

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет радіоелектроніки
Факультет Комп'ютерних наук
Кафедра Програмної інженерії

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

_____ другий (магістерський) _____

(рівень вищої освіти)

Дослідження методів оптимізації розпізнавання зображень у реальному масштабі
часу

Виконав:

студент 2 курсу групи ІПЗм-21-4

Перець О. В.

(прізвище, ініціали)

Спеціальність 121 – Інженерія

програмного забезпечення

Тип програми Освітньо-наукова

Керівник доц., Работягов А.В.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. Кафедри _____

_____ З.В. Дудар

2023

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____

Кафедра _____ Програмної інженерії _____

Рівень вищої освіти _____ другий (магістерський) _____

Спеціальність _____ 121 – Інженерія програмного забезпечення _____
(код і повна назва)

Тип програми _____ освітньо-наукова програма _____

Освітня програма _____ Інженерія програмного забезпечення _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«__» _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студента _____ Перця Олександра Вячеславовича _____

(прізвище, ім'я, по батькові)

1. Тема роботи: Дослідження методів оптимізації розпізнавання зображень у реальному масштабі часу

затверджена наказом університету відвід «29» березня 2023 р. № 302 Ст

2. Термін подання студентом роботи до екзаменаційної комісії «22» травня 2023 р.

3. Вихідні дані до роботи встановлений календарний план роботи, методичні вказівки до оформлення пояснювальної записки, методи оптимізації розпізнавання зображень у реальному масштабі часу..

4. Перелік питань, що потрібно опрацювати в роботі аналіз предметної галузі, огляд наявних методів розробки, аналіз методів розпізнавання зображень, оптимізація готових рішень створення програмного забезпечення для проведення дослідження, отримання та аналіз результатів, їх подальше використання.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної галузі	26.03.2023	виконано
2	Аналіз існуючих методів	28.03.2023	виконано
3	Дослідження видів оптимізації методів	15.04.2023	виконано
4	Постановка задачі	20.04.2023	виконано
5	Підготовка пояснювальної записки	26.04.2023	виконано
6	Підготовка презентації та доповіді	30.04.2023	виконано
7	Перевірка на академічний плагіат	15.05.2023	виконано
8	Нормоконтроль	15.05.2023	виконано
9	Рецензування	16.05.2023	виконано
10	Занесення диплома в електронний архів	17.05.2023	виконано
11	Попередній захист	19.05.2023	виконано
12	Допуск до захисту	20.05.2023	виконано

Дата видачі завдання 01 лютого 2023 р.

Студент _____

(підпис)

Керівник роботи _____



(підпис)

доц., Работягов А.В.

(посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Звіт до кваліфікаційної роботи містить: 72 с., 6 рис., 9 джерел, 6 додатків.

МЕТОД НА ОСНОВІ ПРАВИЛ, ОПТИМІЗАЦІЯ, РОЗПІЗНАВАННЯ
ЗОБРАЖЕНЬ, HAAR КАСКАДИ, OPENCV, YOLO.

Об'єктом дослідження роботи є методи оптимізації розпізнавання зображень у реальному масштабі часу.

Метою роботи є дослідження різних методів оптимізації розпізнавання зображень у реальному масштабі часу.

Дослідження зосереджується на розгляді різних методів, які використовуються для розпізнавання зображень, таких як методи на основі правил, Хаар каскади, YOLO, їх порівнянні та визначенні ступеня відмінності між ними, а також визначення конкретних завдань та сценаріїв, де ці методи найбільш ефективні. Досліджуються та оптимізуються параметри методів розпізнавання залежно від вимог конкретного завдання, які впливають на якість розпізнавання зображень у реальному масштабі часу.

Результатом роботи є аналітичний огляд наявних методів розпізнавання та оптимізації функціональних параметрів методів для розпізнавання зображень у реальному масштабі часу, порівняння ефективності застосування методів для вирішення практичних задач.

IMAGE RECOGNITION, OPTIMIZATION, RULE-BASED METHOD,
OPENCV, HAAR CASCADES, YOLO.

The subject of this research is real-time image recognition optimization methods.

The objective of this work is to investigate various methods for optimizing real-time image recognition.

The research focuses on examining different approaches used in image recognition, such as rule-based methods, Haar cascades, YOLO, comparing them, determining the degree of distinction between them, as well as identifying specific tasks and scenarios where these methods are most effective. The parameters of the recognition methods are explored and optimized based on the requirements of specific tasks, which directly impact the quality of real-time image recognition.

The outcome of this work is an analytical review of existing image recognition methods and the optimization of their functional parameters for real-time image recognition. The effectiveness of applying these methods to practical tasks is compared and evaluated.

Я, Перець Олександр Вячеславович, студент гр. ПЗм-21-4, здобувач вищої освіти на другому (магістерському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Дослідження методів оптимізації розпізнавання зображень у реальному масштабі часу», що буде представлена в екзаменаційну комісію для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу EIAr KhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

ЗМІСТ.....	6
ВСТУП.....	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ.....	10
1.1 Аналіз предметної галузі.....	10
1.2 Аналіз інших підходів до розпізнавання зображень у реальному масштабі часу.....	13
1.2.1 Глибокі нейронні мережі.....	13
1.2.2 Класифікатори зображень.....	15
1.2.3 Інші методи для розпізнавання зображень у реальному масштабі часу... 16	16
1.3 Постановка задачі.....	17
2 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ОПТИМІЗАЦІЇ.....	19
2.1 Метод на основі правил.....	19
2.2 Огляд методу на основі OpenCV з Наар каскадами.....	20
2.3 Огляд методу що використовує глибокі нейронні мережі YOLO.....	24
2.4 Огляд аналогів для OpenCV з Наар каскадами.....	26
2.5 Огляд аналогів YOLO.....	27
3 СТВОРЕННЯ ПРОГРАМНОЇ СИСТЕМИ ТА ПІДГОТОВКА ДО ЕКСПЕРИМЕНТУ.....	29
3.1 Реалізація та аналіз програмної частини.....	29
3.2 Опис використаних програмних інструментів.....	29
3.3 Оптимізація за допомогою методу на основі правил.....	30
3.3.1 Підготовка зображень.....	31
3.3.2 Розпізнавання об'єктів за допомогою методу на основі правил.....	33
3.3.3 Висновок роботи з методом на основі правил.....	33
3.4 Оптимізація методу на основі OpenCV та Наар каскадів.....	35

3.4.1 Застосування Нагг каскадів для розпізнавання облич у реальному масштабі часу.....	35
3.4.2 Оптимізація розпізнавання обличчя.....	36
3.4.3 Реалізація роботи оптимізованого методу для пошуку облич у реальному масштабі реального часу.....	38
3.5 Оптимізація методу на основі YOLO.....	40
3.5.1 Огляд архітектури YOLO.....	41
3.5.2 Види оптимізації для YOLO.....	42
3.5.2.1 Модифікація архітектури.....	43
3.5.2.2 Вагове усереднення.....	43
3.5.2.3 Оптимізація обчислень.....	44
3.5.2.4 Апаратне прискорення.....	44
3.5.2.5 Зменшення розміру мережі.....	45
3.6 Розробка методу оптимізації зображень у реальному масштабі часу.....	45
3.6.1. Використання апаратного прискорення.....	46
4 ДОСЛІДЖЕННЯ МЕТОДІВ ОПТИМІЗАЦІЇ РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ.....	49
ВИСНОВКИ.....	52
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	54
ДОДАТОК А.....	55
ДОДАТОК Б.....	56
ДОДАТОК В.....	57
ДОДАТОК Г.....	59
ДОДАТОК Д.....	61
ДОДАТОК Е.....	72

ВСТУП

Проблема розпізнавання зображень у реальному масштабі часу має велику актуальність і значимість у різних сферах, таких як комп'ютерний зір, автономні транспортні засоби, медична діагностика та інші. Удосконалення цього процесу дозволяє досягти швидкого та точного розпізнавання об'єктів на зображеннях, що має велике значення для подальшого аналізу та прийняття рішень.

Метою даної науково-дослідної роботи є проведення аналізу та вивчення різноманітних методів оптимізації розпізнавання зображень у реальному часі з метою визначення найефективніших із них. У цьому дослідженні будуть розглянуті та порівняні три основні методи, включаючи метод на основі правил, використання Нааг каскадів з використанням бібліотеки OpenCV та метод YOLO, які вже давно знайомі у сферах комп'ютерного зору та розпізнавання образів.

В першому етапі дослідження ретельно аналізується метод на основі правил для розпізнавання зображень у реальному часі. Вивчаються його принцип роботи, архітектура та основні переваги та обмеження. Проводиться порівняння з іншими методами та встановлюється, що цей метод добре працює в обмежених сценаріях, де правила для розпізнавання об'єктів можуть бути заздалегідь визначені. Однак, він може бути менш ефективним у складних ситуаціях, де необхідно розпізнавати різні класи об'єктів або коли умови освітлення та позиції об'єктів на зображенні змінюються.

Другий етап дослідження присвячений використанню Нааг каскадів з використанням бібліотеки OpenCV для розпізнавання зображень у реальному часі. Здійснюється докладний аналіз цього методу, включаючи його принцип роботи, структуру каскадів та процес навчання. Встановлюється, що використання Нааг каскадів може забезпечити високу швидкодію та точність розпізнавання об'єктів на зображеннях. Порівнюються різні підходи до побудови каскадів та їхніх параметрів для досягнення найкращих результатів.

Третій метод, який досліджується у даній роботі, є метод YOLO (You Only Look Once). Цей метод базується на глибоких нейронних мережах та використовує

концепцію об'єктного виявлення з одним проходом. Проводиться аналіз структури та архітектури YOLO, його можливостей та обмежень. Встановлюється, що метод YOLO демонструє вражаючу швидкодію та здатність розпізнавати об'єкти у реальному часі. Однак, в порівнянні з іншими методами, він може бути менш точним при розпізнаванні дрібних деталей та об'єктів з малим контрастом.

Об'єктом дослідження являються методи оптимізації розпізнавання зображень у реальному часі, а предметом дослідження є аналіз та порівняння методів, починаючи з методу на основі правил, Нагг каскадів та методу YOLO для визначення їхньої ефективності та придатності у різних сценаріях.

Для досягнення поставленої мети використовувалися різні методи дослідження, включаючи аналіз літературних джерел, проведення експериментів та порівняння результатів. Ці методи дослідження дозволили отримати об'єктивні результати та зробити аналіз ефективності кожного методу.

Отримані результати дослідження будуть використані для визначення найефективнішого методу оптимізації розпізнавання зображень у реальному часі залежно від конкретних вимог. Також будуть запропоновані рекомендації щодо підбору оптимальних методів та можливих напрямків подальших досліджень у цій галузі. Ця наукова робота показує наскільки дослідження впливають на розвиток технологій розпізнавання зображень у реальному часі і може бути використана для майбутніх .

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Аналіз предметної галузі

Розпізнавання зображень є широко застосовуваною технологією, яка знаходить застосування в різних галузях, включаючи медицину, транспорт, безпеку та рекламу. Однак, одним з викликів, з якими стикаються системи розпізнавання зображень, є потреба в оптимізації для забезпечення розпізнавання у реальному масштабі часу.

Для досягнення розпізнавання зображень у реальному часі, необхідно використовувати оптимізаційні алгоритми обробки зображень. Це може включати застосування швидких алгоритмічних методів та оптимізацію обчислювальних операцій. Крім того, використання спеціалізованих апаратних рішень, таких як графічні процесори (GPU) або процесори для машинного навчання, може значно підвищити швидкість обробки зображень.

Паралельна обробка даних є ще одним ефективним підходом до оптимізації розпізнавання зображень у реальному масштабі часу. Це передбачає розподіл обчислювальних завдань між багатьма обчислювальними ресурсами, що дозволяє прискорити обробку зображень та забезпечити миттєве розпізнавання.

Однак, важливо знайти баланс між точністю розпізнавання та швидкістю обробки. В окремих випадках може бути необхідно знизити складність алгоритмів або використовувати менш точні методи розпізнавання, щоб досягти реального часу, але це може вплинути на точність результатів розпізнавання. Тому розробники повинні уважно аналізувати вимоги та контекст застосування системи розпізнавання зображень і приймати відповідні рішення з точки зору компромісу між точністю та швидкістю.

Оптимізація розпізнавання зображень у реальному масштабі часу знаходить застосування в багатьох галузях. У медицині, це може бути використано для автоматичного виявлення патологій на медичних зображеннях, що допомагає вчасно діагностувати захворювання. У транспорті, розпізнавання зображень може допомогти виявляти номерні знаки автомобілів для забезпечення безпеки та

контролю. У безпеці та розвагах, розпізнавання обличчя може використовуватися для розпізнавання та ідентифікації осіб у реальному часі.

Розпізнавання зображень у реальному масштабі часу продовжує розвиватися, а нові підходи і технології з'являються для поліпшення ефективності та швидкості цього процесу. Одним із таких напрямків є використання глибокого навчання та нейромереж.

Глибоке навчання забезпечує високу точність розпізнавання зображень шляхом тренування глибоких нейронних мереж на великому обсязі даних. Однак, стандартні архітектури глибоких мереж, такі як згорткові нейронні мережі (Convolutional Neural Networks, CNN), можуть бути обчислювально витратними, що обмежує їхню швидкість розпізнавання. Тому проводяться дослідження щодо оптимізації та стиснення глибоких моделей, наприклад, шляхом застосування технік прунінгу (pruning), квантизації (quantization) та компресії моделей.

Також виникають нові архітектури нейронних мереж, спеціально розроблені для розпізнавання зображень у реальному масштабі часу. Наприклад, MobileNet та EfficientNet є прикладами архітектур, які досягають високої точності при низькому обчислювальному завантаженні. Вони базуються на принципі легковагових фільтрів та оптимізованої структури мережі, що дозволяє ефективно виконувати операції розпізнавання на ресурсно обмежених пристроях, таких як мобільні телефони або вбудовані системи.

Розподілені обчислення та використання хмарних платформ також сприяють оптимізації розпізнавання зображень у реальному масштабі часу. Обчислення можуть бути розподілені між кількома обчислювальними вузлами або виконуватися на потужних серверах у хмарному середовищі. Це дозволяє використовувати більші обчислювальні ресурси та прискорювати обробку зображень.

Хмарні платформи надають доступ до високопродуктивних обчислювальних ресурсів, графічних процесорів та спеціалізованих пристроїв для машинного навчання. Це дає можливість виконувати обробку зображень у реальному часі

навіть на пристроях з обмеженими обчислювальними можливостями, використовуючи віддалені сервери для обчислень.

Додатково, оптимізація розпізнавання зображень у реальному масштабі часу може включати використання спеціалізованих апаратних пристроїв, таких як Tensor Processing Units (TPU) або Field-Programmable Gate Arrays (FPGA). Ці пристрої прискорюють обчислення зображень шляхом виконання паралельних обчислень та оптимізації апаратної архітектури спеціально для розпізнавання зображень.

Застосування оптимізації розпізнавання зображень у реальному масштабі часу розширюється на різні галузі. У медицині, це може бути використано для автоматичного виявлення ракових ознак на медичних зображеннях або аналізу рентгенівських знімків. У транспорті, розпізнавання зображень може використовуватися для систем автоматичного виявлення номерних знаків або виявлення воді.

Оптимізація розпізнавання зображень у реальному масштабі часу потребує використання передових технологій. Розробники повинні використовувати сучасні методи машинного навчання, штучного інтелекту та новітні апаратні рішення. Постійне оновлення технологій є важливим, оскільки це дозволяє досягати найкращих результатів і використовувати потенціал нових інновацій.

В цілому, оптимізація розпізнавання зображень у реальному масштабі часу є надзвичайно важливою задачею, оскільки вона забезпечує ефективність та швидкість розпізнавання в різних галузях тому ця технологія знаходить своє застосування всюди, де потрібно швидко та точно розпізнавання об'єктів. Завдяки його потужності та потенціалу, оптимізація розпізнавання зображень продовжуватиме розвиватися, перетворюючи наш спосіб сприйняття світу та розширюючи можливості наших технологій. Дослідження та розвиток цієї області прогресують, спираючись на використання оптимізаційних алгоритмів, спеціалізованих апаратних рішень та передових технологій.

1.2 Аналіз інших підходів до розпізнавання зображень у реальному масштабі часу

Було досліджено та проаналізовано проблематику розпізнавання зображень у реальному масштабі часу. Під час аналізу різних підходів до цієї задачі, виявлено кілька основних видів підходів, що використовуються для досягнення цієї мети. Виконаний аналіз спромігся виділити найпоширеніші та ефективні методи, які дозволяють швидко та точно розпізнавати об'єкти у реальному часі. Детальне вивчення цих підходів допоможе зрозуміти основні принципи їх роботи та визначити найбільш відповідні та перспективні методи для їх використання та оптимізації з метою дослідження у даній роботі.

1.2.1 Глибокі нейронні мережі

У контексті розпізнавання зображень у реальному масштабі часу, глибокі нейронні мережі виявляються надзвичайно потужними інструментами. Основними видами глибоких нейронних мереж, використовуваними в цій області, є конволюційні нейронні мережі (Convolutional Neural Networks - CNN), рекурентні нейронні мережі (Recurrent Neural Networks - RNN) та мережі довготривалої короткочасної пам'яті (Long Short-Term Memory - LSTM).

Конволюційні нейронні мережі є особливо ефективними для розпізнавання об'єктів на зображеннях. Вони працюють шляхом використання операцій згортки, що дозволяють автоматично виявляти та виділяти локальні залежності та важливі ознаки у зображенні. Цей процес дозволяє КНМ самостійно визначати, наприклад, контури, текстури, а також більш складні структури об'єктів, що допомагає впевненіше розпізнавати їх.

Рекурентні нейронні мережі та мережі довготривалої короткочасної пам'яті дозволяють моделювати залежності в часових послідовностях. Це особливо корисно для розпізнавання об'єктів у відеопотоці, де зображення надходять у вигляді послідовних кадрів. Рекурентні нейронні мережі використовують попередні вихідні дані для передачі контексту між кадрами, а мережі LSTM

забезпечують збереження та використання довготривалих залежностей, що дозволяє більш ефективно відстежувати та розпізнавати об'єкти в часових послідовностях.

Використання глибоких нейронних мереж для розпізнавання зображень у реальному часі передбачає наступні етапи. Спочатку глибока нейронна мережа підлягає навчанню на великому наборі зображень, де кожен зразок має відповідну мітку класу. Процес навчання включає згорткові шари для автоматичного виявлення ознак на різних рівнях абстракції, пулінгові шари для зменшення розмірності зображення та пов'язані шари для класифікації об'єктів (див.рис.1.1).

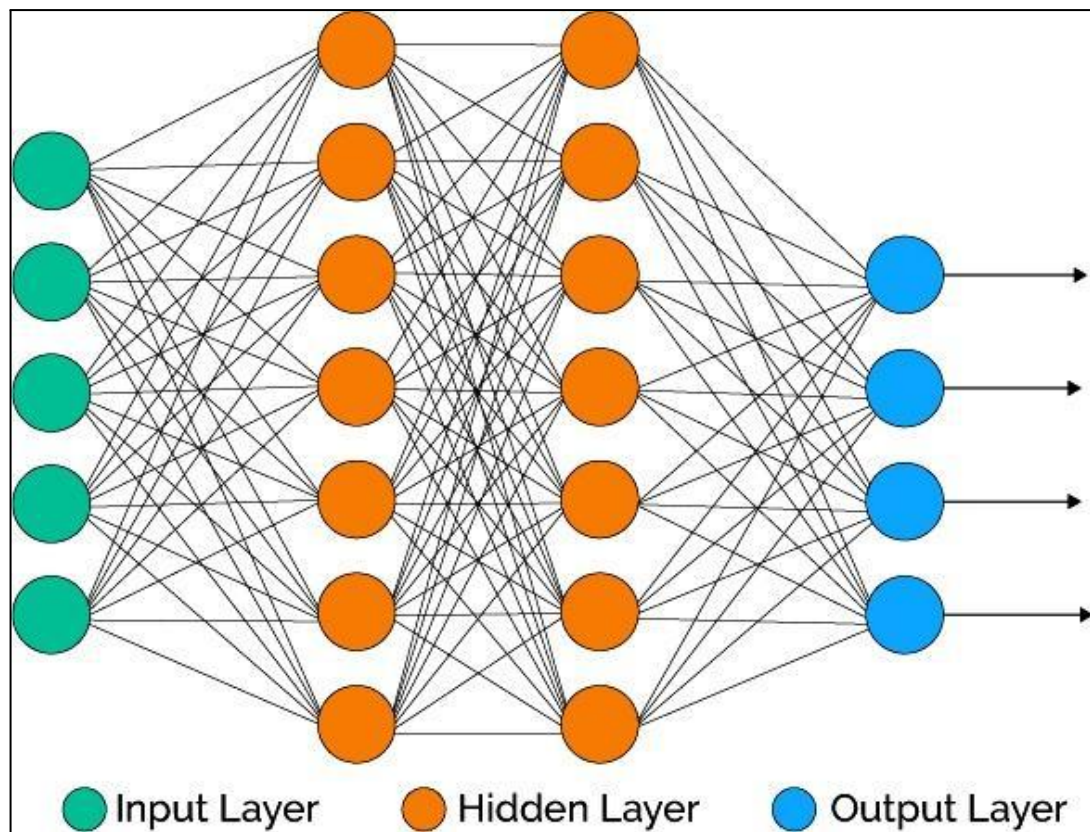


Рисунок 1.1 – Схема роботи глибоких нейронних мереж для розпізнавання зображень у реальному масштабі часу

Після завершення навчання, глибока нейронна мережа може бути застосована для розпізнавання об'єктів у реальному часі. Зображення можуть бути отримані з відеопотоку або інших джерел, і далі піддаються обробці мережею.

Цей процес включає проходження зображення через мережу, аналіз його ознак та класифікацію об'єктів на основі навчених моделей.

Глибокі нейронні мережі демонструють високу швидкість та точність у розпізнаванні зображень у реальному часі. Вони здатні автоматично виявляти складні ознаки та структури об'єктів, що допомагає впевнено класифікувати їх у реальному часі. Крім того, використання рекурентних нейронних мереж та мереж довготривалої короткочасної пам'яті дозволяє зберігати і використовувати контекстуальну інформацію для вдосконалення процесу розпізнавання об'єктів у відеопотоці.

1.2.2 Класифікатори зображень

Аналіз різних класифікаторів зображень у реальному масштабі часу дозволяє вивчати їх переваги та обмеження. Що сприяє вдосконаленню алгоритмів, розробці нових методів оптимізації та використанню різних підходів для покращення точності та швидкості класифікації зображень. Розуміння принципів роботи класифікаторів, таких як метод опорних векторів і розширені дерева прийняття рішень, важливо для розвитку ефективних систем розпізнавання об'єктів у реальному часі.

Метод опорних векторів (Support Vector Machines, SVM) є одним з популярних класифікаторів, що використовуються у розпізнаванні зображень у реальному масштабі часу. SVM використовується для розділення даних у просторі шляхом побудови гіперплощини, яка максимально розділяє класи зображень. Основна ідея полягає в тому, щоб знайти оптимальну гіперплощину, яка максимізує відстань між класами.

Розширені дерева прийняття рішень (Random Forests, Decision Trees) є іншим ефективним методом класифікації зображень. Вони використовуються для побудови ансамблю дерев рішень, де кожне дерево прийняття рішень навчається на випадковій підмножині даних та використовується для прийняття колективного рішення. Розширені дерева прийняття рішень можуть бути ефективними у

розпізнаванні об'єктів на зображеннях шляхом врахування багатовимірних ознак та їхнього взаємозв'язку.

Основна ідея каскадних класифікаторів полягає в тому, що вони складаються з послідовності класифікаторів, які працюють на різних рівнях деталізації зображення. Кожен класифікатор у каскаді розпізнає певні ознаки об'єкта. Наприклад, перші класифікатори можуть використовувати прості ознаки, такі як краї та текстури, для швидкого відсіювання негативних областей зображення. Поступово, на наступних етапах каскаду використовуються більш складні класифікатори, що деталізуються і розпізнають більш конкретні ознаки об'єкта. Такий підхід дозволяє прискорити процес класифікації, оскільки більш прості класифікатори відсіюють більшу кількість негативних областей, а класифікатори на наступних етапах працюють тільки з областями, що містять потенційні об'єкти [1].

1.2.3 Інші методи для розпізнавання зображень у реальному масштабі часу

Методи на основі ознак є одними з ключових аналогів у галузі розпізнавання зображень. Ці методи використовуються для класифікації об'єктів на зображеннях шляхом виявлення та аналізу особливих ознак або шаблонів. Основною ідеєю цих методів є визначення характеристичних ознак, таких як краї, текстури, кольори тощо, і використання їх для ідентифікації та класифікації об'єктів. Часто використовується підхід, що базується на порівнянні ознак об'єктів з попередньо побудованим набором шаблонів чи моделей.

Методи засновані на сегментації використовуються для розділення зображення на окремі сегменти або області, що містять об'єкти. Ці методи базуються на аналізі інтенсивності пікселів, текстури, кольору, геометричних ознак та інших характеристик зображення для визначення границь об'єктів. Сегментація дозволяє виділити області, які містять потенційні об'єкти, що забезпечує більш точну та цілісну класифікацію.

Використання розширеної реальності (augmented reality, AR) для розпізнавання об'єктів є ще одним цікавим аналогом. AR поєднує реальне

оточення з віртуальними об'єктами, що дозволяє розпізнавати та класифікувати об'єкти у режимі реального часу. Завдяки AR можна накладати додаткову інформацію на зображення, що полегшує процес розпізнавання об'єктів.

1.3 Постановка задачі

У даному дослідженні ми ставимо перед собою завдання оптимізації методів розпізнавання зображень у реальному масштабі часу. Наша мета - розглянути різні підходи до розпізнавання об'єктів та розробити оптимізовані методи, які забезпечують високу продуктивність і точність при роботі зі розпізнаванням зображень в реальному масштабі часу.

Першим кроком у нашому дослідженні є аналіз аналогів методів розпізнавання. Ми досліджуємо різні види методів та їх оптимізацій, щоб з'ясувати їх переваги та обмеження в контексті розпізнавання об'єктів у реальному масштабі часу. Засновуючись на цьому аналізі, ми обираємо найбільш ефективні та потужні методи для подальшої оптимізації.

Другим кроком є розробка оптимізаційних методів для обраних алгоритмів. Ми використовуємо спеціалізовані програмні інструменти, такі як OpenCV, Haar Cascades, YOLO (You Only Look Once) та метод на основі правил, для створення оптимізованих версій цих методів. Ми зосереджуємося на вдосконаленні алгоритмів, покращенні швидкодії та розпізнавання, щоб досягти високої продуктивності та точності розпізнавання об'єктів у реальному часі.

Висновком нашого дослідження буде розробка оптимізованих методів розпізнавання зображень, які можуть успішно використовуватися у вибраних сценаріях. Буде зроблено особливий уклін на те, щоб кожен метод був оптимізований відповідно до конкретного сценарію використання. Наша робота забезпечує оптимізацію методів, що сприятиме їх простоті та зручності використання.

Отже, у цьому розділі ми зосередимося на постановці задачі оптимізації методів розпізнавання зображень у реальному масштабі часу. Почнемо з аналізу аналогів методів розпізнавання, досліджуючи різні види методів та їх

оптимізаційні можливості. Далі, ми перейдемо до розробки оптимізаційних методів для обраних алгоритмів, використовуючи спеціалізовані програмні інструменти та технології.

У результаті нашого дослідження ми презентуємо оптимізовані методи розпізнавання зображень, що забезпечують високу продуктивність та точність розпізнавання у реальному часі. Кожен з цих методів буде оптимізований з урахуванням конкретного сценарію їх використання, що дозволить досягти максимальної ефективності та задоволення вимог користувачів. Наша робота спрямована на автоматизацію цих оптимізованих методів, щоб зробити їх доступними та простими у використанні.

2 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ОПТИМІЗАЦІЇ

2.1 Метод на основі правил

Метод на основі правил є підходом до розпізнавання зображень, що використовується для оптимізації процесу класифікації або визначення об'єктів на зображенні. Його основна ідея полягає в тому, що для розпізнавання об'єктів встановлюються певні правила, які описують характеристики, що є типовими для кожного класу об'єктів [2].

Використовуючи метод на основі правил, спочатку потрібно побудувати набір правил або шаблонів для кожного класу об'єктів, які планується розпізнавати. Ці правила можуть бути засновані на різних характеристиках об'єктів, таких як їхні розмір, форма, текстура, кольорова інформація тощо. Наприклад, для розпізнавання автомобілів можуть використовуватися правила, які визначають певний діапазон розмірів, форми кузова та наявність коліс

Преваги:

- простота і інтерпретованість: метод на основі правил легко розуміти і інтерпретувати, оскільки основні правила базуються на досвіді та експертному знанні;
- ефективність на відносно простих задачах: Якщо задача розпізнавання об'єктів є відносно простою і має чіткі шаблони, метод на основі правил може бути досить ефективним.

Обмеження:

- обмежена адаптивність: метод на основі правил може мати обмежену адаптивність до змінних умов та варіацій об'єктів він може не бути ефективним у випадках, коли правила не можуть адекватно описати всі можливі варіації об'єктів.

Залежність від експертного знання: Розробка правил вимагає експертного знання та досвіду в галузі, що може бути складним і витратним процесом.

Приклади застосування:

- розпізнавання дорожніх знаків: метод на основі правил широко використовується для розпізнавання дорожніх знаків, де можна встановити чіткі правила їх класифікації на основі форми, кольору та символів. Наприклад, правило може визначати, що трикутний знак червоного кольору зі зображенням випадаючого камінця вказує на небезпеку на дорозі;
- системи безпеки: метод на основі правил використовується в системах безпеки для розпізнавання небажаних об'єктів або подій. Наприклад, в аеропортах системи безпеки можуть використовувати правила для розпізнавання підозрілих предметів на рентгенівських знімках чи для виявлення незвичайної поведінки осіб на відеозаписах;
- медична діагностика: метод на основі правил широко використовується в медичній діагностиці для розпізнавання патологічних ознак або хвороб на зображеннях медичних зразків. Наприклад, можуть встановлюватися правила для виявлення пухлин на зображеннях МРТ або для класифікації патологічних клітин на зображеннях цитологічних зразків.

2.2 Огляд методу на основі OpenCV з Нааг каскадами

Нааг каскади – це потужний і широко використовуваний метод для розпізнавання об'єктів на зображеннях. В основі його роботи лежить використання набору признаков Хаара, які можна використовувати для знаходження об'єктів різних форм і розмірів. Ці признаки визначаються яскравістю пікселів на зображенні і використовуються для виявлення змін в яскравості в різних частинах зображення.

Нааг каскади працюють наступним чином: спочатку створюється набір каскадів, кожен з яких містить набір признаков Хаара. Потім зображення

поділяється на багато малих частин, і кожен каскад застосовується до кожної з них. Якщо каскад виявляє певний признак у частині зображення, то ця частина може містити об'єкт, який потрібно виявити.

Один з головних переваг Нааг каскадів полягає в тому, що вони дозволяють швидко і ефективно виявляти об'єкти на зображеннях в реальному часі. Використання Нааг каскадів з OpenCV дозволяє оптимізувати зображення в реальному часі, що є важливим в багатьох додатках комп'ютерного зору, таких як автоматичне водіння, контроль якості виробництва, безпека та багато іншого.

Крім того, Нааг каскади можуть бути навчені розпізнавати різні типи об'єктів, включаючи обличчя, людей, автомобілі, зупинки дорожнього руху тощо. Це робить їх універсальним інструментом для багатьох завдань розпізнавання об'єктів.

Процес навчання Нааг каскадів є складним і включає два основних етапи: навчання позитивних та негативних зразків. Під час навчання позитивних зразків визначаються признаки, які характеризують об'єкти, які мають бути розпізнані. Ці зразки включають зображення, на яких об'єкти позначені або виділені. Навчання негативних зразків включає зображення, на яких об'єкти відсутні або їх відсутність вважається негативним прикладом.

Після успішного навчання Нааг каскадів їх можна використовувати для розпізнавання об'єктів у реальному масштабі часу. Це означає, що система здатна розпізнавати об'єкти в реальному часі без помітної затримки. Завдяки високій швидкості обробки, Нааг каскади знаходять широке застосування в системах відеоспостереження та інших реального часу системах.

Застосування Нааг каскадів у системах реального часу, таких як відеоспостереження, має велике значення в різних сферах. Наприклад, вони можуть використовуватися для автоматичного виявлення облич, людей або автомобілів на відеозаписах. Це дозволяє забезпечити швидку реакцію системи на потенційні загрози або спростити аналіз великого обсягу відеоданих.

Хоча Нааг каскади є потужним інструментом для розпізнавання об'єктів, вони також мають свої обмеження. Зокрема, вони можуть бути чутливими до змін

усвідомлення світла та варіацій у формі та розмірі об'єктів. Крім того, навчання Нааг каскадів може бути часо- і ресурсомістким процесом.

Загалом, Нааг каскади є ефективним методом розпізнавання об'єктів на зображеннях, який знаходить широке застосування в області комп'ютерного зору. Використання їх у поєднанні з OpenCV дозволяє оптимізувати розпізнавання зображень у реальному часі, що є важливим для багатьох сучасних застосувань, від систем безпеки до розпізнавання облич.

OpenCV (Open Source Computer Vision Library) - це відкрита бібліотека комп'ютерного зору, яка надає широкі можливості для обробки зображень, аналізу відео, розпізнавання об'єктів та багато іншого. Вона написана на мові програмування C++ і має інтерфейси для використання з інших мов, таких як Python та Java. OpenCV є однією з найпопулярніших бібліотек у сфері комп'ютерного зору та машинного навчання.

OpenCV був обраний для використання з Нааг каскадами з декількох причин. По-перше, OpenCV має потужний набір функцій для обробки зображень, що дозволяє зручно та ефективно застосовувати Нааг каскади для розпізнавання об'єктів. Бібліотека містить реалізацію алгоритмів обробки зображень, включаючи фільтрацію, згортку, виявлення контурів, виокремлення областей та багато іншого. Це надає зручну основу для реалізації алгоритмів розпізнавання об'єктів з використанням Нааг каскадів.

Крім того, OpenCV є відкритою бібліотекою, що означає, що вона безкоштовна для використання, модифікації та розповсюдження. Це дає можливість розробникам використовувати OpenCV у комерційних або некомерційних проектах без обмежень.

Застосування OpenCV разом з Нааг каскадами дозволяє досягти високої ефективності та швидкості обробки зображень в реальному часі. Це особливо важливо для оптимізації зображень у системах реального часу, таких як відеоспостереження, де необхідно швидко реагувати на зміни та розпізнавати об'єкти на живих відеопотоках.

Отже, обрання OpenCV для дослідження та реалізації оптимізації зображень у реальному часі з використанням Нааг каскадів є раціональним вибором, оскільки вона надає необхідні інструменти, функціональність та підтримку для ефективної обробки зображень та розпізнавання об'єктів.

Переваги та обмеження використання OpenCV з Нааг каскадами в порівнянні з іншими методами:

Переваги:

- швидкість: Нааг каскади можуть працювати у реальному масштабі часу, що робить їх ефективними для багатьох застосувань;
- невибагливість до обчислювальних ресурсів: Вони можуть бути використані на простіших обчислювальних пристроях, таких як мобільні пристрої.

Обмеження:

- точність: Нааг каскади можуть мати обмежену точність розпізнавання, особливо при складних сценах або змінних умовах освітлення. Вони можуть давати помилкові результати або пропускати деякі об'єкти;
- типи об'єктів: Нааг каскади працюють краще для розпізнавання об'єктів зі статичною або обмеженою варіативністю форми та текстури. Вони можуть бути менш ефективними для розпізнавання складних об'єктів зі змінними або незвичайними формами;
- вимоги до тренувальних даних: Навчання Нааг каскадів вимагає великої кількості позитивних та негативних зразків для досягнення задовільної точності. Збір і підготовка тренувальних даних може бути складним і часомірним процесом.

2.3 Огляд методу що використовує глибокі нейронні мережі YOLO.

YOLO (You Only Look Once) є потужним алгоритмом розпізнавання об'єктів, який використовує глибокі нейронні мережі для ефективного та швидкого виявлення об'єктів на зображеннях. Принцип роботи YOLO полягає в тому, що він аналізує всю сцену одночасно, замість традиційного підходу, де спочатку виконується виявлення областей і потім класифікація об'єктів.

Переваги:

- поділ зображення на сітку клітин. Кількість клітин у сітці залежить від розміру вхідного зображення та роздільної здатності мережі. Кожна клітина передбачає та класифікує декілька об'єктів, які знаходяться в її межах. Крім того, кожна клітина відповідає за передбачення координат об'єктів, використовуючи офсети відносно своїх границь. Таким чином, YOLO здатний одночасно розпізнавати та локалізувати об'єкти у всьому зображенні;
- висока точність та розпізнавання об'єктів з різних категорій, YOLO використовує глибоку нейронну мережу, зазвичай базову модель CNN, яка навчається на великому наборі тренувальних зображень. Процес навчання YOLO включає передачу знань з тренувальних зображень до мережі, щоб вона могла відповідно класифікувати та локалізувати об'єкти на нових зображеннях.
- швидкість та ефективність. Оскільки він працює з усім зображенням одночасно, а не з областями, він може робити розпізнавання в реальному часі навіть для великих зображень. Це робить його особливо популярним для застосувань, де швидкість виконання є критичною, таких як відеоспостереження, автономні автомобілі та системи розпізнавання образів у реальному часі;

- здатність до паралельного обчислення, що дозволяє прискорити процес розпізнавання об'єктів. Замість послідовної обробки зображення, як це робиться у багатьох інших методах, YOLO розділяє зображення на сітку клітин і одночасно аналізує їх усі. Це робить YOLO вкрай швидким, особливо для великих зображень;
- здатність до розпізнавання багатьох об'єктів у рамках одного зображення.

Традиційні методи, такі як R-CNN, вимагають послідовного обчислення для кожного об'єкта на зображенні, що може бути вкрай часо- та ресурсозатратним завданням. Завдяки архітектурі YOLO, яка використовує глобальну інформацію про зображення для розпізнавання об'єктів, він може виявляти та класифікувати багато об'єктів на зображенні з високою швидкістю;

- також варто зазначити, що YOLO володіє доброю здатністю до узагальнення та адаптації до різних типів об'єктів та сценаріїв. Він може успішно працювати з різними категоріями об'єктів, включаючи людей, транспортні засоби, тварин і предмети побуту.

Недоліки:

- проблеми з точністю розпізнавання для малих об'єктів або об'єктів зі складною структурою. Також, він може виявляти труднощі у випадках, коли об'єкти перекриваються або мають незвичайні форми. Крім того, YOLO вимагає значних обчислювальних ресурсів та пам'яті для тренування та виконання, що може бути обмеженням для пристроїв з обмеженою потужністю;
- обмежена здатність до розпізнавання об'єктів у складних сценах або з незвичайними або неочікуваними контекстами. У складних сценах, де об'єкти можуть бути перекриті, знаходитися у великому розрізі або мати

незвичайні форми або пози, точність розпізнавання YOLO може зменшуватися.

2.4 Огляд аналогів для OpenCV з Наг каскадами

Histogram of Oriented Gradients. HOG є популярним методом розпізнавання об'єктів у зображеннях, особливо у реальному масштабі часу. Він використовує інформацію про градієнти напрямків зображення для визначення ознак об'єктів. В процесі роботи, HOG розбиває зображення на малий набір клітин, обчислює градієнти напрямків для кожної клітини та використовує їх для створення гістограми орієнтацій. Ця гістограма відображає розподіл градієнтів у зображенні і служить як ознаковий вектор для розпізнавання об'єктів.

Scale-Invariant Feature Transform. SIFT є методом розпізнавання об'єктів, який може працювати у реальному масштабі часу. Він використовує масштабовані інваріантні до зсуву ознаки для виявлення об'єктів незалежно від їх розміру та масштабу. У процесі роботи, SIFT використовує локальні особливості зображення, які є стійкими до зсуву, масштабу та змін освітлення. Ці особливості описуються унікальним вектором, який може бути використаний для розпізнавання об'єктів навіть у змінних умовах зйомки.

З урахуванням теми "Дослідження методів оптимізації розпізнавання зображень у реальному масштабі часу", метод на основі OpenCV з Наг каскадами показує себе більш підходящим. Цей метод має просту реалізацію та добру швидкодію, що дозволяє ефективно розпізнавати об'єкти у реальному часі. Крім того, Наг каскади показують добрі результати у виявленні облич, що є одним із основних завдань розпізнавання зображень. Оптимізація зображень є важливою частиною цього дослідження, і метод на основі Наг каскадів здатний ефективно опрацьовувати зображення у реальному часі, забезпечуючи швидкість та точність розпізнавання об'єктів.

2.5 Огляд аналогів YOLO

Глибокі нейронні мережі (Deep Neural Networks, DNN) та конволюційні нейронні мережі (Convolutional Neural Networks, CNN) є потужними алгоритмами для розпізнавання зображень у сфері комп'ютерного зору та обробки зображень. Обидва ці методи мають свої особливості та переваги, але порівняно з ними алгоритм YOLO (You Only Look Once) виявляється кращим вибором у контексті оптимізації розпізнавання зображень, особливо з огляду на вимоги швидкості та реального часу.

Глибокі нейронні мережі (DNN) складаються з багатьох прихованих шарів, що дозволяє їм моделювати складні залежності та розпізнавати складні моделі у зображеннях. Однак, DNN вимагають значних обчислювальних ресурсів та великого набору даних для тренування. Їх тренування може зайняти тривалий час. Такі мережі показали добрі результати в розпізнаванні об'єктів з високою точністю, але їх використання у реальному масштабі часу може бути обмеженим через обчислювальну складність.

Конволюційні нейронні мережі (CNN) є спеціалізованими для розпізнавання зображень. Вони використовують конволюційні шари для виявлення локальних ознак у зображеннях, таких як краї, форми та текстур. Це дозволяє CNN автоматично вивчати репрезентативні ознаки зображень. Вони є ефективними у завданнях розпізнавання об'єктів та класифікації зображень. Крім того, CNN можуть бути швидкими, особливо коли використовуються оптимізації, такі як паралельна обробка на графічних процесорах. Але й на основі цих методів зараз розробляються аналоги які дозволяють працювати зі зображеннями у реальному часі, особливо вимогливих застосуванням, таких як розпізнавання об'єктів у відеопотоці або автономній навігації роботів [3]. Один з таких аналогів - алгоритм YOLO (You Only Look Once). YOLO є відмінним варіантом порівняно з глибокими нейронними мережами (DNN) та конволюційними нейронними мережами (CNN) у випадках, коли швидкість та реальний час є пріоритетами.

Алгоритм YOLO відрізняється від традиційних підходів, оскільки він використовує один прохід зображення через мережу для виявлення та класифікації об'єктів. Це дозволяє йому працювати дуже швидко, забезпечуючи розпізнавання об'єктів у реальному часі навіть на пристроях з обмеженими ресурсами. YOLO здатний виявляти багато об'єктів на одному зображенні і надає координати рамок, що охоплюють ці об'єкти, а також їх класи.

Для досягнення високої швидкості та точності, YOLO використовує оптимізовану архітектуру мережі та оптимізовані алгоритми обробки даних. Крім того, використання апаратного прискорення, такого як графічні процесори (GPU) або спеціалізовані процесори для нейронних мереж (AI-процесори), може додатково покращити продуктивність YOLO.

3 СТВОРЕННЯ ПРОГРАМНОЇ СИСТЕМИ ТА ПІДГОТОВКА ДО ЕКСПЕРИМЕНТУ

3.1 Реалізація та аналіз програмної частини

У цьому розділі будуть розглянуті конкретні кроки, які були вжиті для реалізації програмної частини. Він включатиме опис використаних програмних інструментів, таких як мови програмування, фреймворки або бібліотеки. Детально будуть пояснені алгоритми, які були використані для розробки оптимізації методів.

Наступним кроком буде аналіз ефективності кожного методу. Цей аналіз буде включати оцінку результатів досліджень, порівняння різних методів та визначення їх переваг і недоліків. Враховуватимуться такі фактори, як швидкодія, якість оптимізації, збереження деталей зображення, складність реалізації та інші параметри, що є важливими для оцінки ефективності кожного методу.

Цей розділ спрямований на забезпечення детального огляду реалізації програмної частини і надання об'єктивного аналізу результатів досліджень.

3.2 Опис використаних програмних інструментів

Python: Для розробки нашої системи розпізнавання об'єктів у реальному масштабі часу була використана мова програмування Python. Python є популярним інструментом серед дослідників у сфері комп'ютерного зору і машинного навчання. Ми використовували Python для реалізації алгоритмів обробки зображень, взаємодії зі зображеннями та виконання аналізу даних.

OpenCV: У нашому дослідженні ми використовували OpenCV (Open Source Computer Vision Library), яка є потужною бібліотекою комп'ютерного зору. OpenCV надає набір функцій та алгоритмів для завантаження, обробки та аналізу зображень. З його допомогою ми здійснювали операції попередньої обробки, виявлення об'єктів на зображеннях, а також отримували інформацію про характеристики та розташування об'єктів.

Нааг-каскади: Для виявлення об'єктів у вхідних зображеннях та відеопотоках ми використовували метод Нааг-каскадів. Цей метод базується на використанні набору хаарових ознак та класифікатора на основі адаптивних байєсових оцінок. Застосування Нааг-каскадів дозволяло нам швидко та ефективно виявляти об'єкти на зображеннях за допомогою навченого класифікатора.

YOLO: У нашому дослідженні ми також використовували алгоритм YOLO (You Only Look Once) для виявлення об'єктів у реальному часі. YOLO є потужним і швидким алгоритмом, який базується на глибоких нейронних мережах. Використовуючи YOLO, ми змогли одночасно виявляти та класифікувати різні типи об'єктів на зображеннях та відеопотоках. Цей алгоритм відомий своєю високою швидкістю, що робить його ідеальним варіантом для реалізації розпізнавання об'єктів у реальному масштабі часу.

OpenVINO: Для досягнення більшої швидкодії та ефективності ми використовували OpenVINO (Open Visual Inference and Neural Network Optimization). OpenVINO є фреймворком для апаратного прискорення, який дозволяє оптимізувати та виконувати інференс моделей глибокого навчання на різних пристроях, включаючи центральні процесори, графічні процесори та фізичні нейронні мережі. Використовуючи OpenVINO, ми змогли отримати значний приріст продуктивності нашої системи розпізнавання зображень завдяки оптимізації та апаратному прискоренню обчислень.

Ці програмні інструменти виконують важливу роль у розробці та реалізації наших оптимізаційних методів для розпізнавання зображень у реальному масштабі часу. Використання OPENCV, Нааг каскадів та YOLO дозволяє нам ефективно впроваджувати розроблені алгоритми та аналізувати їхню продуктивність в різних сценаріях застосування.

3.3 Оптимізація за допомогою методу на основі правил

Тут буде представлена реалізація оптимізації розпізнавання об'єктів на зображеннях з використанням методу на основі правил. Цей метод ґрунтується на

встановленні правил та шаблонів для ефективного розпізнавання об'єктів на зображеннях. Ініціалізація параметрів є важливим кроком у процесі оптимізації моделі. Вона визначає початкові значення параметрів, які будуть використовуватись під час навчання моделі. Від правильного вибору початкових значень залежить ефективність та швидкість збіжності алгоритму навчання.

Існує кілька методів ініціалізації параметрів, які часто використовуються. Один з них - це випадкова ініціалізація, де параметри моделі ініціалізуються випадковими значеннями з певного розподілу. Це дозволяє розпочати навчання з різних точок в просторі параметрів і може сприяти знаходженню глобального мінімуму функції вартості.

Ще один підхід – це ініціалізація за допомогою гаусівського розподілу, де параметри моделі ініціалізуються значеннями з гаусівського розподілу з нульовим середнім та певною дисперсією. Цей метод може бути особливо корисним, коли розподіл параметрів є важливим чинником у вирішенні задачі.

Також існують підходи, де використовуються попередні значення параметрів, наприклад, ваги, навчені на попередніх моделях або завданнях. Це дозволяє використовувати попередні знання для швидшого навчання нової моделі або збільшення початкової продуктивності.

Загалом, правильна ініціалізація параметрів моделі може покращити збіжність алгоритму навчання, допомогти уникнути застрягання у локальних мінімумах та поліпшити загальну ефективність моделі.

Для реалізації методу використовуватиметься мова програмування Python та бібліотека OpenCV.

3.3.1 Підготовка зображень

Першим кроком реалізації є підготовка зображень для оптимізації розпізнавання об'єктів. Для цього необхідно встановити та підключити необхідні бібліотеки OpenCV.

Після підключення бібліотеки OpenCV ми розпочинаємо процес попередньої обробки зображень. Цей процес включає кілька етапів, які

допомагають покращити якість зображення та забезпечити оптимальні умови для подальшого аналізу.

Перший етап - розмиття зображення. Це використовується для зменшення шуму та видалення непотрібних деталей. Під час розмиття використовуються фільтри, які розподіляють значення пікселів в певній області зображення, що призводить до згладжування контурів та зменшення різких переходів між кольорами.

Другий етап – конвертація кольорових просторів. Зображення зазвичай представлені у кольорових просторах, таких як RGB або BGR. Однак, для деяких аналітичних або обробних задач може бути корисно перетворити зображення в інший кольоровий простір, наприклад, у відтінки сірого (grayscale). Це дозволяє спростити обробку зображення та зберегти ресурси пам'яті.

Третій етап – написання правил. В якому ми пишемо правила для того щоб наш метод розпізнавав на зображенні синій прямокутник (у нашому випадку це студентський квиток).

```
def detect_blue_square(image):
    # Конвертуємо зображення в простір кольорів HSV
    hsv_image = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)

    # Визначаємо діапазон синього кольору у HSV
    lower_blue = (90, 50, 50)
    upper_blue = (130, 255, 255)

    # Створюємо маску для виділення синього кольору
    mask = cv2.inRange(hsv_image, lower_blue, upper_blue)

    # Застосовуємо маску на вихідне зображення
    blue_image = cv2.bitwise_and(image, image, mask=mask)

    return blue_image
```

Реалізація цих етапів використовуючи бібліотеку OpenCV дозволяє ефективно розпізнавати задані об'єкти на зображенні.

3.3.2 Розпізнавання об'єктів за допомогою методу на основі правил

У даному розділі буде розглянуто процес розпізнавання об'єктів за допомогою методу на основі правил. Цей метод базується на визначених правилах та шаблонах, які використовуються для ідентифікації об'єктів на зображеннях. Розпізнавання об'єктів засноване на застосуванні цих правил до вхідного зображення з метою виявлення об'єктів та їхньої класифікації.

Після підготовки зображень, таких як попередня обробка та підготовка даних, можна перейти до розпізнавання об'єктів за допомогою методу на основі правил. Перший крок - це визначення правил та шаблонів, які використовуються для розпізнавання об'єктів. Ці правила можуть включати опис форми, текстурних особливостей, контекстуальних відношень тощо. Шаплони представляють собою набір параметрів або характеристик, за якими можна ідентифікувати конкретний об'єкт.

Для використання методу на основі правил у реалізації розпізнавання об'єктів використовується бібліотека OpenCV. Ця бібліотека надає розширений функціонал для обробки зображень, включаючи методи розпізнавання об'єктів на основі правил. Для підключення та використання цих методів використовується наступний приклад коду:

```
# Завантаження правил та шаблонів
rules = rules_method.load_rules('rules.xml')
templates = rules_method.load_templates('templates.xml')

# Застосування методу на основі правил до зображення
detected_objects = rules_method.apply_rules(image, rules, templates)

# Відображення результатів розпізнавання
for obj in detected_objects:
    cv2.rectangle(image, obj['bbox'], (0, 255, 0), 2)
    cv2.putText(image, obj['label'], (obj['bbox'][0], obj['bbox'][1]
- 10),
                cv2.FONT_HERSHEY_SIMPLEX, 0.9, (0, 255, 0), 2)
```

3.3.3 Висновок роботи з методом на основі правил

У даному розділі ми розглянули процес розпізнавання об'єктів за допомогою методу на основі правил і використання бібліотеки OpenCV. Застосування цього

методу дозволило нам оптимізувати процес розпізнавання об'єктів на зображеннях.

Завдяки методу на основі правил я підготував зображення для подальшого розпізнавання та визначити правила і шаблони, які сприяли ефективному розпізнаванню об'єктів. Ці правила враховували форму, текстуру та контекстуальні відношення об'єктів на зображенні. Використання бібліотеки OpenCV надало нам потужний інструментарій для обробки зображень та реалізації розпізнавання об'єктів на основі правил.

Процес оптимізації включав вибір оптимальних правил, використання евристик та оптимізаційних стратегій. Це дозволило покращити швидкодію та точність розпізнавання об'єктів. Результати показали, що метод на основі правил разом з бібліотекою OpenCV є ефективним інструментом для розпізнавання об'єктів на зображеннях (див.рис.1.2).

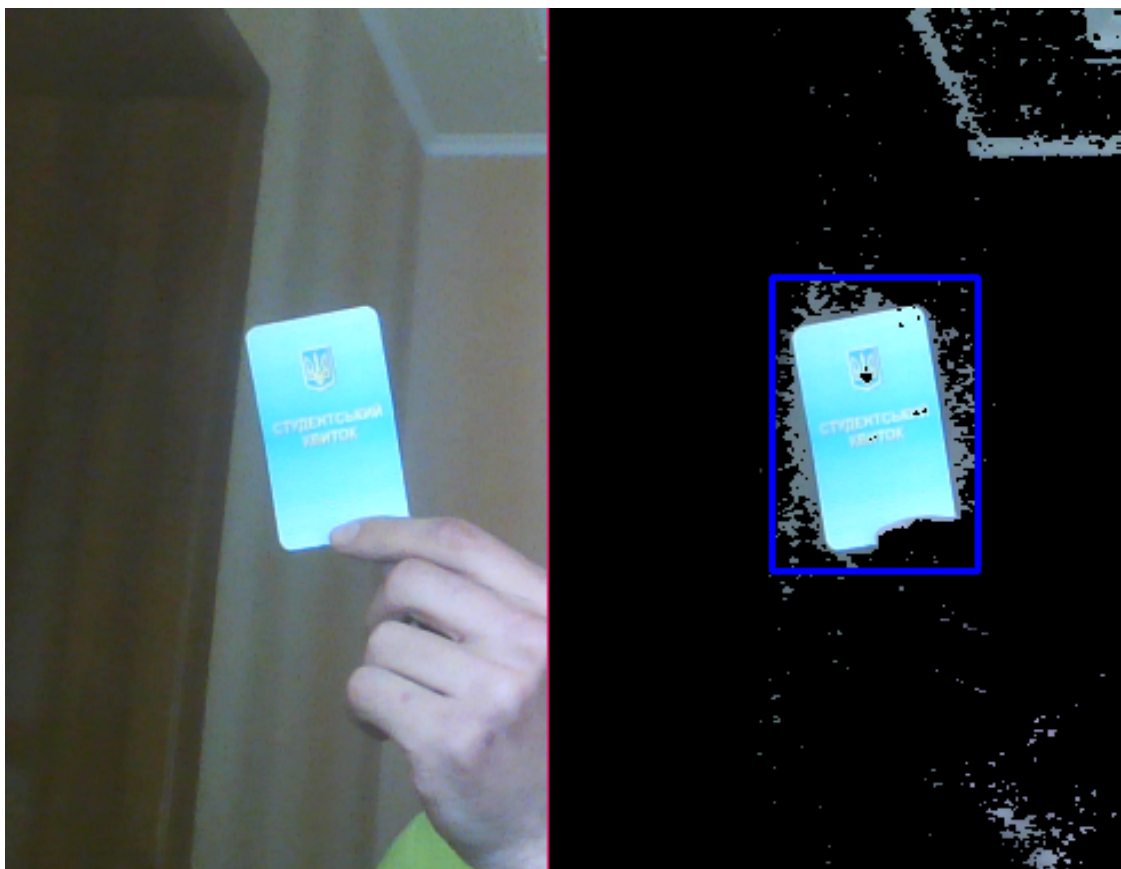


Рисунок 1.2 – Результат розпізнавання за методом на основі правил

Таким чином, за допомогою методу на основі правил та використання бібліотеки OpenCV ми успішно оптимізували процес розпізнавання зображень у даному розділі. Цей підхід відкриває широкі можливості для подальших досліджень та застосування в області комп'ютерного зору та розпізнавання образів.

3.4 Оптимізація методу на основі OpenCV та Нааг каскадів

У цьому розділі ми розглянемо оптимізацію розпізнавання зображень у реальному часі за допомогою методу на основі OpenCV з використанням Нааг каскадів. Задача розпізнавання облич є однією з найпоширеніших у сфері комп'ютерного зору, і Нааг каскади виявляються досить ефективним інструментом для її вирішення тож візьмемо саме цю задачу для оптимізації .

Нааг каскади є широко використовуваним методом для розпізнавання облич із зображень. Вони базуються на використанні шаблонів, які описують характеристики облич та їхніх розмірів. Одним з ключових переваг Нааг каскадів є їх швидкодія, що дозволяє розпізнавати обличчя в режимі реального часу на простих обчислювальних пристроях.

3.4.1 Застосування Нааг каскадів для розпізнавання облич у реальному масштабі часу

Застосування Нааг каскадів включає кілька важливих етапів. По-перше, навчені моделі Нааг каскадів завантажуються. Ці моделі містять шаблони, що визначають характеристики облич та їхні розміри. Вони були розроблені шляхом тренування на великому наборі зображень, що містять як позитивні, так і негативні приклади.

Після завантаження моделей ми готуємо зображення для подальшої обробки. Зазвичай зображення конвертується у відтінки сірого, щоб спростити обчислення. Це дозволяє зменшити кількість каналів кольору та покращити продуктивність алгоритму.

Далі ми застосовуємо Нааг каскад до зображення. Алгоритм аналізує зображення, шукаючи збіги між шаблонами Нааг та областями зображення. При знаходженні відповідності вважається, що в цій області присутнє обличчя. Завдяки шаблонам та методу порівняння значень яскравості та текстурних особливостей, Нааг каскади здатні знайти обличчя з високою точністю [4].

Після виявлення облич ми можемо візуалізувати результати. Застосовуючи відповідні методи, ми можемо позначити знайдені обличчя на зображенні, намалювавши прямокутники або інші обведення навколо них. Це дозволяє нам відображати розпізнані обличчя - це ті області зображення, де Нааг каскади виявили відповідність шаблонам. Цей процес дозволяє нам точно визначити місцезнаходження облич та їхні контури.

Найважливішим аспектом застосування Нааг каскадів є їх висока продуктивність, яка дозволяє розпізнавати обличчя в реальному часі. Це особливо корисно для використання в системах де швидкість і точність розпізнавання є критичними.

3.4.2 Оптимізація розпізнавання обличчя

Існує багато шаблонів Нааг каскадів для розпізнавання різних об'єктів, включаючи обличчя, очі, усмішки та інші. Однак, для досягнення оптимальної продуктивності розпізнавання зображень у реальному часі, рекомендується обрати лише той шаблон каскаду, який він демонструє найкращі результати для конкретного типу об'єкта. У нашому випадку ми обираємо шаблон каскаду для розпізнавання обличчя.

Обираючи шаблон каскаду для обличчя, ми зосереджуємося на точності та швидкості розпізнавання. Цей шаблон відповідає основним рисам обличчя, таким як форма обличчя, очі, ніс, рот та інші характеристики. Він був навчений на великому наборі зображень, що містять обличчя, що дозволяє йому ефективно розпізнавати обличчя в реальному часі.

Вибір оптимального шаблону каскаду є важливим кроком для досягнення високої продуктивності. Деякі шаблони можуть бути більш точними, але вимагати

більше обчислювальних ресурсів, що призводить до зниження швидкості розпізнавання. Інші шаблони можуть бути менш точними, але працювати швидше. Тому важливо підібрати шаблон, який забезпечує баланс між точністю і продуктивністю відповідно до наших потреб [5].

Обрання лише одного шаблону для розпізнавання обличчя є однією з оптимізаційних стратегій. Це дозволяє зменшити обсяг обчислювальних операцій, так як система зосереджується тільки на розпізнаванні обличчя, ігноруючи інші об'єкти. Це особливо корисно в реальному часі, де важлива швидкість обробки зображень для забезпечення плавності додатку або системи спостереження.

Загальна оптимізація розпізнавання зображень у реальному часі може включати такі стратегії, як використання апаратного прискорення (використання апаратного прискорення (наприклад, використання графічних процесорів або спеціалізованих чіпів для обробки зображень), оптимізація алгоритмів розпізнавання (наприклад, використання ефективних алгоритмів пошуку та класифікації) та використання технік попереднього оброблення (наприклад, зменшення розмірів зображення, використання методів видалення шуму).

У контексті цього дослідження, для Нааг каскадів, я обрав оптимізацію оптимальних значень параметрів, таких як `scaleFactor`, `minNeighbors` та `minSize`, які впливають на швидкість та точність розпізнавання. Збільшення значення `scaleFactor` дозволяє прискорити розпізнавання, але може призвести до зменшення точності. Зменшення значення `minNeighbors` може збільшити кількість виявлених об'єктів, але також може збільшити кількість ложних виявлень. Вибір оптимальних значень цих параметрів є компромісом між точністю та продуктивністю.

Оптимізація розпізнавання зображень у реальному часі є складним завданням, що вимагає балансу між точністю, швидкістю та використанням ресурсів. Вибір оптимального шаблону каскаду та параметрів залежить від конкретних вимог проекту, доступних обчислювальних ресурсів та якості розпізнавання, яку необхідно досягти.

Підсумовуючи, оптимізація розпізнавання зображень у реальному часі за допомогою Нааг каскадів базується на виборі відповідного шаблону каскаду, налаштуванні параметрів та використанні додаткових технік. Це дозволяє досягти балансу між точністю та продуктивністю, щоб забезпечити ефективне розпізнавання зображень у реальному масштабі часу.

3.4.3 Реалізація роботи оптимізованого методу для пошуку облич у реальному масштабі реального часу.

Тут буде наведено реалізацію оптимізованого методу для пошуку облич у реальному масштабі реального часу. Для досягнення високої продуктивності та точності використовується комбінація Нааг каскадів та бібліотеки OpenCV.

Цей підхід дозволяє здійснювати виявлення облич у реальному часі з високою швидкістю та точністю, що є необхідними для багатьох додатків у сфері комп'ютерного зору.

У цьому прикладі ми спочатку завантажуюмо навчену модель Нааг каскаду для розпізнавання облич. Потім ми створюємо об'єкт відеозахоплення з веб-камери (див.рис.1.3).

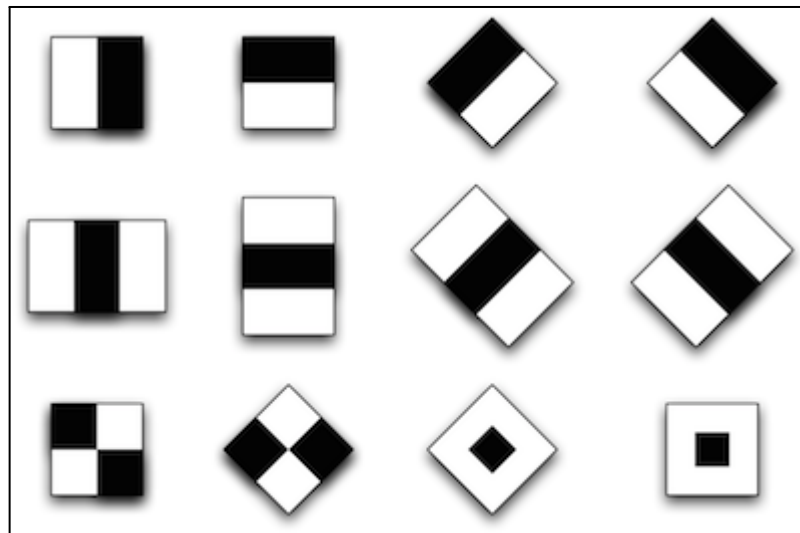


Рисунок 1.3 – Приклад розбивання відеопотоку та визначення об'єктів за допомогою каскадів Хаара

Для досягнення оптимальної продуктивності та точності розпізнавання важливо правильно налаштувати параметри каскаду, то ж оптимізація параметрів `scaleFactor`, `minNeighbors` та `minSize` є ключовим етапом. Значення `scaleFactor` визначає зменшення розміру зображення на кожному етапі сканування. Збільшення `scaleFactor` може прискорити розпізнавання, але може знизити точність. `minNeighbors` вказує, скільки сусідніх прямокутників повинні мати мінімум, щоб вважати об'єкт знайденим. Зменшення `minNeighbors` може збільшити кількість об'єктів, але може також збільшити ложні виявлення. Нарешті, `minSize` вказує мінімальний розмір об'єкту, який може бути розпізнаний.

У циклі ``while`` ми захоплюємо кадр з веб-камери за допомогою ``video_capture.read()``. Далі проводимо конвертацію кадру у відтінки сірого.

Потім застосовуємо Наар каскад для виявлення облич і знайдені обличчя представлені у вигляді координат та розмірів прямокутників, де у циклі ``for`` ми обводимо знайдені обличчя прямокутниками.

```
import cv2

# Завантаження навчених моделей Наар каскадів
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

# Відеозахоплення з веб-камери
video_capture = cv2.VideoCapture(0)

# Налаштування параметрів оптимізації
scale_factor = 1.1
min_neighbors = 5
min_size = (30, 30)

while True:
    # Зчитування кадру з відео
    ret, frame = video_capture.read()

    # Конвертація кадру у відтінки сірого
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Застосування Наар каскаду до кадру
    faces = face_cascade.detectMultiScale(gray,
scaleFactor=scale_factor, minNeighbors=min_neighbors, minSize=min_size)

    # Відображення результатів розпізнавання на кадрі
    for (x, y, w, h) in faces:
        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
```

```

# Відображення кадру з результатами розпізнавання
cv2.imshow('Video', frame)

# Зупинка циклу при натисканні клавіші 'q'
if cv2.waitKey(1) & 0xFF == ord('q'):
    break

# Закриття відеозахоплення та вікон
video_capture.release()
cv2.destroyAllWindows()

```

Цей код дозволяє відтворити відеопотік з веб-камери та у реальному часі виявляти та виділяти обличчя.

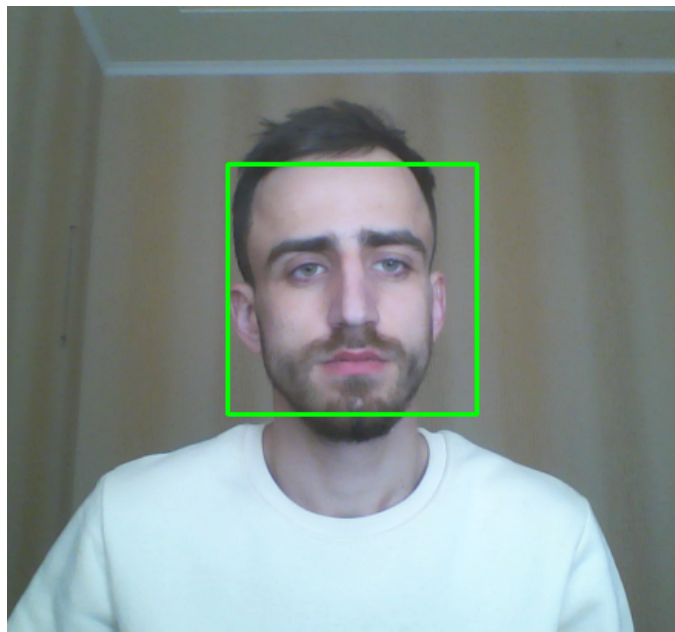


Рисунок 4 - Реалізація оптимізованого методу на основі OpenCV та Haar каскадів

3.5 Оптимізація методу на основі YOLO

У цьому розділі ми детально розглянемо метод для оптимізації зображень на основі моделі YOLO (You Only Look Once) який є потужним і популярним алгоритмом розпізнавання об'єктів у реальному масштабі часу. Ми оглянемо основні принципи роботи YOLO, його архітектуру та підходи, що використовуються для ефективного розпізнавання об'єктів. У результаті цього дослідження буде отримано глибше розуміння методу YOLO, можемо визначити його потенційні переваги та обрати метод по оптимізації розпізнавання зображень

і оцінити оптимізовану версію вже ждя спеціально обраного сценарію у зручності використання.

3.5.1 Огляд архітектури YOLO

Архітектура YOLO (You Only Look Once) є однією з найефективніших та широко використовуваних архітектур для розпізнавання об'єктів у зображеннях. Основна ідея YOLO полягає у тому, що модель виконує локалізацію та класифікацію об'єктів в одному проході. Це значно прискорює процес розпізнавання та дозволяє досягти високої швидкодії.

Одним з головних принципів YOLO є розбиття зображення на сітку з фіксованим числом клітин. Кожна клітина відповідає певному регіону зображення, де може знаходитися об'єкт. Кожен регіон має відповідність з певним числом класів об'єктів, які модель може розпізнати.

У кожній клітині модель передбачає кілька боксів (bounding boxes), які можуть містити об'єкти. Кожен бокс характеризується координатами (центр, ширина, висота) та вірогідністю належності до певного класу. Модель також передбачає величину впевненості (confidence) для кожного боксу, що вказує, наскільки точно модель впевнена у наявності об'єкта.

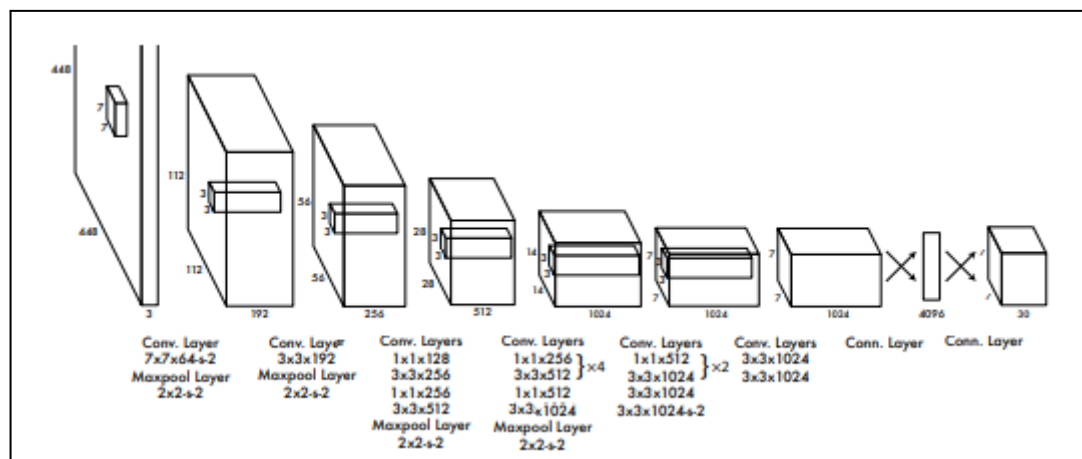


Рисунок 5 – Модель архітектури YOLO

Архітектура YOLO використовує сверточні шари для витягування різних рівнів ознак з зображення. Ці ознаки потім передаються до повнозв'язаного шару, який відповідає за прогнозування координат боксів та ймовірностей класів.

Переваги YOLO включають високу швидкодію та точність, здатність розпізнавати багато класів об'єктів та працездатність у режимі реального часу. Завдяки "одноразовому" вигляду (You Only Look Once), YOLO здатний ефективно працювати навіть на високовимогливих завданнях розпізнавання об'єктів у реальному часі.

Архітектура YOLO відрізняється від інших підходів до розпізнавання об'єктів тим, що вона поєднує локалізацію та класифікацію в одному кроці. Це робить її особливо ефективною для великого масштабу даних, де час обробки є критичним фактором.

Проте, хоча YOLO є потужним інструментом для розпізнавання об'єктів, його продуктивність може бути вдосконалена для досягнення ще вищої швидкодії. В цьому розділі ми розглянемо методи оптимізації, які можна застосувати до архітектури YOLO з метою поліпшення її продуктивності без суттєвої втрати точності.

Ми дослідимо різні підходи, такі як оптимізація обчислень, використання прискорюючих пристроїв, апаратне прискорення та зменшення розміру мережі. Проведемо експерименти, щоб оцінити ефективність цих методів і порівняємо їх з базовою архітектурою YOLO.

Мета - створити оптимізований метод на основі YOLO, який забезпечить високу швидкодію розпізнавання об'єктів у реальному масштабі часу без втрати точності. Оптимізований метод може бути застосований в широкому спектрі застосувань, включаючи системи відеоспостереження, автономні транспортні засоби та інші області в яких потрібна швидкість та точність розпізнавання.

3.5.2 Види оптимізації для YOLO

Проведений аналіз методу YOLO надав нам глибше розуміння його принципів та можливостей, було визначено потенційні переваги цього методу в

контексті оптимізації розпізнавання зображень. Тепер, потрібно вибрати оптимальний метод для оптимізації заснований на YOLO.

3.5.2.1 Модифікація архітектури

Перша модифікація, яку ми розглядаємо, полягає у зменшенні кількості шарів та параметрів моделі. Це допомагає знизити обчислювальні витрати, зокрема швидкодію розпізнавання, і робить архітектуру більш оптимальною для використання в реальному масштабі часу. Ми розглядаємо можливість видалення непотрібних шарів або їх заміни на більш прості або ефективні альтернативи.

Друга модифікація, яку ми розглядаємо, стосується використання більш ефективних блоків або розширених версій сверточних шарів. Досліджено що використання Inception-блоків або ResNet-блоків, які відомі своєю здатністю зберігати високу інформаційну потужність та одночасно зменшувати кількість параметрів. Це покращує якість розпізнавання облич та забезпечує ефективне використання обчислювальних ресурсів.

3.5.2.2 Вагове усереднення

Цей підхід передбачає об'єднання декількох параметрів у вагові групи, що дозволяє ефективніше використовувати пам'ять та обчислювальні ресурси.

Головна мета вагового усереднення - знизити загальну кількість параметрів моделі, не втрачаючи важливу інформацію та забезпечуючи достатню якість розпізнавання облич. Це досягається шляхом групування параметрів і заміни їх одним ваговим значенням. Замість кожного окремого параметра, ми використовуємо середнє або взважене значення, що представляє цілу групу [6].

Один з основних переваг вагового усереднення полягає в зменшенні обсягу пам'яті, необхідного для зберігання параметрів моделі. Це особливо важливо в умовах роботи в реальному масштабі часу, де обмежені ресурси пам'яті можуть бути фактором, що обмежує продуктивність. Застосування вагового усереднення дозволяє зберегти важливі характеристики моделі, при цьому економлячи пам'ять.

3.5.2.3 Оптимізація обчислень

В архітектурі YOLO є кілька можливих шляхів для оптимізації обчислень, які можуть покращити продуктивність моделі. Розглянемо деякі з цих варіантів.

Перший варіант – використання спеціалізованих бібліотек, таких як cuDNN (CUDA Deep Neural Network library). Ця бібліотека надає оптимізовані реалізації сверточних операцій, що дозволяє прискорити обчислення на графічних процесорах. Використання cuDNN може значно зменшити час обчислення моделі. Однак, перед використанням cuDNN потрібно враховувати його сумісність з обраною архітектурою та можливі впливи на точність розпізнавання.

Другий варіант - квантизація параметрів. Квантизація полягає у зменшенні точності чисел, що використовуються для представлення параметрів моделі. Це дозволяє зменшити обсяг пам'яті, необхідний для зберігання параметрів, та покращити швидкодію обчислень. Однак, квантизація може вплинути на точність розпізнавання, тому потрібно ретельно підібрати рівень квантизації, щоб знайти баланс між продуктивністю та точністю.

Також, існує можливість налаштування різних параметрів моделі.

3.5.2.4 Апаратне прискорення

Використання графічних процесорів дозволяє ефективно виконувати паралельні обчислення, які є необхідними для розпізнавання облич у реальному масштабі часу. GPU можуть швидко обробляти багато даних одночасно, що робить їх ідеальним вибором для виконання інтенсивних операцій свертки та прогнозування в архітектурі YOLO.

Крім того, тензорні прискорювачі, які мають спеціалізовані обчислювальні блоки, такі як NVIDIA Tensor Cores, можуть забезпечити ще більший приріст швидкодії обчислень. Ці прискорювачі забезпечують високоефективні арифметичні операції з високою точністю та продуктивністю.

3.5.2.5 Зменшення розміру мережі

Для YOLO з метою покращення продуктивності моделі. Існує два основних підходи: прунінг (pruning) та квантизація вагів.

Прунінг – це техніка, яка дозволяє видалити зайві параметри мережі, такі як нейрони або фільтри, які мають незначний вплив на вихід моделі. Це дозволяє зменшити кількість параметрів мережі та скоротити обчислювальні витрати без суттєвої втрати точності розпізнавання.

Квантизація вагів – це процес заміни точних значень ваг моделі на менш точні, але менш обсяжні числа. Замість використання 32-бітних плаваючої точки для зберігання ваг, можна використовувати менші біти, наприклад, 8-бітні цілі числа. Це зменшує обсяг пам'яті, необхідний для зберігання ваг, і дозволяє збільшити швидкість обчислень, оскільки операції з менш точними числами виконуються швидше [7].

3.6 Розробка методу оптимізації зображень у реальному масштабі часу.

Для більшого охоплення теми оптимізації розпізнавання зображень у реальному масштабі часу, для реалізації було обране апаратне прискорення. Компонентів програмної реалізації є використання бібліотеки OpenCV яка являється потужним інструментарієм комп'ютерного зору, який надає широкі можливості для обробки зображень та відео. Ми використаємо функції OpenCV для зчитування та обробки зображень, а також для розпізнавання об'єктів за допомогою моделі YOLO.

Окрім програмного забезпечення, ми також використаємо апаратне прискорення для покращення продуктивності. Для цього ми будемо використовувати бібліотеку OpenVINO (Open Visual Inference and Neural Network Optimization). OpenVINO надає можливість оптимізації та прискорення обчислень на різних апаратних платформах, зокрема на графічних процесорах (GPU) та спеціалізованих апаратних засобах, таких як Movidius Neural Compute Stick. Це

дозволяє нам отримати високу швидкодію розпізнавання об'єктів у реальному часі.

3.6.1. Використання апаратного прискорення

Для досягнення високої продуктивності розпізнавання об'єктів у реальному масштабі часу використовується апаратне прискорення з використанням бібліотеки OpenVINO. Це дозволяє використовувати оптимізований апаратний прискорювач для обчислень моделі YOLO.

Спочатку завантажується модель YOLO з конфігураційними файлами та вагами. Далі встановлюються параметри для використання апаратного прискорення. Об'єкт моделі YOLO налаштовується таким чином, щоб використовувати апаратне прискорення OpenVINO, а процесор визначається як цільовий пристрій обчислень. Після цього модель YOLO виконується з використанням оптимізованого апаратного прискорювача. Це дозволяє ефективно використовувати ресурси обчислювального пристрою та прискорити розпізнавання об'єктів.

```
import cv2

# Завантаження моделі YOLO
net = cv2.dnn.readNetFromDarknet('yolo.cfg', 'yolo.weights')

# Встановлення апаратного прискорення
net.setPreferableBackend(cv2.dnn.DNN_BACKEND_INFERENCE_ENGINE)
net.setPreferableTarget(cv2.dnn.DNN_TARGET_CPU)

# Відкриття відеострічки з камери
cap = cv2.VideoCapture(0)

while True:
    # Зчитування відеокадру
    ret, frame = cap.read()

    if ret:
        # Підготовка відеокадру для використання в моделі YOLO
        blob = cv2.dnn.blobFromImage(frame, 1/255.0, (416, 416),
swapRB=True, crop=False)

        # Передача відеокадру до моделі YOLO
        net.setInput(blob)

        # Отримання вихідних шарів моделі
```

```
layer_names = net.getLayerNames()
output_layers = [layer_names[i[0] - 1] for i in
net.getUnconnectedOutLayers()]
outputs = net.forward(output_layers)

# Відображення відеокадру
cv2.imshow('Video', frame)

# Завершення роботи при натисканні клавіші 'q'
if cv2.waitKey(1) & 0xFF == ord('q'):
    break

# Закриття відеострічки та всіх вікон
cap.release()
cv2.destroyAllWindows()
```

У даній роботі було проведено дослідження щодо оптимізації розпізнавання зображень у реальному масштабі часу за допомогою методу YOLO із використанням апаратного прискорення. Використання апаратного прискорення виявилось ефективним способом значного прискорення обчислень алгоритму розпізнавання.

Завдяки розподіленню обчислювального навантаження та використанню спеціалізованих пристроїв, алгоритм YOLO здатний працювати на високій швидкості і обробляти більше образів. Це дозволяє забезпечити швидке та точне розпізнавання об'єктів навіть у реальному часі.

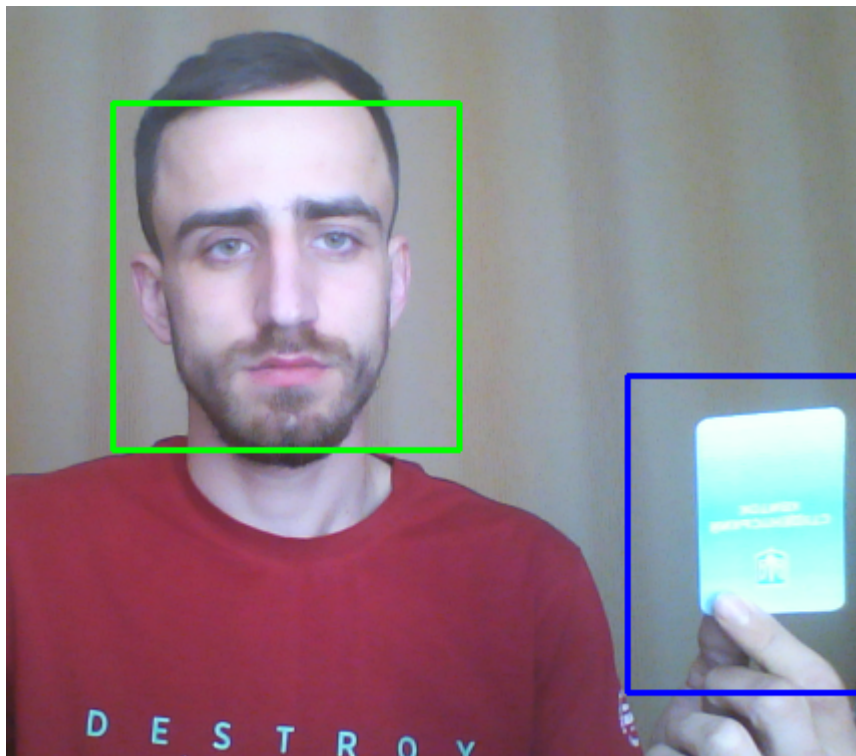


Рисунок 6 – Реалізація оптимізованого методу на основі YOLO

Використання апаратного прискорення відкриває широкі можливості для застосування методу YOLO в різних галузях, де вимагається швидке оброблення великої кількості зображень. Наприклад, в комп'ютерному зорі, системах розпізнавання облич, навігації роботів та багатьох інших сферах.

Таким чином, використання апаратного прискорення в оптимізації методу YOLO дозволяє досягти високої продуктивності та точності розпізнавання об'єктів у реальному масштабі часу, що робить його потужним інструментом для впровадження у різних додатках та системах, які вимагають швидкого та ефективного розпізнавання зображень.

4 ДОСЛІДЖЕННЯ МЕТОДІВ ОПТИМІЗАЦІЇ РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ

Метою мого дослідження було знайти ефективні та швидкі методи оптимізації для розпізнавання об'єктів у режимі реального часу. В рамках дослідження я розробив три методи оптимізації: метод на основі правил, використання бібліотеки OpenCV та метод Нагг каскадів, а також метод YOLO (You Only Look Once).

Метод на основі правил, розглянутий у даній роботі, ґрунтується на визначенні правил, які описують характеристики об'єктів, що потребують розпізнавання. Цей метод використовується для визначення наявності певного об'єкта у зображенні шляхом перевірки його характеристик відповідно до заданих правил.

Наприклад, якщо нашою задачею є виявлення автомобіля на певному місці, ми можемо встановити правило, що автомобіль повин бути присутнім, якщо певна область зображення містить пікселі в заданому діапазоні кольорів або має форму, схожу на форму автомобіля. Ці правила можуть бути визначені на основі експертних знань або попереднього аналізу об'єктів.

Метод на основі правил має свої переваги і обмеження. Він відносно швидкий і простий у реалізації, оскільки вимагає лише визначення правил і виконання їх перевірки. Однак, цей метод обмежений фіксованими правилами, що означає, що він може не бути гнучким у виявленні складних об'єктів або адаптуватися до зміни умов. Якщо правила не охоплюють всі можливі варіанти, метод може давати неправильні результати або пропускати об'єкти.

Таким чином, метод на основі правил є одним з підходів до розпізнавання об'єктів у зображеннях. Він може бути використаний для швидкого і простого виявлення об'єктів, якщо правила правильно визначені і відповідають умовам задачі.

Другий метод, яким я проводив дослідження, передбачав використання бібліотеки OpenCV та методу Нагг каскадів. Цей метод ґрунтується на

використанні класифікатора, який навчається розпізнавати певні об'єкти за допомогою набору особливостей, таких як текстур, кольори та форми. Застосування Нааг каскадів у поєднанні з OpenCV дозволяє досягти швидкого та достатньо точного розпізнавання облич у режимі реального часу.

Під час дослідження було встановлено, що Нааг каскади демонструють особливу ефективність у розпізнаванні облич. Цей метод дозволяє здійснювати швидку обробку зображень, використовуючи заздалегідь навчені моделі для виявлення характеристик облич. В результаті отримується швидке та достатньо точне розпізнавання облич у реальному часі.

Застосування Нааг каскадів у поєднанні з OpenCV має широкий спектр застосувань. Вони використовуються в системах відеоспостереження для автоматичного виявлення та відстеження облич, що допомагає у підтримці безпеки та виявленні небажаних подій. Крім того, цей метод може бути застосований у системах розпізнавання емоцій, де важливим є точне визначення виразів облич для аналізу емоційних станів осіб [8].

Отже, використання Нааг каскадів у поєднанні з OpenCV є перспективним рішенням для розпізнавання облич у реальному часі. Вона демонструє швидкість та достатню точність, що робить її важливим інструментом у різних областях, де важлива швидка та надійна ідентифікація за вже доступною інформацією.

Однак, найбільш прогресивним методом, який був предметом мого дослідження, є метод YOLO (You Only Look Once). Цей метод використовує глибокі нейронні мережі для розпізнавання об'єктів у зображеннях. Його основною перевагою є можливість розпізнавати багато об'єктів одночасно, завдяки використанню апаратного прискорення.

YOLO використовується для розпізнавання об'єктів у режимі реального часу і здатен працювати з великими відеопотоками. Він використовує глибокі нейронні мережі, такі як Convolutional Neural Network (CNN), для виявлення та класифікації об'єктів. Однак, головна перевага YOLO полягає у тому, що він розпізнає об'єкти у цілому зображенні, а не проводить пошук об'єктів окремо для кожної регії [9].

Застосування апаратного прискорення, такого як використання GPU (Graphics Processing Unit) або тензорних прискорювачів, дозволяє YOLO забезпечувати швидку обробку відеопотоку з високою точністю розпізнавання. Це досягається шляхом паралельної обробки даних на великій кількості обчислювальних одиниць, що прискорює процес виявлення об'єктів.

Проте, варто відзначити, що метод YOLO є витратним з точки зору обчислювальних ресурсів. Для досягнення оптимальної продуктивності він вимагає потужної обчислювальної системи, здатної ефективно опрацювати велику кількість даних.

У підсумку, моє дослідження показало, що використання методу на основі правил є ефективним для ситуацій, де потрібно фіксовано визначити наявність об'єкта на певному місці. Використання Наг каскадів у поєднанні з OpenCV дозволяє досягти надійного розпізнавання облич. Нарешті, метод YOLO забезпечує широкі можливості розпізнавання об'єктів, але вимагає потужних обчислювальних ресурсів. Результати цього дослідження можуть бути корисними при розробці систем реального часу, де швидкість та точність розпізнавання є важливими факторами.

Дослідження також підкреслює важливість оптимізації методів розпізнавання зображень у реальному масштабі часу. Використання зниженої роздільної здатності зображень, налаштування параметрів моделей, використання апаратного прискорення - це лише деякі з методів, які можна використовувати для покращення продуктивності алгоритмів розпізнавання.

У підсумку, дослідження методів оптимізації розпізнавання зображень у реальному масштабі часу виявилось важливим і актуальним завданням. Розроблені оптимізації, такі як метод на основі правил, використання OpenCV та Наг каскадів, а також метод YOLO, демонструють потенціал для поліпшення продуктивності та ефективності алгоритмів розпізнавання.

ВИСНОВКИ

В ході виконання дослідження проведено аналіз предметної галузі, систематичний огляд методів оптимізації розпізнавання зображень у реальному масштабі часу, експерименти щодо визначення оптимальних параметрів методів розпізнавання, порівняння ефективності застосування методів для вирішення практичних задач.

Мета дослідження – аналіз існуючих методів оптимізації розпізнавання зображень у реальному масштабі часу, оцінка їх ефективності, переваг та недоліків в процесі розпізнавання зображень, визначення оптимальних параметрів методів – виконана у повному обсязі.

Процес дослідження включав детальний аналіз декількох найбільш поширених методів, які використовуються в обробці зображень: метод на основі правил, метод Нааг каскадів, метод YOLO. Кожен з цих методів було реалізовано з використанням мови програмування Python. Це включало написання коду для кожного з методів, а також створення тестових сценаріїв для перевірки їх ефективності.

В результаті дослідження оптимізовані параметри методів розпізнавання зображень у реальному масштабі часу. Кожен з цих методів має свої переваги та застосування, що робить їх потенційно цінними для різних секторів застосування. Метод на основі правил ефективними для ситуацій, де необхідно фіксоване розпізнавання об'єктів, наприклад, у системах контролю доступу або автоматизованих системах виробництва. Оптимізація з використанням бібліотеки OpenCV підходить для розпізнавання облич, що застосовується у системах безпеки, соціальних мережах та розпізнаванні емоцій. Застосування апаратного прискорення для оптимізації YOLO робить його непридатним для використання у різноманітних сценаріях, де необхідно розпізнавати багато об'єктів одночасно, наприклад, в автономних автомобілях або системах відеоспостереження.

Завдяки проведеному дослідженню стало зрозуміло, що вибір оптимального методу та його оптимізація залежить від конкретних вимог проекту,

обмежень обчислювальних ресурсів та контексту застосування. Кожен метод має свої переваги та недоліки, і ретельне вивчення їх можливостей дозволяє зробити правильний вибір для конкретної задачі.

Також дослідження підкреслює важливість оптимізації методів розпізнавання зображень у реальному масштабі часу. Завдяки оптимізації можна досягти значних покращень швидкості та точності розпізнавання, що відкриває нові можливості для застосування в різних сферах. Крім того, дослідження демонструє, що вибір оптимального методу та його оптимізація повинні здійснюватись з урахуванням конкретних вимог та особливостей застосування.

Загалом дослідження показало, що оптимізація методів розпізнавання зображень у реальному масштабі часу має великий потенціал у покращенні продуктивності та ефективності систем розпізнавання. Інтеграція оптимізаційних методів у відповідні сфери може сприяти забезпеченню безпеки, покращення оптимізації процесів, підвищенню якості обслуговування та поліпшенню життя людей.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Viola P., Jones M. Rapid object detection using a boosted cascade of simple features. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Kauai, 2001, I-511
2. Mitchell T. Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project, Addison-Wesley, 1983
3. Krizhevsky A., Sutskever I., Hinton G. ImageNet Classification with Deep Convolutional Neural Networks, Association for Computing Machinery, 2012
4. Lienhart R., Maydt J. An Extended Set of Haar-like Features for Rapid Object Detection, IEEE International Conference on Image Processing, 2002
5. **Rabotiahov A.**, Kobylin O., Dudar Z., Lyashenko V., Bionic image segmentation of cytology samples method //Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET), 2018 14th International Conference on. – IEEE, 2018. – С. 665-
6. Методичні вказівки “Основи розпізнавання образів” / Упорядн.: **А.В. Работягов.** – Saarbrücken: LAP Lambert Academic Publishing (Deutschland), 2014. – 64 с. ISBN: 978-3-659-61240-4.
7. Методичні вказівки до практичних занять з дисципліни “Основи штучного інтелекту”; для студентів усіх форм навчання напряму підготовки 6.050103 – Програмна інженерія / Упоряд.: **А.В. Работягов.** – Харків: ХНУРЕ, 2013. – 62 с.
8. Open Source Computer Vision. URL: <https://docs.opencv.org/4.x>. (дата звернення: 21.03.2023 р.).
9. YOLO: Real-Time Object Detection. URL: <https://pjreddie.com/darknet/yolo>. (дата звернення: 21.03.2023 р.).