

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
Факультет комп'ютерних наук та технологій
Кафедра комп'ютерних інформаційних технологій

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач кафедри

Аліна САВЧЕНКО

“_____” _____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

(ДИПЛОМНА РОБОТА, ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ

“Магістра”

ЗА ОСВІТНЬО-ПРОФЕСІЙНОЮ ПРОГРАМОЮ “ІНФОРМАЦІЙНІ
УПРАВЛЯЮЧІ СИСТЕМИ ТА ТЕХНОЛОГІЇ”

**Тема: “Розпізнавання образів літаків у реальному часі з
використанням комп'ютерного зору”**

Виконавець: Шемідько Артем Олександрович

Керівник: д.т.н., доцент Савченко Аліна Станіславівна

Нормоконтролер: _____ Ігор РАЙЧЕВ

Київ – 2023

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук та технологій

Кафедра Комп'ютерних інформаційних технологій

Освітній ступінь: "Магістр"

Галузь знань, спеціальність, освітньо-професійна програма: 12 "Інформаційні технології", 122 "Комп'ютерні науки", "Інформаційні управляючі системи та технології".

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Аліна САВЧЕНКО

« ____ » _____ 2023р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи студента

Шемідька Артема Олександровича

(прізвище, ім'я, по батькові)

- 1. Тема роботи:** «Розпізнавання образів літаків у реальному часі з використанням комп'ютерного зору» затверджена наказом ректора від "29" вересня 2023 року за № 1976/ст.
- 2. Термін виконання роботи:** з 02 жовтня 2023р. по 31 грудня 2023р.
- 3. Вихідні дані до роботи:** принципи та методики комп'ютерного зору, структура та використання згорткових нейронних мереж, інтеграція з іншими системами та формування і аугментація датасетів зображень.
- 4. Зміст пояснювальної записки:** вступ, аналітичний огляд і постановка завдання, розгляд основних проблем, підходів та практик з використанням комп'ютерного зору, дослідження архітектурних рішень у побудові нейронних мереж, створення програмного продукту для розпізнавання та навчання згорткової нейромережі, висновки.
- 5. Перелік обов'язкового ілюстративного матеріалу:** інформативні рисунки, графічні скріншоти роботи системи, слайди, презентація.

6. Календарний план-графік

<i>№ n/n</i>	<i>Завдання</i>	<i>Термін виконання</i>	<i>Підпис керівника</i>
1.	Отримання завдання на кваліфікаційну роботу та побудова плану-графіку виконання робіт	02.10.23 – 03.10.23р.	
2.	Огляд та аналіз готових рішень, огляд алгоритмів комп'ютерного зору, аналіз алгоритмів нейронних мереж, та огляд згорткових нейромереж	4.10.23 – 12.10.23р.	
3.	Проведення консультації з науковим керівником щодо плану розділів.	24.10.23 – 28.10.23р.	
4.	Розробка розділу 1	29.10.23 – 11.11.23р.	
5.	Розробка розділу 2	12.11.23 – 20.11.23р.	
6.	Розробка розділу 3	21.11.23 – 29.11.23р.	
7.	Оформлення та друк пояснювальної записки	30.11.23 – 01.12.23р.	
8.	Оформлення супровідних документів. Створення презентації, доповіді та підготовка до захисту кваліфікаційної роботи	04.12.23 – 13.12.23р.	

Дата видачі завдання: 02.10.2023р.

Керівник кваліфікаційної роботи _____ Аліна САВЧЕНКО
(підпис керівника)

Завдання прийняв до виконання _____ Артем ШЕМІДЬКО
(підпис випускника)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи «Розпізнавання образів літаків у реальному часі з використанням комп'ютерного зору» складається із вступу, трьох розділів, висновків, і містить 75 сторінок тексту та 28 рисунків. Список бібліографічних посилань містить 19 найменувань.

Метою кваліфікаційної роботи є розробка та впровадження ефективного алгоритму для розпізнавання образів літаків у реальному часі з використанням методів комп'ютерного зору. Проект спрямований на створення робочого прототипу системи, яка забезпечить автоматизоване та точне виявлення літаків на зображеннях, надходячи в режимі реального часу.

Об'єктом дослідження є процес розробки та оптимізації алгоритмів, які дозволять ефективно та точно виявляти літаки на зображеннях, отриманих в реальному часі з високороздільних джерел.

Предметом дослідження є розробка та оптимізація алгоритмів розпізнавання образів літаків у реальному часі за допомогою комп'ютерного зору та нейронних мереж. Дослідження спрямоване на вдосконалення точності та швидкості розпізнавання в умовах змінюючогося оточення, а також врахування різноманітних типів літаків для підвищення універсальності системи.

Результат проєкту: розробка та впровадження системи розпізнавання образів літаків у реальному часі, використовуючи комп'ютерний зір.

Ключові слова: РОЗПІЗНАВАННЯ ОБРАЗІВ ЛІТАКІВ, КОМП'ЮТЕРНИЙ ЗІР, РЕАЛЬНИЙ ЧАС, ГЛИБОКА НЕЙРОННА МЕРЕЖА (CNN), ЕФЕКТИВНІСТЬ, ІНТЕГРАЦІЯ СИСТЕМ, АНАЛІЗ РЕЗУЛЬТАТІВ, АВТОМАТИЗАЦІЯ, ОПТИМІЗАЦІЯ АЛГОРИТМІВ, КОМП'ЮТЕРНИЙ ЗІР.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ	7
ВСТУП.....	8
РОЗДІЛ 1 ОГЛЯД СУЧАСНИХ ПІДХОДІВ КОМП'ЮТЕРНОГО ЗОРУ	10
1.1. Визначення комп'ютерного зору та його цілі	10
1.2. Модель людського зору і порівняння її із комп'ютерним зором	11
1.2.1. Сприйняття	12
1.2.2. Швидкість та ефективність	13
1.2.3. Послідовність та точність	13
1.2.4. Навчання, розуміння та інтерпретація	13
1.3. Обробка зображень (Image Processing)	14
1.4. Моделі та алгоритми комп'ютерного зору	18
1.4.1. Алгоритм SIFT	18
1.4.2. Алгоритм SURF.....	21
1.4.3. Алгоритм Віюлі-Джонса	22
1.4.4. Підхід до власних граней з використанням методу головних компонент (Principal Component Analysis – PCA).....	25
1.5. Роль комп'ютерного зору в сучасних технологіях	26
1.6. Постановка завдання.....	27
1.7. Висновки до розділу 1	29
РОЗДІЛ 2 ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ	31
2.1. Принципи та структура згорткових нейронних мереж	31
2.1.1. Вхідний шар	34
2.1.2. Згортковий шар	35
2.1.3. Рівень об'єднання (Pooling layer)	38
2.1.4. Шар зведення (вирівнювання) (Flattening layer).....	40
2.1.5. Повністю зв'язний шар (Fully connected layer).....	41
2.1.6. Функції активації (Activation layer).....	42
2.1.7. Шар довільного виключення (Dropout Layer).....	44

2.1.8. Вихідний шар (Output Layer)	45
2.2. Ключові ідеї застосування згорткових нейронних мереж	46
2.3. Недоліки та обмеження згорткових нейромереж	48
2.4. Висновки до розділу 2	49
РОЗДІЛ 3 РОЗРОБКА ЗАСТОСУНКУ ДЛЯ ВИЯВЛЕННЯ ЛІТАКІВ	51
3.1. Використання згорткової нейронної мережі	51
3.1.1. Підготовка даних до навчання.....	52
3.1.2. Архітектура моделі	58
3.1.3. Навчання моделі.....	61
3.2. Використання навченої моделі для виявлення образів	64
3.3. Швидкодія та точність.....	68
3.4. Обмеження.....	69
3.5. Висновки до розділу 3	69
ВИСНОВКИ	71
СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ	73

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

<i>OPENCV</i>	–	Бібліотека функцій та алгоритмів комп'ютерного зору
<i>CNN</i>	–	Згортова нейронна мережа
<i>ReLU</i>	–	Випрямлений лінійний вузол
<i>DROPOUT</i>	–	Філософія ітераційного переходу управління ПЗ
<i>FLATTEN</i>	–	Шар зведення в CNN
<i>POLLING</i>	–	Шар об'єднання в CNN
<i>ACTIVATION</i>	–	Шар активації в CNN
<i>SOFTMAX</i>	–	Нормована експоненційна функція
<i>SIGMOID</i>	–	Неперервно диференційована монотонна нелінійна S-подібна функція
<i>IoU</i>	–	Intersection over Union
<i>ЛА</i>	–	Літальний апарат
<i>CV</i>	–	Комп'ютерний зір

ВСТУП

В сучасному світі, охопленому стрімким технологічним прогресом, розпізнавання образів літаків у реальному часі з використанням комп'ютерного зору стає невід'ємною складовою вирішення важливих завдань в різних областях, включаючи авіаційну безпеку, транспортні системи та моніторинг просторового простору. Здатність автоматично та точно виявляти літаки в режимі реального часу має визначальне значення для оперативної реакції на потенційні загрози, ефективного управління повітряним рухом, а також забезпечення надійності та безпеки авіаційних процесів.

Глибока нейронна мережа (CNN), як ключовий інструмент у сфері комп'ютерного зору, відіграє важливу роль у досягненні високої точності та швидкодії в процесі розпізнавання образів. Ця кваліфікаційна робота присвячена розгляду та розробці ефективної системи розпізнавання літаків, яка працює в реальному часі, використовуючи передові методи комп'ютерного зору та глибокого навчання.

У світлі загостреної потреби у вдосконаленні систем безпеки та управління повітряним простором, дана робота націлена на вирішення актуальних завдань, пов'язаних із застосуванням сучасних технологій для надійного та ефективного розпізнавання літаків в режимі реального часу. Результати даного дослідження можуть стати основою для подальшого розвитку автоматизованих систем, що сприятимуть підвищенню рівня безпеки та ефективності в авіаційній сфері. Експериментальні дослідження полягають в огляді існуючих рішень, їх переваг та недоліків, зосереджуючи увагу на відсутності необхідного функціоналу, який покращить досвід як керівника так і розробника в проєкті та перевірці продуктивності, стабільності та інформативності власних розроблених систем, з урахуванням специфіки діяльності для продукту, шляхом тестування системи, аналізу та відгуку користувачів.

Основні завдання та напрямки роботи включають в себе:

1. *Розробка глибокої нейронної мережі:* створення та налаштування глибокої нейронної мережі, зокрема Convolutional Neural Network (CNN), яка буде здатна ефективно визначати особливості та ознаки, характерні для літаків на зображеннях.

2. *Інтеграція в реальний час:* розробка алгоритму, здатного працювати в режимі реального часу, щоб забезпечити миттєве реагування на нові зображення та миттєве виявлення літаків у реальних умовах.

3. *Оптимізація для різних сценаріїв:* підготовка алгоритму до ефективного використання в різних сценаріях, включаючи різні умови освітлення, погодні умови та варіації у типах літаків, для максимальної універсальності та точності розпізнавання.

4. *Аналіз та оцінка результатів:* проведення аналізу результатів роботи системи, включаючи вивчення метрик ефективності та тестування на реальних даних. Оцінка точності, чутливості та специфічності для забезпечення високоякісного та достовірного розпізнавання літаків.

Мета проекту полягає в створенні інноваційного та надійного рішення для реального часу розпізнавання образів літаків, що може знайти застосування в авіаційній безпеці, навігації та інших областях, де важлива автоматизована система виявлення літаків.

Кваліфікаційна робота має велику практичну значимість, оскільки розроблена система розпізнавання літаків у реальному часі з комп'ютерним зором може знайти застосування в авіаційній безпеці, транспортних системах, медичних доставках та обороні. Результати дослідження визначають нові можливості для оптимізації процесів та підвищення рівня безпеки в різних областях використання.

В результаті проведеної роботи буде отримано систему, яку можна буде повноцінно використовувати в якості програми для розпізнавання ЛА на зображеннях.

РОЗДІЛ 1

ОГЛЯД СУЧАСНИХ ПІДХОДІВ КОМП'ЮТЕРНОГО ЗОРУ

1.1. Визначення комп'ютерного зору та його цілі

Комп'ютерний зір (або комп'ютерний зоровий аналіз) — це галузь інформатики, яка спрямована на те, щоб комп'ютери могли ідентифікувати та розуміти об'єкти та людей на зображеннях і відео. Формально це можна трактувати наступним чином: дозволяти комп'ютеру/машині мати здатність до візуального аналізу. Ця галузь науки вивчає та розробляє методи отримання значущої інформації, алгоритми та системи для автоматичного розпізнавання та обробки з цифрових зображень, відеоданих та інших візуальних вхідних даних, схожих на спосіб, яким функціонує людський зір та в результаті формувати висновки або давати рекомендації на основі сформованої інформації. Комп'ютерний зір ставить перед собою завдання імітувати та розширювати здатність людського зору до обробки великої кількості візуальної інформації, а також виконувати завдання, які для нього важко або неможливо виконати.

Комп'ютерний зір – це мультидисциплінарна галузь, яку в широкому сенсі можна назвати підсферою штучного інтелекту та машинного навчання, яка може передбачати використання спеціалізованих методів і використання загальних алгоритмів навчання. Вона використовує методи для навчання моделей для захоплення та вилучення корисних даних після інтерпретації інформації із зображення чи відеопотоку [1].

Для досягнення поставлених цілей в комп'ютерному зорі поєднуються в собі використання камери, периферійні або хмарні обчислення, програмне забезпечення

Кафедра КІТ (47)				НАУ 23 23 66 000 ПЗ			
Виконав	Шемідько А.О.			ОГЛЯД СУЧАСНИХ ПІДХОДІВ КОМП'ЮТЕРНОГО ЗОРУ	Літера	аркуш	аркушів
Керівник	Савченко А.С.					10	21
Консульт.					УС-211М 122		
Н. контроль	Райчев І.Е.						

та штучний інтелект (AI), щоб дозволити системам «бачити» та ідентифікувати об'єкти. Його концепція базується на навчанні комп'ютерів обробляти зображення на піксельному рівні та розуміти його. Комп'ютер не розглядає зображення як цілісне зображення, як ми, він розбиває зображення на пікселі та використовує значення RGB кожного пікселя, щоб зрозуміти, чи є якась важлива функція, прихована в зображенні. Алгоритми комп'ютерного зору зосереджуються на одній ділянці пікселів за раз і використовують ядро або фільтр, який містить значення множення пікселів для виявлення країв об'єкта. Комп'ютер аналізує зображення за всіма аспектами, такими як кольори, тіні, штрихи тощо, щоб розпізнавати та диференціювати зображення.

1.2. Модель людського зору і порівняння її із комп'ютерним зором

Людський зір, без сумніву, є одним із найважливіших із п'яти почуттів, якими володіє людина, від яких ми залежимо понад усе. Людський зір - це особливе і складне відчуття, яке тісно пов'язане з навколишнім світлом. Люди бачать речі такими, якими вони є: дерева в лісі, книги на полиці, машини на дорозі та хмари в небі. Для інтерпретації кожного об'єкта чи сцени не потрібні очевидні висновки або додаткові зусилля.

Усе це залежить від очей і від того, як вони виявляють світлові візерунки та координують роботу з мозком, щоб перевести світло в зображення, які ми потім бачимо. Людське око є складною оптичною системою, схожою на камеру; світло відбивається від предмета, за яким ведеться спостереження, і потрапляє в очі через рогівку. Далі світло проходить через зіницю та райдужну оболонку, які разом контролюють кількість світла, що надходить в очі. Коли всі вони працюють разом, вони фокусують світло на задній частині ока, яка називається сітківкою. Коли світло потрапляє на сітківку, дрібні клітини, що містяться в сітківці, перетворюють його в електричні сигнали [2].

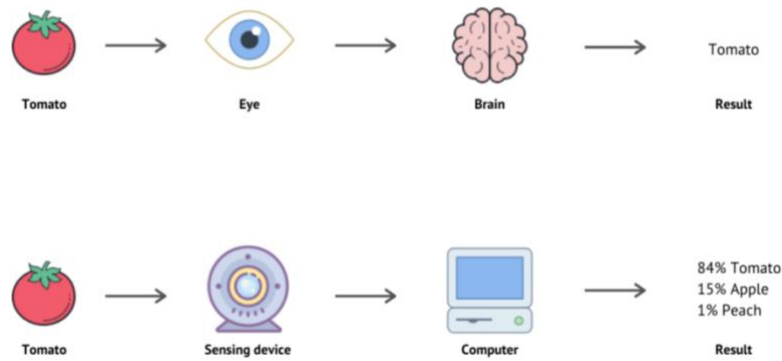


Рис. 1.1. Порівняння моделей обробки зображень

1.2.1. Сприйняття

Люди бачать об'єкти, сцени, візерунки та людей такими, якими вони є, як дерева в пейзажі, книги на полиці, люди в таксі або ключі на ноутбуці. Люди сприймають речі такими, якими вони є, і зберігають те, що вони впізнають, зберігаючи це глибоко в мозку, доки вони знову не зустрінуть ці предмети. Мозок і очі працюють рука об руку, щоб обчислити ці візуальні ефекти без необхідності робити висновки або вимагати додаткових зусиль. Швидкість, з якою відбувається ця інтерпретація, надзвичайно висока, і ми навіть не усвідомлюємо, що це відбувається [2].

Для комп'ютерів це передбачає захоплення зображення чи відео, перетворення його на цифрові дані та обробку цих даних для вилучення значущої інформації. Комп'ютерний зір, з іншого боку, дозволяє комп'ютерам інтерпретувати навколишнє середовище та ідентифікувати речі після того, як реалізовано набір шаблонів і зображень, які комп'ютери «навчені» розпізнавати. Комп'ютерне зір використовує методи та алгоритми машинного навчання для ідентифікації, розрізнення та класифікації об'єктів за розміром або кольором, а також для виявлення та інтерпретації шаблонів у візуальних даних, таких як фотографії та відео. Комп'ютерний зір моделює зір людини, ідентифікуючи об'єкти в його полі зору [3].

1.2.2. Швидкість та ефективність

Людський мозок неймовірно ефективно обробляє візуальну інформацію. Однією з ключових можливостей зорової системи людини є інваріантне розпізнавання об'єктів, тобто люди можуть миттєво й точно ідентифікувати об'єкти в різних варіантах (різне освітлення чи під різними кутами). Люди легко розпізнають об'єкти й не мають проблем з описом об'єктів, навіть якщо вони ніколи раніше не бачили цих об'єктів.

Комп'ютери, з іншого боку, потребують значної обчислювальної потужності для обробки зображень, і хоча вони можуть робити це дуже швидко, вони, як правило, не такі ефективні, як людський мозок. Комп'ютеру потрібно витягти набір характеристик із зображення, щоб створити описи зображення, відмінні від масиву значень пікселів. Розпізнавання тривимірних об'єктів з одного двовимірного зображення є однією з найскладніших проблем комп'ютерного зору [3].

1.2.3. Послідовність та точність

Однією з областей, де комп'ютерний зір може перевершити людський, є послідовність і точність. У той час як люди схильні до помилок і на них можуть впливати такі чинники, як втома чи відволікання, комп'ютери можуть виконувати одне й те саме завдання щоразу послідовно й точно за однакових вхідних даних [1].

1.2.4. Навчання, розуміння та інтерпритація

Людський зір неймовірно гнучкий і адаптивний. Він може розпізнавати об'єкти, навіть якщо вони частково затемнені, при поганому освітленні або під незнайомим кутом. Люди вчаться розпізнавати предмети з самого раннього віку і продовжують вчитися та адаптуватися протягом усього життя. Вони, маючи пару очей, від природи добре сприймають глибину, здатність, відому як стереопсис.

Системи комп'ютерного зору, хоча й вдосконалюються, все ще борються з цими проблемами. Вони потребують навчання на великих вибірках даних, і їм часто

важко розпізнавати об'єкти, розпізнаванню яких вони не були спеціально навчені. Системи комп'ютерного зору також можуть сприймати глибину, але для цього зазвичай потрібні кілька камер або спеціалізовані датчики глибини [2].

Хоча системи комп'ютерного зору можуть розпізнавати об'єкти та візерунки, вони не «розуміють» те, що вони бачать, як люди. Люди не тільки розпізнають об'єкти, але й інтерпретують їх значення та контекст на основі минулого досвіду та знань.

1.3. Обробка зображень (Image Processing)

Зображення можна покращити або отримати важливу інформацію за допомогою обробки зображень. Це форма обробки сигналу, у якій зображення служить вхідним сигналом, а результатом може бути інше зображення, особливості або властивості вхідного зображення. Зараз технологія все частіше включає обробку зображень, а комп'ютерні та інженерні науки використовують її як основну тему вивчення.

Одиницею аналізу в обробці зображень комп'ютерного зору являє собою цифровий об'єкт – зображення, яке визначається як двовимірна функція $F(x,y)$, де x і y — просторові координати, а амплітуда F у будь-якій парі координат (x,y) називається інтенсивністю цього зображення у цій точці. Коли значення x , y та амплітуди F скінченні, це і називається цифровим зображенням.

Іншими словами, зображення може бути визначено двовимірним масивом, спеціально організованим у рядки та стовпці (Рис. 1.2).

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & f(0,2) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & f(1,2) & \dots & f(1,N-1) \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ f(M-1,0) & f(M-1,1) & f(M-1,2) & \dots & f(M-1,N-1) \end{bmatrix}$$

Рис. 1.2. Матричне представлення цифрового зображення

Цифрове зображення складається зі скінченної кількості елементів, кожен з яких має певне значення в певному місці. Ці елементи називаються елементами зображення, вокселями або пікселями. Піксель найбільш широко використовується для позначення елементів цифрового зображення [4].

Зображення також представлено своїми розмірами (висотою та шириною) на основі кількості пікселів. Наприклад, якщо розміри зображення 500 x 400 (ширина x висота), загальна кількість пікселів у зображенні становить 200 000. Піксель — це точка на зображенні, яка набуває певного відтінку, непрозорості або кольору. Зазвичай він представлений в одному з наступних виглядах:

- бінарне зображення (монохромне) – Як випливає з назви, містить лише два піксельні елементи, тобто 0 і 1, де 0 означає чорний колір, а 1 – білий;
- відтінок сірого (*grayscale*) – піксель – це ціле число зі значенням від 0 до 255 (0 – повністю чорний, а 255 – повністю білий);
- *RGB* – піксель складається з 3 цілих чисел від 0 до 255 (цілі числа представляють інтенсивність червоного, зеленого та синього кольорів);
- *RGBA* – це розширення *RGB* з доданим полем альфа, яке представляє непрозорість зображення.

Обробка зображення вимагає фіксованої послідовності операцій, які виконуються з кожним пікселем зображення. Процесор зображень виконує першу

послідовність операцій із зображенням піксель за пікселем. Коли це буде повністю зроблено, він почне виконувати другу операцію тощо (Рис. 1.3). Вихідне значення цих операцій можна обчислити для будь-якого пікселя зображення [5-6].

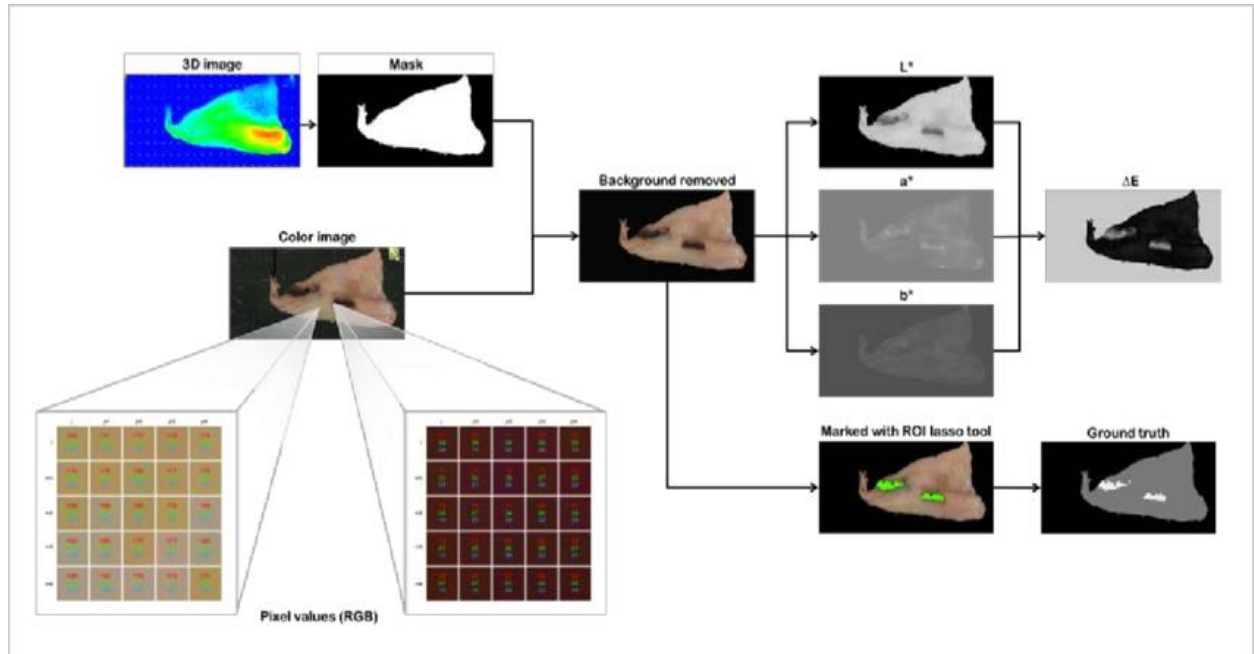


Рис. 1.3. Зображення послідовності операцій обробки зображення

Розробляючи рішення на основі штучного інтелекту при обробці зображень у більшості випадків під час використання комп'ютерного зору моделі не навчаються на кольорових зображеннях, а спочатку перетворюються на зображення у відтінках сірого (grayscale), і лише потім опрацьовуються. Це пов'язано з тим, що кольорові зображення набагато складніші для обробки, і навчання моделі та вимагають значно більше часу.

Зображення зберігаються у формі матриці (2-D масиви/вектори), коли справа доходить до їх обробки. У випадку кольорових зображень кожен елемент цієї матриці містить 3 числа, кожне з яких представляє концентрацію червоного, зеленого та синього кольорів за шкалою від 0 до 255. З іншого боку, кожен елемент зображень у градаціях сірого містить лише одне число в діапазоні від 0 до 255, де 0 означає чорний колір, а 255 — білий. Ці числа називаються колірними каналами. Таким чином, щоб

обробити три колірні канали, потрібно набагато більше обчислювальних ресурсів і часу, і оскільки цього можна легко уникнути, використовуючи зображення у градаціях сірого, більшість моделей, представлених сьогодні, приймають лише зображення у градаціях сірого як вхідні дані [7].

Обробка зображень, як випливає з цього слова, полягає в застосуванні різних алгоритмів і методів для маніпулювання або модифікації зображення, щоб зробити його придатним для поставленого завдання та подальшого використання. Яскравим прикладом використання обробки зображення може бути портретний режим при зйомці, де в процесі створення зображення у нього розмивається фон.

Комп'ютерний зір, з іншого боку, є одним із застосувань штучного інтелекту та використовує алгоритми та методи для визначення шаблонів у даних зображення [8-9].

На прикладі вище згаданого портретного режиму можна пов'язати його з обробкою зображень: у той час як обробка зображень використовується для розмиття фону, штучний інтелект використовується для ідентифікації або розпізнавання фону, і все рішення, розроблене за допомогою поєднання цих двох, буде рішенням комп'ютерного зору (Рис. 3). Тобто відповідь на запитання «Як мобільний телефон дізнається, які об'єкти розмивати, а які не розмивати?» – це штучний інтелект або глибоке навчання, а на питання «Як мобільний розмиває об'єкти?» – це обробка зображень. І відповідь на питання «Як у мобільному телефоні реалізувати портретний режим?» – це комп'ютерний зір [10].



Рис. 3. Діаграма Венна для представлення кореляції між ШІ, обробкою зображень і комп'ютерним зором

Обробка зображення складається з наступних етапів:

- Отримання зображення: це передбачає захоплення зображення за допомогою цифрової камери чи сканера або імпорт існуючого зображення на комп'ютер. Передбачається, що при даному етапі зображення масштабується та відбувається перетворення кольору (наприклад, RGB на сірий)
- Відновлення та покращення зображення: це передбачає покращення візуальної якості зображення, наприклад підвищення контрастності, зменшення шуму та видалення артефактів, усунення розмиття чи спотворення.
- Сегментація зображення: передбачає поділ зображення на області або сегменти, кожен з яких відповідає певному об'єкту чи функції зображення.
- Репрезентація та опис зображення: це передбачає представлення зображення таким чином, щоб його можна було проаналізувати та обробити комп'ютером, а також опис характеристик зображення в стислий та змістовний спосіб.
- Аналіз зображення: це передбачає використання алгоритмів і математичних моделей для отримання інформації із зображення, наприклад розпізнавання об'єктів, виявлення шаблонів і кількісного визначення характеристик.
- Синтез і стиснення зображень: це передбачає створення нових зображень або стиснення існуючих зображень для зменшення вимог до зберігання та передачі.
- Виявлення та розпізнавання об'єктів – це процес, який призначає мітку об'єкту на обробленому цифровому зображенні на основі його дескриптора (опису, що був визначений на етапі репрезентації та аналізу) [9].

1.4. Моделі та алгоритми комп'ютерного зору

1.4.1. Алгоритм SIFT

SIFT був запропонований у 2004 році Девідом Лоу з Університету Британської Колумбії у своїй науковій статті. SIFT або масштабно-інваріантний алгоритм перетворення ознак використовується для виявлення також опису локальних

особливостей у цифровому зображенні. Він знаходить ключові точки та надає їм кількісну інформацію, також відому як дескриптори, які використовуються для виявлення та розпізнавання об'єктів. Дескриптори, отримані за допомогою SIFT, є інваріантними щодо трансформації зображень, завдяки чому зображення виглядають інакше, навіть якщо вони мають однакові об'єкти, але також є стійкими до обертання, освітлення та точки огляду.

SIFT – це 4-етапний алгоритм комп'ютерного зору:

1. Виявлення екстремумів масштабного простору: на цьому кроці алгоритм шукає загальне розташування зображення та масштабує його за допомогою функції різниці Гауса або (DoG - difference-of-Gaussian) для визначення потенційних точок інтересу. Ці точки незмінні щодо масштабу та орієнтації.

2. Локалізація ключових точок: у кожному кандидатському місці підходить детальна модель для визначення розташування та масштабу ключових точок на основі показників їх стабільності.

3. Призначення орієнтації: орієнтації призначаються кожній ключовій точці на основі місцевих напрямків градієнта зображення. Призначені орієнтація, масштаб і розташування кожного елемента на зображенні використовуються в усіх майбутніх операціях над зображенням, які є інваріантними до будь-яких перетворень.

4. Дескриптор ключової точки: локальні градієнти зображення вимірюються у вибраному масштабі навколо кожної ключової точки. Ці градієнти перетворюються на представлення, яке допускає значну зміну освітлення та локальне спотворення форми.

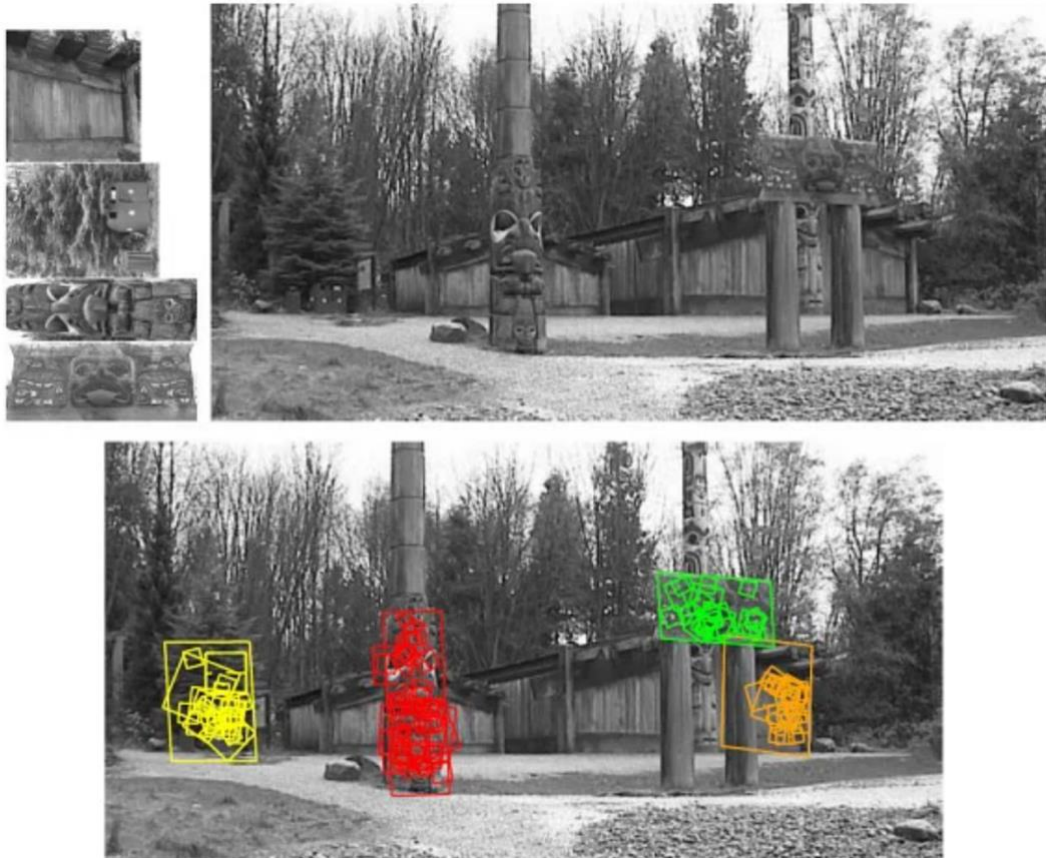


Рис. 1.4. Приклад розпізнавання місцезнаходження за допомогою SIFT.

Верхні ліві зображення (Рис. 1.4) використовуються для навчання. Праве верхнє зображення – це тестове зображення, зроблене з іншої точки огляду. Нижнє зображення показує виявлені місця розташування об'єктів у квадратах і паралелограмах.

До переваг алгоритму можна віднести дескриптори, які є більш точні, ніж інші. SIFT може знаходити відмінні ключові точки, незмінні щодо розташування, масштабу та обертання та стійкі до змін обертання, масштабу, зсуву та положення, що робить SIFT ідеальним алгоритмом для розпізнавання об'єктів. Алгоритму властива локальзованість об'єктів, тому вони стійкі до оклюзії та безладу, що часто спричинено попередньою сегментацією. В загальних випадках має продуктивність, близьку до реального часу.

До недоліків алгоритму SIFT можна віднести повільну роботу при зміні освітлення. Є висока розмірність його дескрипторів, що робить його важко обчислювальним.

SIFT також можна використовувати для розпізнавання об'єктів у 2D-зображеннях, 3D-реконструкції, відстеження руху та сегментації об'єктів, зшивання панорамних зображень тощо. Інші програми включають роботизоване картографування та навігацію, відеоспостереження, індивідуальну ідентифікацію дикої природи, розпізнавання жестів рук тощо [11].

1.4.2. Алгоритм SURF

Прискорені стійкі функції або SURF — це запатентований алгоритм детектора ознак і дескрипторів, який використовується в комп'ютерному зорі головним чином для завдань розпізнавання об'єктів, класифікації, реєстрації зображень і реконструкції. SURF є наближенням SIFT, але він у кілька разів швидший за SIFT і дає кращі результати без будь-якого зниження якості виявлених точок. SURF більш стійкий до трансформації зображень порівняно з SIFT.

SURF — це двоетапний алгоритм комп'ютерного зору:

1. Вилучення функції: точка інтересу на зображенні вибирається за допомогою наближення матриці Гессе (Hessian matrix).
2. Опис функції: дескриптор SURF створюється за допомогою двох кроків. Спочатку фіксується орієнтація на основі інформації про кругову область навколо ключової точки (точка інтересу). Далі створюється квадратна область, вирівняна за орієнтацією, і з якої можна отримати дескриптори.

SURF апроксимує DoG за допомогою «коробкових» фільтрів. Тобто замість Гауссового усереднення зображень для наближення використовуються квадрати. Це дає кращі результати, оскільки згортка з квадратами відбувається швидше, якщо використовується цілісне зображення. Для кожної обраної ключової точки вибирається околиця навколо неї та ділиться на субрегіони. Для кожного субрегіону взяті та представлені вейвлет-відповіді. Це дає дескриптори функцій SURF. Знак

Лапласіана обчислюється на етапі виявлення та використовується для базових точок інтересу. Знак Лапласа розрізняє на зображеннях яскраві плями на темному тлі. Коли зображення порівнюються для відповідності, об'єкти порівнюються, лише якщо вони мають однаковий тип контрасту на основі знака орієнтації. Це дозволяє швидше збігатися.

SURF є швидшим порівняно з SIFT у програмах комп'ютерного зору в реальному часі. Він також має низьку розмірність і менший час обчислення порівняно з SIFT. Проте він не стійкий до обертання. Він не працює належним чином, якщо на зображеннях є проблеми з освітленням.

Алгоритм може бути використаним для пошуку та розпізнавання об'єктів, людей (розпізнавання об'єктів), 3D-реконструкцій, реєстрації та класифікації зображень, відстеження об'єктів тощо [11].

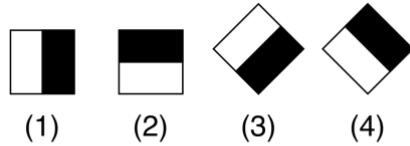
1.4.3. Алгоритм Віоли-Джонса

Алгоритм виявлення об'єктів Віоли-Джонса був розроблений двома дослідниками комп'ютерного зору Полом Віолою та Майклом Джонсом у 2001 році для вирішення проблеми виявлення обличчя, але його також можна навчити виявляти різні класи об'єктів на зображеннях у режимі реального часу. Цей алгоритм повільно навчається для заданого набору даних, але може виявляти обличчя з вражаючою швидкістю та точністю в режимі реального часу.

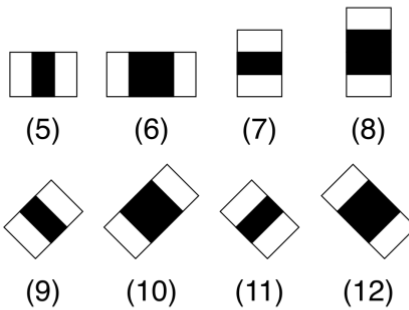
Алгоритм Віоли-Джонса складається з чотирьох основних етапів: для даного зображення (кольорового або сірого зображення) алгоритм розглядає багато менших підобластей на зображенні та намагається знайти обличчя, шукаючи певні особливості в кожній підобласті. Алгоритм повинен перевіряти багато різних масштабів і позицій, оскільки зображення може містити багато граней різного розміру.

1. Вибір Гаароподібних ознак (Рис. 1.5) (ознаки, що можуть характеризувати область цифрового зображення):

Ознаки контурів



Ознаки ліній



Ознаки оточених осередків

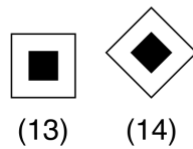


Рис. 1.5. Приклад Гаароподібних ознак

використовує три типи подібних до Гаароподібних ознак: реберні, лінійні та чотиристоронні (Рис. 1.6). Об'єкти ребра та об'єкти лінії використовуються для виявлення ребер та ліній відповідно. Чотиристоронні елементи використовуються для пошуку діагональних елементів на зображенні.



Рис. 1.6. Ознаки, що використовуються алгоритмом Віоли-Джонса

2. Створення цілісного образу: під час створення цілісного зображення значення кожної точки є сумою всіх пікселів вище та ліворуч, включаючи цільовий піксель (Рис. 1.7).

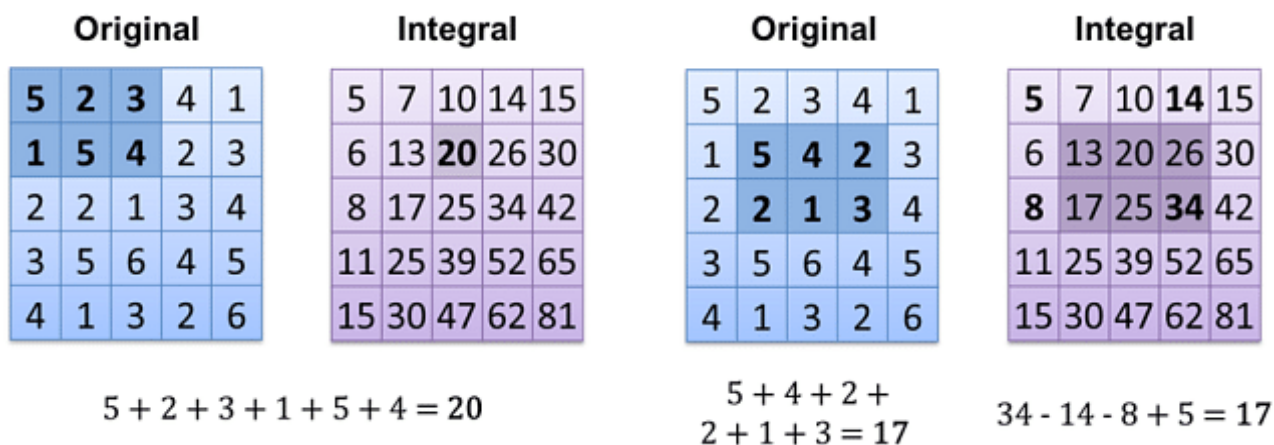


Рис. 1.7. Створення цілісного образу

3. Запуск навчання AdaBoost: алгоритм, що вивчає зображення, які ми надаємо з навчального набору даних, і може визначати помилкові позитивні та справжні негативні результати в наборі даних, роблячи його більш точним. Отримується точна модель, коли розглядаються всі можливі позиції та комбінації функцій.

4. Створення каскадів класифікаторів: каскадування — ще один «хак» для збільшення швидкості та точності алгоритму. У каскадуванні кожен етап складається з сильного класифікатора. Всі ознаки згруповані в кілька етапів. Кожен етап має кілька особливостей. Завдання кожного етапу полягає в тому, щоб визначити, чи дана частинка в зображенні не є обличчям, чи воно може бути обличчям. Ця частинка негайно відкидається як не обличчя, якщо воно не може виявити обличчя на жодному етапі.

Незважаючи на те, що алгоритм Віоли-Джонса є однією з перших систем розпізнавання облич, є досі актуальним. Програми, створені за допомогою цього алгоритму, дають помітні результати визначення обличчя в режимі реального часу.

Алгоритм може бути повільним для навчання зі збільшенням розміру навчального набору даних, але швидкість виявлення обличчя в реальному часі залишається досить вражаючою.

За допомогою цього алгоритму можна побудувати систему розпізнавання обличчя в режимі реального часу, систему відстеження об'єктів, систему відвідуваності в реальному часі за допомогою відеопотоків.

1.4.4. Підхід до власних граней з використанням методу головних компонент (Principal Component Analysis – PCA)

Розпізнавання облич є одним із найуспішніших і широко використовуваних застосувань дослідження комп'ютерного зору. Від використання розпізнавання обличчя для розблокування наших телефонів і ноутбуків до використання розпізнавання обличчя як інструменту для виявлення загроз безпеці в організаціях і оборонних завдань, розпізнавання обличчя є скрізь. Формально можна сказати, що розпізнавання облич є класифікаційним завданням. Вхідними даними алгоритму є зображення, а виходом є список імен людей, ідентифікованих алгоритмом.

Цей підхід використовує концепції лінійної алгебри та зменшення розмірності для розпізнавання облич на зображеннях. Цей метод простий у реалізації та менш витратний з точки зору обчислень. Таким чином, він використовується для розпізнавання рукописного тексту, аналізу медичних зображень, виявлення облич, розпізнавання тощо. Алгоритм власних облич використовує техніку неконтрольованого зменшення розмірності, яка називається PCA або аналізом головних компонентів.

Ідея PCA полягає в тому, що ми хочемо вибрати гіперплощину так, щоб усі точки даних проектувалися на неї, тобто вони були максимально розкидані. Потім ми використовуємо вісь максимальної дисперсії. Ми запозичили термін з лінійної алгебри, який називається власними векторами. Саме звідси власні грані отримали свою назву. Ми обчислюємо коваріаційну матрицю наших даних зображення та

розглядаємо найбільші власні вектори цієї коваріаційної матриці. Це наші головні осі, і ми використовуємо ці осі, щоб проектувати дані та зменшувати розміри.

Цей підхід дозволяє нам брати багатовимірні дані та зменшувати їх до нижчого виміру, вибираючи найбільші власні вектори коваріаційної матриці та проектуючи їх на ці власні вектори. Оскільки ми обчислюємо осі максимального поширення, ми завжди зберігаємо найважливіші аспекти наших даних.

Застосувати цей алгоритм для виявлення облич на відео та зображеннях дуже просто.

Щоб алгоритм давав точні результати, зображення тренувального набору даних потребують правильно відцентрованих облич. Алгоритм також чутливий до освітлення та масштабування.

Підхід власних греней був одним із перших алгоритмів розпізнавання облич, який використовувався на Android. Його можна використовувати для розпізнавання емоцій обличчя. Техніка власних облич не обмежується лише розпізнаванням обличчя, але також може бути розширена для розпізнавання рукописного тексту, аналізу медичних зображень за допомогою читання з губ у лікарнях, розпізнавання голосу, сурдоперекладу, розпізнавання жестів рукою тощо [11].

1.5. Роль комп'ютерного зору в сучасних технологіях

Важливість комп'ютерного зору в сучасних технологіях важко переоцінити, оскільки ця технологія впливає на багато сфер життя та генерує нові інновації та відкриває нові можливості. Ось кілька ключових аспектів важливості комп'ютерного зору в сучасних технологіях:

- *Медицина:* комп'ютерний зір використовується для діагностики та візуального аналізу медичних зображень, таких як рентгенівські знімки, магнітно-резонансна томографія (МРТ) і зображення з камер ендоскопів. Він допомагає лікарям точніше та швидше виявляти патології та хвороби.

- *Автономні автомобілі*: комп'ютерний зір є ключовою технологією для систем автоматичного водіння. Він дозволяє автомобілям сприймати та аналізувати дорожні умови, виявляти перешкоди та інші транспортні засоби.

- *Розпізнавання образів*: використовується для розпізнавання облич, об'єктів, тварин, багатьох інших об'єктів на зображеннях або відео. Це має застосування у відеоспостереженні, безпеці, сортуванні продуктів та інших галузях.

- *Роботизована автоматизація*: дозволяє роботам виконувати завдання та приймати рішення на основі візуального введення. Комп'ютерний зір допомагає роботам та автоматизованим системам виробництва визначати та обробляти об'єкти, контролювати якість продукції та забезпечувати безпеку.

- *Сільськогосподарський моніторинг*: моніторинг росту сільськогосподарських культур, здоров'я худоби та погодних умов за допомогою візуальних даних.

- *Мультимедіа та реклама*: застосування комп'ютерного зору включає в себе ретуш та відновлення зображень, анімацію, аналіз текстур для створення реалістичних ефектів у фільмах, відеоіграх та рекламі.

Це лише кілька прикладів багатьох способів використання комп'ютерного зору сьогодні. Загалом, комп'ютерний зір відкриває широкий спектр можливостей у багатьох галузях та допомагає вирішувати складні завдання, які раніше вважалися неможливими або вимагали б значних людських ресурсів. Він є ключовою складовою сучасних технологій і сприяє прогресу та розвитку в багатьох аспектах нашого життя. Оскільки технологія продовжує розвиватися, очікується, що в майбутньому буде ще більше застосувань і розширюватись вплив комп'ютерного зору на життя.

1.6. Постановка завдання

Даний проект має велике практичне значення в сучасному світі, де технології штучного інтелекту та комп'ютерного зору набувають все більшого застосування в різних сферах. Розпізнавання образів літаків у реальному часі важливо для

забезпечення безпеки, моніторингу повітряного простору, а також для впровадження автономних систем, таких як безпілотні літальні апарати. Ця технологія може знайти застосування в армії, цивільній авіації, логістиці та інших галузях, де необхідно ефективно виявляти та ідентифікувати літаки в реальному часі.

Метою роботи є розробка та впровадження ефективного алгоритму для розпізнавання образів літаків у реальному часі з використанням методів комп'ютерного зору.

Для досягнення поставленої мети необхідно вирішити такі задачі:

- проаналізувати сучасні підходи та методи в області комп'ютерного зору;
- розглянути види та архітектури нейромережі, придатні для розпізнавання образів у реальному часі;
- провести обґрунтований вибір виду нейронної мережі;
- провести підготовку даних до навчання;
- спроектувати архітектуру нейромережі;
- провести навчання моделі нейромережі на тренувальному датасеті;
- провести тестування навченої моделі для розпізнавання образів;
- визначити ефективність побудованої нейромережі за критеріями швидкодії та точності;
- розробити прототип програмної системи для автоматизованого розпізнавання літаків в реальному часі.

В результаті кваліфікаційної роботи буде розроблено прототип програмної системи на базі згорткової нейронної мережі, яка забезпечить автоматизоване та точне виявлення літаків на зображеннях, які надходять в реальному часі.

Такий застосунок, заснований на згортковій нейромережі, вирізняється високою точністю розпізнавання образів літаків, завдяки ефективному використанню локальних особливостей. Ця система демонструє ефективність в реальному часі, забезпечуючи швидке та надійне виявлення літаків в динамічних умовах.

Додатково, застосунок є високоадаптивним до змін умов, таких як зміни освітлення та погодні умови, забезпечуючи стабільність та надійність

функціонування. Крім того, архітектура згорткових нейромереж відзначається можливістю ефективно обробляти великі обсяги зображень, роблячи застосунок придатним для використання в різних сценаріях.

Легкість у використанні та розширенні системи підтримується наявністю передбачених ваг від попередньо навчених моделей, що спрощує процес тренування та дозволяє легко адаптувати систему до нових типів літаків або змінюючихся умов.

1.7. Висновки до розділу 1

У даному розділі було ретельно досліджено та розглянуто різні аспекти комп'ютерного зору та його значення в сучасних технологіях. Починаючи з визначення комп'ютерного зору та його цілей, було проведено порівняльний аналіз між моделлю людського зору і комп'ютерним зором. Особлива увага приділена різниці у сприйнятті, швидкості та ефективності, послідовності та точності, а також можливостям навчання, розуміння та інтерпретації.

Комп'ютерний зір є сферою штучного інтелекту, яка спеціалізується на моделюванні та імітації людського зору за допомогою комп'ютерів. Головні особливості включають в себе обробку зображень, розпізнавання об'єктів, аналіз текстур та форм, а також визначення паттернів. Важливість комп'ютерного зору полягає у здатності точно та швидко обробляти великі обсяги візуальної інформації. Це забезпечує високу ефективність у сферах, де необхідна автоматизована обробка та аналіз зображень, таких як медицина (діагностика), автомобільна промисловість (автономні автомобілі), безпека (відеоспостереження), а також у прогнозуванні та аналізі зображень для вирішення різних завдань у реальному часі.

Процес обробки зображень є ключовим етапом у розвитку комп'ютерного зору і лишається вкрай необхідним для подальшого аналізу та розпізнавання об'єктів.

Алгоритм SIFT вирізняється своєю масштабовано-інваріантною природою, забезпечуючи стійкість до змін розміру та ідеально підходить для складних об'єктів. SURF, зокрема, вразив своєю швидкістю та ефективністю завдяки використанню інтегральних зображень.

Алгоритм Віолі-Джонса, спрямований на обличчярозпізнавання, використовує активний класифікатор та адаптивні фільтри для швидкого та ефективного виділення обличчя на зображенні. Зокрема, цей метод знаходить широке застосування в системах відеоспостереження та безпеки.

Для визначення власних граней, метод головних компонент (РСА) використовує статистичний підхід, що дозволяє зменшити розмірність даних, зберігаючи основні характеристики зображень. Цей підхід стає особливо ефективним у задачах розпізнавання обличчя та інших сценаріях, де важливо визначити ключові компоненти вхідних даних.

Усі ці алгоритми та методи взаємодіють для створення потужних інструментів у галузі комп'ютерного зору, що забезпечує високий рівень точності та швидкості у завданнях обробки та аналізу зображень. Їхнє впровадження у різноманітні сфери застосування свідчить про значущий внесок у вдосконалення технологій та розвиток сучасних систем комп'ютерного зору.

Комп'ютерний зір відіграє ключову роль у сучасних технологіях, забезпечуючи ефективність та точність обробки та аналізу великого обсягу візуальної інформації. Застосування алгоритмів та методів комп'ютерного зору, таких як SIFT, SURF, Віолі-Джонса та РСА, сприяє вдосконаленню систем розпізнавання об'єктів, обличчя, а також забезпечує надійні та швидкі рішення у сферах медицини, безпеки, автомобільної промисловості та інших галузях. Розвиток комп'ютерного зору визначає нові стандарти у використанні візуальної інформації та є ключовим фактором у подальшому розвитку сучасних технологій.

РОЗДІЛ 2

ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ

2.1. Принципи та структура згорткових нейронних мереж

Нейронні мережі є ключовим елементом машинного навчання і виступають фундаментом для алгоритмів глибокого навчання. Складаючись із шарів вузлів, вони включають вхідний рівень, один чи кілька прихованих шарів та вихідний рівень. Кожен вузол взаємодіє з іншим, обладнаний вагою та порогом. Якщо вихід будь-якого вузла перевищить встановлене порогове значення, активується цей вузол, передаючи дані на наступний рівень. В іншому випадку інформація не переходить наступний рівень мережі. Раніше для ідентифікації об'єктів на зображеннях застосовувалися ручні методи виділення ознак, що були часомомісткими. Однак згорткові нейронні мережі тепер забезпечують більш масштабний підхід до задач класифікації та розпізнавання об'єктів. Вони використовують принципи лінійної алгебри, зокрема операції множення матриць, для виявлення шаблонів на зображеннях. Однак, незважаючи на це, їхня висока обчислювальна складність може вимагати використання графічних процесорів (GPU) під час навчання моделей.

Згорткова нейронна мережа (CNN) — це тип штучної нейронної мережі, призначеної для обробки даних структурованої сітки (Рис. 2.1), наприклад зображень. CNN особливо ефективні в задачах комп'ютерного зору, де вхідними даними часто є зображення, а метою є виділення ознак, розпізнавання закономірностей і прогнозування на основі візуальних даних. Даний тип нейронних мереж зазвичай використовується в поєднанні з комп'ютерним зором, який дозволяє комп'ютеру розуміти та інтерпретувати зображення або візуальні дані.

Кафедра КІТ (47)				НАУ 23 23 66 000 ПЗ			
<i>Виконав</i>	<i>Шемідько А.О.</i>			ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ	<i>Літера</i>	<i>аркуш</i>	<i>аркушів</i>
<i>Керівник</i>	<i>Савченко А.С.</i>					31	20
<i>Консульт.</i>					УС-211М 122		
<i>Н. контроль</i>	<i>Райчев І.Е.</i>						

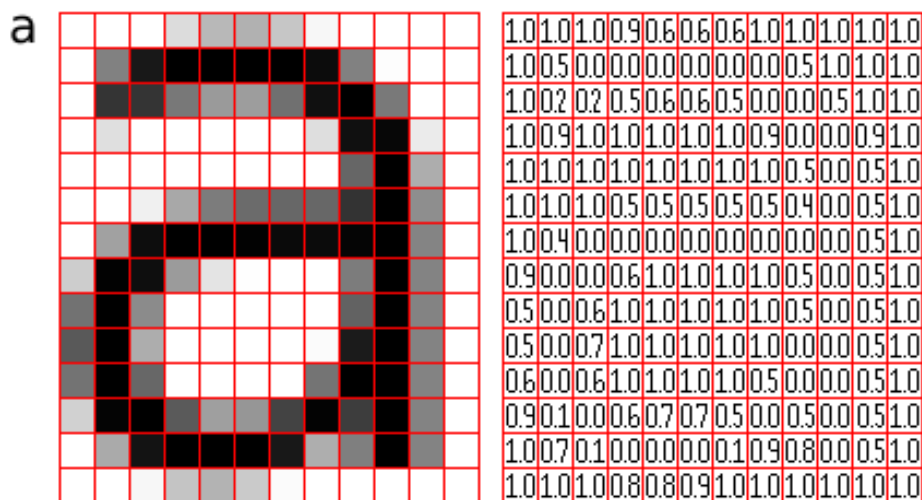


Рис. 2.1. Подання зображення у вигляді сітки пікселів

У типовій архітектурі згорткової нейронної мережі (CNN) проект організований у рівні, які обробляють вхідні дані та поступово витягують функції. Зазвичай мережа починається з вхідного рівня, за яким слідує серія прихованих рівнів і закінчується вихідним рівнем [12].

Вхідний шар служить точкою входу для необроблених вхідних даних, часто представляючи зображення з розмірами, що відповідають його ширині, висоті та кольоровим каналам. Цей початковий шар створює основу для подальшої обробки.

Після вхідного рівня CNN складається з послідовності прихованих шарів, ядро яких утворюють згортковий і активаційний шари. Згорткові шари застосовують фільтри до вхідних даних, фіксуючи просторові моделі та особливості. Функції активації, такі як Rectified Linear Unit (ReLU), вводять нелінійність, щоб дозволити мережі вивчати складні зв'язки всередині даних.

У поєднанні зі згортковими шарами шари об'єднання використовуються для зменшення вибірки просторових розмірів карт функцій. Об'єднання допомагає зменшити обчислювальну складність, зосереджуючись на важливій інформації та сприяючи інваріантності перекладу, зберігаючи важливі функції, відкидаючи менш важливі деталі.

Приховані шари також включають один або кілька наборів повністю підключених і активаційних шарів. Повністю підключені рівні поєднують функції високого рівня з попередніх рівнів, дозволяючи мережі розуміти складні зв'язки в даних. Функції активації додатково вносять нелінійність у вивчені представлення.

Перед підключенням до повністю зв'язаних шарів карти функцій часто трансформуються за допомогою шару зведення, перетворюючи просторову інформацію в одновимірний вектор, придатний для введення в наступні повністю зв'язані шари.

За бажанням можна ввести рівень вилучення для покращення узагальнення шляхом випадкової дезактивації нейронів під час навчання. Це допомагає запобігти переобладнанню, гарантуючи, що мережа не покладається надто сильно на певні функції.

Вихідний рівень служить останнім рівнем мережі та створює прогнози або класифікації. Кількість нейронів у цьому шарі залежить від поставленого завдання: один нейрон для бінарної класифікації або кілька нейронів для багатокласової класифікації. Вибір функції активації на вихідному рівні визначається конкретними вимогами завдання, такими як використання softmax для багатокласової класифікації або sigmoid для бінарної класифікації [13].

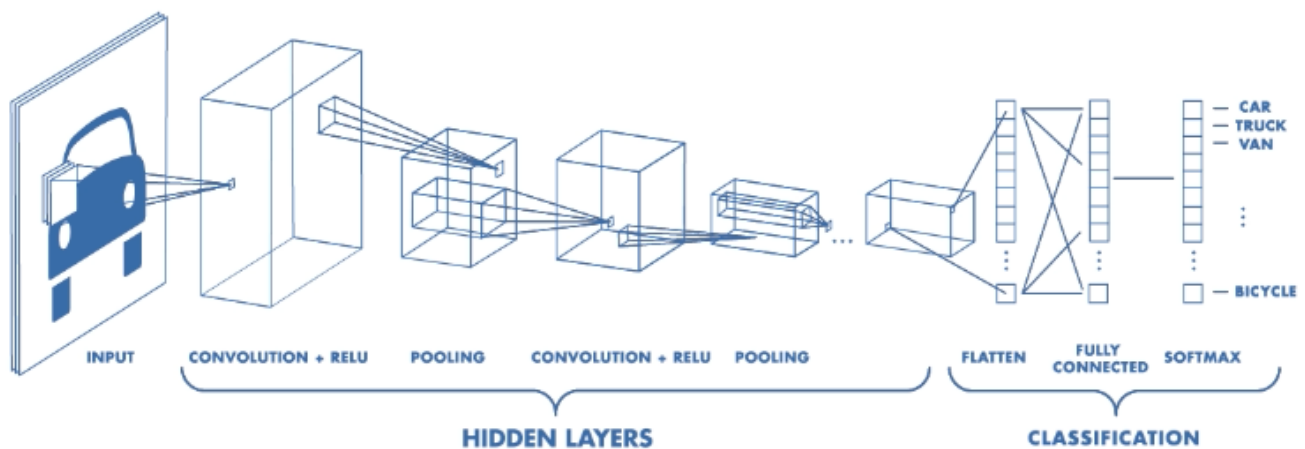


Рис. 2.2 Архітектура згорткової нейромережі

2.1.1. Вхідний шар

Вхідний рівень згорткової нейронної мережі (CNN) служить початковим етапом, який отримує необроблені вхідні дані. У контексті завдань комп'ютерного зору, зокрема класифікації зображень, вхідні дані зазвичай складаються із зображень, а вхідний рівень розроблений для розміщення просторової структури цих зображень.

Розміри вхідного шару визначаються розміром вхідних даних, що представляють такі атрибути, як ширина, висота та кількість кольорових каналів. Наприклад, кольорове зображення з каналами RGB може мати такі розміри, як $224 \times 224 \times 3$, що означає ширину та висоту 224 пікселів кожен і три кольорові канали для червоного, зеленого та синього.

Окрім просторових розмірів, вхідний рівень також обробляє нормалізацію значень пікселів. Нормалізація — це звичайна практика, яка забезпечує послідовне масштабування значень пікселів, часто в діапазоні $[0, 1]$ або $[-1, 1]$. Ця нормалізація сприяє швидшій конвергенції (тобто процесу зближення, сходження) під час процесу навчання.

Етапи попередньої обробки можуть застосовуватися в межах вхідного рівня залежно від конкретного завдання та характеристик вхідних даних. Ці кроки можуть включати зміну розміру зображень до стандартного розміру, обрізання або інші перетворення для забезпечення однорідності набору даних.

Вхідний рівень обробляє вхідні дані у вигляді тензора. У випадку даних зображення цей тензор зазвичай є тривимірним тензором, де розміри відповідають ширині, висоті та кольоровим каналам.

Під час навчання методи розширення даних можуть бути включені у вхідний рівень. Збільшення даних передбачає застосування випадкових перетворень, таких як повороти, перевероти та масштабування, для створення варіацій вхідних даних. Це розширення допомагає моделі краще узагальнювати та стати більш надійною до різних зовнішніх виглядів того самого об'єкта [14].

2.1.2. Згортковий шар

Згортковий рівень є фундаментальним компонентом згорткових нейронних мереж (CNN) і відіграє ключову роль у захопленні просторових моделей і характеристик із вхідних даних, зокрема в задачах комп'ютерного зору, таких як розпізнавання зображень.

Розташований після вхідного рівня, згортковий рівень застосовує згорткові операції до вхідних даних за допомогою навчальних фільтрів або ядер. Кожен фільтр — це невелика матриця, яка ковзає по вхідних даних, виконуючи поелементне множення та підсумовування (Рис. 2.3). Цей процес дозволяє шару виявляти локальні візерунки, такі як краї, текстури або більш складні структури, коли фільтри переміщуються по вхідних даних [14].

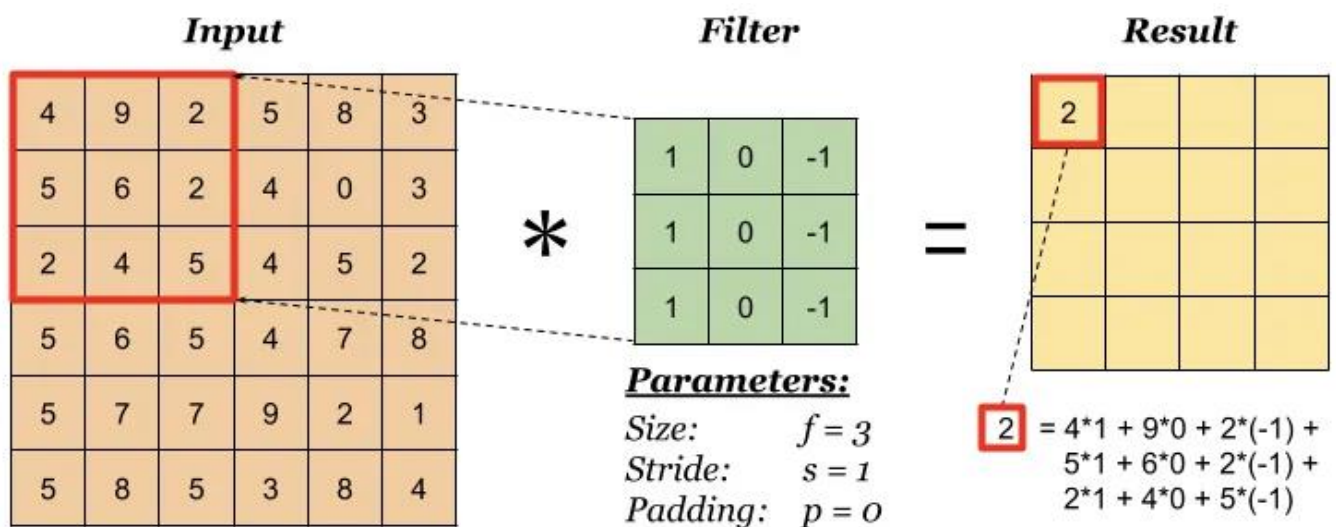


Рис. 2.3. Згортка на двовимірному зображенні / один канал

Функція активації, зазвичай Rectified Linear Unit (ReLU), зазвичай застосовується до виходу згорткового рівня. ReLU вводить нелінійність, дозволяючи мережі вивчати складніші зв'язки в даних.

Згортковий рівень характеризується спільним використанням параметрів, коли однаковий набір ваг (фільтрів) застосовується до різних просторових положень у

вхідних даних. Такий розподіл ваг сприяє здатності моделі розпізнавати шаблони, інваріантні до їх конкретного розташування у вхідних даних, сприяючи інваріантності перекладу.

Кілька фільтрів використовуються паралельно, створюючи набір карт функцій, які представляють різні об'єкти, отримані з вхідних даних. Ці карти ознак спільно фіксують ієрархічні представлення вхідних даних, причому нижчі рівні фіксують прості функції, а вищі рівні об'єднують їх для розпізнавання складніших моделей.

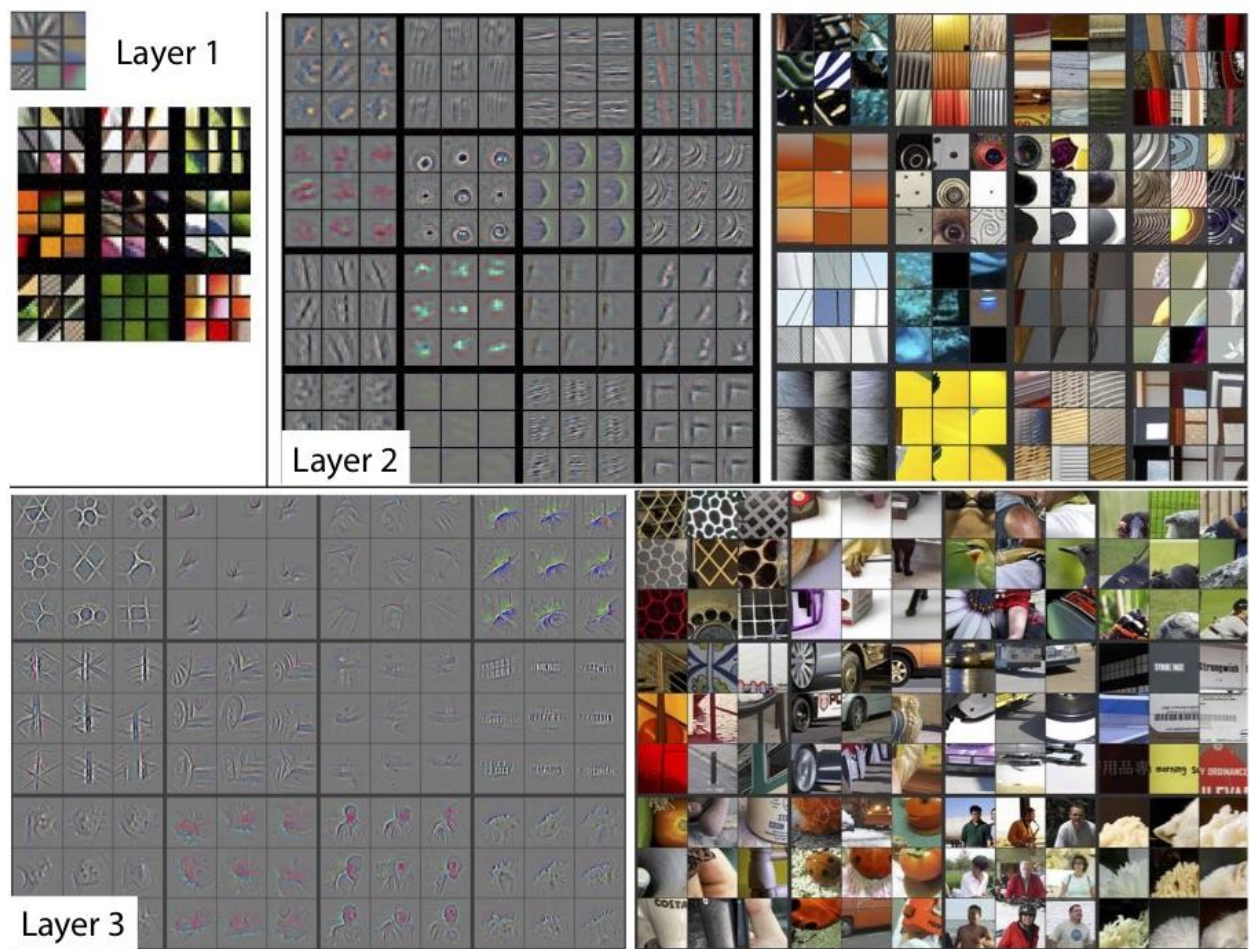


Рис. 2.4. Приклад формування ознак при застосуванні декількох згорткових шарів

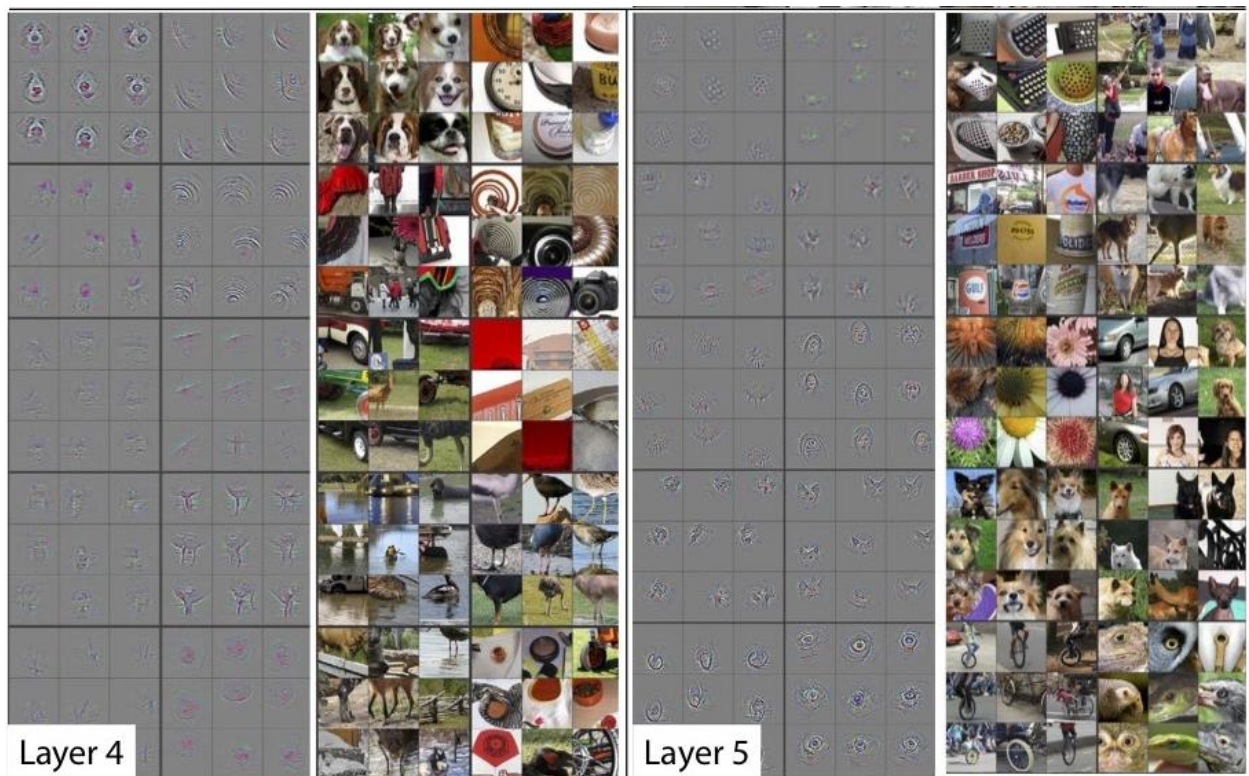


Рис. 2.4. Приклад формування ознак при застосуванні декількох згорткових шарів (продовження).

У деяких архітектурах згорткові рівні можуть мати групи, де фільтри розділені на окремі набори, які працюють з різними частинами вхідних даних. Це може бути корисним для сприяння різноманітності функцій [15].

Поведінка згорткового шару в згортковій нейронній мережі (CNN) регулюється різними гіперпараметрами, які визначають його характеристики та впливають на процес навчання. Ці гіперпараметри включають:

- кількість фільтрів (ядер): кількість фільтрів визначає, скільки функцій згортковий шар зможе охопити. Кожен фільтр вивчає різні моделі, і загальна глибина шару визначається кількістю фільтрів;
- розмір фільтра (розмір ядра): розмір фільтра визначає просторові розміри фільтрів. Загальні варіанти включають 3x3 і 5x5. Фільтри меншого розміру вловлюють дрібніші деталі, тоді як фільтри більшого розміру вловлюють більш глобальні візерунки;

- Крок фільтра: крок визначає розмір кроку, з яким фільтр рухається по вхідних даних. Більший крок призводить до менших карт об'єктів, тоді як менший крок зберігає більше просторової інформації. Крок впливає на просторові розміри результату;

- відступи: передбачає додавання додаткових пікселів навколо вхідних даних, щоб пом'якшити зменшення просторових розмірів, спричинене операцією згортання. Це допомагає підтримувати просторову інформацію на кордонах введення;

- функція активації: вибір функції активації, такої як ReLU (Rectified Linear Unit), вносить нелінійність у вихід згорткового рівня, дозволяючи мережі вивчати складні шаблони;

- ініціалізація ваги: початкова вага фільтрів має вирішальне значення для ефективного навчання. Належні методи ініціалізації ваги, такі як ініціалізація Xavier, можуть вплинути на конвергенцію та продуктивність мережі.

Ці гіперпараметри разом визначають, як згортковий рівень обробляє вхідні дані, витягує ознаки та створює ієрархічні представлення. Правильне налаштування цих гіперпараметрів має важливе значення для ефективної роботи CNN у різних задачах комп'ютерного зору.

2.1.3. Рівень об'єднання (Pooling layer)

У згорткових нейронних мережах (CNN) шар об'єднання — це тип шару, який зазвичай використовується для зменшення просторових розмірів (ширини та висоти) вхідного об'єму. Об'єднання використовується після згорткових шарів для поступового зменшення просторового розміру представлення, тим самим зменшуючи кількість обчислень у мережі та контролюючи надлишок.

Існує два поширених типи шарів об'єднання:

- Об'єднання по максимуму (Рис. 2.6): у цьому об'єднанні для кожного локального регіону на вхідній карті об'єктів береться максимальне значення та використовується для представлення цього регіону на об'єднаній карті об'єктів. Максимальне об'єднання часто використовується для збереження найважливіших

функцій і відкидання менш важливої інформації. Реалізується за допомогою ковзного вікна над входом і вибору максимального значення в кожному вікні.

- Об'єднання по середньому значенню (Рис. 2.6): у цьому об'єднанні для кожного локального регіону на вхідній карті об'єктів обчислюється середнє значення та використовується для представлення цього регіону на об'єднаній карті об'єктів. Середнє об'єднання менш дороге в порівнянні з максимальним об'єднанням. Його можна використовувати для більш плавної дискретизації карти функцій.

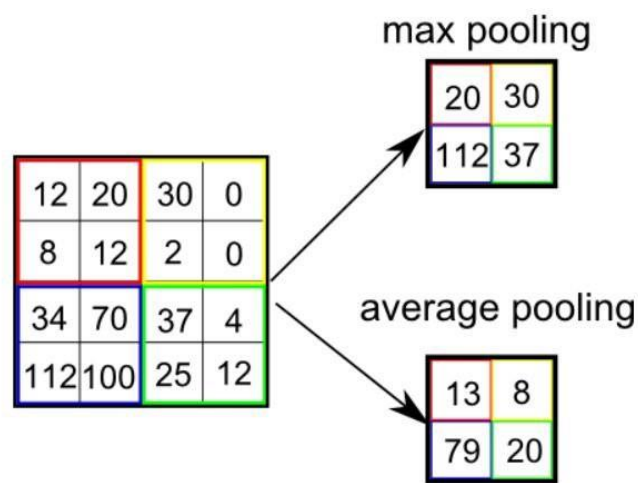


Рис. 2.5. Результат об'єднання по максимуму і середньому значенню

Шари об'єднання важливі з кількох причин:

- зменшення розмірності: завдяки зменшенню просторових розмірів обчислювальна складність мережі зменшується, що робить її ефективнішою з точки зору обчислень;

- інваріантність перекладу: об'єднання допомагає мережі стати певною мірою інваріантною до перекладів у вхідних даних, оскільки об'єднаний вихід визначається наявністю функцій, а не їх конкретним розташуванням;

- зменшення параметрів: Об'єднання допомагає зменшити кількість параметрів і обсяг обчислень у мережі, що є корисним для керування надмірним оснащенням, особливо у великих мережах.

Важливо зазначити, що не всі архітектури CNN використовують шари об'єднання, і деякі сучасні архітектури досліджують альтернативи, такі як поступові згортки, щоб досягти подібних ефектів просторового зменшення [14].

2.1.4. Шар зведення (вирівнювання) (Flattening layer)

Рівень зведення в згорткових нейронних мережах (CNN) служить критичним інтерфейсом між згортковим і об'єднуючим рівнями, які працюють з просторово структурованими даними, і наступними повністю пов'язаними рівнями, які очікують одновимірного введення. Цей шар має стратегічне розташування для перетворення двовимірних карт об'єктів, створених попередніми шарами, у згладжений одновимірний вектор.

Перш ніж вхідні дані можна буде ввести в повністю зв'язані шари, вони повинні пройти цей процес зведення. Метою зведення є перетворення просторової інформації, що міститься в картах об'єктів, у формат, який можна легко інтегрувати в щільно зв'язану структуру повністю зв'язаного шару. Це перетворення, по суті, розкриває просторову ієрархію об'єктів і організовує їх у лінійну послідовність, дозволяючи наступним повністю пов'язаним шарам зафіксувати глобальні взаємозв'язки та комбінації цих об'єктів.

У процесі зведення кожен елемент у двовимірних картах ознак послідовно впорядковується в одновимірний вектор. Отриманий сплющений вектор служить високорівневим абстрактним представленням вхідних даних, причому кожен елемент представляє конкретну особливість або характеристику, витягнуту попередніми шарами згортки та об'єднання.

Варто зазначити, що шар зведення не передбачає жодних параметрів навчання чи обчислень, окрім перевпорядкування даних. Його роль є суто структурною, полегшуючи перехід від просторово організованих карт об'єктів до сплющеного представлення, необхідного для повністю зв'язаних шарів.

Включення шару вирівнювання є стандартною практикою в архітектурах CNN, що забезпечує сумісність між виходом згорткових і об'єднаних шарів і вхідними

очікуваннями повністю з'єднаних шарів. Цей плавний перехід дозволяє нейронній мережі ефективно фіксувати як локальні, так і глобальні закономірності у вхідних даних, завдяки чому вона добре підходить для широкого кола завдань, включаючи класифікацію зображень, виявлення об'єктів і сегментацію [15].

2.1.5. Повністю зв'язний шар (Fully connected layer)

Повністю зв'язний рівень, також відомий як щільний шар, є ключовим компонентом згорткових нейронних мереж (CNN), який полегшує інтеграцію високорівневих функцій, витягнутих за допомогою згорткових і об'єднаних рівнів. Розташований у кінці мережевої архітектури, повністю підключений рівень служить переходом від просторових ієрархій функцій до кінцевого результату, що робить його критичним елементом у прийнятті рішень.

На відміну від згорткових шарів, які зосереджені на локальних шаблонах, повністю зв'язаний рівень з'єднує кожен нейрон з кожним нейроном у попередньому та наступних шарах. Цей комплексний зв'язок дозволяє цьому рівню фіксувати складні взаємозв'язки та комбінації функцій, отриманих за допомогою згорткового рівня та рівня об'єднання. Він служить глобальним агрегатором, поєднуючи локальну інформацію в представлення, придатне для прогнозування або класифікації.

Перед входом у повністю зв'язаний шар вихідні дані з попередніх шарів зазвичай зводять до одновимірного вектора. Це перетворення гарантує належну обробку просторової інформації та введення її в щільно зв'язану структуру повністю зв'язаного рівня. Цей сплющений вектор по суті представляє високорівневе представлення абстрактних ознак вхідних даних.

Кожен нейрон у повністю зв'язаному шарі пов'язаний із ваговим параметром, і під час навчання ці ваги коригуються за допомогою зворотного поширення, щоб мінімізувати різницю між прогнозованим виходом і основною правдою. Крім того, функція активації зазвичай застосовується до виходу кожного нейрона в повністю підключеному шарі, що вносить нелінійність у процес прийняття рішень. Загальні

функції активації включають Rectified Linear Unit (ReLU) для прихованих шарів і Softmax для вихідного шару в завданнях класифікації [15].

Хоча повністю підключений рівень довів ефективність у різних завданнях, він також сприяє загальній складності моделі та вводить велику кількість параметрів. У деяких архітектурах, особливо тих, які розроблені для спеціальних застосувань, таких як класифікація зображень, повністю пов'язані шари супроводжуються додатковими методами регуляризації, такими як вилучення, щоб пом'якшити переобладнання.

2.1.6. Функції активації (Activation layer)

Також важливою частиною згорткового шару є функції активації, відіграють вирішальну роль у штучних нейронних мережах, вносячи в модель нелінійність. Ця нелінійність є важливою для того, щоб мережа вивчала складні закономірності та зв'язки в даних, дозволяючи їй наближено представляти складні функції. Ось деякі ключові моменти щодо функцій активації

Важливість нелінійних функцій активації в нейронних мережах полягає в їх здатності вводити складність, виразність і здатність вивчати складні шаблони з вхідних даних. Ключовою характеристикою нейронних мереж є їх здатність апроксимувати та моделювати нелінійні зв'язки в даних, і нелінійні функції активації є важливими для розблокування цієї можливості.

За відсутності нелінійності нейронна мережа по суті була б зведена до лінійної моделі. Укладання кількох лінійних шарів призведе лише до лінійного перетворення, обмежуючи здатність моделі охоплювати та представляти складні, нелінійні відображення між вхідними та вихідними даними. Справжня сила нейронних мереж полягає в їхній здатності вивчати та розуміти складні особливості, зв'язки та шаблони в даних, чому сприяє введення нелінійності через функції активації.

Розглядаючи сценарій, у якому використовуються лише лінійні функції активації, такі як функція ідентифікації. Композиція цих лінійних функцій все одно призведе до загального лінійного перетворення, нездатного охопити складні зв'язки, які зустрічаються в багатьох проблемах реального світу. Нелінійні функції активації,

такі як широко використовувана випрямлена лінійна одиниця (ReLU), сигмоїда або гіперболічний тангенс (\tanh), дозволяють моделі звільнитися від лінійності та вивчати складні представлення, які є життєво важливими для таких завдань, як розпізнавання зображень, природні обробка мови та інші складні області [16].

ReLU працює, встановлюючи вихід на нуль для будь-яких негативних вхідних значень і пропускаючи позитивні вхідні значення без змін. Математично це можна визначити як $f(x) = \max(0, x)$ (Рис. 2.5). Цей простий пороговий механізм дає кілька переваг у контексті функцій активації нейронної мережі.

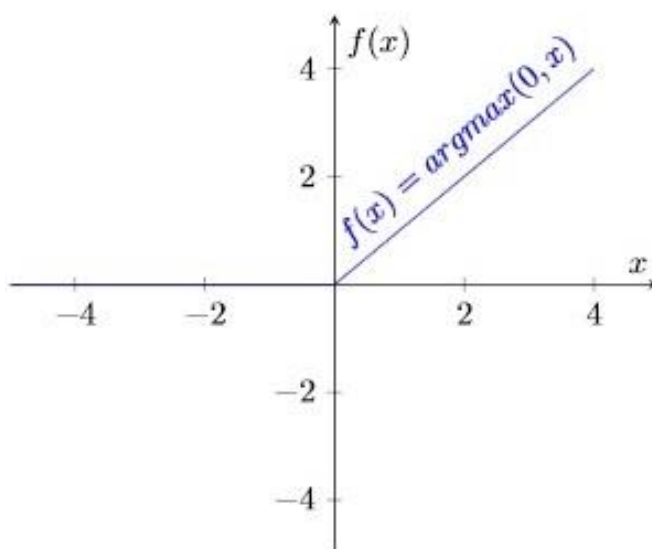


Рис. 2.5. Графік функції активації ReLU

Однією з основних причин використання ReLU є його здатність пом'якшувати проблему зникнення градієнта, яка може виникнути з іншими функціями активації, такими як sigmoid або \tanh . У глибоких мережах під час зворотного поширення градієнти можуть стати надзвичайно малими, перешкоджаючи ефективним оновленням ваги та сповільнюючи навчання. ReLU, дозволяючи проходити позитивні градієнти без змін, допомагає вирішити цю проблему, сприяючи швидшій конвергенції під час навчання.

Крім того, ReLU є обчислювально ефективним і простим у реалізації. Її лінійна природа для додатних вхідних даних спрощує процес навчання, а відсутність

експоненціальних членів зменшує обчислювальне навантаження порівняно з такими функціями, як sigmoid і tanh.

ReLU також пов'язаний з розрідженою активацією, коли для даного вхідного сигналу активується лише підмножина нейронів. Така розрідженість підвищує здатність мережі до вибіркового вивчення функцій, дозволяючи їй зосереджуватися на найбільш релевантній інформації та допомагаючи запобігти переобладнанню.

Хоча ReLU довів свою ефективність у багатьох сценаріях, важливо зазначити, що це не без проблем. Проблема «вмирання ReLU» може виникнути, коли нейрони постійно видають нуль для всіх вхідних даних під час навчання, роблячи їх неактивними. Цю проблему можна вирішити за допомогою таких варіантів, як Leaky ReLU, Parametric ReLU або Exponential Linear Unit (ELU), які вносять незначні зміни до основної функції ReLU.

Необхідність нелінійних функцій активації також стає очевидною під час вирішення проблеми зникаючого градієнта. Під час зворотного поширення градієнти можуть зменшуватися, коли вони поширюються через кілька шарів, особливо у випадку повторюваних лінійних перетворень. Функції нелінійної активації, такі як ReLU, допомагають усунути цю проблему, дозволяючи моделі фіксувати та поширювати більш значні градієнти, сприяючи ефективному оновленню ваги та швидшій конвергенції під час навчання.

2.1.7. Шар довільного виключення (Dropout Layer)

Рівень вилучення є обов'язковим компонентом в архітектурах нейронних мереж, включаючи згорткові нейронні мережі (CNN), призначені для вирішення проблеми перенавчання під час навчання. Перенавчання відбувається, коли модель надто добре вивчає навчальні дані, вловлюючи шум і нерелевантні деталі, що може призвести до зниження продуктивності узагальнення невидимих даних. Вилучення — це техніка регуляризації, запроваджена для пом'якшення цієї проблеми.

У цьому шарі під час навчання випадкова підмножина нейронів «викидається» або тимчасово дезактивується з імовірністю, визначеною гіперпараметром. Це

означає, що для кожного прямого проходу через мережу інший набір нейронів ігнорується. Цей механізм відключення не дозволяє певним нейронам занадто сильно покладатися на певні функції, змушуючи мережу вивчати більш надійні та узагальнені представлення.

Введення відсіву заохочує мережу вивчати надлишкові представлення та зменшує співзалежність між нейронами, роблячи модель більш стійкою до змін у вхідних даних. Цей метод регуляризації ефективно моделює навчання кількох підмереж із спільними вагами, підвищуючи здатність моделі узагальнювати різноманітні шаблони.

У той час як відсівання зазвичай використовується в повністю зв'язаних шарах, його також можна застосувати до згорткових шарів, особливо в сценаріях, де модель демонструє високу складність або велику кількість параметрів. Однак його застосування до згорткових шарів часто є предметом експериментів, оскільки ці шари за своєю суттю вловлюють просторові ієрархії та можуть бути не такими схильними до переобладнання, як повністю з'єднані шари.

2.1.8. Вихідний шар (Output Layer)

Вихідний рівень нейронної мережі, включно з згортковими нейронними мережами (CNN), служить останнім етапом, відповідальним за створення прогнозів або класифікацій на основі вивчених функцій з попередніх рівнів. Характеристики вихідного рівня тісно пов'язані з конкретним завданням, для виконання якого призначена нейронна мережа.

З точки зору архітектури, кількість нейронів у вихідному шарі безпосередньо відповідає кількості класів або категорій, які беруть участь у завданні класифікації. Для бінарної класифікації достатньо одного нейрона, тоді як завдання класифікації з кількома класами передбачають нейрон для кожного окремого класу. Функція активації, обрана для вихідного рівня, залежить від характеру завдання. Для бінарної класифікації сигмоїдна функція активації часто використовується для отримання оцінок ймовірності в діапазоні від 0 до 1. У випадку багатокласової класифікації

зазвичай використовується функція активації softmax, яка дає нормалізовані ймовірності класу, сума яких дорівнює 1.

Вибір відповідної функції втрат, пов'язаної з вихідним рівнем, має вирішальне значення для процесу навчання. Ця функція втрат кількісно визначає розбіжність між прогнозованими результатами та основними мітками істинності, керуючи алгоритмом оптимізації під час навчання. Перехресна втрата ентропії виділяється як поширений вибір для завдань класифікації.

У завданнях регресії, де метою є прогнозування безперервного значення, вихідний рівень зазвичай містить один нейрон. Функція активації в цьому випадку часто є лінійною або невизначеною, оскільки метою є створення прямого числового прогнозу.

Для завдань бінарної класифікації введення порогу прийняття рішення може бути необов'язковим на вихідному рівні. Цей поріг визначає точку, в якій передбачена ймовірність інтерпретується як позитивний або негативний результат, полегшуючи перетворення балів ймовірності в двійкові прогнози.

Нарешті, показники оцінки, які використовуються для оцінки продуктивності моделі, такі як точність, точність, запам'ятовування, оцінка F1 або середня квадратична помилка, покладаються на прогнози, створені вихідним рівнем. По суті, конструкція та конфігурація вихідного рівня відіграють ключову роль у визначенні ефективності та надійності нейронної мережі в призначеному для неї застосуванні.

2.2. Ключові ідеї застосування згорткових нейронних мереж

Згорткові нейронні мережі (CNN) містять кілька ключових ідей, які давно мотивують дослідників комп'ютерного зору. Однією з фундаментальних концепцій є використання локальних рецептивних полів, натхненних ідеєю, що нейрони пов'язані з невеликими локалізованими областями вхідних даних. Це дозволяє мережі зосередитися на захопленні ієрархічних функцій, просуваючись від простих моделей до більш складних структур.

Іншою ключовою ідеєю є розподіл ваги, коли однаковий набір ваг (фільтрів або ядер) застосовується до різних просторових положень у вхідних даних. Ця концепція ґрунтується на меті досягнення інваріантності перекладу, гарантуючи, що мережа може розпізнавати шаблони незалежно від їхнього конкретного розташування.

Поняття ієрархії ознак відіграє значну роль у CNN. Нижні шари призначені для захоплення основних функцій, таких як краї та текстури, тоді як вищі шари поєднують ці функції для представлення більш складних структур. Це ієрархічне виділення ознак відображає людське візуальне сприйняття та розуміння.

Шари об'єднання в CNN спираються на концепцію зменшення вибірки, сприяючи зменшенню просторових розмірів у картах функцій. Об'єднання допомагає контролювати складність обчислень, усувати менш релевантну інформацію та створювати більш абстрактне представлення вхідних даних.

Акцент на інваріантності перекладу узгоджується з ширшою метою досягнення надійності в таких завданнях, як розпізнавання об'єктів, де положення об'єкта в зображенні не повинно впливати на його ідентифікацію [17].

CNN також вирішують проблему високої розмірності, використовуючи спільні ваги та методи зменшення вибірки, що призводить до значного зменшення кількості параметрів порівняно з повністю підключеними мережами. Це підвищує ефективність обчислень.

Збільшення даних, техніка, яка використовується під час навчання, мотивується ідеєю введення варіативності в набір даних. Випадкові перетворення, такі як повороти та перевероти, допомагають моделі краще узагальнювати невидимі дані та ставати надійнішими.

По суті, CNN втілюють ці фундаментальні ідеї щодо автоматичного вивчення та вилучення функцій із візуальних даних, розпізнавання шаблонів і досягнення чудової продуктивності в різних завданнях комп'ютерного зору, включаючи класифікацію зображень, виявлення об'єктів і сегментацію.

2.3. Недоліки та обмеження згорткових нейромереж

Хоча згорткові нейронні мережі (CNN) довели високу ефективність у різних задачах комп'ютерного зору, вони мають певні обмеження, які дослідники та практики мають враховувати. Розуміння цих обмежень має вирішальне значення для розробки надійніших моделей і вивчення альтернативних архітектур. Деякі помітні обмеження CNN включають:

- інваріантність інтерпритації обмежена даними навчання: CNN розроблені таким чином, щоб бути інваріантними до інтерпритації, тобто вони можуть розпізнавати шаблони незалежно від їхньої позиції у вхідних даних. Однак ця інваріантність інтерпритації часто обмежується шаблонами, присутніми в навчальних даних. Якщо об'єкт з'являється в неочікуваному місці чи орієнтації, моделі може бути складно ефективно узагальнити;
- просторові ієрархії та глобальний контекст: незважаючи на те, що CNN вправно вловлюють місцеві шаблони та особливості, вони можуть зіткнутися з проблемами під час включення глобального контексту та розуміння зв'язків між віддаленими елементами зображення. Це обмеження може бути критичним у завданнях, які потребують цілісного розуміння сцени;
- обмежене поводження з варіаціями обертання та масштабу: CNN можуть мати проблеми з варіаціями в масштабі та ротації. Якщо навчальні дані не охоплюють належним чином ці варіації, модель може продемонструвати неоптимальну продуктивність, якщо представити обернені або різномасштабовані екземпляри об'єктів;
- чутливість до вхідних збурень: CNN можуть бути чутливими до невеликих змін у вхідних даних, відомих як змагальні атаки. Ці атаки передбачають внесення непомітних змін до вхідних даних, що змушує модель робити неправильні прогнози. Стійкість змагальності є активною сферою досліджень для усунення цього обмеження;
- велика кількість параметрів: архітектури CNN, особливо глибші, можуть мати велику кількість параметрів. Незважаючи на те, що це дозволяє моделі вивчати

складні представлення, це також збільшує ризик переобладнання, особливо коли навчальні дані обмежені. Щоб пом'якшити цю проблему, часто використовуються методи регуляризації та архітектурні модифікації;

- залежність від позначених даних: CNN покладаються на дані з мітками для контрольованого навчання, і їх продуктивність сильно залежить від якості та кількості навчальних даних. Отримання великих, різноманітних і добре позначених наборів даних для певних завдань може бути складним завданням і може обмежити здатність моделі до узагальнення;

- обчислювальна інтенсивність: навчання глибоких CNN може бути обчислювально інтенсивним, вимагаючи значних обчислювальних ресурсів, особливо для великомасштабних наборів даних і складних архітектур. Це може створити проблеми для розгортання в середовищах з обмеженими ресурсами;

- можливість тлумачення: CNN часто розглядають як моделі чорної скриньки, і інтерпретація вивчених уявлень і процесів прийняття рішень може бути складною. Розуміння того, чому CNN робить певний прогноз, має вирішальне значення, особливо в програмах, де інтерпретація є важливою, наприклад, у сфері охорони здоров'я чи фінансів.

Дослідники та практики продовжують усувати ці обмеження шляхом постійного вдосконалення архітектури моделей, стратегій навчання та розробки гібридних моделей, які поєднують сильні сторони CNN з іншими методами. Незважаючи на свої обмеження, CNN залишаються основоположними та потужними інструментами в області комп'ютерного зору.

2.4. Висновки до розділу 2

В даному розділі були розглянуті основні принципи та структура CNN, які визначають їхню ефективність у розв'язанні завдань машинного навчання. Було висвітлено значущі аспекти, такі як локальність ієрархічної інформації, які використовуються у шарах згортки, що дозволяє нейронним мережам ефективно впізнавати та адаптуватися до різних характеристик вхідних даних.

Підводячи підсумок, архітектура CNN слідує послідовному потоку від вхідного рівня через приховані рівні (згортковий, активаційний, об'єднаний, повністю підключений) до вихідного рівня, причому кожен рівень відіграє окрему роль у вилученні функцій і прийнятті рішень.

Згортковий рівень у CNN є ключовим елементом для виділення ознак, використовуючи навчальні фільтри для захоплення просторових шаблонів у вхідних даних. Його здатність виявляти локальні особливості та ієрархічно будувати представлення робить його ключовим компонентом успіху CNN у задачах комп'ютерного зору.

Використання функцій активації надає необхідною нелінійність мережі, допомагаючи їй краще адаптуватися до складних взаємозв'язків в даних. Такі функції активації є незамінними в нейронних мережах, оскільки вони надають необхідні інструменти для моделі для захоплення та вивчення складних моделей і зв'язків у даних. Їх впровадження дозволяє нейронним мережам досягти успіху в широкому спектрі складних завдань, що робить їх потужними інструментами в сфері машинного навчання та штучного інтелекту.

Rectified Linear Unit (ReLU) є широко поширеною функцією активації завдяки своїй здатності вводити нелінійність, пом'якшувати проблему зникаючого градієнта та сприяти ефективному та вибірково навчанню функцій у нейронних мережах, що робить її ключовим елементом для успішного навчання моделі та продуктивності.

Повністю зв'язаний рівень у CNN діє як міст між просторовою ієрархією характеристик, отриманих попередніми рівнями, і остаточним вихідним рівнем, дозволяючи мережі робити прогнози або класифікації на основі комплексної комбінації функцій, отриманих із вхідних даних. Його підключення та нелінійність роблять його ключовим елементом успіху CNN у різних програмах машинного навчання та комп'ютерного зору.

РОЗДІЛ 3

РОЗРОБКА ЗАСТОСУНКУ ДЛЯ ВИЯВЛЕННЯ ЛІТАКІВ

3.1. Використання згорткової нейронної мережі

В сучасному дослідженні та розробці систем розпізнавання об'єктів за допомогою комп'ютерного зору, особливу увагу приділяють вибору оптимального алгоритму для досягнення найвищої точності та надійності в різних умовах. У контексті виявлення літальних апаратів, використання глибоких згорткових нейронних мереж (CNN) виявляється особливо вдалим вибором, і це обумовлено численними перевагами цього підходу:

1. Здатність до ієрархічного вивчення ознак: CNN володіють унікальною здатністю автоматично визначати та використовувати ієрархічні ознаки на зображеннях. Вони можуть розпізнавати високорівневі абстракції, такі як форми та структури, а також низькорівневі деталі, що дозволяє ефективно адаптуватися до великого різноманіття літальних апаратів.

2. Автоматичне визначення характеристик: CNN можуть автоматично вивчати необхідні характеристики об'єктів під час навчання, що вирішує проблему ручного визначення ознак та спрощує процес розробки системи розпізнавання.

3. Ефективність у вирішенні завдань зорового сприймання: завдяки своїй спроможності пристосовуватися до локальних та глобальних змін на зображеннях, CNN виявляються ефективними в умовах зміни масштабу, поворотів, а також при наявності об'єктів на фоні різних зовнішніх факторів.

4. Здатність до масштабованої обробки даних: CNN можуть успішно працювати з великим обсягом даних, що робить їх ідеальними для використання у завданнях, пов'язаних із зображеннями літальних апаратів, де доступні великі набори

Кафедра КІТ (47)				НАУ 23 23 66 000 ПЗ			
Виконав	Шемідько А.О.			РОЗРОБКА ЗАСТОСУНКУ ДЛЯ ВИЯВЛЕННЯ ЛІТАКІВ	Літера	аркуш	аркушів
Керівник	Савченко А.С.					51	20
Консульт.					УС-211М 122		
Н. контроль	Райчев І.Е.						

даних для навчання.

5. Автоматична локалізація об'єктів: CNN взаємодіють із зображеннями як цілісними об'єктами, що дозволяє їм ефективно локалізувати та розпізнавати об'єкти на зображеннях, включаючи розпізнавання літальних апаратів та їхніх складових частин.

Зазначена комбінація переваг робить CNN потужним інструментом для завдань розпізнавання об'єктів на зображеннях, зокрема для ефективного розв'язання важливих проблем в області безпеки, пов'язаних із виявленням літальних апаратів.

3.1.1. Підготовка даних до навчання

Створення комплексного набору даних для навчання згорткової нейронної мережі (CNN) передбачає два важливі кроки: отримання відповідних зображень і ретельне їх анотування. Ці кроки мають ключове значення для забезпечення того, щоб набір даних був не тільки різноманітним і репрезентативним, але й точно позначеним, що сприяє ефективному навчанню та узагальненню CNN у сценаріях реального світу.

Коли зображення отримано, процес анотації стає вирішальним. Кожне зображення проходить ретельну анотацію, позначаючи точні координати об'єктів, зокрема місця розташування літальних апаратів. Використання таких інструментів, як обмежувальні рамки або багатокутники, забезпечує точну локалізацію об'єктів. Встановлення чітких стандартів анотації та вказівок має важливе значення для підтримки узгодженості в наборі даних, що дозволяє CNN вивчати надійні шаблони під час навчання.

Вибір відповідних інструментів анотації – це наступне питання. Ці інструменти мають полегшити ефективне маркування об'єктів, пропонуючи такі функції, як малювання обмежувальних рамок (Рис. 3.1), маркування та експорт анотованих даних у стандартизованих форматах. У випадках із великою кількістю зображень використання автоматизованих інструментів або служб анотацій може допомогти прискорити процес анотації, зберігаючи при цьому точність [18].

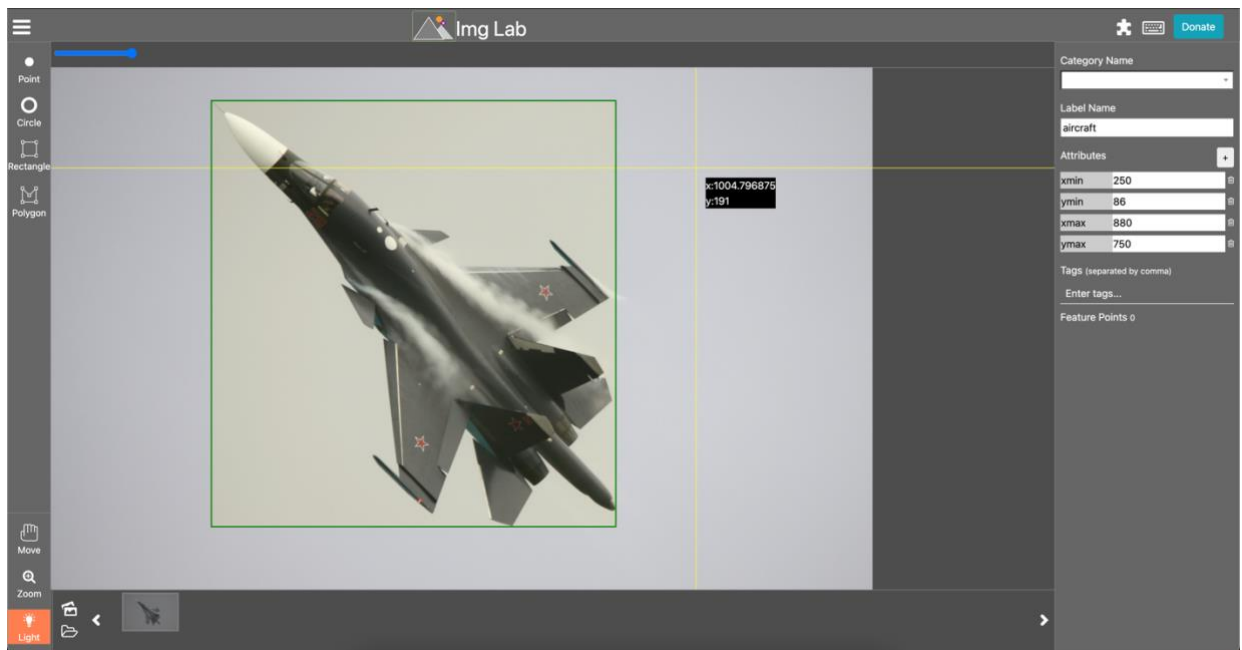


Рис. 3.1. Приклад формування обмежувальних рамок для зображення

Перед інтеграцією зображень у набір даних проводиться ретельний перегляд анотацій, щоб виявити та виправити будь-які неточності чи невідповідності. Цей крок забезпечує високу якість і надійність анотованих даних. Подвійна перевірка того, що анотації відповідають заздалегідь визначеним стандартам, включаючи правильне маркування та точні межі об'єктів, має вирішальне значення для успіху процесу навчання CNN.

В результаті використання таких застосунків анотації формується вихідний файл текстового, або табличного (.csv) форматів, в якій записується основна інформація необхідна для формування навчальних датасетів нейромережі (Рис. 3.2).

	A	B	C	D	E	F	G	H
1	filename	width	height	class	xmin	ymin	xmax	ymax
2	0a3346cf65f6534ecc81600b51d31b8d	650	369	J20	264	138	420	252
3								

Рис. 3.2.

Документація відіграє ключову роль у процесі створення набору даних. Створення детальної документації щодо процесу анотації, включаючи стандарти анотації, використовувані інструменти та конкретні міркування щодо набору даних, служить цінним довідником для майбутньої роботи та забезпечує прозорість у процесі створення набору даних.

Нарешті, створення набору даних розглядається як ітеративний процес. Періодична перевірка та вдосконалення анотацій на основі відгуків і нових вимог сприяє постійному покращенню якості та відповідності набору даних. Ретельно збираючи різноманітні зображення та створюючи точні анотації, фахівці-практики закладають основу для високоякісного набору даних, необхідного для ефективного навчання CNN. Цей ретельний підхід гарантує, що мережа може вивчати надійні функції та добре узагальнювати, що в кінцевому підсумку призводить до підвищення продуктивності в розпізнаванні та класифікації літальних апаратів.

Однак перед початком формування датасету в нашому випадку потрібно здійснити повне об'єднання всіх анотацій в один файл анотації. Для цього було використано невеликий пайтон-скрипт (Рис. 3.3), що об'єднує вміст файлів починаючи з другого рядку, але через певні поточні обмеження розробленої мережі до вибірки попали лише зображення, що містять один літальний апарат на зображенні.

```

1 import os
2 import csv
3
4
5 usage
6 def merge_csv_files(input_dir, output_file):
7     with open(output_file, 'w', newline='') as out_csvfile:
8         writer = csv.writer(out_csvfile)
9
10        do_once = False
11
12        for filename in os.listdir(input_dir):
13            if filename.endswith('.csv'):
14                file_path = os.path.join(input_dir, filename)
15
16                with open(file_path, 'r') as in_csvfile:
17                    reader = csv.reader(in_csvfile)
18
19                    all_rows = []
20                    for row in reader:
21                        all_rows.append(row)
22                    if len(all_rows) != 2:
23                        continue
24
25                    if do_once == False:
26                        writer.writerow(all_rows[0])
27                        do_once = True
28                    writer.writerow(all_rows[1])
29
30        print(f"Merge data from CSV files in {output_file}")
31
32 if __name__ == '__main__':
33     input_directory = 'dataset'
34     output_csv_file = 'annotation.csv'
35     merge_csv_files(input_directory, output_csv_file)

```

	A	B	C	D	E	F	G	H
1	filename	width	height	class	xmin	ymin	xmax	ymax
2	ce0ef71518135727e09a984f07aa26	1600	1075	SR71	216	345	1195	586
3	769c27a7b2c804a73458175fd13a3a53	916	316	J20	303	183	684	282
4	ac6c99e73c2b4a6af7a9d4667209bd32	6016	4016	F4	1	1759	6016	4016
5	f8dcb25ac0f69b2a7b4408e9481cedc	1041	1314	F18	252	16	750	1312
6	9944e1a41b1fb0e33f665844209698e	1406	1757	B2	164	335	1190	1633
7	27af5e27f0875d39fb3d94cf39b8	1536	1097	F35	580	318	959	744
8	db14871376b9377a30737ce4d995a15	1920	1230	XB70	1	535	1920	1011
9	4a6b0a99c9e8c0b93ec1b6d46573b60	4096	2731	P3	372	599	4096	1556
10	414162b1c5414b5b3a2622a993d7ac1e	1280	854	J20	308	276	888	645
11	cd1659b5aac08ec32d4c42e03f4eca9	4096	2318	US2	1	1	4096	2318
12	28edf9a9a04732efc5401b722daf411e	1200	675	F15	362	247	499	294
13	1449ed1726c3f56ff6518244afcc828	1920	1049	AG600	413	564	1436	904
14	5ac91f8e70885f3f56662c42c3ea190	1724	2560	YF23	106	490	1613	1620
15	122d7e3c1a2931db9859248a41d004	2048	1368	P3	743	336	1412	516
16	049059fac3670dfe4590d712dfc57	720	720	B1	1	1	642	720
17	8604a8374312c60089119172617da70a	700	348	Tu160	18	67	700	258
18	594db96ab6413152a0e24f3ba438990	1800	1200	F16	597	368	1123	551
19	eb144e965b19e2e3020a0255ec64b7	1280	1920	F15	402	577	995	739
20	fh6dd4r177nareha2whh372he7h1fa2R	дiаr	3072	рi	1516	947	3141	1603

Рис. 3.3. Скрипт для формування кінцевого файлу анотацій (зліва) та результат його виконання (справа)

Згодом етап попередньої обробки даних має вирішальне значення для стандартизації набору даних і підготовки його для введення в CNN. Зміна розміру зображень до єдиного розміру забезпечує узгодженість, тоді як нормалізація значень пікселів до попередньо визначеного діапазону, зазвичай $[0, 1]$ або $[-1, 1]$, сприяє стабільному навчанню та конвергенції (Рис. 3.4). Крім того, методи розширення даних, такі як випадкове обрізання, обертання, зміщення та масштабування, можуть бути застосовані для розширення набору даних і покращення здатності моделі обробляти варіації в реальному середовищі.

```

1 usage
15 def parse_dataset():
16     batch_images = []
17     batch_labels = []
18
19     for _, row in df.iterrows():
20         image_path = os.path.join(image_folder, row['filename'])
21         image = cv2.imread(image_path + ".jpg")
22         original_height, original_width, _ = image.shape
23         image = cv2.resize(image, image_size)
24         # image = image.reshape((-1, image_size[0], image_size[1], 3))
25         image = image / 255.0 # Image data normalization [0, 1]
26
27         batch_images.append(image)
28
29         width = row['xmax'] - row['xmin']
30         height = row['ymax'] - row['ymin']
31
32         box = np.array([
33             row['xmin'] * image_size[0] / original_width,
34             row['ymin'] * image_size[1] / original_height,
35             width * image_size[0] / original_width,
36             height * image_size[1] / original_height
37         ])
38         label = np.append(box, values: 1)
39
40         batch_labels.append(label)
41
42     return batch_images, batch_labels

```

Рис. 3.4. Скрипт для формування стандартизаційного набору вхідних даних

В поточній реалізації формування вхідного датасету використовуючи підготовлений файл анотацій відбувається покрокова підготовка всіх зображень, що в ньому згадувались. За допомогою модулю OpenCV зображення відкривається, перетворюючись у форму матриці [ширина]x[висота]x[3 кольорові канали]. Далі змінюємо розмір зображення за допомогою того самого модуля в той, що підходить для нашої моделі і наостанок нормалізуємо цей масив даних зображення в діапазон [0;1]. Наступна частина використовуючи дані з файлу анотацій формує дані про розташування об'єкта на зображенні, а також адаптує це розташування під зміну розміру зображення.

Наступний крок передбачає випадкове поділ набору даних на набори для навчання та перевірки, необхідні для точної оцінки продуктивності моделі під час навчання та забезпечення її хорошого узагальнення для невидимих даних (Рис. 3.5.). Збалансування розподілу класів в обох наборах допомагає запобігти упередженням і сприяє надійному вивченню функцій, пов'язаних з кожним класом.


```

2 usages
45 def split_dataset():
46
47     dataset = parse_dataset()
48
49     # Extracting features and labels
50     features = dataset[0]
51     labels = dataset[1]
52
53     # Splitting the dataset into train and test
54     features_train, features_test, labels_train, labels_test = train_test_split(*arrays: features, labels, test_size=0.2, random_state=42)
55
56     # Further splitting the test set into validation and test
57     features_val, features_test, labels_val, labels_test = train_test_split(*arrays: features_test, labels_test, test_size=0.5, random_state=42)
58
59     # Print the sizes of the datasets
60     print(f"Train set size: {len(features_train)}")
61     print(f"Validation set size: {len(features_val)}")
62     print(f"Test set size: {len(features_test)}")
63
64     def to_tensor(np_feature, np_label):
65         #X = np.expand_dims(np_feature, axis=0)
66         X = tf.convert_to_tensor(np_feature, dtype=tf.float32)
67         np_label = tf.convert_to_tensor(np_label, dtype=tf.float32)
68
69         result = tf.data.Dataset.from_tensor_slices((X, np_label))
70         return result
71
72     return to_tensor(features_train, labels_train), to_tensor(np.array(features_val), np.array(labels_val)), to_tensor(np.array(features_test), np.

```

Рис. 3.5. Скрипт для розділення датасету на три вибірки (навчальну, валідаційну та тестову)

Техніки збільшення даних ще більше підвищують різноманітність набору даних шляхом застосування випадкових перетворень до зображень під час навчання. Цей крок має вирішальне значення для підвищення стійкості моделі до змін масштабу, обертання та інших факторів середовища.

Використання таких інструментів, як ImageDataGenerator від TensorFlow Keras, автоматизує створення доповнених пакетів зображень під час навчання, зменшуючи вимоги до пам'яті та оптимізуючи процес навчання для ефективної обробки великих наборів даних.

Забезпечення збалансованого представлення кожного класу в наборі даних має вирішальне значення для того, щоб модель не віддала перевагу одному класу над іншим під час навчання. Незбалансовані набори даних можуть призвести до упереджених моделей, тому слід ретельно розглянути питання збалансування класів, особливо в сценаріях, коли певні класи, наприклад літальні апарати, можуть бути менш поширеними.

3.1.2. Архітектура моделі

Вибір архітектури згорткової нейронної мережі (CNN) для цього завдання ґрунтується на ретельному розгляді, спрямованому на оптимізацію продуктивності для конкретного завдання розпізнавання літальних апаратів. Обрана архітектура є результатом стратегічного балансу між складністю моделі, обчислювальною ефективністю та здатністю фіксувати складні ієрархічні особливості, притаманні аерофотознімкам.

Обґрунтування вибору саме цієї архітектури CNN полягає в її доведеній ефективності при виконанні складних завдань візуального розпізнавання. Властива архітектурі здатність автоматично витягувати ієрархічні характеристики із зображень бездоганно узгоджується зі складними візерунками та різноманітними зовнішніми ознаками, пов'язаними з літальними апаратами. Його згорткові шари вправно вловлюють локальні шаблони, тоді як наступні шари агрегують ці функції ієрархічно, дозволяючи мережі розпізнавати складні деталі та просторові відносини, важливі для точного розпізнавання.

Крім того, вибрана архітектура демонструє універсальність у адаптації до варіацій у масштабі, орієнтації та фоні, що важливо під час роботи з різними повітряними сценаріями. Її глибока багаторівнева структура дозволяє вивчати абстрактні представлення, сприяючи надійності моделі в розпізнаванні літальних апаратів у різних умовах.

Архітектура вибраної згорткової нейронної мережі (CNN) характеризується чітко визначеною структурою, ретельно розробленою для оптимізації продуктивності для завдання розпізнавання літальних апаратів. Вхідний шар нейромережі визначений наступними параметрами: на вхід подається масив, що репрезентує зображення шириною і висотою 224 та 3 канали кольорів (Рис.3.6.). Далі знаходяться 5 згорткових шарів, за кожним з яких йдуть шари об'єднання, після яких слідує повністю зв'язаний шар, створюючи ієрархічну мережу, здатну вивчати складні характеристики з аерофотознімків [19].

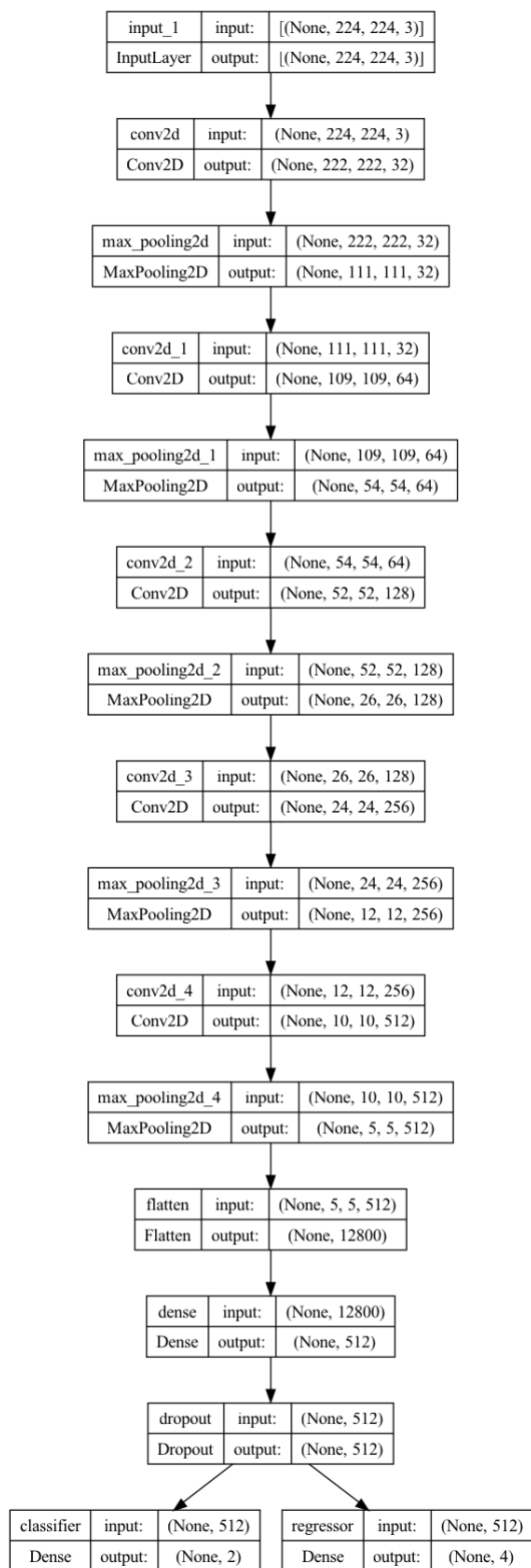


Рис. 3.6. Структура згорткової нейромережі

Вкінці мережі знаходиться шар довільного виключення, який призначений для того, щоб не виникало перенавчання мережі, за яким знаходяться 2 виходи

нейромережі – класифікатор, що встановлює чи є на зображенні літальний апарат, та регресор, задача якого полягає у виявленні розташування літального апарату на зображенні.

Початкові згорткові шари служать екстракторами функцій, використовуючи фільтри для виявлення локальних шаблонів і особливостей у вхідних зображеннях. Ці шари є інструментальними для захоплення низькорівневих деталей, пов'язаних із літальними апаратами, таких як краї, текстури та основні форми. Операції переміщення та об'єднання ще більше зменшують дискретизацію просторових розмірів, підвищуючи здатність мережі зосереджуватися на основних функціях, одночасно зменшуючи обчислювальну складність.

Після згорткових шарів шари об'єднання стратегічно вкраплюються, щоб поступово зменшувати просторові розміри карт функцій, сприяючи вилученню ієрархічних функцій. Ця просторова абстракція має вирішальне значення для того, щоб модель розрізняла моделі в різних масштабах, орієнтаціях і положеннях на аерофотознімках.

Повністю пов'язані рівні в кінці архітектури служать класифікаторами, поєднуючи абстрактні функції, щоб робити прогнози щодо присутності конкретних літальних апаратів. Рівень виводу зазвичай використовує відповідну функцію активації, таку як `softmax`, для отримання розподілу ймовірностей по різних класах, сприяючи ефективній класифікації.

З точки зору кількості рівнів, архітектура врівноважує глибину та складність для досягнення оптимальної продуктивності. Це включає в себе розумний вибір згорткових і об'єднаних шарів із варіаціями розмірів фільтрів і глибин каналів для захоплення різноманітних особливостей, притаманних аерофотознімкам. Таким чином розроблена структура містить 8125766 параметрів, що можуть використовуватись для навчання (Рис. 3.7).

```
=====  
Total params: 8125766 (31.00 MB)  
Trainable params: 8125766 (31.00 MB)  
Non-trainable params: 0 (0.00 Byte)  
=====
```

Рис. 3.7. Висновок по структурі мережі

Загальна структура моделі розроблена таким чином, щоб її можна було адаптувати, що дозволяє ефективно вивчати та узагальнювати різноманітні повітряні сценарії. Завдяки модульній природі архітектури можна безперешкодно інтегрувати навчання з передаванням, використовуючи попередньо підготовлені моделі на великомасштабних наборах даних для підвищення можливостей розпізнавання навіть з обмеженими даними, пов'язаними з конкретними завданнями.

Таким чином, обрана структура моделі CNN є добре організованою композицією згорткових, об'єднаних і повністю пов'язаних шарів, що створює ієрархічну мережу, здатну вивчати та узагальнювати складні характеристики з аерофотознімків. Ретельно збалансована архітектура забезпечує ефективне розпізнавання, адаптивність до різноманітних умов і полегшує включення стратегій перенесення навчання для підвищення надійності моделі.

3.1.3. Навчання моделі

Вибрані параметри навчання розроблені для досягнення балансу між швидкою конвергенцією та уникненням надмірного оснащення.

Швидкість навчання: критичний гіперпараметр, встановлено на оптимальне значення для регулювання розміру кроку під час градієнтного спуску. Вибирається помірна швидкість навчання, щоб запобігти перевищенню глобального мінімуму та забезпечити постійну конвергенцію, особливо враховуючи потенційну складність завдання розпізнавання літального апарату.

Кількість епох: ретельно визначається шляхом експериментів, збалансовуючи час навчання моделі та запобігаючи переналаштуванню. Вибирається достатня, але не надмірна кількість епох, щоб дозволити моделі вивчати різноманітні характеристики та шаблони з набору даних без ризику запам'ятовування.

Розмір партії: впливає на кількість оброблених зразків перед оновленням ваг моделі, вибирається на основі обчислювальної ефективності та обмежень пам'яті. Вибирається розмір партії, який оптимально використовує доступні ресурси, одночасно сприяючи надійному навчанню.

Інші параметри: Додаткові параметри, такі як імпульс і зниження ваги, точно налаштовані, щоб доповнити швидкість навчання та підвищити стабільність тренування. Ці параметри сприяють зниженню ризику застрягання моделі в локальних мінімумах і допомагають досягти кращого узагальнення [18].

Функція втрати: вибір відповідної функції втрат є ключовим у скеруванні моделі під час навчання до мінімізації помилок передбачення. Для завдання розпізнавання літальних апаратів використовується категорійна функція крос-ентропійних втрат. Ця функція добре підходить для завдань багатокласової класифікації, накладаючи більші санкції на модель за більші відхилення від основної істини.

Обґрунтування вибору категоріальної крос-ентропії полягає в її ефективності в навчанні CNN для класифікації зображень. Заохочуючи модель призначати вищі ймовірності для правильних класів, ця функція втрат узгоджується з метою точного розпізнавання різноманітних літальних апаратів.

Використовуючи алгоритм що формує датасет отримуємо три набори даних: навчальний, валідаційний та тестовий. В даній ітерації навчання кількість віх визначена в 1000 разів, а розмір партії вираховується з загальної кількості вхідних даних поділених на BATCH_SIZE, що визначений константою 32. При навчанні результат найкращої віхи буде автоматично зберігатись беручи до уваги метрику втрати, тобто коли її значення буде найменшим, тоді модель буде презбережена (Рис. 3.8.).

```

138 train_df, valid_df, test_df, tr_num, val_num, test_num = split_dataset()
139
140 train_df = tune_training_ds(train_df)
141 valid_df = tune_validation_ds(valid_df)
142 test_df = tune_test_ds(test_df)
143
144 epochs = 1000
145 checkpoint = ModelCheckpoint('airplane_detector_model.h5',
146                             monitor='val_loss',
147                             save_best_only=True,
148                             mode='min', verbosity=1)
149
150 history = model.fit(
151     train_df,
152     steps_per_epoch=tr_num // BATCH_SIZE,
153     epochs=epochs,
154     validation_data=valid_df,
155     validation_steps=val_num // BATCH_SIZE,
156     callbacks=[checkpoint]
157 )
158
159 plot_training(history)
160
161 # Evaluate the model
162 train_score = model.evaluate(train_df, steps=test_num // BATCH_SIZE, verbose=1)
163 valid_score = model.evaluate(valid_df, steps=test_num // BATCH_SIZE, verbose=1)
164 test_score = model.evaluate(test_df, steps=test_num // BATCH_SIZE, verbose=1)
165
166 print("Train Loss: ", train_score[0])
167 print("Train Accuracy: ", train_score[1])
168 print('-' * 20)
169 print("Validation Loss: ", valid_score[0])
170 print("Validation Accuracy: ", valid_score[1])
171 print('-' * 20)
172 print("Test Loss: ", test_score[0])
173 print("Test Accuracy: ", test_score[1])

```

Рис. 3.8. Скрипт навчання моделі на визначеному датасеті

В результаті навчання формуються результати у вигляді графіків. На них визначаються як змінюються втрати і точність моделі на двох вибірках: навчальній і валідаційній (Рис. 3.9).

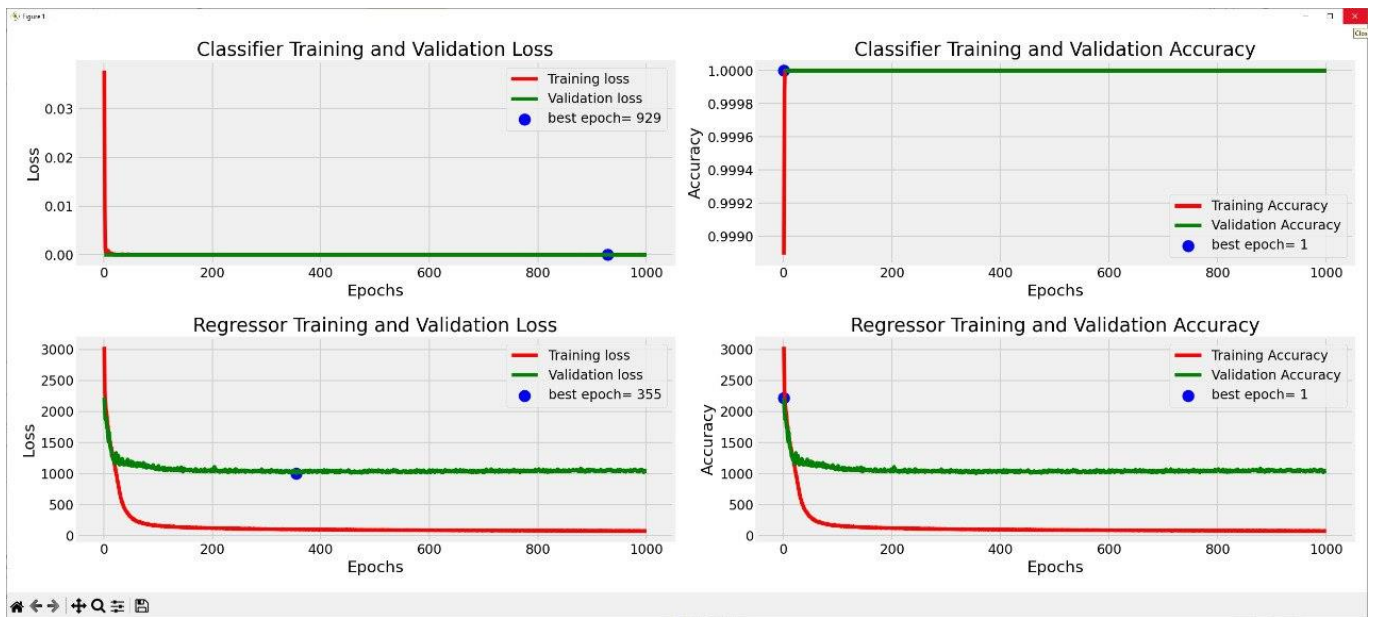


Рис. 3.9. Графіки результатів навчання моделі за 1000 віх

Підсумовуючи, параметри навчання вибрано продумано, щоб забезпечити стабільну конвергенцію та запобігти переобладнанню, з урахуванням швидкості навчання, кількості епох і розміру партії. Категоріальна функція перехресної ентропійної втрати використовується для того, щоб скерувати модель до точної багатокласової класифікації, узгоджуючи її з тонкощами завдання розпізнавання літальних апаратів. Ця параметризація сприяє ефективному навчанню CNN для оптимальної продуктивності в реальних сценаріях.

3.2. Використання навченої моделі для виявлення образів

Додаток, розроблений на Python, плавно інтегрує навчену модель для ефективного виявлення зображень. Завдяки можливості відкривати та обробляти зображення, програма використовує навчену згортову нейронну мережу (CNN), щоб робити прогнози та ідентифікувати присутність літаків. Вхідні зображення піддаються необхідним перетворенням для узгодження з конкретними вхідними вимогами мережі, забезпечуючи оптимальну продуктивність.

Процес відкриття зображення здійснюється функцією, що використовує системний інтерфейс для вибору файлів, зазначивши фільтр у вигляді файлів-зображень (Рис. 3.10).

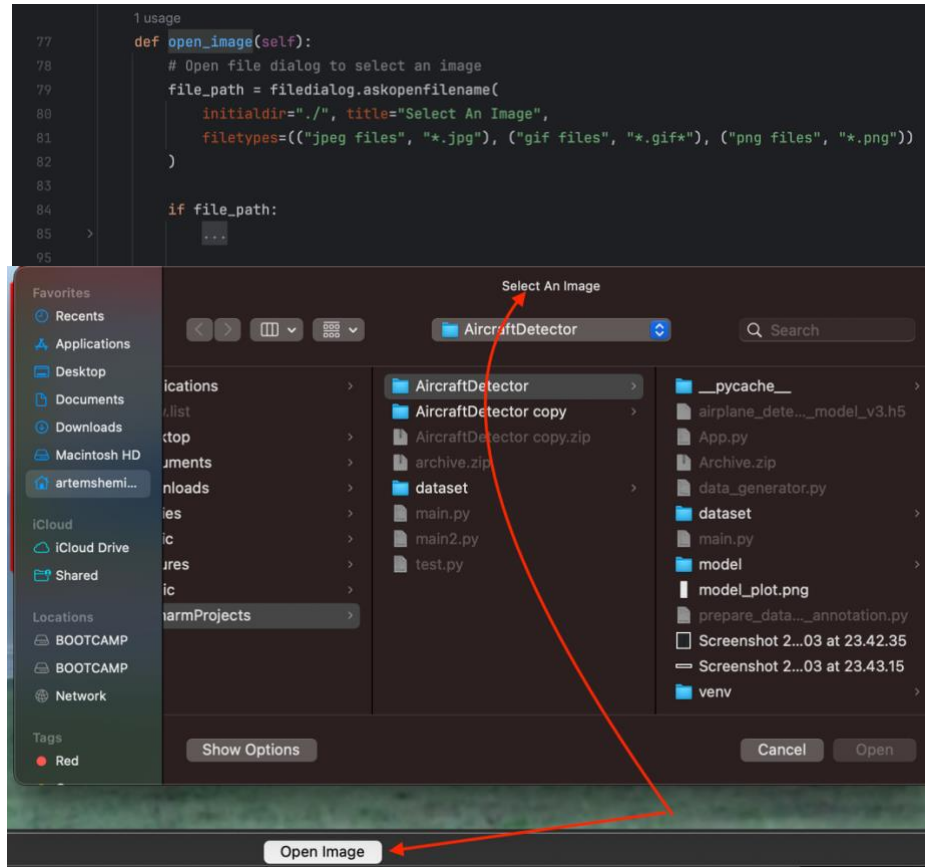


Рис. 3.10. Скрипт для формування панелі вибору зображень та результат роботи в застосунку

Програма відкриває вхідні зображення, готуючи етап для процесу виявлення. Ці зображення можуть бути отримані з різних каналів, таких як завантаження користувачів або канали в реальному часі

Після вибору зображення і перед тим як відправити його на опрацювання нейромережею воно піддається змінам аналогічним, як ті, що були при навчанні моделі (Рис. 3.11), а саме: зображення підганяють під розміри вхідного шару нейромережі, та нормалізують величину значень пікселів.

```

1 usage
44 def preprocess_image(image_path):
45     img = image.load_img(image_path, target_size=image_size)
46     img_array = image.img_to_array(img)
47     img_array = np.expand_dims(img_array, axis=0)
48     img_array /= 255.0 # Normalize pixel values if needed
49     return img_array
50

```

Рис. 3.11. Скрипт для обробки зображення перед подачею в неймережу

Після цього дані подаються в неймережу, а на виході з якої отримуються результат «передбачення». Модель використовує свої вивчені функції, щоб робити прогнози щодо присутності та розташування літаків на зображеннях. Цей результат надалі адаптується під екранні розміри вхідного зображення, та відмальовується прямокутник зеленого кольору, позицію та розміри якого були отримані з результату роботи неймережі та адаптовані під поточне зображення (Рис. 3.12).

```

97 predictions = self.model.predict(img_array)
98
99 bounding_box_coordinates = predictions[1]
100
101 # Load the original image for visualization
102 self.original_image = cv2.imread(file_path)
103 self.original_image = cv2.cvtColor(self.original_image, cv2.COLOR_BGR2RGB) # Convert from BGR to RGB
104
105 # Get the original image shape
106 original_shape = self.original_image.shape
107
108 # Rescale bounding box coordinates to the original image size
109 x, y, width, height = bounding_box_coordinates[0]
110 x *= original_shape[1] / image_size[0]
111 y *= original_shape[0] / image_size[1]
112 width *= original_shape[1] / image_size[0]
113 height *= original_shape[0] / image_size[1]
114
115 # Convert to integers
116 x, y, width, height = int(x), int(y), int(width), int(height)
117
118 # Draw the bounding box on the image
119 self.original_image = cv2.rectangle(self.original_image, (x, y), (x + width, y + height), (0, 255, 0), 4)
120

```

Рис. 3.12. Скрипт, що викликає обробку зображення мережею, та адаптує результат

Після висновку моделі виконуються кроки постобробки для уточнення прогнозів. Обмежувальні прямокутники часто використовуються для візуалізації та точної локалізації виявленого літака на зображеннях. Ці обмежувальні прямокутники інкапсулюють просторову протяжність розпізнаних об'єктів, пропонуючи візуальне представлення результатів виявлення моделі.

В результаті роботи застосунку маємо наступний вивід програми:

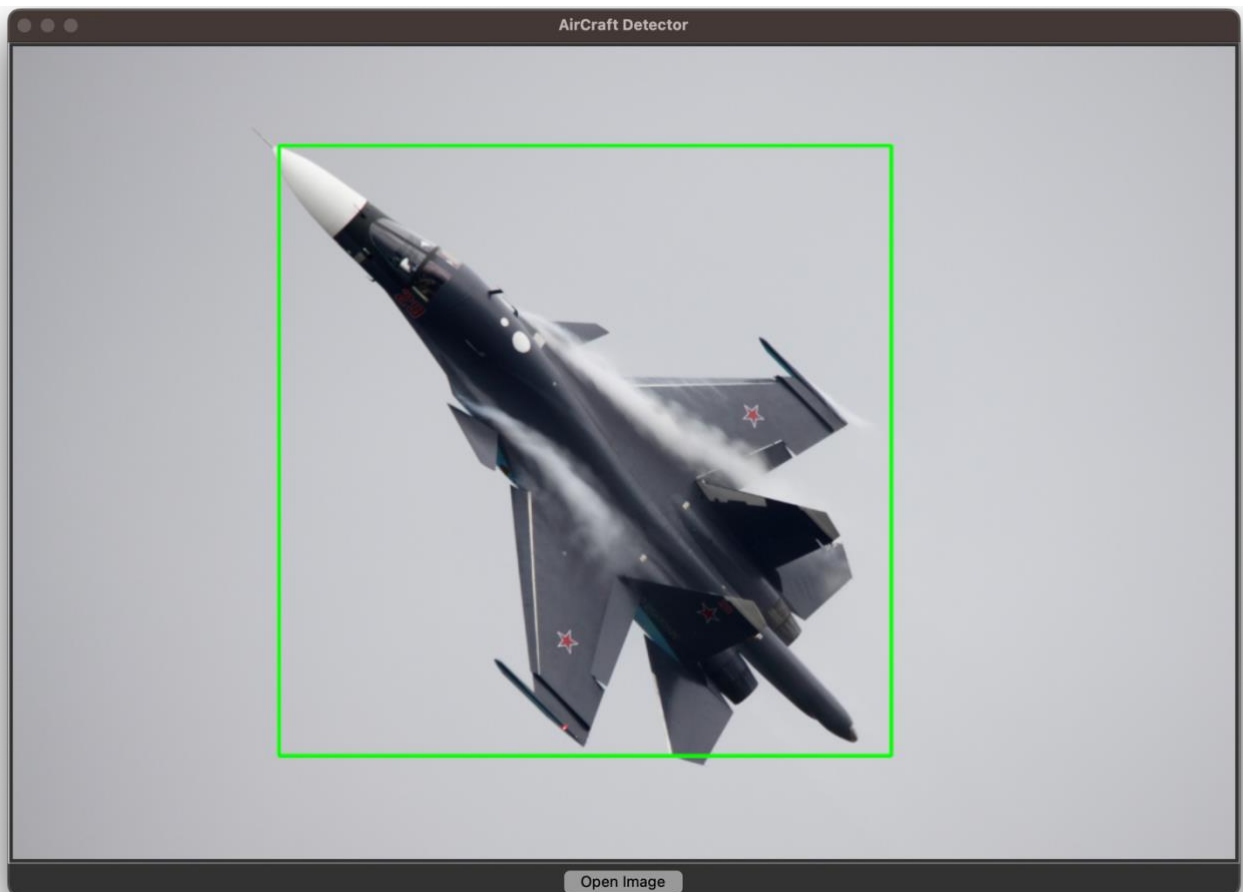


Рис. 3.13. Результат роботи застосунку

На додаток до процесу локалізації, програма обчислює коефіцієнт перетину через з'єднання (IOU). Ця метрика оцінює перекриття між прогнозованими обмежувальними рамками та обмежувальними рамками правдивості на землі, забезпечуючи кількісну міру точності моделі в просторовій локалізації (Рис. 3.14). Дане обчислення можливе лише у випадку наявності файлу з анотаціями, що розташовується поряд зі зображенням, що подавалось на вхід мережі.

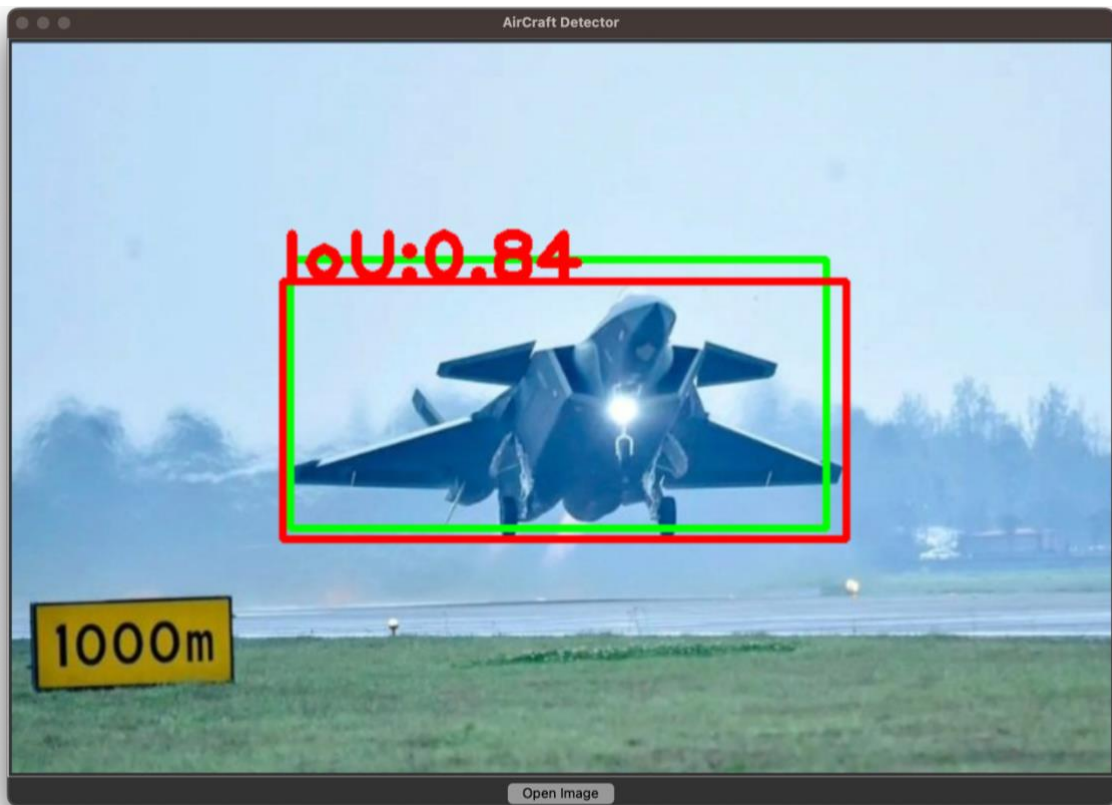


Рис. 3.14. Результат обчислення IoU для справжніх та отриманих моделю результатів

Остаточний результат програми включає візуально анотовані зображення з обмежувальними рамками навколо виявлених літаків. Цей вихід служить не лише практичною демонстрацією можливостей виявлення моделі, але й дає цінну інформацію про просторову точність прогнозів за допомогою коефіцієнтів IOU.

3.3. Швидкодія та точність

Незважаючи на властиву обчислювальну складність і допустиму похибку в обчисленнях, система демонструє вражаючу швидкість, досягаючи виявлення зображення в реальному часі. Зокрема, програма швидко обробляє зображення високої роздільної здатності (1600×1068) із часом виявлення 25 мілісекунд на зображення. Ця висока швидкість має вирішальне значення для реальних додатків,

гарантуючи, що система може ефективно аналізувати та реагувати на вхідні дані майже в реальному часі.

Здатність системи підтримувати такий швидкий час обробки, навіть з обчислювальними вимогами до зображень із високою роздільною здатністю, підкреслює її ефективність і оперативність. Середній бал Intersection over Union (IOU) від 50% до 70% відображає надійність системи в точному визначенні літальних апаратів на зображеннях. Зусилля щодо оптимізації балансу між обчислювальною складністю та швидкою обробкою сприяють загальній ефективності системи, роблячи її цінним інструментом для швидкого й точного виявлення літаків на основі зображень.

3.4. Обмеження

Даний застосунок, що базується на власній згортковій нейромережі, має деякі обмеження. Зокрема, архітектура мережі містить однооб'єктність виявлення, тобто на зображенні може виявляти лише один літальний апарат, що може стати проблемою в умовах, де присутні більше ніж один літак на зображенні. Крім того, система може демонструвати низьку точність на низькороздільних зображеннях, що може ускладнити правильне розпізнавання об'єктів, особливо в сценаріях з обмеженою якістю зображень.

Також важливо враховувати, що застосунок може виявити себе менш ефективним у випадках обмеженої обчислювальної потужності, оскільки згорткові нейромережі можуть вимагати значних обчислювальних ресурсів для навчання та впровадження в реальні умови.

3.5. Висновки до розділу 3

Використання добре налагодженої архітектури згорткової нейромережі полегшує процес навчання та підвищуючи здатність моделі ефективно

узагальнювати, навіть з обмеженими даними, пов'язаними з конкретними завданнями. Вибрана архітектура CNN забезпечує ефективність, демонструючи здатність ефективно вивчати та узагальнювати складні характеристики з зображень літальних апаратів. Його підтверджена успішність у подібних завданнях, адаптованість до різноманітних умов і здатність до передачі навчання разом роблять його ідеальним вибором для досягнення оптимальної продуктивності в конкретному завданні розпізнавання літальних апаратів.

Під час оцінки навчання моделі були використані ключові метрики, що визначають якість та ефективність розпізнавання літаків. Однією з таких метрик є функція втрат, яка відображає різницю між прогнозованими та фактичними значеннями.

Окрім того, аналізувалася метрика точності, яка визначає, яку частину зображень модель правильно визнала. Графіки точності демонстрували стабільне підвищення впродовж процесу навчання, що підтверджує успішне вдосконалення моделі та її здатність давати точні прогнози.

Результати навчання відображають високий рівень адаптації моделі до різних умов та обставин. Графіки метрик свідчать про те, що модель не лише ефективно навчається наявним даним, але й має потенціал виявлення літаків на нових зображеннях, що підтверджує її придатність для застосування в реальному часі.

Розроблена програма Python ефективно використовує потужність навченої моделі CNN для виявлення літаків на зображеннях, демонструючи її корисність у сценаріях реального світу. Інтегроване обчислення IOU покращує прозорість програми, пропонуючи користувачам кількісну оцінку точності просторової локалізації моделі. Загалом програма являє собою плавне поєднання обробки зображень, глибокого навчання та оцінки продуктивності, що робить її цінним інструментом для виявлення літальних апаратів.

ВИСНОВКИ

Під час виконання кваліфікаційного проєкту було глибоко досліджено актуальність проблеми виявлення та розпізнавання літальних апаратів на зображенні з цілю підвищення безпеки повітряного простору.

Кваліфікаційна робота присвячена розробці та впровадженню системи розпізнавання образів літаків у реальному часі з використанням комп'ютерного зору. У ході дослідження було вивчено та вдосконалено глибоку нейронну мережу (CNN) для ефективного виявлення літаків на зображеннях.

Використання глибоких нейронних мереж, зокрема заснованих на архітектурі CNN, має визначальне значення в даній роботі. Під час навчання моделі для розпізнавання образів літаків, CNN дозволяє автоматично вивчати важливі особливості та закономірності в зображеннях. Це забезпечує ефективний та точний процес навчання, дозволяючи моделі адаптуватися до різноманітних умов та типів об'єктів.

Окрім того, CNN відзначається великою ефективністю у підготовці даних. Здатність автоматично визначати важливі функціональні зв'язки в зображеннях робить процес обробки та аналізу великих обсягів даних більш простим та швидким.

Швидкодія CNN в даній роботі визначається не лише швидкістю навчання, але й здатністю ефективно виявляти літаки на нових зображеннях у реальному часі. Це робить систему придатною для використання в сценаріях, де важлива швидка реакція та висока точність виявлення об'єктів.

В порівнянні з іншими способами та алгоритмами розпізнавання, CNN вирізняється здатністю автоматично вивчати представлення відразу з даних, що робить його особливо ефективним у роботі з великими та складними зображеннями. Це дозволяє підвищити якість розпізнавання та адаптуватися до різноманітних умов в реальному світі. Таким чином, використання CNN в даній роботі визначає його успішність та практичну застосованість у завданні розпізнавання образів літаків.

Основні кроки роботи включали в себе підготовку та аугментацію даних, вибір архітектури моделі, оптимізацію параметрів навчання, використання відомих

функцій втрат та валідацію результатів. Модель була успішно навчена на реалістичних даних, забезпечуючи високий рівень точності та швидкодії виявлення літаків.

Важливим аспектом дослідження є його практична значимість. Система дозволяє автоматично виявляти літаки в режимі реального часу, що має ряд перспективних застосувань у сферах авіаційної безпеки, транспортних системах, рятувальних операціях та інших областях. Застосування передових методів комп'ютерного зору виявляється у високій точності виявлення об'єктів та низьких часових затримках.

Отже, дана кваліфікаційна робота не лише розширює розуміння принципів глибокого навчання в аспекті розпізнавання образів, але і надає конкретні рішення для впровадження цих принципів у реальній практиці з високою якістю результатів.

СПИСОК БІБЛЮГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. [Електронний ресурс] Getting Started with Computer Vision: Basics and Starter Projects. Режим доступу: <https://www.analyticsvidhya.com/blog/2022/03/getting-started-with-computer-vision-basics-and-starter-projects/> (дата звернення: 6.10.2023р) – Назва з екрану.
2. [Електронний ресурс] The Difference Between Computer Vision and Human Vision. Режим доступу: <https://visionaisuite.net/blog/the-difference-between-computer-vision-and-human-vision> (дата звернення: 4.11.2023р) – Назва з екрану.
3. [Електронний ресурс] Computer Vision vs. Human Vision: A Comparative Look at Seeing Machines and Seeing Humans. Режим доступу: <https://www.linkedin.com/pulse/computer-vision-vs-human-comparative-look-seeing-sundar-balamurugan/> (дата звернення: 4.11.2023р) – Назва з екрану.
4. [Електронний ресурс] Computer Vision - A Modern Approach. Режим доступу: https://cdn.preterhuman.net/texts/science_and_technology/artificial_intelligence/Computer%20Vision%20A%20Modern%20Approach%20-%20Forsyth%20,%20Ponce.pdf (дата звернення: 21.10.2023р) – Назва з екрану.
5. [Електронний ресурс] What Is Image Processing : Overview, Applications, Benefits, and More. Режим доступу: <https://www.simplilearn.com/image-processing-article> (дата звернення: 14.10.2023р) – Назва з екрану.
6. [Електронний ресурс] Computer Vision. Режим доступу: <https://www.geeksforgeeks.org/computer-vision/> (дата звернення: 14.10.2023р) – Назва з екрану.
7. [Електронний ресурс] Difference between Image Processing and Computer Vision. Режим доступу: <https://www.geeksforgeeks.org/difference-between-image-processing-and-computer-vision/> (дата звернення: 13.10.2023р) – Назва з екрану.
8. [Електронний ресурс] Introduction to Basic Computer Vision & Image Processing. Режим доступу: <https://bishalbose294.medium.com/introduction-to-basic-computer-vision-image-processing-f692aa1a4f18> (дата звернення: 14.10.2023р) – Назва з екрану.

9. [Электронный ресурс] A Gentle Introduction to Computer Vision. Режим доступа: <https://machinelearningmastery.com/what-is-computer-vision/> (дата звернения: 6.11.2023р) – Назва з екрану.

10. [Электронный ресурс] Image processing and Computer Vision. Режим доступа: <https://medium.com/mlearning-ai/image-processing-and-computer-vision-a90ee7403001> (дата звернения: 11.11.2023р) – Назва з екрану.

11. [Электронный ресурс] Computer Vision: Algorithms and Applications to Explore in 2023. Режим доступа: <https://www.projectpro.io/article/computer-vision-algorithms-and-applications/514> (дата звернения: 21.11.2023р) – Назва з екрану.

12. [Электронный ресурс] What are convolutional neural networks?. Режим доступа: <https://www.ibm.com/topics/convolutional-neural-networks> (дата звернения: 16.11.2023р) – Назва з екрану.

13. [Электронный ресурс] Convolutional neural networks: an overview and application in radiology. Режим доступа: <https://insightsimaging.springeropen.com/articles/10.1007/s13244-018-0639-9> (дата звернения: 21.11.2023р) – Назва з екрану.

14. [Электронный ресурс] Introduction to Convolutional Neural Networks (CNN). Режим доступа: <https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/> (дата звернения: 24.11.2023р) – Назва з екрану.

15. [Электронный ресурс] Convolutional Neural Networks, Explained. Режим доступа: <https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939> (дата звернения: 18.11.2023р) – Назва з екрану.

16. [Электронный ресурс] Best Practices for Preparing and Augmenting Image Data for CNNs. Режим доступа: <https://machinelearningmastery.com/best-practices-for-preparing-and-augmenting-image-data-for-convolutional-neural-networks/> (дата звернения: 27.11. 2023р) – Назва з екрану.

17. [Электронный ресурс] Image Classification Using CNN Режим доступа: <https://www.analyticsvidhya.com/blog/2021/01/image-classification-using-convolutional-neural-networks-a-step-by-step-guide/> (дата звернения: 26.11.2023р) – Назва з екрану.

18. [Електронний ресурс] Convolutional Neural Network for Object Recognition and Detection. Режим доступу: <https://medium.com/@ringlayer/convolutional-neural-network-for-object-recognition-and-detection-126a22af8975> (дата звернення: 25.11.2023р) – Назва з екрану.

19. [Електронний ресурс] Deep Learning for Image Classification. Режим доступу: <https://medium.com/swlh/deep-learning-for-image-classification-creating-cnn-from-scratch-using-pytorch-d9eeb7039c12> (дата звернення: 21.11.2023р) – Назва з екрану.