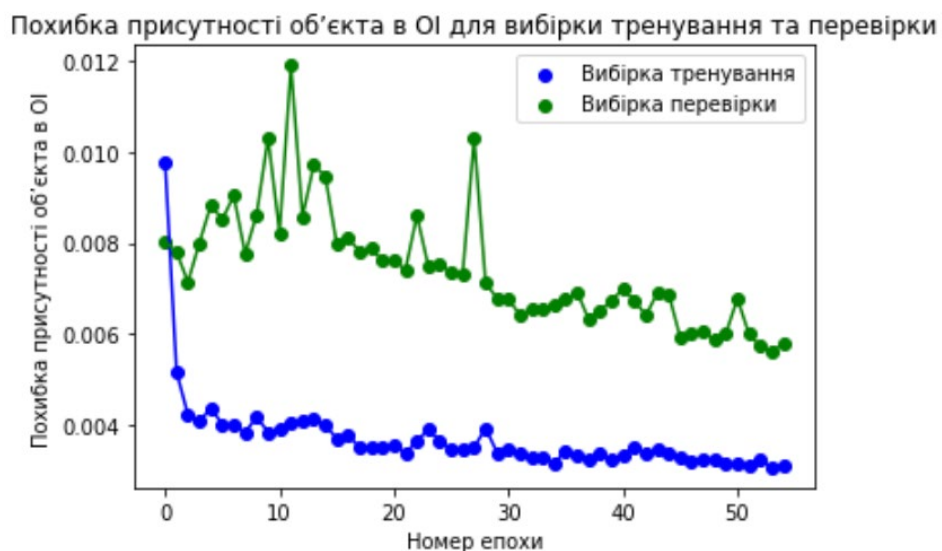


Рисунок 2.10 – Графік зміни похибки об'єктності для вибірки тренування та перевірки



Рисунок 2.12 – Зображення літальних апаратів для тестування

Після проведення тестування були отримані результати, які відображені у таблиці 2.1.

											Арк.
											44
Змн.	Арк.	№ докум.	Підпис	Дата							

ДП.КН.8091602.077.ПЗ

Таблиця 2.1 – Результати тестування модуля МЗ

Вхідне зображення для ОІ-БН	Результат	Правильно / Ні
	PH-HJ	Ні
	PH-RRT	Правильно
	9a-ctM	Правильно
	+oh-Lei	Ні
	F-Gugp	Правильно
	I-RIZA	Правильно
	EC-BIQ	Правильно
	G-FLTC	Правильно
	YU-AOK	Правильно
	VR-CYB	Правильно

З десяти зображень ОІ-БН, що були подані на обробку модулю МЗ, вдалося успішно класифікувати вісім зображень. Проте одне з некласифікованих зображень містить символ «+» на початку БН, оскільки система ОРС помилково ідентифікувала прапор Фінляндії як математичний символ «плюс».

Щодо другого некласифікованого зображення, воно містить більшість правильних символів, але система ОРС не змогла розпізнати символи «VJ».

Отже, проведене тестування продемонструвало, що точність модуля МЗ становить 80 %.

Клас Errors розроблений з метою відстеження помилок у програмній системі під час процесу ідентифікації літального апарату. Він допомагає локалізувати фрагмент коду, де відбулась помилка, та створити відповідь з відповідним повідомленням для користувача. Клас містить лише один атрибут `errorcode`, метод ініціалізації та метод `raiseError()`.

Клас `OCRdetector` представляє абстрактну сутність моделі модуля М3. Він відповідає за завантаження програмного модуля `EasyOCR` та містить атрибути для збереження результату класифікації. Клас містить метод ініціалізації, метод `predict()` для здійснення класифікації та метод `get_results()` для виведення результатів класифікації.

Принцип роботи прикладних програмних інтерфейсів є універсальним для більшості мов програмування та систем [29]. ППІ працює у двох різних станах: стан підготовки та стан обробки запиту. У стані підготовки ППІ налаштовує свої внутрішні компоненти, конфігурації та ресурси, гарантуючи, що все готово до обробки вхідних запитів. Після завершення стану підготовки ППІ переходить у стан обробки запитів, де він очікує на вхідні запити від користувачів або клієнтських програм. У цьому стані ППІ прослуховує вхідні запити, отримує та аналізує запити та виконує відповідну логіку для їх обробки. Маючи ці два стани, ППІ забезпечує спрощений і ефективний робочий процес. Стан підготовки дозволяє виконувати будь-які необхідні завдання ініціалізації та налаштування один раз, зменшуючи надлишкові операції для кожного вхідного запиту. Перебуваючи в стані обробки запитів, ППІ може зосередитися виключно на обробці запитів користувачів, максимізуючи продуктивність і швидкість реагування [30].

Створення класів для серверного компоненту ППІ було виконано відповідно до критеріїв, описаних вище. Розподіл методів підготовки від методів реальної детекції забезпечив зменшення часу виконання запиту клієнта від 30 секунд до 0.8 секунди. Причиною такої великої різниці є те, що моделі модулів М1, М2 та М3 є великими та вимагають багато часу для їх завантаження з файлової системи. Розділивши процес на два етапи, у разі надходження запиту до серверної частини

					ДП.КН.8091602.077.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		48

ППП, необхідно лише провести детекцію, що займає мало часу.

Для реалізації серверного компоненту прикладного програмного інтерфейсу для ідентифікації літальних апаратів за допомогою глибоких НМ у режимі детекції було використано наступні програмні середовища та мову програмування:

- Anaconda;
- Microsoft Visual Studio Code;
- Python.

Anaconda – це система розповсюдження та керування програмним забезпеченням, призначена для завдань з обробки даних і машинного навчання [20]. Середовище надає повний набір програмних пакетів та бібліотек із відкритим вихідним кодом та інструментів, які полегшують розробку, розгортання та керування проектами з обробки даних. Одна з переваг програмного середовища є створення віртуальних середовищ. Це дозволяє ефективно керувати всіма встановленими програмними бібліотеками та сортувати їх в залежності від проекту. Для кожного типу проекту створюється своє віртуальне середовище, що зменшує ймовірність виникнення помилок сумісності програмних пакетів у випадку збільшення їхньої кількості в одному віртуальному середовищі.

Microsoft Visual Studio Code (VS Code) – популярний редактор програмного коду, розроблений компанією Microsoft. Це редактор, який підтримує широкий спектр мов програмування та надає багатий набір функцій для редагування коду, налагодження та контролю версій програмних продуктів [21].

Для реалізації серверного компоненту ППП було використано мову програмування Python [22]. Python широко використовується для створення прикладних програмних інтерфейсів та моделювання моделей машинного навчання завдяки:

- легкості написання та розуміння – Python має простий і зрозумілий синтаксис написання коду;
- великому набору модулів та бібліотек – ці модулі та бібліотеки, спеціально розроблені для веб-дизайну, створення прикладних програмних інтерфейсів та створення базових моделей машинного навчання;

					ДП.КН.8091602.077.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		49

- можливості інтеграції – Python легко інтегрується з іншими мовами програмування та системами, що робить його чудовим вибором для створення ППІ, які взаємодіють з різними технологіями;

- наявності покращених бібліотек для машинного навчання – Python став де-факто мовою математичного моделювання даних та машинного навчання. Він пропонує найбільший вибір бібліотек та інструментів для обробки даних, візуалізації, статистичного аналізу та машинного навчання [27, 28]. Зокрема бібліотека PyTorch використовується як в режимі навчання, так і в режимі детекції при розробці даного прикладного програмного інтерфейсу для ідентифікації літальних апаратів. Для навчання моделей ЗГНМ для модулів М1 та М2 було використано низку програмних продуктів, що включають у себе розширену бібліотеку PyTorch.

PyTorch є однією з ключових бібліотек, яка забезпечує широкий спектр функціональності для тренування та застосування моделей ЗГНМ, зокрема моделі YOLOv7. Функції пакету включають виконання операцій глибинного навчання, автоматичне диференціювання, оптимізацію градієнта, обробку даних, інтеграцію з іншими бібліотеками, інтуїтивний та гнучкий інтерфейс програмування для зручного тренування моделей та ефективного використання отриманих результатів [23].

Для реалізації серверного компонента ППІ в режимі детекції програмний код було організовано у п'ять класів. У додатку В представлено код програми серверного компонента ППІ.

3.2 Інтерфейс користувача

Інтерфейс користувача ППІ складається з двох частин. Інтерфейс користувача серверної частини ППІ буде використовуватися лише для запуску сервера, тому він не містить ніякого функціоналу. Він представлений у вигляді командної стрічки, адже ніхто окрім адміністратора сервера не може мати доступ до серверної частини. Для запуску серверної частини ППІ необхідно прописати

					ДП.КН.8091602.077.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		50

лише одну команду: `uvicorn api:app --reload`. Ця команда запускає сервер та ініціалізує процес підготовки сервера (завантаження моделей). Коли сервер запустився, він очікує запити від користувачів.

На рисунку 3.2 зображено повідомлення у командній стрічці при успішному запуску серверної частини ППІ.

```
PS C:\Users\vtu20\Projects\Plane_Recon_v6\pipeline> uvicorn api:app --reload
INFO: Will watch for changes in these directories: ['C:\\Users\\vtu20\\Projects\\Plane_Recon_v6\\pipeline']
INFO: Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO: Started reloader process [31044] using WatchFiles
YOLOR 2023-5-8 torch 2.0.0+cu117 CUDA:0 (NVIDIA GeForce RTX 3060 Laptop GPU, 6143.5MB)
Model: Tail detection model is loaded.
YOLOR 2023-5-8 torch 2.0.0+cu117 CUDA:0 (NVIDIA GeForce RTX 3060 Laptop GPU, 6143.5MB)
Model: Tail number detection model is loaded.
INFO: Started server process [10776]
INFO: Waiting for application startup.
INFO: Application startup complete.
█
```

Рисунок 3.2 – Повідомлення про успішний запуск серверної частини ППІ

Інтерфейс користувача клієнта працює з використанням документаційної сторінки програмної бібліотеки для створення ППІ FastAPI [24]. Зазвичай, клієнти ППІ – це веб-орієнтовані сторінки, які використовують безліч різних ППІ для надання користувачам веб-орієнтованих сторінок необхідних послуг. Такі сторінки можуть розроблятися на будь-якій мові програмування та використовувати будь-які програмні бібліотеки, адже комунікація з ППІ відбувається за допомогою визначених стандартів. Формат JSON використовується як формат даних для надсилання та отримання даних між клієнтом, наприклад, веб-браузером, мобільним додатком, і сервером. Це дозволяє будь-якому програмному продукту взаємодіяти з ППІ без необхідності знати деталі його внутрішньої реалізації. Для симуляції поведінки клієнта FastAPI пропонує інтерфейс можливих запитів до ППІ за відповідною веб-адресою.

На рисунку 3.2 третій рядок повідомлення вказує локальну адресу, за якою можна надсилати запити до ППІ. Для прикладного програмного інтерфейсу, що проводить ідентифікацію літальних апаратів, запити можна надсилати на адресу `http://127.0.0.1:8000/`, де 8000 після двокрапки – це номер порту. Таким чином, якщо розмістити серверну частину та дозволити приймати запити із зовнішньої мережі,

									Арк.
									51
Змн.	Арк.	№ докум.	Підпис	Дата					

користувачі можуть надсилати запити з будь-якого браузера у будь-якій точці світу.

Перейшовши за веб-адресою розміщення ППІ для ідентифікації літальних апаратів та добавивши вкінці символи «docs#», користувач буде перенаправлений веб-браузером на сторінку, зображену на рисунку 3.3. Варто зазначити, що це працює тільки з пристроїв локальної мережі.

FastAPI 0.1.0 OAS3

/openapi.json

default

POST /getnumber Get Number

Schemas

Body_get_number_getnumber_post >

HTTPValidationError >

ValidationError >

Рисунок 3.3 – Сторінка документації ППІ

На веб-сторінці зображені всі запити, які може приймати ППІ за вказаною веб-адресою. Запит POST є одним із методів HTTP, який використовується для надсилання даних на сервер. Під час виконання запиту POST дані включаються в тіло запиту, а не в саму URL-адресу. Це дозволяє надсилати більший обсяг даних та зберігати їх у безпеці, оскільки вони не містяться безпосередньо в URL-адресі.

Запит /getnumber – це запит, що приймає серверний компонент прикладного програмного інтерфейсу для ідентифікації літальних апаратів. Якщо ППІ приймає

									Арк.
									52
Змн.	Арк.	№ докум.	Підпис	Дата					

ДП.КН.8091602.077.ПЗ

й інші запити, то це буде відображено на веб-сторінці документації. Розгорнувши спадний список запиту /getnumber, веб-сторінка відображає деталі форматування запиту та можливість протестувати процес надсилання запиту, що зображено на рисунку 3.4.

Для того, щоб надіслати запит і отримати бажаний результат, серверна частина повинна бути готова до приймання запитів від користувачів.

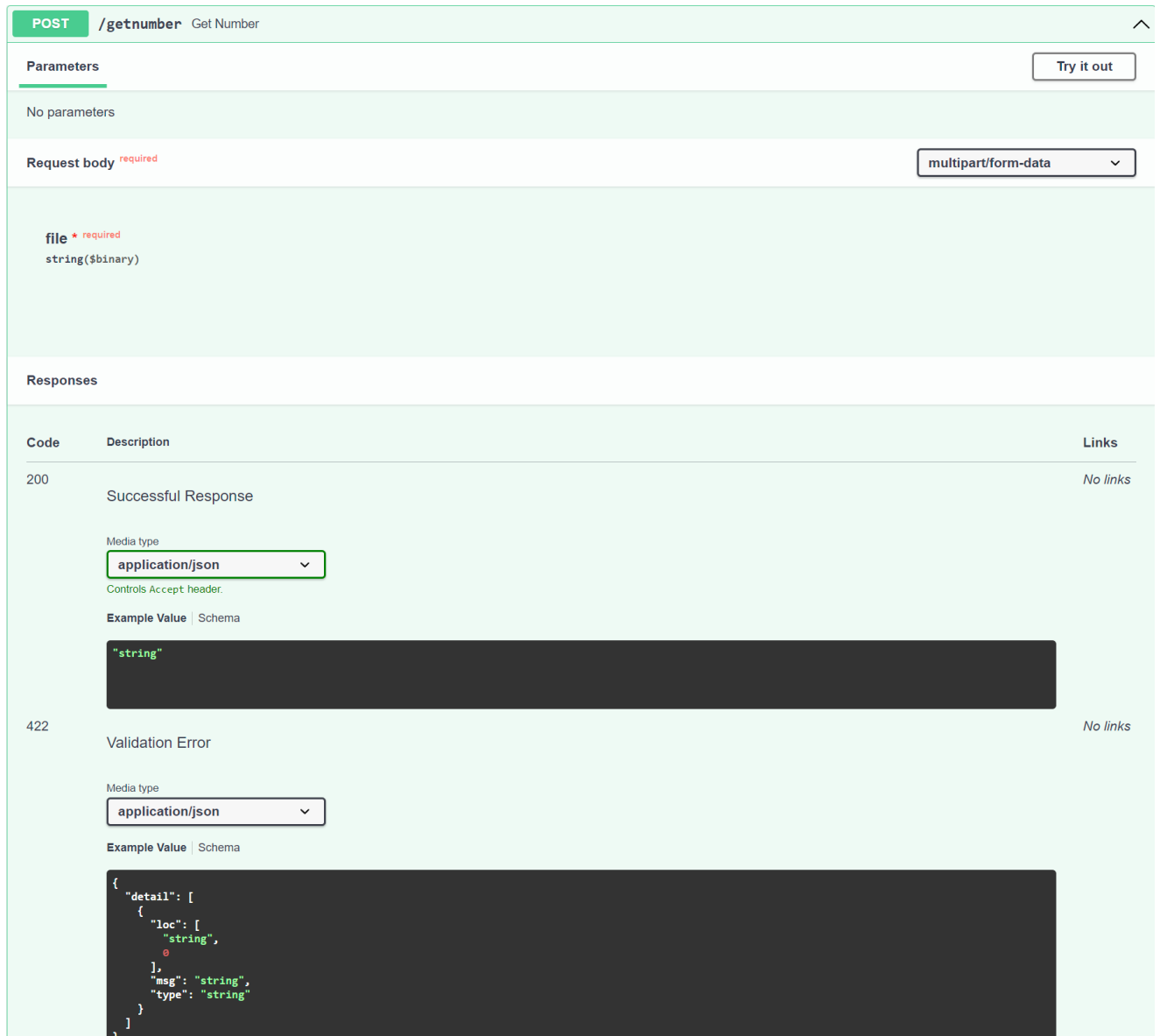


Рисунок 3.4 – Параметри спадного списку запиту /getnumber

Для надсилання запитів за допомогою сторінки документації необхідно натиснути на кнопку «Choose file» та вибрати зображення у меню файлового провідника (рисунок 3.5). Обрано зображення «1296245.jpg» (рисунок 3.6).

									Арк.
									53
Змн.	Арк.	№ докум.	Підпис	Дата	ДП.КН.8091602.077.ПЗ				

Натиснувши кнопку «Open» з рисунку 3.5, зображення завантажувється на сторінку документації ППІ для ідентифікації літальних апаратів.

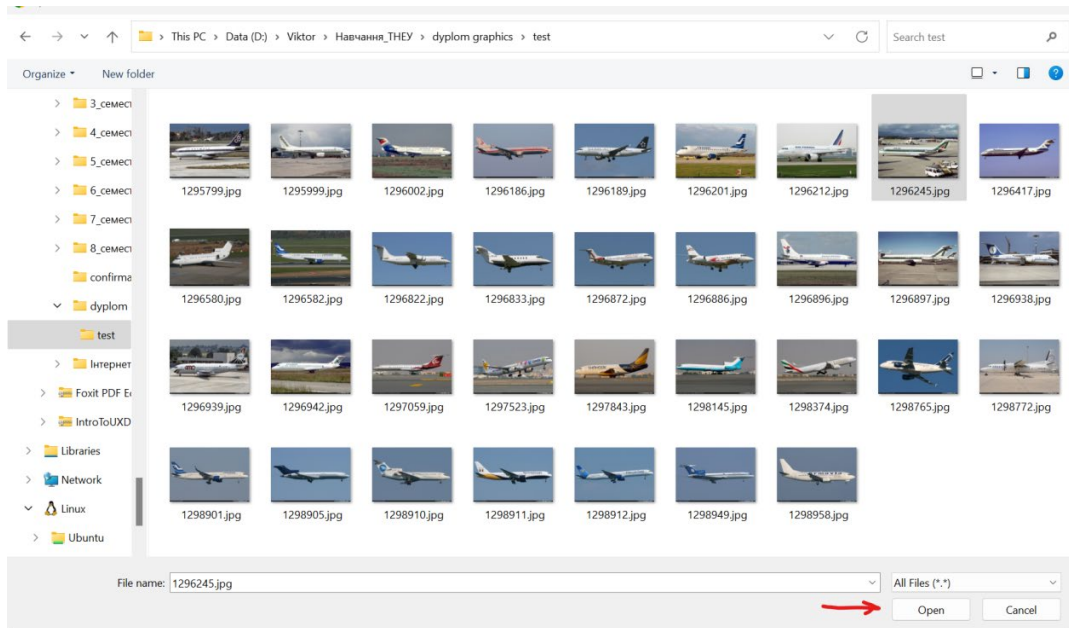


Рисунок 3.5 – Меню файлового провідника Windows



Рисунок 3.6 – Зображення «1296245.jpg»

На рисунку 3.7 представлено результат завантаження зображення «1296245.jpg». Натиснувши кнопку «Execute», запит із вибраним зображенням буде надіслано до серверного компоненту прикладного програмного інтерфейсу. Після отримання запиту від користувача інтерфейс користувача серверної частини

									Арк.
									54
Змн.	Арк.	№ докум.	Підпис	Дата	ДП.КН.8091602.077.ПЗ				

ППП виводить повідомлення про отриманий запит та інформацію про виконану детекцію.

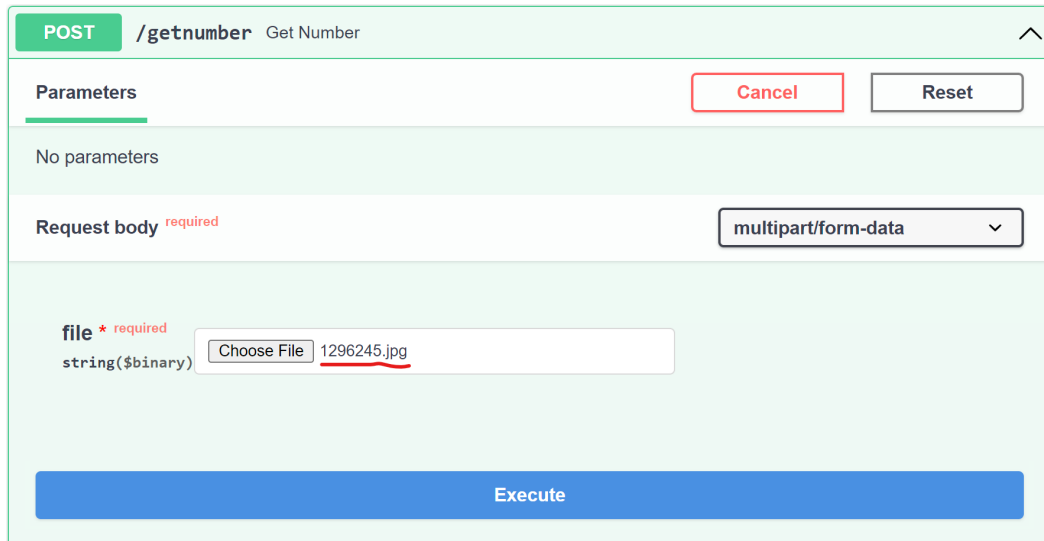


Рисунок 3.7 – Результат завантаження зображення «1296245.jpg»

На рисунку 3.8 зображено повідомлення від серверного компонента про успішну детекцію.

```
INFO: Application startup complete.
1 tail, Done. (13.8ms) Inference, (4.6ms) NMS
Done. (0.198s)
1 aircraft_number, Done. (12.4ms) Inference, (1.0ms) NMS
Done. (0.246s)
INFO: 127.0.0.1:60495 - "POST /getnumber HTTP/1.1" 200 OK
```

Рисунок 3.8 – Повідомлення від серверного компонента про успішну детекцію

У разі виникнення помилки під час детекції модулем M1, M2 або M3 буде виведено відповідне повідомлення. Після проведення детекції серверний компонент формує повідомлення для клієнта у форматі JSON та відправляє назад клієнту.

Повідомлення з рисунку 3.8 сповіщає, що було знайдено один хвіст літака за 0.198 секунди, один бортовий номер літака за 0.246 секунди та серверий компонент переходить у режим очікування запитів від клієнта.

На стороні клієнта веб-браузер отримав відповідь від сервера (рисунок 3.9).

```

Server response
Code      Details
-----
200      Response body
          {
            "aircraft-number": "I-RIZA",
            "time": 1.263047456741333
          }
          Response headers
          content-length: 53
          content-type: application/json
          date: Mon, 29 May 2023 02:54:55 GMT
          server: uvicorn

```

Рисунок 3.9 – Відповідь від серверного компоненту

Відповідь від сервера представлена у форматі JSON та містить два поля:

- «aircraft-number»: I-RIZA;
- «time»: 1.3.

Перше поле – це бортовий номер літального апарату, друге поле – час виконання запиту сервером у секундах.

Відповідь також містить інформацію про довжину повідомлення, формат відповіді, час та дату її отримання відповіді та назву програмного модуля, за допомогою якого здійснюється комунікація. Отримавши відповідь, клієнт успішно ідентифікував літальний апарат.

3.3 Тестування прикладного програмного інтерфейсу

Після завершення розробки ППІ для ідентифікації літальних апаратів, наступним етапом є проведення тестування. Тестування, як етап розробки ППІ, відіграє важливу роль у забезпеченні якості його роботи.

Одна з поширених помилок, виявлених при тестуванні, трапляється при надсиланні запиту до сервера, який не запущений, або не готовий приймати запити від користувачів. У випадку FastAPI, якщо сервер не знаходиться в робочому стані,

доступитися до веб-адреси сторінки документації буде неможливо. Якщо надіслати запит до ППІ, який працює в режимі офлайн, зазвичай буде отримано повідомлення про помилку, що сервер недоступний або що не вдалося встановити з'єднання. Конкретне повідомлення про помилку може відрізнятися залежно від програмної бібліотеки або інструменту клієнтської сторони, які використовуються для надсилання запиту. Важливо обробляти такі помилки в коді програми клієнта, щоб виявити їх та вивести відповідне повідомлення користувачеві, наприклад, що ППІ наразі недоступний, або запропонувати альтернативні дії.

Під час виконання детекції може виникнути багато помилок пов'язаних із завантаженням зображень, детекцією або комунікацією між модулями М1, М2 та М3. При виникненні помилки, яка може спричинити до припинення роботи ППІ, пріоритетним завданням є її виявлення. Після виявлення помилки ППІ повинен бути в змозі повернутись до стану нормальної роботи, забезпечуючи обробку запитів інших користувачів.

У таких ситуаціях важливо використовувати механізми обробки помилок, щоб знаходити та обробляти виняткові ситуації, що можуть виникнути під час виконання ідентифікації літальних апаратів. Механізми обробки помилок повинні включати функції здійснення необхідних корекцій, відновлення збережених станів, логування подій та повідомлень про помилку, а також інформування користувачів про стан системи. Такий підхід дозволяє забезпечити відновлення роботи ППІ після виникнення помилки та збереження продуктивності та доступності для інших користувачів.

Реалізація серверного компоненту прикладного програмного інтерфейсу для ідентифікації літальних апаратів за допомогою глибоких НМ у режимі детекції містить клас Errors, описаний у 3.1, який забезпечує виявлення помилок та формування відповідних повідомлень користувачу.

Для тестування ППІ потрібно розробити декілька сценаріїв поведінки користувача, які б симулювали його дії в реальному світі.

Першим сценарієм поведінки користувача є випадок, коли користувач надсилає зображення, на якому відсутній літальний апарат. На рисунку 3.10

					ДП.КН.8091602.077.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		57

представлено зображення автомобіля (car.jpg), яке буде надіслане до ППІ для тестування.



Рисунок 3.10 – Зображення автомобіля (car.jpg) [25]

У результаті надсилання такого запиту у полі «aircraft-number» повідомлення сервера у форматі JSON представлено причину виникнення помилки, а не бортовий номер. Знімок екрану сторінки документації, зображений на рисунку 3.11, представляє зміст поля «aircraft-number»: «No aircraft tail detected in the image received», що перекладається з англійської мови, як «На отриманому зображенні не виявлено хвоста літака».

Таким чином, коли клієнт (у даному випадку браузер) отримає наступне повідомлення від серверного компоненту ППІ, він може сповістити користувача про проблему, яка виникла при детекції. Коли сервер не виявить проблему під час виклику модуля М1 і безпосередньо після цього викличе модуль М2, останній не знайде відповідного зображення для детекції, адже модуль М1 не виявив хвостової частини літака. Це може бути причиною виникнення серйозних наслідків для роботи сервера. У такому сценарії сервер втратить здатність обробляти запити від інших користувачів, оскільки така помилка приведе до завершення його виконання.

Цей сценарій також може мати значний вплив на продуктивність та доступність сервера, оскільки його здатність приймати та обробляти запити буде тимчасово припинена. Як наслідок, інші користувачі не матимуть можливості звертатися до сервера для отримання необхідної інформації чи виконання запитів.

										Арк.
										58
Змн.	Арк.	№ докум.	Підпис	Дата						

Для запобігання таким ситуаціям та збереження неперервної роботи сервера важливо встановити механізми обробки помилок, які зможуть виявляти виняткові ситуації, такі як відсутність відповідного зображення для детекції, та приймати необхідні заходи для відновлення стабільної роботи сервера.

file * required
string(\$binary) Choose File car.jpg

Execute Clear

Responses

Curl

```
curl -X 'POST' \
'http://127.0.0.1:8000/getnumber' \
-H 'accept: application/json' \
-H 'Content-Type: multipart/form-data' \
-F 'file=@car.jpg;type=image/jpeg'
```

Request URL

```
http://127.0.0.1:8000/getnumber
```

Server response

Code	Details
200	<p>Response body</p> <pre>{ "aircraft-number": "No aircraft tail detected in the image received.", "time": 0.26307010650634766 }</pre>

Рисунок 3.11 – Відповідь від ППІ на запит користувача із зображенням car.jpg

Така ж помилка можлива з модулем М2 у випадку, коли бортовий номер на хвостовій частині літального апарату перекритий іншими елементами літака (рисунок 3.12).

На зображенні 3.12 видно хвостову частину літального апарату, проте неможливо визначити його бортовий номер через ракурс, з якого зображення було захоплене.



Рисунок 3.12 –Зображення літального апарату Aircraft-side.jpg

При надсиланні такого зображення сервер повертає відповідь, що представлена на рисунку 3.13. Зміст повідомлення: «На отриманому зображенні не виявлено бортового номера літака».

file * required
string(\$binary) Choose File Aircraft-side.jpg

Execute Clear

Responses

Curl

```
curl -X 'POST' \  
'http://127.0.0.1:8000/getnumber' \  
-H 'accept: application/json' \  
-H 'Content-Type: multipart/form-data' \  
-F 'file=@Aircraft-side.jpg;type=image/jpeg'
```

Request URL

```
http://127.0.0.1:8000/getnumber
```

Server response

Code	Details
200	<p>Response body</p> <pre>{ "aircraft-number": "No aircraft tail detected in the image received.", "time": 0.22068357467651367 }</pre>

Рисунок 3.13 – Відповідь від ППІ на запит користувача із зображенням Aircraft-side.jpg

Користувач клієнта, отримавши таке повідомлення, повинен змінити ракурс

									Арк.
									60
Змн.	Арк.	№ докум.	Підпис	Дата					

та захопити нове зображення літального апарату.

Подібним чином працює модуль МЗ, надсилаючи користувачу відповідні повідомлення про виявлені помилки.

Завдяки розробці алгоритмів ідентифікації помилок під час детекції модулями М1, М2, МЗ та інших дій на стороні сервера, створена надійна система, що забезпечує безперебійну роботу ППІ, навіть у випадках виникнення помилок.

Шляхом ретельного аналізу та розробки механізмів обробки помилок, серверна частина ППІ здатна виявляти та обробляти виняткові ситуації, що можуть виникнути під її час виконання. Наявність алгоритмів виявлення помилки та повідомлення про неї забезпечує зручну взаємодію з сервером та забезпечує високу продуктивність і надійність ППІ, що, в свою чергу, сприяє задоволенню потреб користувачів прикладного програмного інтерфейсу для ідентифікації літальних апаратів за допомогою глибоких НМ.

Результати тестування показали коректну роботу прикладного програмного інтерфейсу при ідентифікації літального апарату, а також відповідну поведінку системи при виникненні помилок.

					ДП.КН.8091602.077.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		61

ВИСНОВКИ

Під час виконання дипломного проекту:

1. Проаналізовано сучасні літальні апарати, особливості зовнішньої конструкції та позначення літака, як об'єкта дослідження у межах дипломного проекту. Встановлено, що за допомогою бортового номера можливо ідентифікувати повітряне судно та дізнатися інформацію критичної важливості.

2. Розглянуто підходи до ідентифікації літального апарату, здійснено аналіз існуючих методів ідентифікації, які використовують радары та системи навігації літальних апаратів, виявлено відсутність використання систем для візуальної ідентифікації літального апарату.

3. Розглянуто підходи до візуальної ідентифікації об'єктів в суміжних сферах, зокрема: автомобіля за його номерним знаком, наземних роботів та дронів при доставці кореспонденції і товарів, об'єктів дорожнього руху автопілотом автомобіля; проаналізовано задачу класифікації літер та цифр поштової кореспонденції, банківських чеків та розпізнавання тексту на зображеннях. Визначено альтернативний метод ідентифікації літального апарату за допомогою бортового номера з використанням глибоких нейронних мереж.

4. Розроблено структуру прикладного програмного інтерфейсу для ідентифікації літального апарату, що складається з трьох модулів, а саме: детекції ОІ-ХВ, детекції ОІ-БН та класифікації БН.

5. Для реалізації модулів детекції М1 та М2 використано згорткову нейронну мережу YOLOv7 та для М3 – готовий програмний модуль технології оптичного розпізнавання символів.

6. Розроблено алгоритм ППІ для ідентифікації літальних апаратів за допомогою глибоких нейронних мереж.

7. Розроблено алгоритм навчання глибоких нейронних мереж модулів детекції хвостової частини та бортового номера.

8. Обрано та проанотовано набір зображень літальних апаратів шляхом аналізу декількох існуючих вибірок, які містять літальні апарати, що дозволило

					ДП.КН.8091602.077.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		62

зібрати достатню кількість зображень для навчання моделей глибоких нейронних мереж, створених для ідентифікації визначених областей інтересу.

9. Перенавчено моделі YOLOv7 шляхом опису процесу перенавчання у програмній бібліотеці машинного навчання Pytorch , що дозволило провести серію експериментальних досліджень та визначити похибку ідентифікації на високошвидкісному графічному процесорі Nvidia GeForce RTX 3060.

10. Побудовано UML діаграму класів серверного компоненту ППІ для ідентифікації літальних апаратів за допомогою глибоких НМ у режимі детекції.

11. Здійснено програмну реалізацію з використанням Anaconda, Microsoft Visual Studio Code, Python. Реалізовано прикладний програмний інтерфейс, який містить модулі з навченими моделями, шляхом створення абстрактних класів у мові програмування Python, що дозволило об'єднати усі модулі у єдину структуру та дозволило прикладному програмному інтерфейсу отримувати запити на ідентифікацію літальних апаратів на зображеннях.

12. Розроблено інтерфейс користувача, який складається з клієнтської та серверної частини, та дає змогу відстежувати процес ідентифікації за заданим зображенням.

13. Проведено тестування надійності та функціональності прикладного програмного інтерфейсу шляхом випробування системи на визначених сценаріях. Результати тестування показали коректну роботу прикладного програмного інтерфейсу при ідентифікації літального апарату, а також відповідну поведінку системи при виникненні помилок.

14. Прикладний програмний інтерфейс може використовуватись для ідентифікації літальних апаратів на спеціалізованих контрольних пунктах аеропортів та прикордонних об'єктів, для забезпечення моніторингу повітряного простору у разі відмови інших систем ідентифікації літальних апаратів.

15. Запропонований підхід на основі глибоких згорткових нейронних мереж може бути використаний для ідентифікації об'єктів в інших сферах за різними областями інтересу (додаток Г).

					ДП.КН.8091602.077.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		63

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Aircraft-Wikipedia URL: <https://en.wikipedia.org/wiki/Aircraft>
2. An Introduction to Aircraft Inspections URL: <https://www.flyingmag.com/an-introduction-to-aircraft-inspections/>
3. Every Tail Tells a Tale URL: <https://flyavl.com/article/every-tail-tells-tale>
4. A Tale of Tail Numbers URL: <https://www.planesense.com/blog/tail-numbers>
5. How Things Work: Aircraft Identification. URL: <https://www.smithsonianmag.com/air-space-magazine/how-things-work-aircraft-identification-12752594/>
6. Радар. URL: <https://uk.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B4%D0%B0%D1%80>
7. Border Drug Smuggling. URL: <https://smallwarsjournal.com/jrnl/art/mexican-cartel-tactical-note-32-ultralight-aircraft-and-border-drug-smuggling>
8. Usama M., Anwar H., Anwar A., Anwar S. Vehicle and License Plate Recognition with Novel Dataset for Toll Collection. ArXiv, 2022. <https://doi.org/10.48550/arXiv.2202.05631>
9. Japan post delivery robot. URL: <https://www.japantimes.co.jp/news/2020/10/07/business/corporate-business/japan-post-delivery-robot/>
10. Amazon Prime Air prepares for drone deliveries. URL: <https://www.aboutamazon.com/news/transportation/amazon-prime-air-prepares-for-drone-deliveries>
11. Tesla Autopilot. URL: https://en.wikipedia.org/wiki/Tesla_Autopilot
12. OCR. URL: <https://www.accusoft.com/resources/blog/ocr-vs-icr-whats-the-difference/>
13. MNIST database. URL: https://en.wikipedia.org/wiki/MNIST_database
14. EasyOCR. URL: <https://pypi.org/project/easyocr/>
15. Wang C., Bochkovskiy A., Liao H. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. ArXiv, 2022. <https://doi.org/10.48550/arXiv.2207.02696>

										ДП.КН.8091602.077.ПЗ	Арк.
											64
Змн.	Арк.	№ докум.	Підпис	Дата							

Посібник для студентів вищих навчальних закладів. Київ: Центр учбової літератури, 2020. 204 с.

34. Torre D., Genero M., Labiche Y., Elaasar M. How consistency is handled in model-driven software engineering and UML: an expert opinion survey. *Software Quality Journal*, 2023. Vol. 31. Pp. 1-54. <https://doi.org/10.1007/s11219-022-09585-2>

35. Комар М.П., Саченко А.О., Васильків Н.М., Гладій Г.М., Коваль В.С. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» за першим (бакалаврським) рівнем вищої освіти.– Тернопіль: ЗУНУ, 2021. – 56 с.

					ДП.КН.8091602.077.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		66