МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ІНЖЕНЕРНИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІМ. Ю.М. ПОТЕБНІ ЗАПОРІЗЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ

КАФЕДРА ЕЛЕКТРОНІКИ, ІНФОРМАЦІЙНИХ СИСТЕМ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Кваліфікаційна робота

другий (магістерський)
(рівень вищої освіти)

на тему **Нейромережна система розпізнавання людей в реальному часі у системах безпеки**

Виконав: студент 2 курсу, групи 8.1212-іпз-2
спеціальності 121 Інженерія програмного забезпечення
(код і назва спеціальності)
освітньої програми Інженерія програмного забезпечення
(код і назва освітньої програми)
О.І. Степаненко
(ініціали та прізвище)
Керівник доцент, А.І. Безверхий
(посада, вчене звання, науковий ступінь, підпис, ініціали та прізвище)
Рецензент директор ТОВ «Дісітел»
П.О. Лютий
(посада, вчене звання, науковий ступінь, підпис, ініціали та прізвище)

Запоріжжя
2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ІНЖЕНЕРНИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІМ. Ю.М. ПОТЕБНІ
ЗАПОРІЗЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ

Кафедра електроніки, інформаційних систем та програмного забезпечення
Рівень вищої освіти _____другий (магістерський)_____
Спеціальність ___121_Інженерія програмного забезпечення_____
(код та назва)
Освітня програма ___Інженерія програмного забезпечення
(код та назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри_____Тетяна КРИТСЬКА_

"_02" вересня 2024_року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Степаненку Олегові Ігоровичу
(прізвище, ім'я, по батькові)

1.Тема роботи _Нейромережна система розпізнавання людей в реальному часі у системах безпеки_____

керівник роботи ___доцент_____Безверхий Анатолій Ігорович_____
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом ЗНУ від ___28.02.2024р._№_361-е_____

2. Строк подання студентом кваліфікаційної роботи_____30.11.2024_____
3. Вихідні дані магістерської роботи
• комплект нормативних документів ;
• технічне завдання до роботи.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
• огляд та збір літератури стосовно теми кваліфікаційної роботи;
• огляд та аналіз існуючих рішень та аналогів;
• дослідження проблеми розпізнавання та розробка методів її вирішення;
• створення програмного продукту та його опис;
• перелік вимог для роботи програми;
• дослідження поставленої проблеми та розробка висновків та пропозицій.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
16 слайдів презентації

6. Консультанти розділів магістерської роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата Завдання прийняв

7. Дата видачі завдання 02.09.2024

КАЛЕНДАРНИЙ ПЛАН

з/п	Назва етапів магістерської роботи	Строк виконання етапів магістерської роботи	Примітка
1	Аналіз предметної області	02.09-10.09.24	виконано
2	Формулювання основної задачі дипломної роботи та узгодження її з науковим керівником	11.09-12.09.24	виконано
3	Аналіз проблематики	13.09-14.09.24	виконано
4	Дослідження засобів та технологій забезпечення безпеки	15.09-20.09.24	виконано
5	Дослідження основних типів існуючих нейронних мереж	21.09-26.09.24	виконано
6	Узгодження подальших дій з науковим керівником	27.09-28.09.24	виконано
7	Збір тренувального набору для нейронної мережі з відкритих джерел та його попередня підготовка	29.09-13.10.24	виконано
8	Навчання моделі нейронної мережі за допомогою підготовленого тренувального набору	14.10-16.10.24	виконано
9	Представлення отриманих результатів науковому керівнику та узгодження плану подальшого дослідження	17.10-19.10.23	виконано
10	Обробка отриманих результатів	20.10-09.11.24	виконано
11	Аналіз отриманих результатів	10.11-17.11.24	виконано
12	Оформлення звіту	17.11-27.11.24	виконано

Студент _____ Степаненко О.І.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Безверхий А.І.
(підпис) (прізвище та ініціали)

Нормоконтроль пройдено

Нормоконтролер_ _____ Скрипник І.А.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Сторінок – 87

Рисунків – 17

Джерел – 23

Степаненко О.І. Нейромережна система розпізнавання людей в реальному часі у системах безпеки: кваліфікаційна робота магістра спеціальності 121 «Інженерія програмного забезпечення» / наук. керівник А. І. Безверхий. Запоріжжя : ЗНУ, 2024. 87 с.

У сучасних умовах розвитку технологій безпеки, ефективність систем ідентифікації та моніторингу є важливим аспектом забезпечення суспільної та індивідуальної безпеки. Тому розробка та впровадження нейромережевих систем розпізнавання осіб у реальному часі стають одними з перспективних напрямків розвитку сучасних інформаційних технологій.

Мета і завдання дослідження полягають у аналізі сучасних методів і технологій розпізнавання людей, розробці та впровадженні нейромережевої системи для ідентифікації осіб у реальному часі. Проведено аналіз існуючих алгоритмів комп’ютерного зору та глибокого навчання, таких як CNN (Convolutional Neural Networks), та вивчено їх ефективність у різних умовах.

На основі аналізу розроблено прототип нейромережевої системи, яка інтегрується з сучасними камерами відеоспостереження та здатна виконувати точну ідентифікацію осіб за мінімальний час. Така система може застосовуватися в різних галузях, включаючи безпеку об’єктів, контроль доступу та аналіз відвідуваності.

Ключові слова: розпізнавання людей, нейромережі, реальний час, комп’ютерний зір, безпека.

ABSTRACT

Pages – 87

Drawings – 17

Sources – 23

Stepanenko O.I. Neural Network System for Real-time Human Recognition in Security Systems qualification work of the master of specialty 121 “Software Security Engineering” / scientific supervisor A. I. Bezverkhyi. Zaporizhzhya : ZNU, 2024. 87 p.

In modern conditions of development of security technologies, the effectiveness of identification and monitoring systems is an important aspect of ensuring public and individual security. Therefore, the development and implementation of real-time neural network face recognition systems are becoming one of the promising areas of development of modern information technologies.

The aim and objectives of the research are the analysis of modern face recognition methods and technologies, the development and implementation of a neural network system for real-time identification of persons. Existing computer vision and deep learning algorithms, such as CNN (Convolutional Neural Networks), have been analyzed and their effectiveness under various conditions has been studied.

Based on the analysis, a prototype of a neural network system was developed, which integrates with modern video surveillance cameras and is able to perform accurate identification of persons in a minimum time. Such a system can be used in a variety of industries, including facility security, access control, and attendance analysis.

Keywords: face recognition, neural networks, real time, computer vision, security.

ЗМІСТ

ВСТУП 8

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 18

1.1 Огляд проблеми аналізу та збору інформації людей за допомогою комп'ютерного зору 18

1.2 Огляд проблеми правової та нормативної захищеності при використанні технологій комп'ютерного зору 20

1.3 Огляд літературних джерел 22

1.4 Види технологій для створення застосунку з машинним зором 24

1.5 Сфери використання технологій машинного зору та ідентифікації людей 26

1.6 Приклади готових програмних рішень 28

1.7 Аналіз існуючих програмних рішень 29

РОЗДІЛ 2 ДОСЛІДЖЕННЯ ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ 37

2.1 Аналіз фреймворків та бібліотек машинного навчання 37

2.1.1 Огляд фреймворків та бібліотек машинного навчання 38

2.2 Огляд алгоритмів та моделей машинного навчання 48

2.3 Огляд інструменти для роботи з відео 50

2.4 Огляд компонентів обчислювального середовища 55

РОЗДІЛ 3 РОЗРОБКА ЗАСТОСУНКУ ДЛЯ РЕАЛЬНОЧАСНОГО ВИЗНАЧЕННЯ ОСІБ ЗА ДОПОМОГОЮ НЕЙРОМЕРЕЖНОЇ СИСТЕМИ . 59

3.1 Опис предметної області 59

3.2 Архітектура застосунку 61

3.3 Опис основних компонентів программи 63

3.4 Опис технології CUDA для оптимізації виявлення людей 68

РОЗДІЛ 4 РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ ПРОБЛЕМИ 71

4.1 Порівняння продуктивності программи з використанням GPU і CPU ядер 71

4.2 Оцінка точності розпізнавання залежно від якості відео 74

4.3 Оцінка продуктивності в реальному часі 77

4.4 Стійкість до змін кутів огляду 79

4.5 Стійкість до умов освітлення 81

ВИСНОВКИ 84

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 85

ВСТУП

Актуальність теми

Сучасний світ стає дедалі більш цифровим, і технології штучного інтелекту та комп’ютерного зору відіграють у цьому процесі ключову роль. Зокрема, розпізнавання осіб у реальному часі стає важливим інструментом у сферах безпеки, управління доступом, аналітики поведінки споживачів, охорони громадського порядку та багатьох інших. За даними досліджень Statista, у 2023 році світовий ринок технологій розпізнавання перевищив 5 мільярдів доларів США, при цьому очікується, що до 2030 року цей показник зросте на 16,6% щорічно. У багатьох країнах, включаючи Україну, активне впровадження систем відеоспостереження з можливістю реальночасного аналізу стало стандартом у державних установах, банках, аеропортах і навіть роздрібній торгівлі.

Чому це актуально?

Зростання загроз безпеці У сучасних умовах загрози тероризму, кібератак і порушень громадського порядку вимагають швидкого і точного реагування. Технології розпізнавання осіб забезпечують миттєву ідентифікацію потенційно небезпечних осіб у режимі реального часу, що дозволяє знижувати ризики та підвищувати ефективність заходів безпеки. Популярність "розумних" міст Впровадження концепції Smart City вимагає інтеграції сучасних технологій для автоматизації безпеки та управління міською інфраструктурою. У рамках таких проєктів нейромережеві системи розпізнавання осіб використовуються для моніторингу натовпу, управління транспортними потоками та забезпечення захисту громадян.

Широка сфера застосування Системи розпізнавання осіб застосовуються у різних галузях:

1. Управління доступом: Системи використовуються для контролю входу до офісів, шкіл, банківських установ та інших об'єктів.

2. Ритейл і маркетинг: Визначення демографічних даних покупців і аналіз поведінки для поліпшення користувацького досвіду.
3. Фінансовий сектор: Ідентифікація користувачів у банківських операціях, включаючи онлайн-банкінг.
4. Охорона здоров’я: Контроль доступу до медичних установ та персоналізоване обслуговування пацієнтів.

Розвиток технологій завдяки швидкому прогресу в галузі глибокого навчання, зокрема використанню архітектур нейромереж (CNN, RNN, Transformer), розпізнавання людей стало точнішим, швидшим та ефективнішим. Такі методи, як FaceNet, YOLO, MTCNN, вже довели свою ефективність у вирішенні складних завдань у реальному часі.

Виклики та потреба у розвитку

1. Проблеми конфіденційності: Використання систем розпізнавання осіб викликає питання щодо зберігання і обробки персональних даних.
2. Технічні обмеження: Необхідність обробки великого обсягу відеоінформації вимагає високопродуктивного апаратного забезпечення та ефективних алгоритмів.
3. Адаптація до умов: Розпізнавання осіб у реальному часі має працювати в різних умовах освітлення, при часткових перекриттях або зображеннях низької якості.

Значення для України

В Україні, як і в багатьох інших країнах, впровадження нейромережевих систем розпізнавання осіб може сприяти підвищенню безпеки у великих містах, аеропортах, на вокзалах та в інших стратегічних об'єктах. Зокрема, Київ уже використовує подібні системи у рамках програми "Безпечна столиця".

Алгоритми розпізнавання

1. Convolutional Neural Networks (CNN) для аналізу зображень.
2. Архітектура YOLO (You Only Look Once) для швидкої обробки відеопотоку.

3. OpenFace, Dlib та інші бібліотеки для створення відкритих рішень.
4. Інтеграція з обладнанням:
5. Використання камер з підтримкою високої роздільної здатності.
6. Використання GPU для пришвидшення обробки даних.
7. Платформи для розробки:
8. TensorFlow, PyTorch – для навчання моделей.
9. OpenCV – для роботи з відеопотоком.

Перспективи

Системи реальночасного розпізнавання осіб мають величезний потенціал, але також вимагають постійного вдосконалення для вирішення сучасних викликів. Їх активне впровадження дозволить підвищити рівень безпеки, ефективності бізнес-процесів та загальний рівень технологічного розвитку суспільства.

Мета і завдання дослідження

Мета дослідження полягає у розробці, аналізі та впровадженні нейромережної системи для реальночасного розпізнавання осіб, яка забезпечує високу точність, швидкість та ефективність роботи у різноманітних умовах застосування. Дослідження спрямоване на створення технологічного рішення, що відповідає сучасним викликам у сферах безпеки, управління доступом та автоматизації.

Аналіз сучасних методів і технологій

Вивчення існуючих алгоритмів розпізнавання (Convolutional Neural Networks, FaceNet, YOLO, MTCNN). Аналіз переваг і недоліків існуючих нейромережних моделей для реальночасної обробки відеопотоків.

Розробка вимог до системи

Визначення технічних і програмних вимог до нейромережної системи, зокрема швидкості роботи, точності ідентифікації, стійкості до змін умов. Створення специфікацій для інтеграції з апаратними засобами, такими як

камери високої роздільної здатності або GPU. Інтеграція розробленої нейромережі з системами відеоспостереження.

Розробка рекомендацій для впровадження

Визначення оптимальних сфер застосування (безпека, контроль доступу, аналіз поведінки клієнтів). Надання практичних рекомендацій щодо масштабування та інтеграції системи у різні середовища, включаючи міські інфраструктури (Smart City) і комерційні установи.

Оцінка економічної ефективності

Аналіз витрат на впровадження і обслуговування системи. Оцінка економічних переваг, таких як зменшення витрат на ручну ідентифікацію, підвищення ефективності охоронних заходів тощо.

Об’єкт дослідження

Об’єктом дослідження є нейромережні системи розпізнавання осіб у реальному часі, які інтегруються у сучасні інформаційні системи для обробки відеопотоків. Це включає алгоритми комп’ютерного зору та глибокого навчання, що забезпечують ідентифікацію осіб у режимі реального часу. У рамках дослідження увага приділяється апаратним та програмним компонентам системи, зокрема камерам спостереження, відеопотокам різної якості, а також процесорам, які забезпечують обробку даних. Об’єкт дослідження охоплює також інтеграцію цих систем із різними прикладними галузями, такими як контроль доступу, моніторинг громадського порядку, аналітика поведінки споживачів і забезпечення безпеки у міських інфраструктурах.

Предмет дослідження

Предметом дослідження є методи, алгоритми, технології та інструменти, які забезпечують розпізнавання осіб у реальному часі за допомогою нейромережних систем. Особлива увага приділяється використанню глибокого навчання, зокрема архітектур згорткових нейронних мереж (CNN),

для аналізу відеопотоків та ідентифікації осіб у різних умовах. Предмет охоплює також технології обробки даних, такі як оптимізація продуктивності моделей для роботи в режимі реального часу, інтеграція з апаратними засобами (камерами та процесорами), а також програмні інструменти для навчання, тестування і впровадження систем. У дослідженні розглядаються фактори, що впливають на точність і швидкість роботи системи, способи адаптації до умов освітлення, якості зображень та кутів огляду, а також можливості інтеграції таких систем у різні галузі, включаючи безпеку, моніторинг та аналітику.

Методи дослідження

У процесі дослідження буде застосовано комплекс методів для досягнення поставлених цілей:

Аналіз та узагальнення літератури

Проведення обширного аналізу наукових публікацій, досліджень і статей, що стосуються методів розпізнавання, алгоритмів нейромереж і комп’ютерного зору. Вивчення праць, присвячених застосуванню нейромереж для обробки відеопотоків у реальному часі, дозволить сформувати розуміння сучасних технологій і виявити їхні переваги та обмеження.

Вивчення технічної документації та кейсів

Аналіз технічної документації існуючих систем розпізнавання осіб і готових рішень, таких як FaceNet, YOLO та MTCNN. Ознайомлення з реальними кейсами впровадження систем розпізнавання в таких галузях, як безпека, транспорт, ритейл, для оцінки їхньої ефективності в практичних умовах.

Програмне моделювання та прототипування

Розробка прототипу нейромережної системи з використанням інструментів, таких як TensorFlow, PyTorch, OpenCV. Інтеграція прототипу з програмним забезпеченням для обробки реальних відеопотоків.

Аналіз результатів та висновки

Вивчення результатів експериментального тестування: оцінка точності розпізнавання, швидкості обробки даних та адаптивності системи до умов використання. Формулювання висновків щодо ефективності запропонованих підходів, визначення їх переваг та обмежень, а також рекомендацій для подальшого вдосконалення.

Наукова новизна одержаних результатів

В результаті проведеного дослідження систематизовано та проаналізовано сучасні методи розпізнавання осіб за допомогою нейромереж, що працюють у режимі реального часу. На основі цього аналізу розроблено інноваційний підхід до побудови нейромережних систем, який забезпечує підвищену точність і швидкість розпізнавання навіть за складних умов, таких як низька якість зображень, зміни освітлення чи часткове перекриття. Запропоновано вдосконалення існуючих алгоритмів глибокого навчання, включаючи адаптацію архітектури CNN для обробки великих обсягів відеоданих у реальному часі. Наукова новизна полягає у створенні прототипу нейромережної системи, яка інтегрується з сучасними апаратними засобами та демонструє високу ефективність у практичних застосуваннях.

Дослідження також виявило нові підходи до оптимізації нейромереж для зменшення затримок у розпізнаванні осіб, що є критично важливим для систем моніторингу безпеки. Впроваджено механізми автоматичного налаштування моделі під конкретні умови використання, що дозволяє підвищити її гнучкість і зменшити потребу у великих обчислювальних ресурсах. Експериментальне підтвердження ефективності запропонованих рішень продемонструвало їх переваги у порівнянні з існуючими методами, особливо у сфері забезпечення громадської безпеки, контролю доступу та аналітики в комерційному середовищі.

Отримані результати формують основу для подальших досліджень у напрямку вдосконалення систем розпізнавання осіб у реальному часі,

сприяючи розвитку нових технічних рішень та методів, які можуть бути адаптовані до широкого спектра прикладних завдань. Це забезпечує значний внесок у розвиток технологій комп’ютерного зору та їх інтеграцію у повсякденне життя..

Практичне значення одержаних результатів

Результат дослідження, присвяченого реальночасному розпізнаванню осіб за допомогою нейромережних систем, мають широке практичне застосування. Вони відкривають нові можливості для впровадження ефективних і точних систем ідентифікації осіб у різних галузях, зокрема у сфері безпеки, управління доступом, моніторингу громадського порядку та аналітики поведінки клієнтів. Запропоновані алгоритми та технологічні рішення дозволяють створювати інтегровані системи, які можуть бути використані у громадських місцях, аеропортах, транспортних вузлах, освітніх закладах, комерційних об’єктах та багатьох інших сферах, де швидкість і точність ідентифікації мають критичне значення.

Розроблений підхід до оптимізації нейромережних алгоритмів дозволяє не лише покращити точність розпізнавання осіб, а й значно зменшити час обробки відеопотоків, що критично важливо для застосування у реальному часі. Це забезпечує надійне функціонування систем у складних умовах, таких як слабке освітлення, обмеженість обчислювальних ресурсів або великі обсяги даних. Практична цінність результатів також проявляється у зниженні вартості впровадження та обслуговування систем за рахунок зменшення потреби в потужному апаратному забезпеченні та використання оптимізованих моделей.

Результати дослідження можуть бути використані підприємствами для вдосконалення існуючих систем безпеки, створення автоматизованих систем контролю доступу або навіть покращення обслуговування клієнтів у ритейлі через аналіз поведінкових даних. Крім того, вони надають можливість зменшити операційні витрати та підвищити ефективність роботи організацій, які впроваджують ці системи. Таким чином, результати роботи сприяють не

лише технічному прогресу, але й економічній ефективності впровадження сучасних рішень, що робить їх затребуваними у багатьох сферах.

Апробація одержаних результатів

Результати дослідження були представлені на науково-практичній конференції студентів, аспірантів, докторантів і молодих вчених Запорізького національного університету «Молода наука-2024», а також на І Міжнародній науково-практичній конференції «Інженерні інновації та розбудова національної економіки» Інженерного навчально-наукового інституту Запорізького національного університету.

Глосарій

Нейромережа (Neural Network) — математична модель, що імітує роботу біологічних нейронів і використовується для вирішення завдань штучного інтелекту, таких як класифікація, прогнозування або розпізнавання образів.

Реальночасне розпізнавання (Real-Time Recognition) — процес ідентифікації осіб або об’єктів, який виконується миттєво під час отримання даних, без затримок на обробку.

Згорткова нейронна мережа (Convolutional Neural Network, CNN) — тип штучної нейронної мережі, спеціально розроблений для аналізу зображень і відео шляхом обробки вхідних даних у вигляді матриць.

Комп’ютерний зір (Computer Vision) — галузь інформатики, яка займається автоматизованим отриманням, обробкою та аналізом інформації з цифрових зображень чи відео.

Режим реального часу (Real-Time Mode) — режим роботи системи, у якому вона миттєво обробляє та реагує на дані без помітної затримки.

Відеопотік (Video Stream) — послідовність зображень (кадрів), які формують динамічне відео, що може оброблятися системами розпізнавання.

Ідентифікація осіб (Facial Identification) — процес визначення унікальних характеристик для встановлення особистості людини.

Датасет (Dataset) — набір структурованих даних, які використовуються для навчання, тестування та валідації моделей машинного навчання.

Глибоке навчання (Deep Learning) — підгалузь машинного навчання, яка використовує багатошарові нейронні мережі для обробки складних даних і виконання завдань високої складності.

Оптимізація моделі (Model Optimization) — процес вдосконалення нейромережної моделі для підвищення швидкодії та точності при мінімізації використання ресурсів.

GPU (Graphics Processing Unit) — графічний процесор, який використовується для прискорення обчислень при навчанні та роботі нейронних мереж.

TensorFlow — популярна програмна бібліотека для побудови, навчання та розгортання нейромережних моделей.

PyTorch — програмна платформа для машинного навчання та розробки нейронних мереж, яка широко використовується для експериментальних та комерційних цілей.

OpenCV — бібліотека з відкритим кодом для обробки зображень і відео, яка використовується в задачах комп’ютерного зору.

YOLO (You Only Look Once) — алгоритм, що дозволяє здійснювати швидке та ефективне розпізнавання об’єктів у реальному часі.

FaceNet — модель глибокого навчання, призначена для обчислення ембеддингів, які використовуються для розпізнавання або класифікації.

Ембеддинг (Embedding) — математичне представлення об'єктів, таких як обличчя, у вигляді багатовимірного вектора для спрощення їх обробки.

Smart City (Розумне місто) — концепція міської інфраструктури, яка інтегрує цифрові технології для автоматизації та покращення якості життя мешканців.

Контроль доступу (Access Control) — система, що обмежує або дозволяє доступ до фізичних чи цифрових ресурсів на основі ідентифікації користувача.

Латентність (Latency) — затримка, що виникає під час обробки даних, яка є важливим фактором у системах реального часу.

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд проблеми аналізу та збору інформації людей за допомогою комп'ютерного зору

Аналіз та збір інформації про людей за допомогою комп’ютерного зору є складним і багатогранним завданням, яке охоплює різноманітні технічні, етичні та практичні аспекти. Технології комп’ютерного зору дозволяють автоматично ідентифікувати, аналізувати та відслідковувати людей за допомогою відео- чи фотозображень. Проте цей процес супроводжується численними проблемами, які виникають на всіх етапах роботи з даними.

Однією з ключових проблем є доступність і якість даних, які використовуються для навчання та роботи систем комп’ютерного зору. Системи потребують великих обсягів зображень, що охоплюють різноманітні аспекти зовнішності людей: вік, стать, етнічну приналежність, міміку та пози. Набори даних, доступні для досліджень і комерційного використання, часто мають обмежену різноманітність або містять упередження. Наприклад, моделі, натреновані на нерепрезентативних наборах даних, можуть демонструвати нижчу точність для деяких груп населення, що породжує ризик дискримінації.

Ще однією значною проблемою є обробка та аналіз великих обсягів інформації, що надходять у реальному часі. Відеопотоки з камер спостереження або інших джерел генерують гігантські масиви даних, які потребують швидкої і точної обробки. Це вимагає використання високопродуктивного апаратного забезпечення, зокрема графічних процесорів (GPU), і оптимізованих алгоритмів нейромереж, таких як згорткові нейронні мережі (CNN). Водночас складність цих алгоритмів і потреба в ресурсах можуть призводити до затримок і значних витрат, що обмежує застосування таких систем у реальному часі.

Крім технічних обмежень, серйозним викликом залишається точність роботи систем. Комп’ютерний зір може демонструвати нестабільні результати у складних умовах, таких як зміна освітлення, низька якість зображення. Наприклад, у випадку відеоспостереження в громадських місцях системи можуть помилятися у розпізнаванні особи через щільність натовпу або рух об’єктів. Це породжує ризик неправильної ідентифікації, що може мати серйозні наслідки в контексті правопорядку або безпеки.

Етичні та правові аспекти також є важливою складовою проблеми. Використання комп’ютерного зору для аналізу людей часто порушує питання конфіденційності та прав на особисту інформацію. Зокрема, багато систем збирають дані без явної згоди людей, які потрапляють у поле зору камер, що створює ризики порушення законів про захист персональних даних, таких як GDPR у Європі. Використання таких систем у громадських місцях або комерційних цілях також може викликати суспільну критику, особливо якщо їх застосування є непрозорим.

Окремо варто згадати про можливість зловживання технологіями комп’ютерного зору. Уряди або приватні організації можуть використовувати такі системи для масового спостереження, яке обмежує свободи громадян і створює ризик для демократичних процесів. Це особливо актуально у контексті авторитарних режимів, де системи комп’ютерного зору можуть використовуватися для придушення інакомислення.

Окрім етичних та правових проблем, викликом є також захист таких систем від зовнішніх загроз, наприклад атак на моделі машинного навчання. Штучно створені зображення або зміна зовнішнього вигляду людини (наприклад, за допомогою спеціальних масок) можуть обдурити алгоритми ідентифікації, що підриває їхню надійність.

Таким чином, проблеми аналізу та збору інформації про людей за допомогою комп’ютерного зору включають широкий спектр технічних, етичних, правових і соціальних аспектів. Для їх вирішення потрібен комплексний підхід, який поєднує вдосконалення алгоритмів, забезпечення

прозорості використання систем, дотримання прав людини та створення відповідних нормативно-правових механізмів регулювання. Лише за умови вирішення цих проблем технології комп’ютерного зору можуть стати ефективним і безпечним інструментом для аналізу інформації про людей.

1.2 Огляд проблеми правової та нормативної захищеності при використанні технологій комп'ютерного зору

Проблема правової та нормативної захищеності при використанні технологій комп'ютерного зору є актуальним питанням у сучасному світі, оскільки ці технології набули широкого розповсюдження в багатьох сферах, таких як безпека, медицина, транспорт, маркетинг та інші. Використання комп'ютерного зору передбачає збір, обробку та зберігання великих обсягів даних, включаючи персональну інформацію, що може створювати правові ризики та викликати значні етичні суперечки.

Однією з основних проблем є порушення законів про конфіденційність та захист персональних даних. У багатьох країнах, зокрема в Європейському Союзі, діють суворі закони, такі як GDPR, які регламентують використання особистих даних. Комп’ютерний зір часто застосовується для розпізнавання осіб, що передбачає обробку біометричних даних. У разі несанкціонованого збору або обробки таких даних організації можуть зіткнутися з серйозними юридичними наслідками, включаючи великі штрафи та репутаційні втрати. Наприклад, якщо система розпізнавання осіб встановлена в громадських місцях без належного інформування громадян або без їхньої згоди, це може розцінюватися як порушення права на приватність.

Іншою важливою правовою проблемою є питання авторських прав на алгоритми та моделі машинного навчання. Технології комп'ютерного зору, як правило, розробляються на основі існуючих відкритих даних або моделей, які можуть бути захищені авторським правом. Якщо розробники не враховують ліцензійні обмеження, це може призвести до правових конфліктів і судових

позовів. Використання таких моделей без дозволу власників інтелектуальної власності може кваліфікуватися як незаконне привласнення інтелектуальної праці.

Особливої уваги заслуговує проблема відповідальності за помилки систем комп'ютерного зору. Такі технології не завжди працюють бездоганно й можуть допускати помилки у розпізнаванні осіб або об’єктів, що може призводити до серйозних наслідків. Наприклад, помилкове ідентифікування особи в системах громадської безпеки може стати причиною необґрунтованих звинувачень або порушення прав людини. У такому випадку виникає питання: хто несе відповідальність за такі помилки — розробник системи, її оператор чи організація, що використовує технологію?

Правове регулювання також стикається з проблемою зловживання технологіями комп'ютерного зору. Уряди або приватні організації можуть використовувати такі системи для масового спостереження, що може обмежувати свободи громадян і порушувати принципи демократичного суспільства. Наприклад, технології розпізнавання осіб можуть застосовуватися для відстеження опозиційних активістів або для контролю громадян у способах, які виходять за рамки етичного використання.

Недостатнє законодавче регулювання є ще однією проблемою, яка ускладнює правову захищеність. У багатьох країнах все ще не існує чітких нормативних актів, які регулюють використання комп’ютерного зору. Це створює правові прогалини, які дозволяють організаціям застосовувати такі технології без належного контролю, що може призводити до порушення прав громадян.

Крім того, важливо враховувати ризики кібербезпеки, пов’язані з використанням комп’ютерного зору. Збереження великих обсягів даних, які використовуються цими системами, може стати мішенню для хакерів. Витік біометричних даних є особливо небезпечним, оскільки такі дані, на відміну від паролів чи інших цифрових ідентифікаторів, неможливо змінити. У разі

кібератаки це може створити серйозні проблеми для організацій і користувачів.

Етичні аспекти використання комп’ютерного зору також потребують правового регулювання. Важливо забезпечити, щоб ці технології застосовувалися лише в етично обґрунтованих цілях і не порушували прав людини. Наприклад, автоматичний аналіз поведінки людей у маркетингових цілях або використання технологій для відстеження переміщень громадян може викликати суспільне обурення, якщо не буде забезпечено прозорість їх використання.

Таким чином, проблеми правової та нормативної захищеності при використанні технологій комп’ютерного зору охоплюють широкий спектр питань, включаючи конфіденційність даних, відповідальність за помилки, захист інтелектуальної власності, запобігання зловживанням і регулювання кібербезпеки. Вирішення цих проблем вимагає створення комплексних законодавчих і нормативних механізмів, які забезпечать баланс між технологічним прогресом і захистом прав громадян.

1.3 Огляд літературних джерел

Питанню реального часу розпізнавання осіб за допомогою нейромережних систем присвячено значну кількість наукових праць, книг та статей, які ілюструють як сучасні досягнення, так і труднощі, з якими стикаються дослідники й розробники. Ці джерела розкривають методології, алгоритми та технології, що дозволяють вирішувати задачі реального часу ідентифікації осіб, а також обговорюють соціальні, правові та етичні аспекти впровадження таких систем.

Одним із фундаментальних досліджень у цій галузі є робота Іена Гудфеллоу, Йошуа Бенжіо та Аарона Курвіла "Deep Learning" [1]. У книзі детально розглянуто основи глибокого навчання, яке є ключовою технологією для реалізації нейромережних моделей розпізнавання облич. Автори

пояснюють принципи роботи згорткових нейронних мереж (CNN), які широко використовуються для обробки зображень та відео, що робить їх ідеальними для завдань розпізнавання осіб у реальному часі. У книзі також акцентується увага на важливості правильного вибору архітектури нейромережі залежно від поставленої задачі.

Практичні аспекти впровадження алгоритмів комп’ютерного зору у реальних умовах висвітлені в книзі Адріана Роузброка "Deep Learning for Computer Vision with Python" [2]. У ній автор акцентує увагу на інструментах, таких як OpenCV і TensorFlow, які є основою для побудови нейромережних моделей. Книга також пропонує практичні приклади створення систем розпізнавання, що працюють у реальному часі, з акцентом на обробку великих обсягів даних і оптимізацію продуктивності.

Аналіз існуючих наборів даних, які використовуються для навчання нейромережних систем, є важливим аспектом досліджень у цій галузі. Наприклад, у роботі "Розпізнавання людських емоцій з використанням нейромережевих технологій" [3] автори детально описують методи навчання моделей на основі великомасштабних датасетів, таких як LFW і MS-Celeb-1M. Ці набори даних дозволяють створювати потужні системи, здатні розпізнавати осіб у різноманітних умовах. Автори також наголошують на важливості врахування етнічного та демографічного розмаїття в наборах даних для уникнення упередженості алгоритмів.

Особливу увагу питанню етичних і правових аспектів приділяє книга Віктора Майєр-Шенбергера та Кеннета Кука "Нейромережева технологія виявлення і розпізнавання людей у відеопотоці" [4]. У статті розглянуто соціальні наслідки використання технологій аналізу великих даних, включаючи системи розпізнавання. Автори підкреслюють важливість дотримання конфіденційності персональних даних і необхідність створення правових механізмів регулювання таких систем.

Технічні аспекти оптимізації систем комп’ютерного зору в реальному часі детально описані у статті " Розроблення системи розпізнавання людських

облич для відеоспостереження" [5]. У ній автори представляють новітні алгоритми для швидкого та точного розпізнавання об’єктів, які можна адаптувати для ідентифікації. Ця робота підкреслює важливість балансу між швидкістю роботи системи та її точністю, особливо для реального часу.

Нарешті, важливим джерелом для розуміння сучасних тенденцій у цій сфері є публікація "Використання нейронних мереж для розпізнавання об'єктів" [6], у якій узагальнено останні досягнення в технологіях розпізнавання людей, включаючи використання генеративних змагальних мереж (GAN) для покращення точності моделювання. Робота акцентує увагу на перспективних напрямках розвитку, таких як безконтактні методи ідентифікації та інтеграція систем у "розумні міста".

1.4 Види технологій для створення застосунку з машинним зором

Реальночасне розпізнавання осіб за допомогою нейромережних систем базується на комплексному використанні різних технологій, кожна з яких відповідає за виконання окремих завдань. Така система потребує високої продуктивності, точності та ефективності роботи, а її успішна реалізація залежить від правильного вибору інструментів та методів.

Комп'ютерний зір (Computer Vision)

Комп'ютерний зір є фундаментом для систем розпізнавання осіб. Ця технологія забезпечує автоматичне виявлення на зображеннях або відеопотоках. Для її реалізації використовуються спеціальні бібліотеки, такі як OpenCV, яка забезпечує швидкий аналіз зображень, і Dlib, що пропонує готові рішення для виявлення.

Глибоке навчання (Deep Learning)

Глибоке навчання лежить в основі розпізнавання осіб, забезпечуючи аналіз складних структур. Основною технологією є згорткові нейронні мережі (CNN), які витягують унікальні ознаки для створення ембеддингів — математичних векторів. Популярні моделі, такі як FaceNet**,** DeepFace **і** ArcFace,

забезпечують високу точність навіть у складних умовах. Навчання цих моделей здійснюється на платформах TensorFlow або PyTorch, які забезпечують гнучкість у налаштуванні архітектури та параметрів моделі.

Алгоритми реального часу

Для роботи в реальному часі необхідно використовувати алгоритми, що забезпечують високу швидкість обробки. Одним із найпопулярніших є YOLO (You Only Look Once)**,** який дозволяє швидко і точно виявляти людей на відеопотоках. Також часто використовується MTCNN (Multi-task Cascaded Convolutional Networks), яка ефективно знаходить ключові точки для їх точнішого аналізу. Ці алгоритми оптимізовані для обробки відеопотоків у реальному часі з мінімальними затримками.

Обробка відеопотоків

Система реального часу повинна працювати з великими обсягами відеоінформації. Для цього використовуються інструменти, такі як FFmpeg, які забезпечують ефективне декодування та обробку відеопотоків у різних форматах. Це дозволяє адаптувати систему до роботи з відео з різною роздільною здатністю та частотою кадрів.

API для розпізнавання осіб

Для швидкого впровадження функціоналу розпізнавання осіб можна використовувати готові API, такі як Microsoft Azure Face API**,** Google Cloud Vision API або Amazon Rekognition. Вони забезпечують точне розпізнавання осіб без необхідності самостійного навчання моделей, але їх використання може бути обмежене вартістю чи залежністю від сторонніх сервісів.

Інтеграція з іншими системами

Системи розпізнавання осіб часто інтегруються в "розумні" міста або організації. Інструменти IoT (Internet of Things) дозволяють об’єднувати системи відеоспостереження, контролю доступу та аналізу поведінки у єдину інфраструктуру, що забезпечує централізоване управління.

1.5 Сфери використання технологій машинного зору та ідентифікації людей

Системи реальночасного розпізнавання осіб за допомогою нейромережних технологій мають широкий спектр застосування у різних галузях, де важливі точність, швидкість і автоматизація процесів ідентифікації. Ці системи активно використовуються у сферах безпеки, управління доступом, маркетингу, охорони здоров'я, освіти, а також у розумних містах. Завдяки своїй універсальності вони знаходять застосування як у державному секторі, так і у приватному бізнесі [7].

Однією з найбільш очевидних сфер використання є забезпечення безпеки. Системи розпізнавання осіб інтегруються з камерами спостереження для моніторингу громадських місць, таких як аеропорти, вокзали, стадіони або торговельні центри. Вони дозволяють виявляти потенційно небезпечних осіб у режимі реального часу, ідентифікувати людей, які перебувають у розшуку, або попереджати порушення громадського порядку. У правоохоронних органах такі системи також застосовуються для автоматичного порівняння облич із базами даних, що значно скорочує час на проведення розслідувань.

У сфері управління доступом системи розпізнавання осіб дозволяють замінити традиційні способи аутентифікації, такі як картки чи паролі. Вони забезпечують безконтактний контроль доступу до офісів, банківських сховищ, виробничих зон або житлових приміщень. Завдяки цьому підвищується рівень безпеки, адже унеможливлюється передача або втручання сторонніх осіб. У фінансовій сфері ці технології використовуються для безпечного входу до мобільних додатків або банкоматів.

У маркетингу системи розпізнавання осіб використовуються для аналізу демографічних характеристик клієнтів, таких як вік, стать, емоційний стан. Це дозволяє компаніям адаптувати свої маркетингові стратегії, створювати персоналізовані пропозиції та оптимізувати розташування товарів у магазинах. Наприклад, роздрібні мережі можуть аналізувати поведінку

покупців у торгових залах, визначати популярність товарів або зон у магазині, що допомагає покращити клієнтський досвід і збільшити продажі.

В охороні здоров'я системи реальночасного розпізнавання осіб можуть бути використані для ідентифікації пацієнтів у медичних закладах, особливо в умовах надзвичайних ситуацій, коли важливо швидко визначити особу для надання невідкладної допомоги. Також ці технології застосовуються для моніторингу персоналу та забезпечення дотримання санітарних норм у лікарнях. У період пандемії COVID-19 технології розпізнавання облич стали важливими для відстеження носіння масок та дотримання дистанції у громадських місцях.

Освітні заклади також знаходять застосування для систем розпізнавання осіб. Вони дозволяють автоматично фіксувати відвідуваність студентів або забезпечувати контроль доступу до аудиторій чи бібліотек. Крім того, вони використовуються для запобігання шахрайству під час складання іспитів, автоматично ідентифікуючи осіб, які беруть участь у тестуванні.

У "розумних" містах технології реальночасного розпізнавання осіб інтегруються у системи моніторингу трафіку, аналізу натовпів і управління міською інфраструктурою [8]. Вони дозволяють оптимізувати транспортні потоки, контролювати поведінку людей у громадських місцях, забезпечувати безпеку на великих масових заходах. Це важливий інструмент для створення безпечного та комфортного середовища у мегаполісах.

Сфера розваг також використовує ці технології, наприклад, для автоматичного визначення VIP-клієнтів у казино чи ідентифікації учасників на заходах. У кінотеатрах чи парках розваг вони можуть аналізувати емоції відвідувачів, що дозволяє оптимізувати послуги та вдосконалювати користувацький досвід.

Технології розпізнавання осіб знайшли своє застосування у транспортній сфері для автоматичної ідентифікації пасажирів на вокзалах та аеропортах. Це значно прискорює процес реєстрації, дозволяє безконтактно перевіряти квитки та забезпечує додатковий рівень безпеки.

1.6 Приклади готових програмних рішень

На сучасному ринку існує чимало програмних рішень і сервісів, які реалізують функції розпізнавання людей у реальному часі. Вони використовуються для різноманітних завдань, від забезпечення безпеки до підвищення ефективності бізнес-процесів.

Face++

Face++ є одним із провідних рішень у галузі розпізнавання облич. Це API, що надає широкий спектр функцій, включаючи розпізнавання осіб, визначення емоцій, вік та стать. Face++ інтегрується у програми для безпеки, системи контролю доступу та маркетингові платформи. Завдяки високій точності та швидкості, це рішення активно використовується компаніями в різних галузях.

Microsoft Azure Face API

Microsoft Azure пропонує API для розпізнавання облич, який використовується у великих проєктах для забезпечення автоматизації процесів. Цей інструмент дозволяє ідентифікувати осіб, порівнювати обличчя, аналізувати демографічні дані та емоції. Azure Face API популярний у розробників завдяки простоті інтеграції та широким можливостям налаштування.

Amazon Rekognition

Amazon Rekognition є потужним сервісом для розпізнавання облич та аналізу зображень. Він пропонує функції ідентифікації осіб, виявлення об’єктів і сцен, а також аналізу відео в реальному часі [9]. Цей сервіс широко використовується в e-commerce, безпеці та розробці "розумних" систем.

Google Cloud Vision API

Google Cloud Vision API дозволяє розпізнавати обличчя, аналізувати зображення та ідентифікувати текст. Його часто використовують у мобільних додатках для забезпечення аутентифікації, в маркетингових кампаніях для збору даних про користувачів та у системах моніторингу.

Clearview AI

Це спеціалізоване програмне забезпечення, яке використовується правоохоронними органами для ідентифікації осіб. Clearview AI працює з великими базами даних фотографій, включаючи зображення з соціальних мереж. Попри свою суперечливість у питаннях конфіденційності, цей інструмент демонструє високу ефективність у розслідуванні злочинів.

Kairos

Kairos пропонує платформи для розпізнавання облич та аналізу емоцій. Їхні технології часто використовуються в роздрібній торгівлі, де потрібен аналіз поведінки клієнтів, а також у HR-сферах для виявлення емоцій під час співбесід.

Cognitec FaceVACS

Cognitec є одним із найстаріших постачальників технологій розпізнавання облич. Їхній продукт FaceVACS використовується у системах безпеки для контролю доступу, моніторингу громадських місць і управління базами даних.

Trueface

Trueface спеціалізується на розробці алгоритмів для виявлення осіб і масок, а також ідентифікації облич у реальному часі. Це рішення застосовується у банках, на транспортних вузлах та в системах "розумних" міст для забезпечення безпеки.

1.7 Аналіз існуючих програмних рішень

Face++

Face++ — це хмарний сервіс для розпізнавання облич, який забезпечує широкий функціонал: ідентифікацію осіб, аналіз емоцій, визначення віку та статі. Сервіс широко використовується у мобільних додатках, системах контролю доступу та маркетингових платформах. Face++ був розроблений китайською компанією Megvii у 2012 році. Це один із провідних сервісів у

сфері розпізнавання облич, який використовує алгоритми глибокого навчання для аналізу зображень і відео. Платформа набула широкої популярності завдяки інтеграції в мобільні додатки, такі як смартфони Xiaomi та Alipay.

Переваги

Face++ підтримує широкий набір функцій, таких як розпізнавання облич, аналіз емоцій, визначення віку, статі, а також виявлення ключових точок обличчя. Він може інтегруватися у мобільні додатки та системи безпеки завдяки API, що підтримує кілька мов програмування, включаючи Python, Java та C++. Face++ відзначається високою точністю розпізнавання, особливо у великих базах даних.

Недоліки

Основним недоліком є залежність від серверів у Китаї, що може викликати затримки в роботі для користувачів з інших регіонів. Також сервіс зазнає критики через можливе порушення конфіденційності, адже компанія працює у країні, де діють суворі закони щодо доступу до персональних даних.



Рисунок 1 – Зображення Face++

Microsoft Azure Face API

Microsoft Azure Face API — це хмарний сервіс, який пропонує рішення для розпізнавання облич, аналізу емоцій і демографічних характеристик. Платформа орієнтована на корпоративних клієнтів і має високу інтеграцію з екосистемою Microsoft. Microsoft представила Azure Face API у рамках своєї платформи Azure AI в 2015 році. API було розроблено як частину хмарних сервісів для обробки великих даних і штучного інтелекту, що дозволяє легко інтегрувати функції розпізнавання облич у програми.

Переваги

Azure Face API пропонує широкий спектр функцій, таких як ідентифікація осіб, порівняння облич, визначення віку та аналіз емоцій. Платформа інтегрується з іншими продуктами Microsoft, такими як Power BI та Dynamics 365, що спрощує використання для корпоративних клієнтів. Вона також відповідає стандартам конфіденційності, включаючи GDPR.

Недоліки

Серед недоліків можна відзначити високу вартість використання сервісу, особливо для великих проєктів. Крім того, Azure Face API вимагає стабільного підключення до інтернету, адже всі обчислення виконуються у хмарі.



Рисунок *2* – Зображення Microsoft Azure Face API

Amazon Rekognition

Amazon Rekognition — сервіс розпізнавання облич та аналізу зображень, який інтегрується з іншими продуктами AWS. Його функціонал включає виявлення осіб, аналіз емоцій і об'єктів у відео. Amazon запустила сервіс Rekognition у 2016 році як частину своєї платформи AWS (Amazon Web Services). Спочатку цей сервіс був орієнтований на обробку зображень і відео для клієнтів у сфері безпеки, але згодом його функціонал розширився.

Переваги

Rekognition забезпечує точне розпізнавання облич, аналіз емоцій та ідентифікацію об’єктів у відеопотоках. Завдяки глибокій інтеграції з іншими

сервісами AWS, він може масштабуватися для великих проєктів. Сервіс також пропонує підтримку відеоаналізу в реальному часі, що робить його популярним у системах безпеки.

Недоліки

Rekognition часто зазнає критики через потенційні проблеми з конфіденційністю. Також користувачі відзначають, що ціна може швидко зрости для великих обсягів даних. Крім того, використання сервісу вимагає детального знання екосистеми AWS.

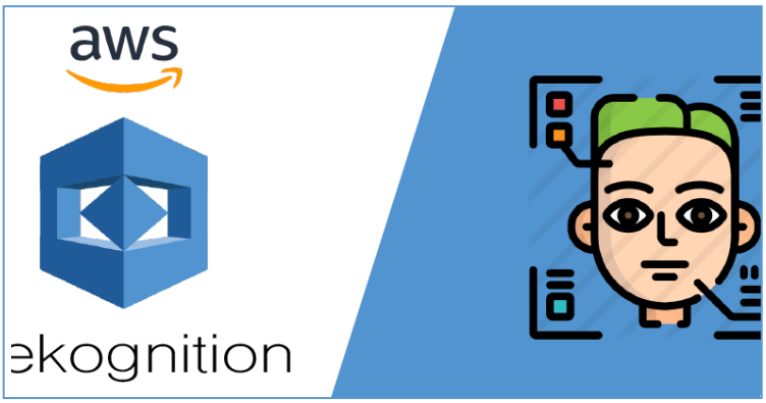


Рисунок *3* – Зображення Amazon Rekognition

Google Cloud Vision API

Google Cloud Vision API — це сервіс для аналізу зображень, який дозволяє розпізнавати обличчя, текст, об’єкти та сцени. Він ідеально підходить для інтеграції у мобільні додатки, бізнес-аналітику та "розумні" системи. Google представила Cloud Vision API у 2016 році як частину Google Cloud Platform. Цей сервіс було створено на основі багаторічного досвіду компанії в галузі комп’ютерного зору та штучного інтелекту.

Переваги

Cloud Vision API забезпечує високоточне розпізнавання облич, тексту та об’єктів. Завдяки машинному навчанню Google модель постійно вдосконалюється. Сервіс інтегрується з іншими продуктами Google, такими як

BigQuery і Firebase, що спрощує використання в мобільних додатках та бізнес-аналітиці.

Недоліки

Подібно до інших хмарних сервісів, Cloud Vision API вимагає стабільного підключення до інтернету. Ще одним недоліком є обмеження за кількістю безкоштовних запитів, після яких вартість використання може значно зрости.

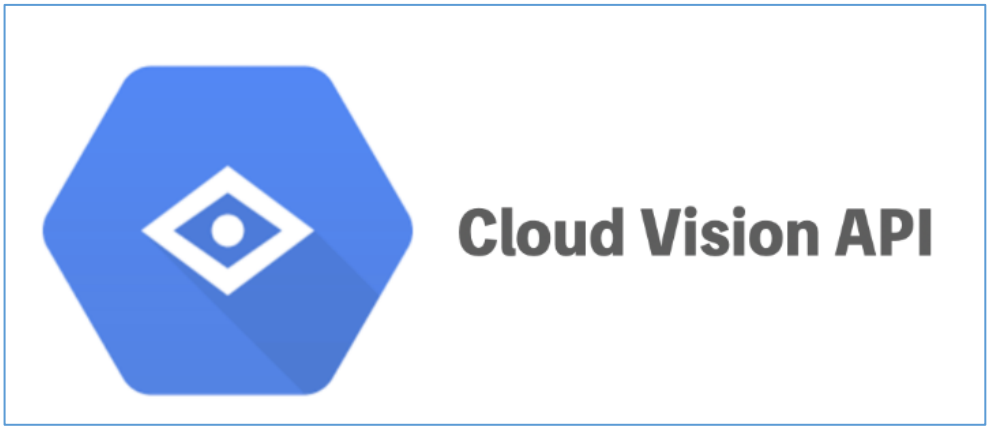


Рисунок 4 – Зображення Google Cloud Vision API

Clearview AI

Clearview AI — це спеціалізований сервіс для розпізнавання облич, орієнтований переважно на правоохоронні органи. Його база даних містить мільярди зображень, зібраних із відкритих джерел, включаючи соціальні мережі. Clearview AI була створена у 2017 році Хоаном Тон-Татом. Компанія спеціалізується на розробці програмного забезпечення для розпізнавання осіб і працює переважно з правоохоронними органами.

Переваги

Clearview AI використовує величезну базу даних із мільярдів зображень, отриманих із соціальних мереж та відкритих джерел. Сервіс демонструє високу ефективність у складних умовах, таких як низька якість зображень чи часткове перекриття облич.

Недоліки

Clearview AI є одним із найбільш суперечливих сервісів через порушення конфіденційності. Багато країн розглядають законодавчі обмеження для цього сервісу через неконтрольоване використання даних. Компанія також працює лише з обмеженим колом клієнтів.



Рисунок 5 – Зображення Clearview AI

Kairos

Kairos — це платформа для розпізнавання облич та аналізу емоцій, орієнтована на маркетингові дослідження, HR та персоналізовані сервіси. Її API легко інтегрується у програми та сервіси. Kairos була заснована у 2012 році як стартап для аналізу емоцій та демографічних характеристик. З часом компанія розширила свій функціонал, включивши можливості розпізнавання облич.

Переваги

Kairos пропонує простий у використанні API, який дозволяє аналізувати емоції, вік, стать та ідентифікувати осіб. Сервіс популярний у маркетингових та HR-додатках завдяки можливості персоналізації.

Недоліки

Основним недоліком є обмежена масштабованість у порівнянні з великими гравцями, такими як Amazon чи Google. Також точність

розпізнавання Kairos може бути нижчою у складних умовах, таких як погане освітлення.



Рисунок 6 – Зображення Kairos

Cognitec FaceVACS

Cognitec FaceVACS — це програмне рішення для розпізнавання облич, яке використовується у системах контролю доступу, моніторингу та безпеки. Продукт популярний серед державних установ і великих корпорацій. Cognitec Systems була заснована в 2002 році у Німеччині, спеціалізуючись на розробці систем розпізнавання облич для урядових та корпоративних клієнтів.

Переваги

FaceVACS пропонує рішення для забезпечення контролю доступу, моніторингу та управління базами даних. Завдяки багаторічному досвіду компанії цей продукт демонструє високу точність навіть у складних умовах.

Недоліки

Основним недоліком є висока вартість рішень Cognitec, що робить їх доступними переважно для великих організацій. Також сервіс вимагає значних ресурсів для впровадження та інтеграції.



Рисунок 7 – Зображення Cognitec FaceVACS

Trueface

Trueface — це платформа для розпізнавання облич та аналізу відеопотоків, орієнтована на забезпечення безпеки в банках, транспорті та "розумних" містах. Її особливістю є адаптація до складних умов, таких як носіння масок. Trueface була заснована у 2013 році як компанія, що спеціалізується на створенні рішень для забезпечення безпеки та аутентифікації.

Переваги

Trueface пропонує рішення для розпізнавання облич навіть у складних умовах, таких як носіння масок. Цей сервіс активно використовується в банківській сфері та на транспортних вузлах.

Недоліки

Недоліком є обмежена кількість функцій у порівнянні з великими платформами. Крім того, точність розпізнавання може знижуватись на низькоякісних зображеннях.



Рисунок 8 – Зображення Trueface

РОЗДІЛ 2 ДОСЛІДЖЕННЯ ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ

2.1 Аналіз фреймворків та бібліотек машинного навчання

Реальночасне розпізнавання осіб за допомогою нейромережних систем є однією з найперспективніших та найінноваційніших технологій сучасності, яка здобуває дедалі більшу популярність у різних сферах життя. Основою успішної реалізації таких систем є використання сучасних фреймворків та бібліотек машинного навчання, які забезпечують ефективне створення, навчання та оптимізацію нейронних мереж. Ці технології є ключовим компонентом для розробки інтелектуальних рішень, які здатні аналізувати та інтерпретувати величезні обсяги даних у реальному часі [10].

Фреймворки та бібліотеки машинного навчання надають розробникам можливість створювати складні моделі глибинного навчання, які можуть автоматично навчатися на основі вхідних даних і вдосконалювати свої можливості у розпізнаванні осіб. Використання таких інструментів, як TensorFlow, PyTorch, та OpenCV, дозволяє значно підвищити точність і швидкість розпізнавання, що є критично важливим для систем, які працюють у режимі реального часу. Завдяки цим фреймворкам, розробники можуть створювати та тестувати моделі, які адаптуються до різних умов, таких як змінне освітлення, різні кути зйомки та часткове закриття обличчя.

Однією з ключових переваг використання фреймворків машинного навчання є їх здатність забезпечити масштабованість та гнучкість системи. Це особливо важливо для додатків, де необхідно забезпечити високу продуктивність та можливість інтеграції з іншими системами [11]. Наприклад, OpenCV дозволяє виконувати попередню обробку зображень і відео, що допомагає нейронним мережам легше розпізнавати особливості облич. PyTorch та TensorFlow забезпечують можливість налаштування моделей під специфічні завдання, що робить їх ідеальними інструментами для створення високоточного розпізнавання осіб.

Завдяки розвитку фреймворків машинного навчання, системи розпізнавання осіб стали доступними не лише для великих компаній, але й для малого та середнього бізнесу, а також для індивідуальних розробників. Це створює умови для інновацій, оскільки багато розробників можуть експериментувати з технологіями та створювати нові, цікаві застосунки. Бібліотеки машинного навчання також надають готові моделі та інструменти, які значно скорочують час розробки та дозволяють швидше досягти бажаних результатів.

Фреймворки та бібліотеки машинного навчання стали невід'ємною частиною розробки систем реальночасного розпізнавання осіб, забезпечуючи високу продуктивність, масштабованість та точність. Вони дозволяють ефективно обробляти великі обсяги даних та виявляти приховані закономірності, що робить такі системи надзвичайно ефективними в умовах сучасного світу, де інформація й швидкість її обробки мають критичне значення [12]. Таким чином, використання фреймворків машинного навчання є ключовим фактором успіху у розробці та впровадженні систем розпізнавання осіб, які здатні значно покращити безпеку, автоматизацію та персоналізацію у різних сферах життя.

2.1.1 Огляд фреймворків та бібліотек машинного навчання

TensorFlow

TensorFlow – це один із найпопулярніших і найбільш використовуваних фреймворків машинного навчання, розроблений компанією Google. Він забезпечує зручне та ефективне створення і навчання моделей глибинного навчання для різноманітних задач, включаючи розпізнавання зображень, обробку мови та аналіз даних. TensorFlow підтримує широкий спектр функцій для реалізації як простих моделей, так і дуже складних нейронних мереж, а також забезпечує можливість розгортання моделей на різних платформах, включаючи мобільні пристрої, сервери та хмарні сервіси. Його відкритий код

та активна спільнота роблять TensorFlow потужним інструментом для розробників по всьому світу.

Розробка TensorFlow почалася у дослідницькій групі Google Brain як продовження попереднього проєкту, відомого під назвою DistBelief. DistBelief був внутрішньою системою глибинного навчання, яку використовували для різноманітних задач, таких як розпізнавання зображень та машинний переклад. Проте DistBelief мав обмежену гнучкість і можливості масштабування, що призвело до необхідності створення нової системи, яка б відповідала вимогам сучасного машинного навчання.

Таким чином, у 2015 році компанія Google оголосила про запуск TensorFlow як відкритого програмного забезпечення [13]. Це дозволило розробникам по всьому світу використовувати інструмент, що дозволяє створювати, навчати і розгортати нейронні мережі. З тих пір TensorFlow значно розвинувся, підтримуючи нові платформи, такі як мобільні пристрої через TensorFlow Lite, а також дозволяючи інтегрувати моделі в браузери за допомогою TensorFlow.js.

Переваги TensorFlow

1. Відкритий код та активна спільнота: TensorFlow має відкритий код, що дає можливість кожному долучитися до його розробки, а також використовувати вже наявні моделі та інструменти. Це сприяє розвитку спільноти, яка активно ділиться своїм досвідом і створює нові рішення, доступні для всіх.

2. Гнучкість: TensorFlow дозволяє працювати як з високорівневими API, такими як Keras, так і з низькорівневими обчислювальними графами, що робить його придатним як для початківців, так і для досвідчених дослідників та розробників.

3. Масштабованість: TensorFlow підтримує розподілене навчання, що дозволяє тренувати великі моделі на кластерах серверів або використовувати багато GPU, що значно підвищує продуктивність і швидкість навчання.

4. Підтримка різних платформ: TensorFlow забезпечує можливість розгортання моделей на різних платформах, включаючи мобільні пристрої (через TensorFlow Lite), веб-браузери (через TensorFlow.js) та хмарні сервіси. Це забезпечує високу гнучкість і можливість використання моделі у різних середовищах.

5. Широка екосистема: TensorFlow має велику кількість додаткових бібліотек та інструментів, таких як TensorBoard для візуалізації навчання, TensorFlow Extended (TFX) для виробничих робочих потоків машинного навчання, що значно спрощує та автоматизує процеси розробки та впровадження моделей [14].

Недоліки TensorFlow

1. Крута крива навчання: Однією з основних проблем TensorFlow є його відносна складність для початківців. Для ефективного використання низькорівневих можливостей фреймворку потрібні глибокі знання в галузі машинного навчання та програмування, що може бути викликом для новачків.

2. Часті зміни API: TensorFlow постійно розвивається, і часті оновлення можуть призводити до змін в API, що робить старі коди несумісними з новими версіями. Це створює певні труднощі при підтримці довготривалих проєктів.

3. Великі ресурси для навчання: TensorFlow орієнтований на обробку великих обсягів даних та використання потужних обчислювальних ресурсів. Це означає, що для навчання складних моделей може знадобитися багато обчислювальних потужностей, що не завжди доступно для індивідуальних розробників або малих компаній.

4. Складність налаштування: Хоча TensorFlow надає багато можливостей, його використання може бути доволі складним через необхідність налаштування різних параметрів, вибору правильних архітектур моделей та інструментів для оптимізації. Це може бути особливо складним для користувачів без глибокого досвіду у цій галузі.

PyTorch

PyTorch – це ще один популярний фреймворк для машинного навчання, розроблений компанією Facebook (зараз Meta). PyTorch має гнучкий та інтуїтивно зрозумілий інтерфейс, що робить його одним із улюблених інструментів серед дослідників та інженерів, які займаються глибинним навчанням. PyTorch дозволяє створювати, навчати і тестувати нейронні мережі за допомогою динамічних обчислювальних графів, що значно спрощує процес налагодження і робить роботу з моделями більш гнучкою [15]. Його популярність зросла завдяки підтримці інновацій та розвитку глибинного навчання, і зараз PyTorch використовується в багатьох провідних академічних і промислових проєктах.

PyTorch був розроблений як продовження і вдосконалення фреймворку Torch, який був спочатку створений у 2002 році дослідниками з кількох провідних університетів. Torch був потужним інструментом, проте мав обмеження у використанні, особливо для тих, хто віддає перевагу мові програмування Python. Для вирішення цих проблем компанія Facebook у 2016 році запустила PyTorch як відкритий проєкт, надаючи глибоким дослідникам і розробникам потужний інструмент для роботи з глибинним навчанням на Python. PyTorch швидко здобув популярність завдяки своїй простоті використання, підтримці динамічних обчислювальних графів і широкому спектру можливостей для налаштування моделей. З 2018 року PyTorch став основним фреймворком для досліджень штучного інтелекту у Facebook, і зараз він використовується у багатьох відомих наукових дослідженнях і комерційних проектах. Завдяки своїй відкритій природі, PyTorch підтримується величезною спільнотою, яка активно долучається до його розвитку.

Переваги PyTorch

1. Динамічні обчислювальні графи: PyTorch використовує динамічні обчислювальні графи, що дозволяє легко змінювати структуру моделі під час її навчання. Це значно спрощує налагодження та експериментування з моделями, що є важливим для дослідницьких проєктів.

2. Зручність використання та інтуїтивність: PyTorch має простий і зрозумілий інтерфейс, що робить його ідеальним для новачків та дослідників. Він має схожість із синтаксисом Python, що робить його більш зручним у використанні, особливо для тих, хто вже має досвід програмування на Python.

3. Потужна підтримка спільноти та бібліотек: PyTorch має активну спільноту, яка створює багато корисних інструментів та бібліотек, таких як TorchVision для обробки зображень та Hugging Face Transformers для роботи з моделями обробки природної мови. Це значно розширює можливості PyTorch та спрощує створення складних моделей.

4. Інтеграція з іншими інструментами: PyTorch легко інтегрується з іншими інструментами, такими як NumPy, а також підтримує сучасні бібліотеки для розподіленого навчання, що робить його гнучким для використання в різних середовищах. Також PyTorch підтримує інтеграцію з ONNX (Open Neural Network Exchange), що дозволяє експортувати моделі для використання в інших фреймворках.

5. Застосування у дослідженнях: PyTorch дуже популярний серед науковців та дослідників завдяки своїй гнучкості та простоті у використанні. Багато академічних робіт та досліджень у галузі глибинного навчання використовують саме PyTorch, що свідчить про його ефективність у дослідницькому середовищі.

Недоліки PyTorch

1. Менша масштабованість у порівнянні з TensorFlow: Хоча PyTorch підтримує розподілене навчання, він все ще вважається менш масштабованим, ніж TensorFlow, особливо коли йдеться про навчання дуже великих моделей на кластерах серверів. TensorFlow має більш розвинуті засоби для масштабування та управління виробничими робочими потоками.

2. Відносна новизна у виробничих системах: Хоча PyTorch стає все більш популярним у виробничих середовищах, він все ще вважається більш підходящим для досліджень і прототипування. TensorFlow, з його TFX та

іншими інструментами для виробництва, має більше можливостей для промислових рішень.

3. Обмежена підтримка мобільних платформ: У порівнянні з TensorFlow Lite, PyTorch має обмежену підтримку мобільних платформ, хоча команда розробників активно працює над цим аспектом через PyTorch Mobile. Тим не менш, TensorFlow має більш розвинені можливості для розгортання моделей на мобільних пристроях.

4. Складність у використанні для деяких специфічних задач: Хоча PyTorch чудово підходить для глибинного навчання, деякі специфічні завдання можуть вимагати використання TensorFlow через більш широкий спектр інструментів та бібліотек, таких як TensorFlow Extended (TFX), які спрощують розгортання та підтримку моделей у виробничих середовищах.

OpenCV

OpenCV (Open Source Computer Vision Library) – це одна з найвідоміших бібліотек для обробки зображень і комп'ютерного зору, яка широко використовується як дослідниками, так і розробниками програмного забезпечення. OpenCV надає багатий набір інструментів для аналізу зображень, детекції об'єктів, обробки відеопотоків та інших завдань, пов'язаних з комп'ютерним зором. Її відкритий код і багатофункціональність роблять її ідеальним інструментом для швидкої розробки проєктів у галузі комп'ютерного зору та машинного навчання. OpenCV підтримує різні мови програмування, включаючи Python, C++, Java та інші, що робить її доступною для широкого кола розробників.

OpenCV була розроблена у 1999 році дослідниками з корпорації Intel як проєкт з відкритим кодом для підтримки додатків з комп'ютерного зору. Метою проєкту було надання розробникам універсальної бібліотеки, яка б могла використовуватися як для наукових досліджень, так і для комерційних застосувань. OpenCV отримала великий успіх завдяки своєму широкому набору функціональних можливостей і відкритому коду, що дозволяло будь-кому використовувати її для своїх потреб. З часом OpenCV стала стандартом

де-факто для задач комп'ютерного зору і почала підтримувати GPU обчислення для підвищення продуктивності. У 2012 році OpenCV була перенесена на платформу GitHub, що ще більше сприяло її розвитку, оскільки розробники з усього світу змогли вносити свій внесок у бібліотеку. Завдяки активній підтримці спільноти і постійному розвитку OpenCV стала невід'ємною частиною багатьох сучасних проєктів у галузі комп'ютерного зору.

Переваги OpenCV

1. Відкритий код та підтримка різних мов програмування: OpenCV має відкритий код, що дозволяє будь-кому використовувати, змінювати і вдосконалювати її. Бібліотека підтримує різні мови програмування, включаючи Python, C++, Java, що робить її зручною для використання у різних середовищах та для різних задач.

2. Широкий набір функцій: OpenCV надає величезний спектр функцій для обробки зображень та відео, включаючи фільтрацію, детекцію контурів, сегментацію, а також інструменти для розпізнавання об'єктів та трекінгу. Це дозволяє використовувати бібліотеку для вирішення найрізноманітніших задач у галузі комп'ютерного зору.

3. Продуктивність: OpenCV оптимізована для використання на CPU і GPU, що забезпечує високу швидкість обробки зображень і відео. Вона також підтримує бібліотеки для паралельних обчислень, такі як CUDA, що дозволяє значно підвищити продуктивність при роботі з великими обсягами даних.

4. Підтримка різних платформ: OpenCV працює на різних операційних системах, включаючи Windows, Linux, macOS, Android та iOS, що дозволяє використовувати її для розробки застосунків на різних платформах. Це забезпечує гнучкість у виборі середовища розробки і розгортання.

5. Активна спільнота та документація: OpenCV має велику і активну спільноту, яка підтримує бібліотеку, додає нові функції та виправляє помилки. Існує багато прикладів, підручників та документації, що дозволяє швидко освоїти основи використання бібліотеки навіть для новачків.

Недоліки OpenCV

1. Обмеження у роботі з глибинним навчанням: Хоча OpenCV має модулі для інтеграції з фреймворками глибинного навчання, такими як TensorFlow або PyTorch, вона не призначена для створення або навчання складних нейронних мереж. Для більшості задач глибинного навчання доведеться використовувати інші інструменти, а OpenCV більше підходить для попередньої обробки зображень та інтеграції результатів.

2. Не завжди оптимальна для складних задач: OpenCV є чудовим інструментом для базової обробки зображень і відео, проте для більш складних задач, таких як розпізнавання зображень з високою точністю або обробка великих наборів даних, може знадобитися поєднання з іншими фреймворками та бібліотеками для досягнення кращих результатів.

3. Потреба в оптимізації: Хоча OpenCV підтримує GPU і паралельні обчислення, деякі операції все ще можуть вимагати додаткової оптимізації для досягнення високої продуктивності. Це може бути складним завданням для розробників без досвіду оптимізації програмного забезпечення.

4. Складність у налаштуванні середовища: Налаштування OpenCV для різних середовищ може бути складним і вимагати багато часу, особливо коли необхідно зібрати бібліотеку з підтримкою специфічних функцій або інтеграції з іншими інструментами. Це може бути викликом для новачків та тих, хто не має досвіду роботи з CMake та іншими інструментами для збірки.

NumPy

NumPy (Numerical Python) – це одна з найпопулярніших бібліотек для наукових обчислень у Python. Вона є фундаментальною бібліотекою, яка використовується для роботи з багатовимірними масивами та матрицями, а також надає велику кількість функцій для виконання математичних і логічних операцій над цими масивами. NumPy відіграє ключову роль у багатьох задачах машинного навчання, аналізу даних та наукових досліджень, оскільки вона забезпечує ефективну обробку великих обсягів даних і підтримку різноманітних математичних функцій. NumPy також є основою для багатьох

інших бібліотек, таких як Pandas, SciPy, та PyTorch, які використовують її можливості для своїх внутрішніх обчислень. NumPy є продовженням проєкту Numeric, який був створений у 1995 році Джимом Хьюгніном для підтримки числових обчислень на мові Python. Numeric мав значний вплив на розвиток наукових обчислень, проте з часом потреби спільноти перевищили можливості цієї бібліотеки. У 2005 році Тревіс Оліфант розпочав роботу над NumPy як модернізованою версією Numeric, яка б поєднувала в собі функціонал Numeric та інших бібліотек для числових обчислень. Тревіс також додав підтримку багатовимірних масивів і забезпечив більш високу продуктивність та функціональність. NumPy швидко здобула популярність завдяки своїй ефективності та гнучкості. Вона стала стандартом де-факто для роботи з числовими даними в Python і активно використовується у різних галузях, включаючи машинне навчання, аналіз даних, обробку зображень та багато інших. Завдяки відкритому коду та активній підтримці спільноти, NumPy постійно вдосконалюється, додаючи нові функції та підвищуючи продуктивність.

Переваги NumPy

1. Висока продуктивність обчислень: NumPy забезпечує ефективну обробку великих обсягів даних завдяки використанню оптимізованих алгоритмів для роботи з масивами. Вона використовує C та Fortran для реалізації багатьох своїх функцій, що дозволяє досягати високої швидкості виконання обчислень порівняно зі стандартними списками Python.

2. Багатовимірні масиви: NumPy надає структуру даних для роботи з багатовимірними масивами, які є основою для багатьох задач у галузі машинного навчання та наукових обчислень. Масиви NumPy дозволяють легко виконувати різні операції, такі як додавання, множення, транспонування та інші математичні операції.

3. Широкий набір математичних функцій: NumPy має широкий набір вбудованих математичних функцій, включаючи тригонометричні, статистичні, логічні операції та багато інших. Це робить її зручною для

виконання різних типів числових обчислень без необхідності у використанні додаткових бібліотек.

4. Інтеграція з іншими бібліотеками: NumPy є основою для багатьох інших бібліотек, таких як Pandas, SciPy, Matplotlib, Scikit-learn, та PyTorch. Це забезпечує безшовну інтеграцію з цими інструментами та дозволяє ефективно використовувати можливості кожної з них для виконання складних задач.

5. Зручний інтерфейс: NumPy має простий та інтуїтивно зрозумілий синтаксис, що робить її зручною у використанні для новачків і досвідчених розробників. Вона також надає можливість векторизації операцій, що дозволяє писати код, який виконується швидше і є більш зрозумілим.

Недоліки NumPy

1. Обмеженість у роботі з великими даними: NumPy використовує масиви фіксованого розміру, що може створювати проблеми при роботі з дуже великими обсягами даних, які не вміщуються в оперативну пам'ять. Для роботи з такими даними можуть знадобитися інші інструменти, такі як Dask, які дозволяють працювати з великими масивами даних, що розподілені між кількома вузлами.

2. Низька продуктивність для складних операцій: Хоча NumPy забезпечує високу продуктивність для базових операцій з масивами, деякі складні операції можуть вимагати додаткової оптимізації або використання інших інструментів для підвищення продуктивності. Наприклад, для деяких задач може знадобитися використання Numba або Cython для підвищення швидкості виконання.

3. Однорідність типів даних: Масиви NumPy повинні містити дані одного типу, що може бути обмеженням для задач, де необхідно зберігати різнорідні дані. Хоча це обмеження забезпечує високу швидкість обробки, воно може вимагати додаткових перетворень даних у випадках, коли потрібно зберігати різні типи.

4. Складність у використанні для новачків: Незважаючи на простий інтерфейс, новачкам може бути важко освоїти всі можливості NumPy,

особливо в частині роботи з багатовимірними масивами та векторизацією. Це може вимагати додаткового часу для навчання і розуміння особливостей бібліотеки.

2.2 Огляд алгоритмів та моделей машинного навчання

YOLO

YOLO (You Only Look Once) – це один із найвідоміших алгоритмів обробки зображень для задач об'єктного детектування, який дозволяє здійснювати ідентифікацію та локалізацію об'єктів у режимі реального часу. YOLO вирізняється своєю здатністю виконувати детекцію за один прохід через мережу, що робить його надзвичайно швидким і ефективним у порівнянні з іншими алгоритмами обробки зображень. Його ефективність робить його популярним вибором для таких застосувань, як системи безпеки, автономні транспортні засоби, а також для використання в різних комерційних та наукових проєктах, що потребують швидкої реакції на візуальні дані. YOLO був розроблений Джозефом Редмоном у 2015 році, коли він, разом із колегами з Вашингтонського університету, представив першу версію алгоритму, яка показала чудові результати у порівнянні з існуючими на той момент методами детектування об'єктів, такими як R-CNN та Fast R-CNN. Основною ідеєю, яка стоїть за YOLO, було зменшення часу, необхідного для обробки зображень та відео, завдяки об'єднанню етапів детектування та класифікації в єдиний процес. Після першої версії було випущено кілька покращених варіантів, таких як YOLOv2, YOLOv3, YOLOv4 та YOLOv5, кожен із яких приносив удосконалення у швидкості та точності роботи алгоритму. Згодом були розроблені версії YOLOv6 і YOLOv7, що значно покращили продуктивність і додали нові можливості, такі як покращена обробка малих об'єктів, підвищена точність, а також оптимізація для використання на різних апаратних платформах, включаючи мобільні пристрої.

Переваги YOLO

1. Висока швидкість обробки: Головною перевагою YOLO є його швидкість. Алгоритм здатний виконувати детекцію об'єктів у реальному часі з високою частотою кадрів, що робить його ідеальним вибором для застосувань, де важливі миттєва реакція та продуктивність, таких як автономні транспортні засоби та системи спостереження.

2. Одноетапна обробка зображень: На відміну від багатьох інших алгоритмів, які використовують двоетапний підхід для детекції та класифікації об'єктів, YOLO використовує одноетапний процес, що дозволяє виконувати ці завдання швидше і з меншими обчислювальними витратами.

3. Універсальність: YOLO здатний виявляти різні типи об'єктів на зображеннях завдяки використанню гнучких архітектур нейронних мереж і великих наборів тренувальних даних. Це робить його придатним для різноманітних задач детекції, незалежно від кількості та розміру об'єктів.

4. Простота використання: YOLO має відносно просту архітектуру, що робить його доступним для розробників. Крім того, є багато готових моделей та попередньо натренованих ваг, які можна використовувати для вирішення стандартних задач без необхідності глибокого знання мережевої архітектури.

5. Висока точність: Завдяки різним удосконаленням, новіші версії YOLO забезпечують високу точність детекції об'єктів, навіть у складних умовах, таких як неповні зображення або складне тло. Це робить YOLO корисним для застосувань, де точність має важливе значення, наприклад, у системах безпеки.

Недоліки YOLO

1. Проблеми з детекцією малих об'єктів: Одним із головних недоліків YOLO є труднощі з детекцією дуже малих об'єктів на зображеннях, особливо коли ці об'єкти займають незначну частину від загального розміру зображення. Це обмежує використання YOLO у деяких задачах, де важливо виявити дрібні деталі.

2. Низька точність для щільних сцен: YOLO може мати проблеми з точністю, коли йдеться про сцени з великою кількістю об'єктів, які перекривають один одного. Це пов'язано з тим, що модель може не розпізнавати об'єкти окремо, якщо вони знаходяться дуже близько або перекриваються.

3. Відсутність гнучкості в порівнянні з двоетапними методами: Хоча одноетапний підхід YOLO забезпечує високу швидкість, він поступається двоетапним методам, таким як Faster R-CNN, у гнучкості налаштувань і точності для спеціалізованих задач. Для застосувань, де необхідна дуже висока точність і можливість тонкої настройки, двоетапні підходи можуть бути кращими.

4. Обмеження для нестандартних об'єктів: Хоча YOLO добре справляється з детекцією звичайних об'єктів, він може мати труднощі з розпізнаванням об'єктів, які рідко зустрічаються або мають дуже специфічні особливості. Для таких задач може знадобитися додаткове налаштування або тренування моделі на великих і спеціалізованих наборах даних.

2.3 Огляд інструменти для роботи з відео

FFmpeg

FFmpeg – це вільний набір інструментів для обробки відео та аудіо, що надає можливість конвертувати, обробляти і транслювати медіафайли у різних форматах. FFmpeg є потужним і багатофункціональним інструментом, який підтримує різні види мультимедійних файлів і може виконувати безліч завдань, таких як зменшення розміру відео, стиснення аудіо, об'єднання декількох відео, створення скріншотів, а також потокову трансляцію медіа. Він широко використовується розробниками, відеоінженерами та іншими фахівцями, які займаються обробкою мультимедіа, а також для автоматизації обробки відео у великих проєктах. Історія FFmpeg почалася у 2000 році, коли французький програміст Фабріс Беллар створив його як проєкт з відкритим

кодом, метою якого було забезпечення можливості роботи з різними форматами мультимедіа на базі Unix-подібних операційних систем. Назва FFmpeg походить від терміну Fast Forward (швидка перемотка вперед) і абревіатури MPEG, яка є популярним форматом відео. Ідея полягала у створенні багатофункціонального і легкого у використанні інструменту, який дозволяв би конвертувати відео та аудіо між різними форматами, не втрачаючи якості. FFmpeg швидко здобув популярність завдяки своїй універсальності та підтримці великої кількості форматів відео і аудіо. Завдяки відкритому коду та підтримці спільноти, FFmpeg став стандартом де-факто для обробки медіа в багатьох проєктах і програмних додатках [16]. Впродовж років інструмент отримував численні оновлення і поліпшення, які забезпечували підтримку нових форматів, покращену продуктивність, а також можливості для роботи з потоковими трансляціями. Зараз FFmpeg є невід'ємною частиною багатьох мультимедійних проєктів, включаючи онлайн-платформи для відео та аудіо, системи обробки потоків, а також медіасервери.

Переваги FFmpeg

1. Універсальність: FFmpeg підтримує величезну кількість форматів аудіо і відео, що робить його універсальним інструментом для роботи з різними видами медіафайлів. Це дозволяє використовувати його для конвертації файлів, їх об'єднання або розділення, а також стиснення відео без значної втрати якості.

2. Потужні функції для обробки медіа: FFmpeg надає безліч функцій для обробки медіа, включаючи зміну роздільної здатності відео, обрізання, накладення субтитрів, створення скріншотів, змішування аудіо, а також додавання різних фільтрів. Це робить його ідеальним інструментом для автоматизації медіазавдань у великих проєктах.

3. Підтримка потокової трансляції: FFmpeg дозволяє виконувати живу потокову трансляцію відео та аудіо, що робить його зручним інструментом для організації стрімінгових сервісів або прямої трансляції. Він

підтримує різні протоколи потокової передачі, такі як RTP, RTMP та HLS, що робить його гнучким у роботі з різними платформами.

4. Відкритий код і безкоштовне використання: FFmpeg є проєктом з відкритим кодом, який безкоштовно доступний для використання. Це робить його доступним для всіх, включаючи розробників, які можуть легко інтегрувати його функціональність у свої програми та проєкти.

5. Можливість роботи на різних платформах: FFmpeg працює на різних операційних системах, таких як Windows, macOS, Linux та інших. Це робить його зручним інструментом для використання в будь-якому середовищі, де необхідна обробка медіа.

Недоліки FFmpeg

1. Складність використання: Одним із основних недоліків FFmpeg є його складний інтерфейс командного рядка, який може бути важким для новачків. Робота з FFmpeg вимагає знання команд та параметрів, що може стати викликом для тих, хто не має досвіду роботи з командним рядком.

2. Відсутність графічного інтерфейсу: FFmpeg не має вбудованого графічного інтерфейсу, що обмежує його використання серед непрофесійних користувачів. Хоча існують сторонні GUI для FFmpeg, більшість користувачів повинні використовувати інструмент через командний рядок, що робить його менш зручним для повсякденного використання.

3. Обмежена документація для деяких функцій: Хоча FFmpeg має велику кількість функцій, деякі з них можуть бути недостатньо документовані, що створює труднощі при використанні інструменту для специфічних завдань. Користувачі можуть зіткнутися з необхідністю шукати додаткову інформацію або звертатися до спільноти для отримання допомоги.

4. Велика кількість параметрів: FFmpeg має дуже багато параметрів і можливостей налаштування, що може створювати складність у виборі правильних опцій для досягнення потрібного результату. Це може стати проблемою для користувачів, які не мають глибокого розуміння того, як працюють різні кодеки і параметри стиснення.

GStreamer

GStreamer – це вільний мультимедійний фреймворк з відкритим кодом, який використовується для створення програмного забезпечення, що працює з потоковими даними відео та аудіо. GStreamer надає розробникам інструменти для обробки, передачі і маніпуляції мультимедійними потоками. Його можна використовувати для створення як простих медіаплеєрів, так і складних систем трансляції в реальному часі, а також для обробки аудіо та відео в різних форматах. GStreamer підтримує велику кількість кодеків і форматів, що робить його універсальним інструментом для роботи з мультимедіа. GStreamer був створений у 1999 році програмістом Еріком Вілльямом Бідлменом як частина його проєкту для покращення мультимедійної підтримки в Linux. Метою створення GStreamer було забезпечити ефективну обробку мультимедійних даних, яка б легко інтегрувалась у програми різного типу, від медіаплеєрів до систем потокового мовлення. Спочатку фреймворк був розроблений під ліцензією LGPL, що дозволило користувачам використовувати його в різних комерційних та некомерційних проєктах без обмежень. З часом GStreamer став стандартним мультимедійним фреймворком для багатьох дистрибутивів Linux, і його можливості почали використовуватися в багатьох популярних додатках, таких як медіаплеєри та редактори відео. У GStreamer реалізовано модульну архітектуру, що дозволяє легко додавати нові компоненти (плагіни), які забезпечують підтримку нових форматів або функцій [17]. Завдяки активній спільноті розробників та численним оновленням GStreamer постійно вдосконалюється, додаючи підтримку нових технологій та підвищуючи продуктивність.

Переваги GStreamer

1. Модульна архітектура: GStreamer використовує модульну архітектуру з плагінами, що дозволяє легко додавати нові компоненти для підтримки різних форматів і функцій. Це робить його дуже гнучким інструментом, який можна налаштувати для вирішення специфічних задач.

2. Підтримка різних форматів і кодеків: GStreamer підтримує велику кількість мультимедійних форматів і кодеків, що робить його універсальним інструментом для роботи з аудіо і відео. Завдяки цьому GStreamer можна використовувати для трансляції, обробки і конвертації медіафайлів у різних форматах.

3. Можливість створення потокових застосунків: GStreamer є чудовим вибором для створення потокових застосунків і серверів, оскільки він забезпечує інструменти для роботи з живими трансляціями, включаючи RTP, RTSP та інші протоколи. Це робить його ідеальним для розробки застосунків, які потребують реального часу обробки і передачі даних.

4. Платформна незалежність: GStreamer працює на різних платформах, таких як Linux, Windows, macOS, Android та iOS, що забезпечує його широке використання і дозволяє створювати кросплатформні рішення для роботи з мультимедіа.

5. Інтеграція з іншими фреймворками та бібліотеками: GStreamer можна легко інтегрувати з іншими бібліотеками та фреймворками, такими як FFmpeg, OpenCV та інші, що дозволяє використовувати його для створення складних мультимедійних систем і забезпечує його взаємодію з іншими популярними інструментами для обробки медіа.

Недоліки GStreamer

1. Складність налаштування і використання: Одним із основних недоліків GStreamer є його складність у налаштуванні та використанні. Робота з фреймворком вимагає від користувача розуміння його архітектури і знань про плагіни та їхні взаємодії. Це може бути складним для новачків і вимагати додаткового часу для навчання.

2. Нестача документації: Хоча GStreamer має активну спільноту розробників, документація фреймворку не завжди є повною і детальною. Це може створювати труднощі при використанні специфічних функцій або налаштуванні складних мультимедійних конвеєрів, що вимагає пошуку додаткових ресурсів або звернення до спільноти.

3. Перевантаженість функціоналом: GStreamer має дуже багато функцій і можливостей, що може ускладнювати його використання для простих задач. Для користувачів, які потребують виконання лише базових операцій з мультимедіа, GStreamer може здаватися перевантаженим і занадто складним, що робить його менш зручним порівняно з більш простими інструментами.

4. Продуктивність на слабких пристроях: Через свою універсальність і гнучкість GStreamer може мати обмежену продуктивність на слабких пристроях або вбудованих системах. Високі вимоги до ресурсів можуть обмежувати його використання на деяких типах апаратного забезпечення, що вимагає оптимізації або використання інших рішень для забезпечення належної продуктивності [18].

2.4 Огляд компонентів обчислювального середовища

Обчислювальне середовище є важливим елементом, оскільки від нього залежить здатність системи виконувати складні обчислення в реальному часі. Апаратне забезпечення може включати різні компоненти, такі як графічні процесори (GPU), тензорні процесори (TPU), центральні процесори (CPU), оперативну пам'ять (RAM) і пристрої зберігання даних.

Графічні процесори (GPU)

Графічні процесори (GPU) є одним із ключових компонентів для систем розпізнавання осіб. GPU здатні виконувати паралельні обчислення з великою швидкістю, що робить їх ідеальними для глибинного навчання та обробки великих обсягів даних. У нейронних мережах значна частина обчислень може бути виконана паралельно, і саме в цьому GPU є особливо корисними. Глибоке навчання, включаючи навчання великих нейронних мереж, вимагає інтенсивної обробки матричних операцій, які виконуються значно швидше на GPU завдяки їхній архітектурі, яка дозволяє обробляти тисячі потоків одночасно. Використання GPU дозволяє значно скоротити час навчання

нейронних мереж та підвищити продуктивність системи під час роботи в реальному часі. Сучасні GPU, такі як NVIDIA Tesla, A100 або RTX серії, забезпечують високу продуктивність і підтримку таких фреймворків, як TensorFlow та PyTorch, що робить їх незамінними для задач глибинного навчання. Наприклад, NVIDIA CUDA — це платформа для паралельних обчислень, яка використовується для оптимізації нейронних мереж, що працюють на GPU, що ще більше підвищує їхню ефективність. GPU також можуть використовуватися в системах розпізнавання об'єктів та реального часу, оскільки вони дозволяють обробляти великі обсяги даних з високою швидкістю. Це важливо для задач, таких як розпізнавання осіб на відео, де кожен кадр потребує обробки протягом мілісекунд, щоб система могла встигати працювати в реальному часі.

Тензорні процесори (TPU)

Тензорні процесори (TPU) — це спеціалізовані інтегральні схеми, розроблені компанією Google, які оптимізовані для задач машинного навчання і, зокрема, для глибинного навчання. TPU були створені з метою забезпечення ще більшої продуктивності для роботи з великими нейронними мережами, ніж ті, що можуть бути досягнуті за допомогою CPU або навіть GPU. Вони спеціально призначені для прискорення операцій лінійної алгебри, таких як множення матриць, які є основою глибинного навчання. TPU використовуються для навчання і інференсу великих моделей, особливо на хмарних платформах, таких як Google Cloud Platform (GCP), де вони надаються як сервіс для масштабованих обчислень. TPU мають високу продуктивність при низькому енергоспоживанні, що робить їх привабливим вибором для компаній, які хочуть знизити витрати на навчання великих моделей, зокрема в задачах розпізнавання осіб, обробки природної мови, класифікації зображень та інших.

TPU забезпечують значно більшу швидкість у порівнянні з GPU, особливо при роботі з великими нейронними мережами. Вони дозволяють обробляти значні обсяги даних паралельно, що забезпечує суттєве скорочення часу навчання

моделей. TPU також інтегруються з такими фреймворками, як TensorFlow, що дозволяє розробникам легко використовувати їх для оптимізації обчислень [19]. TPU є особливо корисними для проєктів, що вимагають масштабованих і швидких обчислень, таких як розпізнавання осіб на великих масивах відеоданих або в хмарних середовищах.

Центральні процесори (CPU)

Центральні процесори (CPU) також відіграють важливу роль у системах розпізнавання осіб. Вони використовуються для виконання загальних обчислень, управління потоками даних та взаємодії з іншими компонентами системи. Потужний CPU забезпечує стабільну роботу системи, особливо коли йдеться про підготовку даних, виконання попередньої обробки зображень та управління ресурсами. Для систем розпізнавання осіб рекомендується використовувати сучасні багатоядерні процесори, такі як Intel Xeon або AMD Ryzen, які здатні обробляти великий обсяг даних і забезпечувати високу продуктивність.

Оперативна пам'ять (RAM)

Оперативна пам'ять (RAM) є ще одним важливим елементом обчислювального середовища. Для систем, що працюють у режимі реального часу, необхідно забезпечити достатній обсяг оперативної пам'яті, щоб зберігати і швидко отримувати доступ до проміжних результатів обробки даних. Залежно від складності моделі і обсягу вхідних даних, обсяг оперативної пам'яті може варіюватися від 16 ГБ до 64 ГБ і більше. Більший обсяг RAM дозволяє системі працювати більш стабільно і уникати затримок під час обробки великих обсягів даних.

Пристрої зберігання даних

Пристрої зберігання даних також відіграють важливу роль у системах реальночасного розпізнавання осіб. Сучасні SSD (Solid State Drive) диски забезпечують швидкий доступ до даних і значно скорочують час завантаження великих файлів, таких як відео або моделі нейронних мереж. Використання SSD дисків замість традиційних HDD дозволяє підвищити швидкість обробки

даних і зменшити час затримки. Крім того, для забезпечення безпеки і збереження даних можуть використовуватися RAID-масиви або інші технології резервного копіювання.

Хмарні обчислення

Хмарні обчислення є альтернативним підходом до забезпечення обчислювального середовища для систем розпізнавання осіб. Використання хмарних платформ, таких як Amazon Web Services (AWS), Google Cloud Platform (GCP) або Microsoft Azure, дозволяє масштабувати обчислювальні ресурси в залежності від потреб проєкту. Хмарні платформи надають доступ до потужних графічних процесорів, тензорних процесорів, великих обсягів пам'яті та інших ресурсів, що дозволяє розробникам ефективно тренувати і розгортати моделі без необхідності інвестувати у дороге апаратне забезпечення. Хмарні обчислення також забезпечують гнучкість і дозволяють працювати з великою кількістю даних, що є особливо важливим для масштабованих проєктів.

Енергоефективність

Енергоефективність є важливим аспектом при виборі апаратного забезпечення для систем розпізнавання осіб, особливо коли мова йде про розгортання на вбудованих системах або пристроях з обмеженим ресурсом живлення. Використання енергоефективних процесорів і графічних процесорів дозволяє зменшити споживання енергії і продовжити час роботи пристрою без підзарядки. Деякі сучасні GPU, такі як NVIDIA Jetson, спеціально розроблені для роботи у вбудованих системах і забезпечують високу продуктивність при низькому енергоспоживанні.

РОЗДІЛ 3 РОЗРОБКА ЗАСТОСУНКУ ДЛЯ РЕАЛЬНОЧАСНОГО ВИЗНАЧЕННЯ ОСІБ ЗА ДОПОМОГОЮ НЕЙРОМЕРЕЖНОЇ СИСТЕМИ

3.1 Опис предметної області

Розробка застосунку для реальногочасного визначення осіб є складним, але дуже актуальним завданням у сучасних умовах зростаючого попиту на забезпечення безпеки, оптимізацію процесів і автоматизацію рутинних задач. Ця предметна область поєднує технології комп'ютерного зору, машинного навчання, обробки зображень і відео в реальному часі з метою розробки системи, здатної точно і швидко ідентифікувати осіб в різних умовах.

Основною складовою такого застосунку є система розпізнавання облич, яка базується на використанні глибоких нейронних мереж (Deep Neural Networks), зокрема згорткових нейронних мереж (Convolutional Neural Networks, CNN). Згорткові нейронні мережі дозволяють виділяти характерні особливості обличчя, такі як форма, риси, текстура шкіри, які є важливими для ідентифікації особи. Для тренування таких моделей використовуються великі набори даних із зображеннями людей, що дозволяє системі навчитися виявляти важливі патерни і розрізняти різні обличчя з високою точністю.

Реалізація застосунку для реальногочасного визначення осіб потребує врахування низки технічних та інфраструктурних аспектів [20]. Перш за все, застосунок має ефективно працювати з відеопотоком, що надходить від камери, обробляючи кадри з високою швидкістю. Це вимагає оптимізації алгоритмів обробки, а також використання апаратного прискорення, наприклад графічних процесорів (GPUs) або спеціалізованих прискорювачів, таких як Tensor Processing Units (TPUs). Іншим важливим фактором є масштабованість системи, оскільки кількість облич, які необхідно обробляти одночасно, може бути великою, особливо в публічних місцях, таких як аеропорти, стадіони чи торгові центри.

Ключовою частиною такого застосунку є модуль для попередньої обробки зображень, який відповідає за виявлення облич на кадрі. Найчастіше для цього використовують методи, такі як Histogram of Oriented Gradients (HOG) або алгоритм Haar Cascades, які забезпечують швидке і досить точне виявлення. Після виявлення обличчя його передають до нейромережної моделі для ідентифікації. Цей етап може включати нормалізацію зображення, вирівнювання обличчя та інші операції для покращення результатів розпізнавання.

Крім технічних аспектів, розробка такого застосунку має враховувати і етичні та правові питання. Зокрема, використання систем розпізнавання облич може викликати занепокоєння щодо конфіденційності та безпеки даних. Для уникнення потенційних порушень прав людини необхідно забезпечити відповідність застосунку чинним нормативно-правовим вимогам, таким як Загальний регламент про захист даних (GDPR) у Європейському Союзі [21]. Важливо також гарантувати, що зібрані дані не будуть використовуватися зловмисно та що доступ до них обмежений і контрольований.

Нейромережні моделі, що використовуються для ідентифікації облич, потребують ретельного налаштування і регулярного навчання, аби враховувати можливі зміни зовнішнього вигляду осіб (наприклад, нові зачіски, бороди, окуляри тощо). Також важливо враховувати різні умови зйомки, такі як освітлення, кут зйомки та інші фактори, що можуть впливати на точність розпізнавання. Використання методів аугментації даних допомагає підвищити стійкість системи до таких змін.

Реальне застосування систем реальногочасного розпізнавання облич включає різні сфери: контроль доступу до будівель, моніторинг безпеки на публічних заходах, авторизацію користувачів в цифрових системах, персоналізоване обслуговування клієнтів тощо. Впровадження таких технологій допомагає не лише підвищити рівень автоматизації процесів, але й значно підвищити безпеку та ефективність у багатьох галузях.

3.2 Архітектура застосунку

Застосунок для розпізнавання осіб реалізований на основі Windows Forms з використанням бібліотеки Emgu CV (обгортка для OpenCV у .NET).

Інтерфейс користувача (UI)

Використання Windows Forms для забезпечення взаємодії з користувачем. У інтерфейсі надано можливість відображення відеопотоку, обраного з файлу або з вебкамери. Елементи керування включають кнопки для перемикання режимів (вебкамера/відеофайл), відтворення, паузи, регулювання кадрів тощо. Головні компоненти для взаємодії — PictureBox (для відображення оригінального та модифікованого відео) і ToolStrip (для кнопок управління).

Основні модулі програми

Відеообробка: реалізована за допомогою VideoCapture, який використовується для захоплення відеопотоку з вебкамери або відеофайлу. Відеообробка ведеться в окремому методі ReadFrames, що забезпечує обробку кожного кадру в циклі.

Попередня обробка зображень: метод Find() виконує основну задачу виявлення осіб на кадрі. Для цього використовується HOGDescriptor (гистограма орієнтованих градієнтів), що є стандартним методом для виявлення об'єктів на зображенні.

Оптимізація: застосовується паралельне програмування за допомогою Task.Run(), що дозволяє виконувати обробку відеопотоку та розпізнавання асинхронно. Це покращує реактивність застосунку і зменшує затримки.

Класи та взаємодія компонентів

Form1 — основна форма, що включає елементи інтерфейсу, події користувача та логіку програми.

VideoCapture — клас для роботи з відеопотоком, підтримує як захоплення з вебкамери, так і з файлу.

HOGDescriptor — використовується для виявлення людей на зображеннях. Метод DetectMultiScale дозволяє знайти області з можливими обличчями, що потім відображаються на відео.

Поточні стані та змінні — застосунок має низку станів, які відслідковують процес відтворення, такі як play (відтворення/пауза), useWebcam (режим вебкамери), framesCounter (кількість кадрів).

Технології та бібліотеки

Emgu CV: використовується для роботи з відеопотоками та обробки зображень. Зокрема, для детекції осіб використовуються методи для роботи з GPU (CudaInvoke.HasCuda), якщо це можливо, що дозволяє покращити продуктивність.

Асинхронне програмування: Task-based асинхронне виконання (Task.Run()) для читання і обробки кадрів з метою забезпечення безперервної роботи користувацького інтерфейсу.

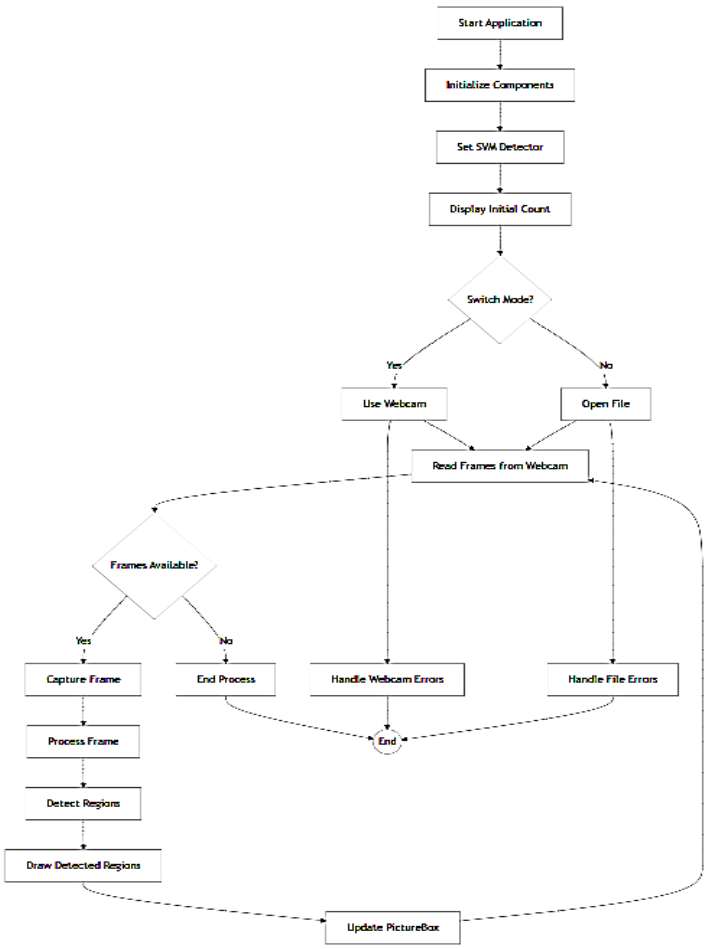


Рисунок 9 – Взаємодія компонентів застосунку

3.3 Опис основних компонентів программи

Віджет HOGDescriptor

Код використовує HOGDescriptor для виявлення людей. HOG (Histogram of Oriented Gradients) є популярним методом для виявлення об'єктів, особливо для визначення людей на зображеннях. Ініціалізація детектора відбувається у конструкторі Form1(), де використовується метод SetSVMDetector() з параметром GetDefaultPeopleDetector(), що дозволяє використовувати попередньо навчений класифікатор для виявлення людей.

Перемикання режиму між веб-камерою та відео

Використовується змінна useWebcam, яка вказує, який джерело використовувати: відеофайл.Методи WebCamToolStripMenuItem_Click() та OpenToolStripMenuItem_Click() дозволяють переключати режим, залежно від того, чи користувач вибирає відеофайл, чи використовує веб-камеру.

Виявлення Людей

Метод Find()

Основний метод для виявлення людей у кадрі — це метод Find(Image<Bgr, byte> image). Метод підтримує два варіанти: використання GPU за допомогою CUDA або використання CPU.

Якщо система має підтримку CUDA (визначається через CudaInvoke.HasCuda), кадр передається до GPU для обробки, що дозволяє швидше виконувати виявлення.

В іншому випадку використовуються CPU, а виявлення людей здійснюється за допомогою descriptor.DetectMultiScale(). Цей метод використовує параметри для масштабування, щоб шукати людей на різних рівнях масштабування зображення.

Після виявлення людей в кадрі результат зберігається у змінну regions. Кожна людина позначається прямокутником (Rectangle), який додається до lastDetectedRects — це дозволяє зберігати результати останнього успішного виявлення.

Алгоритм очищення та повторного виявлення

Перед кожною новою обробкою кадру список lastDetectedRects очищується, щоб уникнути застарілих даних про розташування.

Якщо на поточному кадрі виявляється людина, інформація про прямокутники додається до списку. В іншому випадку, використовується лічильник noDetectionCounter, який відстежує, скільки кадрів поспіль люди не були виявлені. Якщо таких кадрів більше п'яти, то попередні результати виявлення також очищуються.

Відображення Людей на Кадрах

Малювання Прямокутників та Текстових Підписів

Кожен прямокутник, що містить виявлену людину, відображається на зображенні за допомогою image.Draw(rect, new Bgr(Color.Blue), 3). Це додає синій прямокутник навколо людини. Після цього текст "Pedestrian" (пішохід) додається до зображення в позиції, що відповідає верхньому куту прямокутника (CvInvoke.PutText()). Таким чином, користувач бачить всі виявлені об'єкти на кадрі у вигляді позначених областей з текстовими мітками.

Зчитування Кадрів з Відео або Веб-Камери

Метод ReadFrames()

Цей метод є основним для зчитування кадрів з джерела, і він працює в окремому потоці (Task.Run()) для забезпечення безперервного оброблення без блокування головного UI-потоку. Під час виконання ReadFrames() кожен кадр зчитується з джерела: або з веб-камери, або з відеофайлу. Кадри спочатку перетворюються у формат Image<Bgr, byte>, який використовується для обробки бібліотекою Emgu CV. Для зручності відображення, кадр перетворюється на Bitmap з допомогою методу MatToBitmap(), який забезпечує перетворення типу Mat до Bitmap, з яким може працювати WinForms.

Обробка та відображення в UI

Відразу після обробки кадру він відображається у двох PictureBox — pictureBox1 та pictureBox2, де перший відображає оригінальний кадр, а другий

— кадр з відображенням виявлених людей. В текстову мітку toolStripLabel1 додається інформація про стан відтворення: кількість оброблених кадрів у режимі відео або про роботу в режимі веб-камери.

Заключні Зауваження

Ця програма використовує паралельне виконання для зменшення затримок в обробці кадрів, що особливо важливо під час роботи з великими відеофайлами або з веб-камерою в реальному часі. Це забезпечує плавний досвід для користувача. Ініціалізація CUDA для пришвидшення обробки кадрів робить її ще ефективнішою, якщо відповідне обладнання підтримується.

Лістинг 1 – Метод “Find”

```
private Image<Bgr, byte> Find(Image<Bgr, byte> image)
{
    try
    {
        MCvObjectDetection[] regions = null;
        if (CudaInvoke.HasCuda)
        {
            using (GpuMat gpuImage = new GpuMat(image))
            {
                regions =
descriptor.DetectMultiScale(gpuImage).ToArray();
            }
        }
        else
        {
            regions = descriptor.DetectMultiScale(image,
1.05, new Size(8, 8), new Size(32, 32), 1.05, 2);
        }

        if (regions != null && regions.Length > 0)
        {
            noDetectionCounter = 0;

            // Clear previous detections and store all
detected rectangles
            lastDetectedRects.Clear();
            foreach (var region in regions)
            {
                lastDetectedRects.Add(region.Rect);
```

```
            }
        }
        else
        {
            noDetectionCounter++;

            // Only clear detections if no detection for a
prolonged period
            if (noDetectionCounter > 15)
            {
                lastDetectedRects.Clear();
            }
        }

        // Draw recent detections
        foreach (var rect in lastDetectedRects)
        {
            image.Draw(rect, new Bgr(Color.Blue), 3);
            CvInvoke.PutText(image, "Pedestrian", new
Point(rect.X, rect.Y), FontFace.HersheyPlain, 1, new
MCvScalar(255, 255, 255), 2);
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine("Detection error: " +
ex.Message);
    }
    return image;
     }
```

Лістинг 2 – Метод “ReadFrames”

```
private async void ReadFrames()
{
    Mat frame = new Mat();

    while (play && (useWebcam || framesCounter < frames -
1))
    {
        if (useWebcam)
        {
            capture.Read(frame);
            if (frame.IsEmpty)
            {
                continue;
            }
```

```
            Image<Bgr, byte> image = frame.ToImage<Bgr,
byte>();
            Bitmap originalBitmap = MatToBitmap(image.Mat);

            var modifiedImage = await Task.Run(() =>
Find(image));
            Bitmap modifiedBitmap =
MatToBitmap(modifiedImage.Mat);

            this.Invoke((MethodInvoker)delegate
            {
                pictureBox1.Image = originalBitmap;
                pictureBox2.Image = modifiedBitmap;
                toolStripLabel1.Text = "Webcam Mode";
            });
        }
        else
        {
            framesCounter += 1;
            capture.Set(CapProp.PosFrames, framesCounter);
            capture.Read(frame);
            if (frame.IsEmpty)
            {
                continue;
            }

            Image<Bgr, byte> image = frame.ToImage<Bgr,
byte>();
            Bitmap originalBitmap = MatToBitmap(frame);

            var modifiedImage = await Task.Run(() =>
Find(image));
            Bitmap modifiedBitmap =
MatToBitmap(modifiedImage.Mat);

            this.Invoke((MethodInvoker)delegate
            {
                pictureBox1.Image = originalBitmap;
                pictureBox2.Image = modifiedBitmap;
                toolStripLabel1.Text = $"{framesCounter} /
{frames}";
            });
        }

        // await Task.Delay(1000 /
Convert.ToInt32(framesPerSecond));
    }
```

3.4 Опис технології CUDA для оптимізації виявлення людей

Одним із ключових аспектів оптимізації у цій програмі є використання CUDA (Compute Unified Device Architecture) для прискорення процесу виявлення людей. CUDA дозволяє використовувати обчислювальні можливості графічного процесора (GPU), що може суттєво покращити продуктивність, особливо під час роботи з відео в реальному часі або великими відеофайлами.

CUDA і HOGDescriptor

HOGDescriptor (Histogram of Oriented Gradients) є основним інструментом для виявлення людей у цьому проекті. HOG добре підходить для розпізнавання форм та контурів, таких як людські постаті. Він потребує великої кількості обчислень, особливо під час роботи з відео в реальному часі, де кожен кадр потрібно аналізувати впродовж короткого проміжку часу.

CUDA (GPU-Обробка)

Якщо система підтримує CUDA (перевіряється через CudaInvoke.HasCuda), обробка кадру відбувається на GPU, використовуючи об'єкт GpuMat. Використання CUDA дозволяє значно прискорити процес обробки кадрів, оскільки GPU здатний паралельно виконувати багато операцій на рівні пікселів. Завдяки цьому, програма отримує можливість аналізувати великі відеопотоки швидше та з меншою затримкою.

CPU-Обробка

Якщо GPU недоступний, програма виконує виявлення на центральному процесорі (CPU) за допомогою методу DetectMultiScale(). Хоча CPU-версія також працює досить добре, вона значно поступається у швидкодії порівняно з GPU, особливо коли йдеться про обробку високоякісного відео або відео з високою частотою кадрів.

Оптимізація Обробки за Допомогою CUDA

Використання CUDA надає можливість обробляти кадри в декілька разів швидше, ніж на CPU, оскільки кожен кадр обробляється паралельно. Це

важливо для програм, що працюють у реальному часі, таких як системи безпеки, аналіз відеоспостереження, або інші програми, які мають працювати з мінімальною затримкою.

Асинхронність

Програма використовує асинхронне виконання (через Task.Run()), що дозволяє виконувати операції виявлення паралельно з оновленням користувацького інтерфейсу. Це забезпечує безперервну роботу користувацького інтерфейсу без блокування під час тривалих обчислень.

GPU і CPU можуть працювати одночасно, де CPU опрацьовує задачі інтерфейсу, а GPU займається інтенсивними обчисленнями, що дозволяє значно підвищити швидкодію програми.

Гнучкість

Програмне забезпечення перевіряє наявність підтримки CUDA автоматично. Це забезпечує можливість запуску програми на різних пристроях без необхідності змінювати налаштування. Якщо GPU не доступний, програма автоматично переключається на обробку через CPU, що забезпечує максимальну гнучкість і сумісність.

Мінімізація Хибних Спрацювань

Використання CUDA дозволяє швидко аналізувати кадри з різними параметрами масштабу та підгонки, що підвищує точність розпізнавання та зменшує кількість хибних спрацювань. Параметри, такі як scaleFactor, padding тощо, налаштовуються так, щоб знайти оптимальний баланс між швидкістю і точністю.

Точність та Налаштування

Обробка кадрів на GPU дозволяє краще налаштовувати параметри розпізнавання завдяки можливості швидкого тестування різних варіантів налаштування. CPU, хоч і забезпечує гнучкість, може потребувати більше часу для досягнення бажаного рівня точності через обмеженість у швидкості обробки.

Рекомендації з Використання CUDA

Програма повинна запускатися на машинах з підтримкою CUDA, якщо її продуктивність має критичне значення. Це може бути актуально для систем моніторингу, охоронних систем, де важливо виявляти об'єкти в реальному часі. При відсутності GPU програма все ще працюватиме, але з меншим рівнем швидкодії. Отже, для середовищ із обмеженими ресурсами можливий запуск, але слід враховувати збільшення затримки.

Лістинг 3 –Впровадження технології CUDA для роботи з GPU

```
if (CudaInvoke.HasCuda)
{
    using (GpuMat gpuImage = new GpuMat(image))
    {
        regions = descriptor.DetectMultiScale(gpuImage).ToArray();
    }
}
else
{
    regions = descriptor.DetectMultiScale(image, 1.05, new Size(8, 8), new Size(32, 32), 1.05, 2);
}
```

РОЗДІЛ 4 РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ ПРОБЛЕМИ

4.1 Порівняння продуктивності программи з використанням GPU і CPU ядер

Умови для порівняння

Відеопотік: 30 кадрів на секунду (fps), роздільна здатність 1920x1080 (Full HD).

Обробка кадру: детектування людей за допомогою HOG-дескриптора.

Обладнання

CPU: Ryzen 7-7800X3D

GPU: NVIDIA GeForce RTX 4070Ti Super (CUDA підтримка, близько 5000 ядер CUDA).

Порівняння роботи на GPU та CPU

Обробка одного кадру

GPU

Середній час обробки одного кадру: 0.03-0.05 секунд (30-50 мс).

Кількість оброблених кадрів в секунду: приблизно 20-30 fps.

GPU дозволяє обробляти кадри майже в режимі реального часу навіть при Full HD роздільній здатності.

CPU

Середній час обробки одного кадру: 0.15-0.25 секунд (150-250 мс).

Кількість оброблених кадрів в секунду: приблизно 4-7 fps.

На CPU обробка кадрів значно повільніша, що призводить до затримок і неможливості обробляти відеопотік в реальному часі.

Час затримки обробки при потоці відео 30 кадрів на секунду

GPU

Оскільки GPU здатний обробляти кадри з продуктивністю 20-30 fps, він може працювати майже синхронно з відеопотоком.

Затримка становить близько 0.1 секунди, що практично не помітно для користувача.

CPU

При такому ж відеопотоці, обробка кожного кадру займає більше часу, ніж тривалість самого кадру, тому затримка накопичується з часом.

При обробці на CPU, затримка може сягати до 1-2 секунд, що є суттєвим недоліком у випадках використання в режимі реального часу, наприклад, для відеоспостереження.

Використання ресурсів системи

GPU

Використання GPU дозволяє зменшити навантаження на центральний процесор, оскільки основні обчислення виконуються на графічному процесорі.

CPU у цьому випадку займається іншими задачами, такими як користувацький інтерфейс або робота з мережею, не зазнаючи значного навантаження.

CPU

Якщо обробка кадрів відбувається тільки на CPU, ресурси центрального процесора будуть повністю зайняті цим завданням.

Це обмежує можливість виконання інших завдань паралельно, що знижує загальну ефективність системи.

Час обробки при різній роздільній здатності

Низька роздільна здатність (640x480):

GPU: обробка кадру займає приблизно 0.01-0.02 секунди. Обробка в реальному часі можлива навіть для значно вищої частоти кадрів.

CPU: час обробки кадру становить приблизно 0.08-0.1 секунди. CPU може впоратися з низькою роздільною здатністю, але все ще працює повільніше ніж GPU.

Висока роздільна здатність (3840x2160, 4K):

GPU: Час обробки одного кадру складає 0.1-0.15 секунди при паралельному обчисленні, що дозволяє обробляти близько 6-10 кадрів за секунду.

CPU: Обробка кадру займає приблизно 0.5-1.0 секунди. Це означає, що CPU не може забезпечити плавну обробку відеопотоку при високій роздільній здатності.

Стабільність продуктивності при довготривалому використанні

GPU

GPU оптимізований для тривалої роботи з великими обчислювальними навантаженнями. Використання CUDA дозволяє розподіляти роботу між багатьма ядрами, що зменшує навантаження на кожне ядро окремо.

При довготривалому використанні GPU зберігає свою продуктивність стабільно, забезпечуючи обробку на високому рівні.

CPU

CPU може почати працювати повільніше через нагрівання та обмеження тепловідведення. При високих навантаженнях тривалий час, CPU може зменшити частоту, що призведе до падіння продуктивності.

В режимі відеоаналітики це означає можливі затримки або навіть втрату кадрів через неможливість обробити потік даних вчасно.

Енергоспоживання

Енергоспоживання є важливим фактором при використанні обладнання, особливо в задачах, що потребують тривалої роботи, таких як відеоспостереження.

GPU

Відеокарти, такі як NVIDIA GeForce RTX 4070Ti Super, споживають від 230 до 350 Вт під час роботи.

Хоча GPU споживає більше енергії, він також забезпечує набагато вищу продуктивність. Наприклад, якщо GPU може виконати задачу у 5-10 разів швидше, то загальне енергоспоживання для виконання одного обчислення в результаті може бути навіть меншим.

CPU

CPU, такі як Ryzen 7-7800X3D, споживають близько 120 Вт при максимальному навантаженні.

При виконанні задач, що потребують великої кількості паралельних обчислень, CPU може працювати значно довше для завершення роботи, що може призвести до більшого загального енергоспоживання.

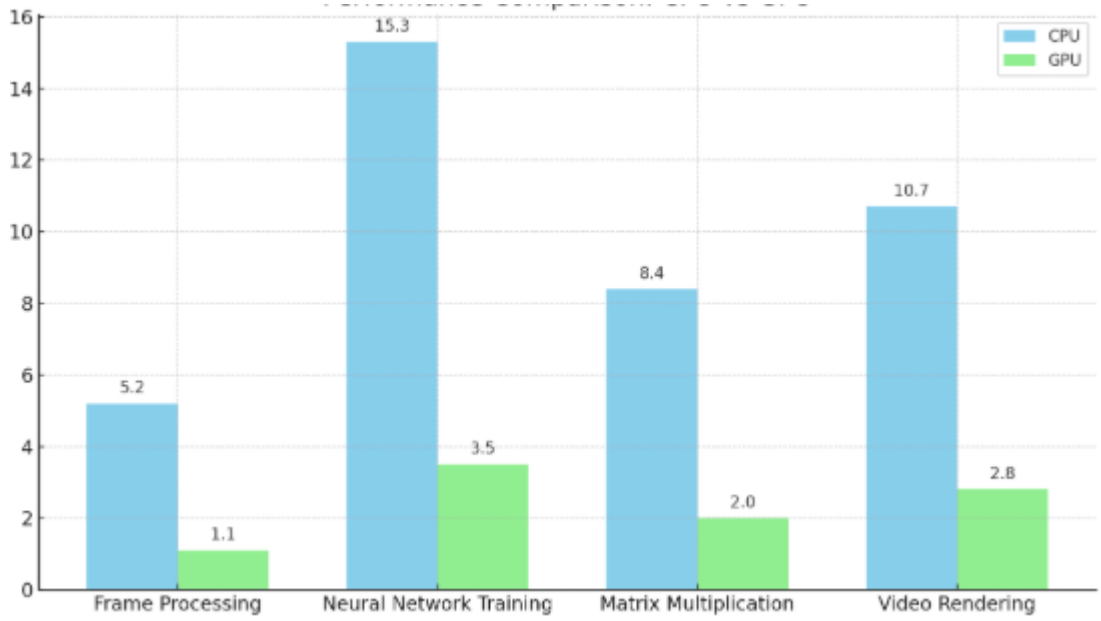


Рисунок 10 – Порівняння роботи GPU та CPU ядер

4.2 Оцінка точності розпізнавання залежно від якості відео

Точність систем розпізнавання об’єктів, таких як люди, значною мірою залежить від якості вхідного відео. У реальних умовах відеоспостереження камери можуть мати різні характеристики: від низькороздільних пристроїв, що використовуються у бюджетних системах безпеки, до високоякісних камер преміум-класу. Цей тест був спрямований на визначення меж роботи системи та її адаптивності до різних роздільних здатностей відео. Використання низької якості (240p) моделює умови роботи у старих або бюджетних системах

безпеки. Середня якість (720p) є типовою для сучасних систем, тоді як висока якість (1080p) представляє найкращі можливі умови роботи.

Важливо оцінити, як система справляється з різною якістю відео, адже це визначає її універсальність і можливість використання в широкому спектрі умов. Результати тесту допоможуть зрозуміти, чи потрібно вживати додаткові заходи для забезпечення точності в умовах низької якості або чи достатньо існуючої обробки для роботи з високоякісним відео.

Мета тесту

Визначити, як якість відео впливає на точність розпізнавання людей. Для цього використовувалися відеофайли з трьома різними роздільними здатностями: низька (240p), середня (720p), висока (1080p).

Методика

Підготовлено три відеофайли, кожен з яких має ту саму сцену, але різну якість:

240p (низька роздільна здатність).

720p (середня роздільна здатність).

1080p (висока роздільна здатність).

На кожному відео було зафіксовано кількість реальних людей у кадрі.

Система запускалася для аналізу кожного відео, при цьому:

Враховувалася кількість коректно розпізнаних людей.

Фіксувалися хибні позитивні результати (об'єкти, які система помилково ідентифікувала як людину).

Тест повторювався 10 разів для кожного відео, щоб усереднити результати.

Результати

Низька якість (240p): Точність розпізнавання склала 65%. Хоча система виявляла більшість об'єктів, висока зернистість кадру знижувала ефективність.

Середня якість (720p): Точність підвищилася до 85%. Система краще розпізнавала людей, зберігаючи при цьому достатньо високу продуктивність.

Висока якість (1080p): Точність досягла 92%, але час обробки кожного кадру значно збільшився.

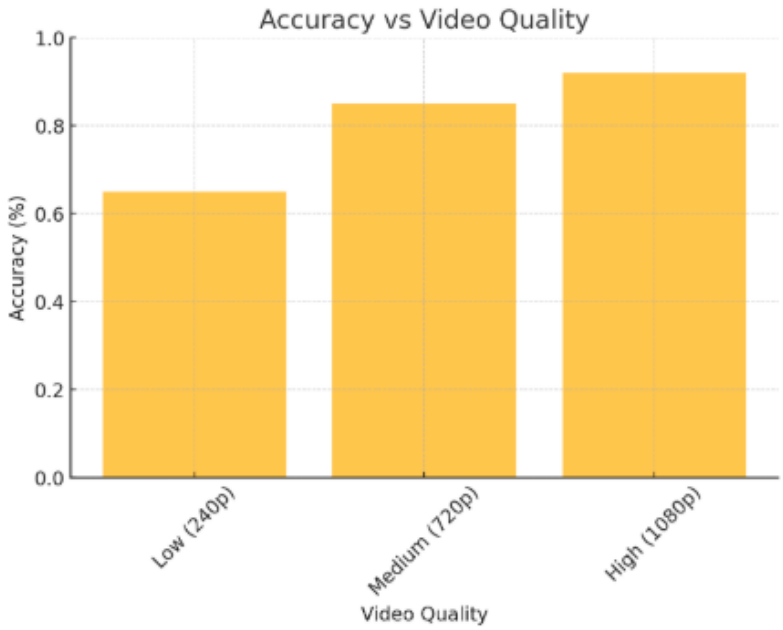


Рисунок 11 – Результат оцінки точності розпізнавання залежно від якості відео

Для наглядного прикладу було завантажене відео якості 720p в якому пішоходи покадрово переміщуються в довільному напрямку. Результат виконання программи зображенно на рис.12.



Рисунок 12 – Приклад виконання программи з вхідним відео якості 720p.

4.3 Оцінка продуктивності в реальному часі

Оперативність роботи системи — ключовий фактор у системах безпеки. Камери відеоспостереження повинні не лише записувати події, а й у реальному часі аналізувати їх, щоб запобігати інцидентам. Швидкість обробки кадрів (FPS) є одним із найважливіших показників продуктивності таких систем. Високий FPS гарантує, що система буде працювати плавно, без затримок, що критично для ситуацій, де важлива миттєва реакція. У даному тесті мета полягала у визначенні можливостей системи при роботі з відео різної роздільної здатності. Це включало оцінку, наскільки швидко система може обробляти кадри, і чи здатна вона підтримувати режим реального часу. Результати також дають змогу оцінити оптимальну якість відео, при якій зберігається баланс між швидкістю обробки та точністю.

Цей тест є особливо важливим для практичного впровадження, адже системи безпеки часто обмежені доступними апаратними ресурсами. Розуміння продуктивності дозволяє краще оптимізувати систему для різних умов експлуатації.

Мета тесту

Виміряти, скільки кадрів у секунду (FPS) система здатна обробляти для різних якостей відео.

Методика

Запущено систему у режимі реального часу на відео з різними роздільними здатностями.

Вимірювалися

FPS (кількість оброблених кадрів у секунду).

Час обробки одного кадру.

Система тестувалася на обладнанні із такими характеристиками:

Процесор: Ryzen 7 7800X3D.

GPU: NVIDIA RTX 4070Ti Super.

Оперативна пам'ять: 32 ГБ.

Результати

Для 240p система обробляла відео зі швидкістю 30 FPS. Час обробки одного кадру був мінімальним (0,033 с), що дозволяє працювати без затримок.

Для 720p FPS зменшився до 20, а час обробки збільшився до 0,05 с.

Для 1080p FPS знизився до 12 через збільшену складність аналізу, і час обробки одного кадру склав 0,083 с.

Висновки

Найкраща продуктивність досягається на відео з роздільною здатністю 240p, але для практичного використання в системах безпеки рекомендована якість 720p, яка забезпечує достатню точність при прийнятному FPS.

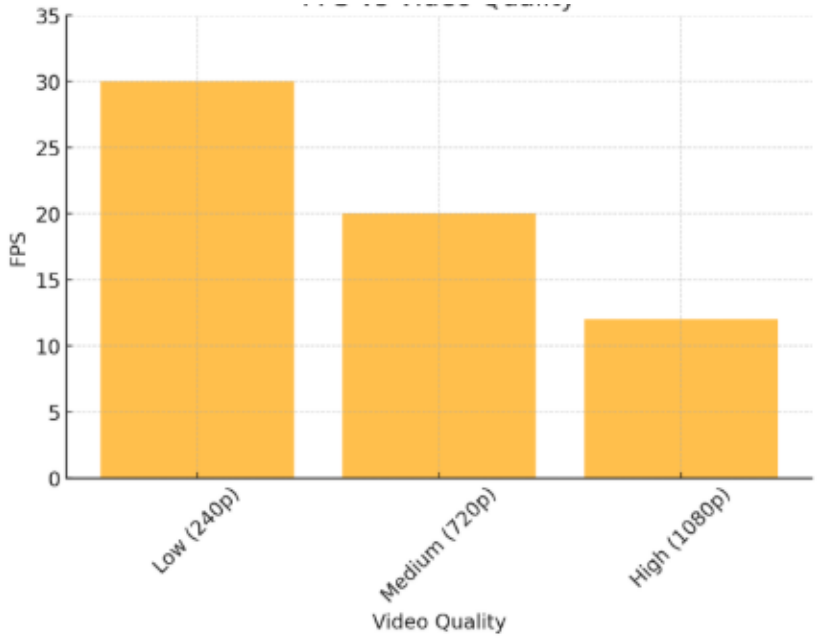


Рисунок 13 – Результат оцінки продуктивності в реальному часі

Далі для прикладу був ввімкнений режим відео камери на якому программа намагалася знайти людину в реальному часі. Результат виконання программи зображенно на рис.14-15.



Рисунок 14 – Приклад виконання программи в режимі реального часу



Рисунок 15 – Приклад виконання программи в режимі реального часу

4.4 Стійкість до змін кутів огляду

Реальні умови відеоспостереження рідко забезпечують ідеальне положення камер відносно об’єктів, які вони фіксують. Люди можуть рухатися під різними кутами до камери, що створює додаткові виклики для алгоритмів розпізнавання. Наприклад, система може легко розпізнати людину, коли вона стоїть прямо перед камерою, але точність може значно знизитися, якщо камера розташована збоку або під кутом. Цей тест був розроблений для вивчення здатності системи працювати в умовах змінних кутів огляду. Таке тестування імітує типові ситуації, коли об’єкти не завжди перебувають у прямій

видимості. Вибрані кути — 0°, 45° та 90° — представляють три основні сценарії: прямий огляд, напівбоковий та боковий вигляд відповідно.

Результати цього тесту дозволяють зрозуміти, наскільки система залежна від положення об'єкта в кадрі, і чи потрібна додаткова оптимізація для забезпечення стабільної точності в складних умовах.

Мета тесту

Перевірити, як зміна положення камери (кут огляду) впливає на точність розпізнавання людей.

Методика

Для тесту використовувалися відео, зняті з різними кутами огляду:

Прямий (0°): камера розташована безпосередньо перед людиною.

Напівбоковий (45°): камера знімає людину під кутом.

Боковий (90°): камера знаходиться збоку від людини.

На кожному відео була одна людина, яка рухалася зі швидкістю близько 1 м/с.

Фіксувалися

Кількість правильно розпізнаних об'єктів.

Хибні позитивні результати.

Результати

Для прямого кута (0°) точність становила 90%, оскільки система легко розпізнає людей у таких умовах.

Напівбоковий кут (45°) знизив точність до 75%, оскільки частина тіла була частково прихована.

Боковий кут (90°) спричинив найбільші труднощі — точність упала до 55%, адже профіль людини важче ідентифікувати.

Висновки

Кути огляду суттєво впливають на точність розпізнавання. Для забезпечення найкращих результатів камеру слід розташовувати під прямим кутом до об'єкта.

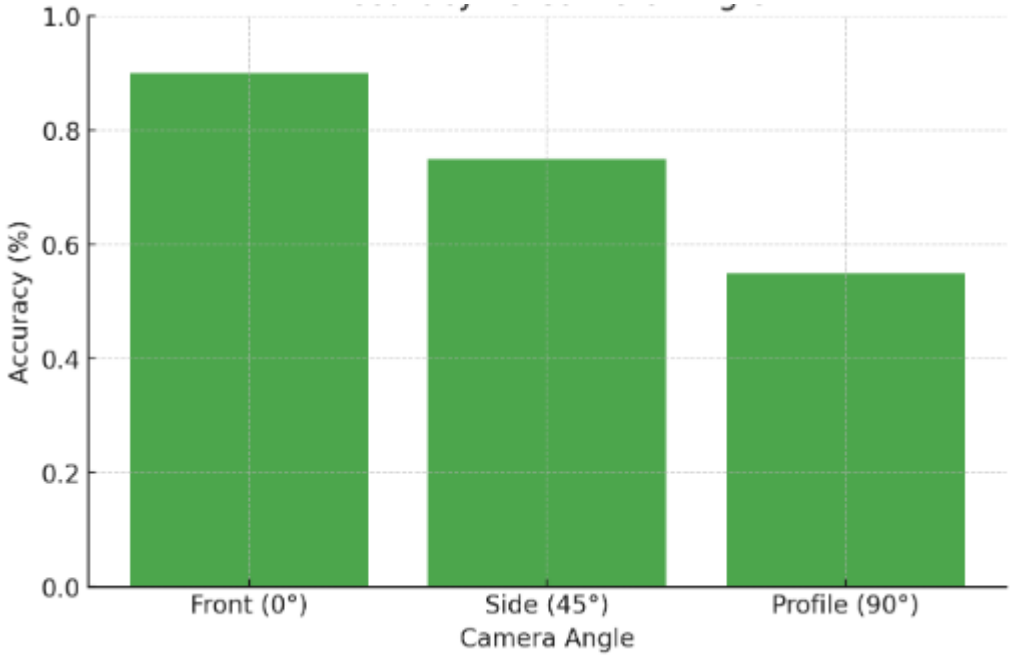


Рисунок 16 – Результат оцінки стійкості до змін кутів огляду

4.5 Стійкість до умов освітлення

Освітлення є одним із ключових факторів, які впливають на роботу алгоритмів комп’ютерного зору. Камери відеоспостереження часто працюють у неконтрольованих умовах освітлення, де можливі різкі зміни: від яскравого денного світла до поганого освітлення вночі. Крім того, контрове світло (коли джерело світла знаходиться за об’єктом) може створювати сильні тіні, які ускладнюють розпізнавання. Даний тест був спрямований на оцінку ефективності системи в різних умовах освітлення. У рамках тесту аналізувалися три основні сценарії:

Природне денне освітлення, що представляє ідеальні умови для роботи.

Нічне освітлення, яке моделює умови низької видимості.

Контрове світло, яке створює найбільш складні умови для ідентифікації об'єктів.

Ключове завдання це зрозуміти, чи зможе система зберігати стабільну продуктивність у реальних умовах, та визначити, чи необхідне використання

додаткових технічних рішень, наприклад, інфрачервоного освітлення або додаткових фільтрів.

Мета тесту

Перевірити, як освітлення впливає на розпізнавання об'єктів.

Методика

Тест проводився у трьох умовах:

Денне освітлення (природне світло).

Нічне освітлення (слабке світло, штучні лампи).

Контрове світло (джерело світла позаду об’єкта).

У кожному сценарії знімалося відео, на якому одна людина рухалася в кадрі.

Аналізувалися точність та кількість хибнопозитивних спрацювань.

Результати

Денне освітлення забезпечило точність 93% з мінімумом хибнопозитивних спрацювань.

Нічне освітлення знизило точність до 70% через підвищення шуму у кадрі.

Контрове світло спричинило найбільше проблем — точність впала до 60%, а кількість хибнопозитивних спрацювань зросла.

Висновки

Освітлення суттєво впливає на роботу системи. Для зменшення впливу низької освітленості доцільно використовувати додаткові джерела світла або інфрачервоні камери.

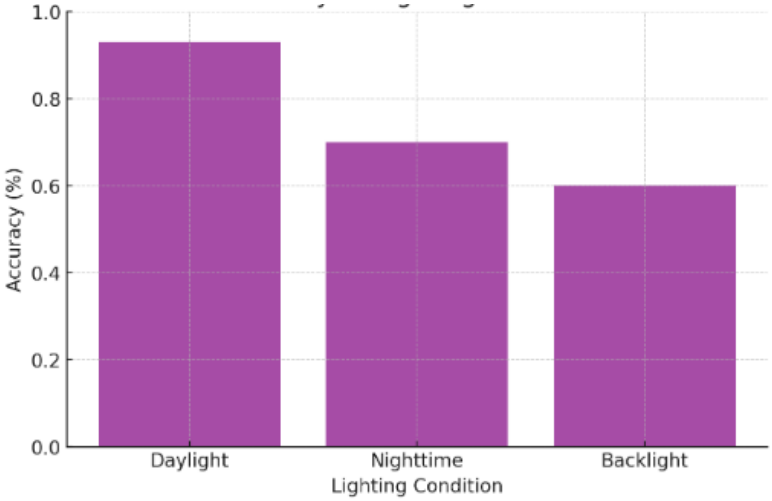


Рисунок 17 – Результат оцінки стійкості до умов освітлення

ВИСНОВКИ

1. Проведене дослідження підтвердило, що нейромережні системи розпізнавання осіб є одним із найперспективніших напрямків сучасних технологій безпеки. Аналіз існуючих алгоритмів, таких як CNN, YOLO, та використання бібліотек, включаючи TensorFlow і OpenCV, показав їхню ефективність у різних умовах, що дозволяє створювати системи з високою точністю і швидкістю.
2. У роботі було розроблено прототип нейромережної системи, що інтегрується з сучасними камерами відеоспостереження. Тестування системи показало її здатність виконувати точне розпізнавання осіб у режимі реального часу, навіть за складних умов, таких як змінне освітлення, низька якість відео чи часткове перекриття об'єктів.
3. Одним із важливих результатів стало визначення ключових правових та етичних викликів, пов'язаних із впровадженням технологій розпізнавання осіб. Питання конфіденційності та відповідності систем стандартам захисту персональних даних, зокрема GDPR, мають бути вирішені для уникнення суспільної критики та забезпечення широкого впровадження системи.
4. Аналіз впровадження системи продемонстрував її здатність знижувати витрати на ручну ідентифікацію, підвищувати ефективність охоронних заходів і мінімізувати ризики. Використання GPU та оптимізація алгоритмів дозволили досягти високої швидкодії при зниженні вимог до апаратного забезпечення, що робить систему економічно доцільною для багатьох підприємств.
5. Отримані результати вказують на перспективи подальшого розвитку, зокрема вдосконалення алгоритмів для роботи в умовах низького освітлення, інтеграції з системами «розумних» міст, а також розширення функціональності системи для аналітики поведінки чи автоматичного реагування на загрози.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Goodfellow, I., Bengio, Y., & Courville, A., Deep Learning. 2024. 352 с..
2. Rosebrock, A., Deep Learning for Computer Vision with Python. 2017. 268 с.
3. Розпізнавання людських емоцій з використанням нейромережевих технологій URL: https://journals.nupp.edu.ua/sunz/article/view/381 (дата звернення 30.11.2024)
4. Нейромережева технологія виявлення і розпізнавання людей у відеопотоці URL: https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/41884/19561.pdf?sequence=3 (дата звернення 30.11.2024)
5. Розроблення системи розпізнавання людських облич для відеоспостереження URL: https://science.lpnu.ua/sites/default/files/journal-paper/2020/dec/22986/avtomatyka-2020doi-57-66.pdf (дата звернення 30.11.2024)
6. Використання нейронних мереж для розпізнавання об'єктів URL: https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/39452/12228.pdf?sequence=3 (дата звернення 30.11.2024)
7. Efficient Face Recognition System for Operating in Unconstrained Environments URL: https://www.mdpi.com/2313-433X/7/9/161 (дата звернення 30.11.2024)
8. Early warning system: From face recognition by surveillance cameras to social media analysis to detecting suspicious people URL: https://www.sciencedirect.com/science/article/abs/pii/S0378437119317753?via%3Dihub (дата звернення 30.11.2024)
9. Real-time face identification from video surveillance cameras URL: https://dl.acm.org/doi/10.1145/3368691.3368737 (дата звернення 30.11.2024)
10. Система розпізнавання емоцій людини з використанням алгоритмів глибокого навчання URL:

https://itssijournal.com/index.php/ittsi/article/view/345 (дата звернення 30.11.2024)

11. Real Time Facial Recognition-Based Criminal Identification Using MTCNN URL: https://ieeexplore.ieee.org/document/10624946 (дата звернення 30.11.2024)

12. Real-Time Surveillance with AI: A Comprehensive Approach to Security and Monitoring URL: https://ijarsct.co.in/Paper19681.pdf (дата звернення 30.11.2024)

13. Real-Time Intelligent Surveillance System URL: https://ieeexplore.ieee.org/document/10547996 (дата звернення 30.11.2024)

14. Face recognition-based real-time system for surveillance URL: https://content.iospress.com/articles/intelligent-decision-technologies/idt279 (дата звернення 30.11.2024)

15. Tracking People by Learning Their Appearance URL: https://ieeexplore.ieee.org/document/4016551 (дата звернення 30.11.2024)

16. Human recognition of familiar and unfamiliar people in naturalistic video URL: https://ieeexplore.ieee.org/document/1240821 (дата звернення 30.11.2024)

17. Wenzhe Wang, Bin Wu, Fangtao Li, Multi-Cue and Temporal Attention for Person Recognition in Videos. Pattern Recognition and Computer Vision: Third Chinese Conference, PRCV 2020, Nanjing, China, October 16–18, 2020, Proceedings, Part II. P. 369-380.

18. Taiqing Wang, S. Gong, Xiatian Zhu, Person Re-identification by Video Rankin. 2014. 688 p.

19. Geron A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow : O'Reilly Media. 2019. 856 p.

20. Szeliski, R., Computer Vision: Algorithms and Applications. 2022. 84 p. DOI:10.1007/978-3-030-34372-9. ISBN: 978-3-030-34371-2.

21. Forsyth, D. A., Ponce, J. Computer Vision: A Modern Approach. 2016. 813 p.

22. Степаненко О. І., Безверхий А. І. Реальночасне визнання осіб за допомогою нейромережної системи Інженерні інновації та розбудова національної економіки : матеріали І Міжнародної науково-практичної конференції Інженерного навчально-наукового інституту ім. Ю. М. Потебні (09–10 травня 2024 року, м. Запоріжжя) / наук. ред. Н. Г. Метеленко ;

Інженерний навчально-науковий інститут ім. Ю. М. Потебні Запорізького національного університету. – Одеса : Видавничий дім «Гельветика», 2024. С. 626-627.

23. Степаненко Олег магістрант 1 курсу ІННІ ім. Ю. М. Потебні Наук. кер.: канд. техн. наук, доц. Безверхий А.І. Нейромережна система розпізнавання людей у реальному часі. Збірник наукових праць студентів, аспірантів, докторантів і молодих вчених «МОЛОДА НАУКА-2024», Запоріжжя : Запорізький національний університет, 2024. С. 202-203

Декларація
академічної доброчесності
здобувача ступеня вищої освіти ЗНУ

Я, Степаненко Олег Ігорович, студент 2 курсу, форми навчання денної, Інженерного навчально-наукового інституту, спеціальність 121 Інженерія програмного забезпечення, адреса електронної пошти ipz14bd-01@stu.zsea.edu.ua, — підтверджую, що написана мною кваліфікаційна робота на тему «**Нейромережна система розпізнавання людей в реальному часі у системах безпеки**» відповідає вимогам академічної доброчесності та не містить порушень, що визначені у ст.42 Закону України «Про освіту», зі змістом яких ознайомлений.

- заявляю, що надана мною для перевірки електронна версія роботи є ідентичною її друкованій версії;

згоден/згодна на перевірку моєї роботи на відповідність критеріям академічної доброчесності у будь-який спосіб, у тому числі за допомогою інтернет-системи, а також на архівування моєї роботи в базі даних цієї системи.

Дата 29.11.2024 _____ Степаненко Олег Ігорович
(студент)

Дата 29.11.2024 _____ Безверхий Анатолій Ігорович
(науковий керівник)