

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
Факультет комп'ютерно-інтегрованих технологій, мехатроніки і
робототехніки
Кафедра робототехніки, електроенергетики та автоматизації
ім. проф. Б.Б. Самотокіна

ПОЯСНЮВАЛЬНА ЗАПИСКА
до кваліфікаційної роботи магістра
на тему: **«Автоматизована системи контролю безпеки з використанням**
нейронних мереж»

Виконав студент 2 курсу, групи АТ-28м
спеціальності 174 “ Автоматизація,
комп'ютерно-інтегровані технології
та робототехніка”

Огар Вадим Васильович

(прізвище та ініціали студента)

Керівник: д.т.н., проф. Безвесільна О.М.

(науковий ступінь, вчене звання, посада,
прізвище та ініціали)

Рецензент: к.т.н., доц. Чепюк Л.О.

(науковий ступінь, вчене звання, посада,
прізвище та ініціали)

Житомир – 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ “ЖИТОМИРСЬКА ПОЛІТЕХНІКА”
Факультет комп’ютерно-інтегрованих технологій, мехатроніки і
робототехніки
Кафедра робототехніки, електроенергетики та автоматизації
ім. проф. Б.Б. Самотокіна

Спеціальність 174 “Автоматизація, комп’ютерно-інтегровані технології
та робототехніка”

ЗАТВЕРДЖУЮ
Завідувач кафедри
робототехніки, електроенергетики
та автоматизації
ім. проф. Б.Б. Самотокіна
к.т.н., доц. Громовий О.А.
_____ к.т.н., доц. Громовий О.А.
« _____ » _____ 2024р.

ЗАВДАННЯ

на кваліфікаційну роботу магістра

Студент Огар Вадим Васильович

1. Тема роботи: Автоматизована система контролю безпеки доступу людей до будівлі або кімнати на основі алгоритмів комп’ютерного зору та нейронних мереж

Затверджена Наказом по університету від «12» листопада 2024 р. № 646/с

2. Термін здачі закінченої роботи на кафедру «18» грудня 2024 р.

3. Вихідні дані до роботи опис загальної роботи, огляд систем аналогів, огляд обладнання, одно-платні комп’ютери, камери

4. Зміст пояснювальної записки (перелік питань, які підлягають розробці)
У роботі буде проведено аналіз сучасного стану систем контролю безпеки, методів моніторингу та огляд існуючих систем. Здійснено підбір обладнання для реалізації системи. Розроблено алгоритми роботи та структурні схеми. Створено власну базу даних і написано програмне забезпечення для реалізації функціоналу. Проведено аналіз результатів виконання програми

5. Перелік презентативного матеріалу (з точним визначенням обов’язкових слайдів/креслень) технічне завдання, огляд та аналіз існуючих систем, розробка структурної схеми та алгоритму роботи

6. Консультанти з кваліфікаційної роботи магістра із зазначенням розділів роботи, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1, 2, 3, 4	Безвесільна О.М., д.т.н., професор	11.09.2024	11.09.2024

7. Дата видачі завдання 11.09.2024

Календарний план

№ з/п	Назва етапів виконання кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1.	Підготувати звіт про переддипломну практику	1.11.2024 – 1.12.2024	виконано
2.	Аналіз сучасного стану систем контролю безпеки	20.09.2024 – 20.10.2024	виконано
3.	Підбір обладнання	21.10.2024 – 30.10.2024	виконано
4.	Створення алгоритмів роботи та структурної схем	1.11.2024 – 02.12.2024	виконано
5.	Розробка власної бази даних	02.12.2024 – 10.12.2024	виконано
6.	Написання програмного забезпечення	11.12.2024 – 15.12.2024	виконано
7.	Аналіз результату виконання програми	15.12.2024 – 18.12.2024	виконано

Студент _____
(підпис, дата)

В.В. Огар
(прізвище та ініціали)

Керівник роботи _____
(підпис, дата)

О.М. Безвесільна
(прізвище та ініціали)

АНОТАЦІЯ

Магістерська робота присвячена розробці автоматизованої системи контролю безпеки, яка базується на використанні штучних нейронних мереж для виявлення потенційних загроз у режимі реального часу. У роботі проведено аналіз сучасних методів відеоаналітики, розглянуто архітектури згорткових нейронних мереж (CNN), їх застосування для розпізнавання об'єктів та поведінкових аномалій у системах відеоспостереження.

Запропоновано структуру системи, що включає модулі збору даних, попередньої обробки, інтелектуального аналізу та сигналізації. Розроблено алгоритм функціонування системи на основі навченої нейромережевої моделі, проведено моделювання та тестування точності розпізнавання ситуацій загрози. Отримані результати підтверджують ефективність запропонованого підходу для підвищення рівня безпеки об'єктів.

Дипломна робота містить 57 сторінок, 15 рисунків, 3 таблиці та 14 використаних джерел та додатку.

THE ABSTRACT

The master's thesis is devoted to the development of an automated security control system based on the use of artificial neural networks for detecting potential threats in real time. The study analyzes modern video analytics methods and examines convolutional neural network (CNN) architectures and their application in object recognition and behavioral anomaly detection in video surveillance systems.

A system structure is proposed that includes modules for data collection, preprocessing, intelligent analysis, and alerting. An algorithm for the system's operation based on a trained neural network model has been developed, along with modeling and testing of threat detection accuracy. The obtained results confirm the effectiveness of the proposed approach in enhancing the security level of facilities.

The thesis consists of 57 pages, 15 figures, 3 tables, 14 references, and an appendix.

РЕФЕРАТ

В кваліфікаційній роботі магістра було створено автоматизовану контролю безпеки доступу людей до будівлі або кімнати на основі алгоритмів комп'ютерного зору та нейронних мереж.

В розділі «Аналіз сучасного рівня систем контролю безпеки» розглянуто особливості різних систем контролю безпеки та зроблено огляд сучасних алгоритмів та методів в даних системах.

В розділі «Загальні положення технологій комп'ютерного зору» розглянуто детально алгоритми на основі машинного навчання та глибинного навчання. Також було виконано огляд різних архітектур глибинного навчання, після чого було проведено аналіз алгоритмів на основі нейронних мереж. Також була поставлена мета та задачі дипломного проєктування.

В розділі «Алгоритмічне забезпечення систем контролю безпеки з використанням нейронних мереж» було виконано підбір апаратної бази, після чого було виконано розробку алгоритму та структурної схем системи контролю безпеки.

В розділі «Програмне забезпечення» наведено кроки створення власної моделі на основі відібраного набору даних. Також реалізовано перевірку роботи моделі та наведено можливу інтеграцію моделі.

ЗМІСТ

Вступ.....	7
Розділ 1. АНАЛІЗ СУЧАСНОГО РІВНЯ СИСТЕМ КОНТРОЛЮ БЕЗПЕКИ	8
1.1. Особливості систем контролю безпеки.....	8
1.2. Сучасні алгоритми та методи в системах контролю безпеки	9
РОЗДІЛ 2. ЗАГАЛЬНІ ПОЛОЖЕННЯ ТЕХНОЛОГІЙ КОМП'ЮТЕРНОГО ЗОРУ	14
2.1. Алгоритми на основі машинного навчання	14
2.2. Глибинне навчання та архітектури	19
2.3. Алгоритми на основі нейронних мереж	26
2.4. Мета та задачі дипломного проєкту	32
Розділ 3. АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМ КОНТРОЛЮ БЕЗПЕКИ З ВИКОРИСТАННЯМ НЕЙРОННИХ МЕРЕЖ.....	34
3.1. Вибір апаратної бази	34
3.2. Алгоритм та структурні схеми системи контролю безпеки.....	41
Розділ 4. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.....	44
4.1. Створення власного набору даних	44
4.2. Результати виконання програми	48
ВИСНОВКИ	51
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	52
ДОДАТКИ	55

ВСТУП

На сьогодні питання безпеки та контролю доступу до приміщень є надзвичайно важливими як для державних установ, так і для приватних організацій. Надійна система контролю доступу дозволяє забезпечити належний рівень безпеки, обмеживши несанкціонований вхід сторонніх осіб. Традиційні методи, такі як використання охоронців або пропускних систем на основі карток чи біометричних даних, мають певні недоліки: високі експлуатаційні витрати, людський фактор та можливість обману систем.

Одним із перспективних підходів до вирішення даної проблеми є розробка автоматизованих систем контролю доступу на основі комп'ютерного зору та машинного навчання. Такі системи здатні в режимі реального часу здійснювати ідентифікацію людей за візуальними ознаками з використанням цифрових камер та алгоритмів розпізнавання образів. Завдяки застосуванню нейронних мереж та методів глибинного навчання можливо досягти високої точності автоматизованого розпізнавання обличч, виявлення підозрілої поведінки чи виявлення інших небажаних ситуацій.

Актуальність даної проблематики зумовлена необхідністю підвищення рівня безпеки при одночасному зниженні експлуатаційних витрат та людського фактору. Аналіз останніх досліджень у цій галузі демонструє наявність низки успішних розробок, однак поряд з цим існують відкриті питання щодо забезпечення надійності, швидкодії та масштабованості подібних систем.

Відтак, розробка та всебічне експериментальне дослідження ефективних автоматизованих систем контролю доступу на основі комп'ютерного зору, глибинного навчання та нейронних мереж є актуальною науково-прикладною задачею. Успішне вирішення даної проблеми дозволить підвищити рівень безпеки та оптимізувати витрати на системи контролю доступу.

Розділ 1. АНАЛІЗ СУЧАСНОГО РІВНЯ СИСТЕМ КОНТРОЛЮ БЕЗПЕКИ

1.1. Особливості систем контролю безпеки

Системи контролю безпеки відіграють вирішальну роль у забезпеченні захисту життя та майна людей, а також у підтримці стабільного функціонування різноманітних об'єктів та організацій. Ці системи розробляються з метою своєчасного виявлення потенційних загроз, запобігання інцидентам та мінімізації можливих наслідків у разі їх виникнення.

Забезпечення контролю безпеки можна розділити на:

- Контроль за допомогою фізичних елементів або людей;
- Цифрові засоби контролю безпеки [1].

Контроль за допомогою фізичних елементів або людей:

Дані методи контролю безпеки використовують різні елементи та предмети, що захищають фізичне середовище і запобігають несанкціонованому доступу до об'єктів. Загалом контроль за допомогою фізичних елементів або людей поділяють на:

1. Контроль доступу:

- Огорожа периметру – встановлення кордону навколо об'єктів для запобігання несанкціонованому проникненню;
- Замки та ключі – захищають двері, ворота та шафи;
- Картки контролю доступу та біометричні дані – обмежують доступ до приміщення тільки для уповноваженого персоналу.

2. Системи відеоспостереження:

- Камери спостереження – відстежують і записують дії в приміщенні та навколо нього;

- Датчики виявлення вторгнення – вмикають тривогу при виявленні несанкціонованого доступу.

3. Персонал охорони:

- Охоронці – фізична присутність для відлякування зловмисників та реагування на інциденти;
- Політика та процедури безпеки – визначають ролі, обов’язки та протоколи дій у надзвичайних ситуаціях.

4. Управління відвідувачами:

- Журнали відвідувачів – документується час входу та виходу відвідувачів;
- Політика супроводу – відвідувачі у супроводі уповноваженого персоналу.

Цифрові засоби контролю безпеки:

Охоплюють такі механізми, як імена користувачів і паролі, розумні системи і т.д., наприклад:

1. Системи відеоспостереження (CCTV) – цифрові камери, які записують відео та зображення, є важливими засобами контролю за фізичною безпекою в приміщеннях та на вулицях.
2. Системи контролю доступу – вони використовуються для обмеження доступу до певних зон або ресурсів. Це може бути біометричний сканер, карткові системи або кодові замки.

1.2. Сучасні алгоритми та методи в системах контролю безпеки

Забезпечення безпеки доступу до приміщень є однією з ключових задач в сучасних умовах, особливо в установах, де потрібно контролювати доступ до обмежених зон. Для досягнення цієї мети використовуються різноманітні алгоритми та методи, які поєднують в собі технології

біометричної ідентифікації, електронних систем керування доступом, а також методи аналізу та моніторингу.

Перед подальшим розглядом систем контролю доступу важливо розуміти ключові терміни: ідентифікація, аутентифікація та авторизація. Ці терміни використовуються в контексті забезпечення безпеки та контролю доступу до ресурсів, і кожен з них відображає певний аспект процесу ідентифікації користувача [2].

Ідентифікація – це перший етап в процесі розпізнавання особи або сутності. Ідентифікація полягає в наданні унікального ідентифікатора або набору даних, які можуть ідентифікувати користувача. Цей ідентифікатор може бути ім'ям користувача, номером ідентифікації (наприклад, логін або електронна пошта), біометричними даними (такими як відбитки пальців або обличчя), або комбінацією цих факторів.

Аутентифікація – це процес перевірки ідентифікованої особи або сутності для підтвердження їхньої ідентичності. Аутентифікація включає перевірку того, що особа дійсно є тим, ким вона стверджує себе бути. Для цього зазвичай використовуються паролі, підписи, біометричні сканування або інші методи ідентифікації, які є унікальними для користувача.

Авторизація – це останній етап в процесі контролю доступу, який визначає, чи має допущена особа (після аутентифікації) доступ до певних ресурсів або функціональності. Авторизація визначає, які дії або ресурси можуть бути доступні конкретному користувачеві після успішної аутентифікації. Наприклад, після введення правильного пароля (аутентифікація), користувач може бути авторизований для перегляду конфіденційної інформації або виконання певних операцій.



Рис.1.1. Схема різниці між поняттями

Біометрична ідентифікація

Один з найбільш надійних методів ідентифікації особи – це біометрична аутентифікація (рис.1.2.). Серед найпоширеніших технологій біометричної ідентифікації в системах контролю доступу є сканування відбитків пальців, розпізнавання обличчя, сканування радужки ока та голосова ідентифікація. Алгоритми цих технологій базуються на унікальних характеристиках людини, що забезпечує високий рівень точності та надійності.



Рис.1.2. Біометрична ідентифікація

Електронні системи керування доступом

Електронні системи керування доступом використовуються для управління правами доступу до приміщень за допомогою електронних ключів, карток, мобільних додатків або інших електронних пристроїв. Ці системи включають електронні замки, читачі карток, контролери доступу та програмне забезпечення для управління правами доступу. Алгоритми в цих системах забезпечують автоматичне відкриття та закриття дверей, а також контроль над часом та місцем доступу користувачів.



Рис.1.3. Приклади електронних систем керування доступом

Методи аналізу та моніторингу

Системи контролю доступу також включають методи аналізу та моніторингу для виявлення ненормальних або підозрілих активностей. Ці методи базуються на аналізі даних з електронних систем керування доступом, відеоспостереження та інших джерел. Вони дозволяють виявляти спроби несанкціонованого доступу, намагання обходу систем безпеки та інші загрози безпеці.

Інтеграція з іншими системами безпеки

Сучасні системи контролю доступу можуть інтегруватися з іншими системами безпеки, такими як системи відеоспостереження, системи виявлення вторгнень та системи автоматичного пожежогасіння. Це дозволяє створювати комплексні системи безпеки, які забезпечують високий рівень захисту приміщень та їхніх мешканців.

Використання штучного інтелекту та аналітики даних

Штучний інтелект та аналітика даних можуть бути використані для покращення систем контролю доступу. Алгоритми машинного навчання можуть використовуватися для аналізу поведінки користувачів та виявлення відхилень від звичайного шаблону. Аналітика даних дозволяє здійснювати прогнозування ризиків та вчасно реагувати на потенційні загрози.



Рис.1.4. Приклад системи контролю безпеки на основі AI

Розділ 2. ЗАГАЛЬНІ ПОЛОЖЕННЯ ТЕХНОЛОГІЙ КОМП'ЮТЕРНОГО ЗОРУ

2.1. Алгоритми на основі машинного навчання

Машинне навчання є широким терміном, який охоплює різноманітні алгоритми та методи, що дозволяють системам автоматично вчитися та покращувати свою продуктивність на основі досвіду, не будучи явно запрограмованими для вирішення конкретного завдання. Основна ідея полягає в тому, що замість написання явних правил та інструкцій, система аналізує великі обсяги даних, виявляє закономірності та тенденції, і будує власну модель для прогнозування або прийняття рішень.

Категорії машинного навчання:

- Навчання з учителем (Supervised Learning);
- Навчання без учителя (Unsupervised Learning);
- Навчання з підкріпленням (Reinforcement Learning) [3].

Навчання з учителем (Supervised Learning)

Навчання з учителем – це тип машинного навчання, в якому система навчається на позначених даних, де вхідні дані (ознаки) і відповідні вихідні дані (мітки або цільові значення) відомі. Основна мета полягає в тому, щоб побудувати модель, яка може прогнозувати вихідні дані для нових вхідних даних. В цьому контексті дані, на яких навчається модель, називаються навчальним набором.

Процес навчання з учителем включає наступні кроки:

1. Підготовка даних – обробка та підготовка навчальних даних, така як очищення даних, видалення аномалій, кодування категоріальних ознак та інші операції для покращення якості даних.

2. Вибір моделі – вибір алгоритму чи моделі, яка буде використовуватися для навчання на даних. Це може бути лінійна регресія, дерево рішень, випадковий ліс, нейронні мережі та інші.

3. Навчання моделі – застосування вибраної моделі до навчального набору даних з метою навчання. Модель адаптується до даних, використовуючи методи оптимізації для мінімізації помилки прогнозування.

4. Оцінка моделі – оцінка ефективності моделі на навчальному наборі даних за допомогою метрик якості, таких як точність, відхилення, чутливість тощо. Це дозволяє зрозуміти, наскільки добре модель працює на навчальних даних.

5. Перевірка моделі – перевірка ефективності моделі на тестовому наборі даних, який модель раніше не бачила. Це дозволяє оцінити загальну загальну здатність моделі до узагальнення на нових даних.

Навчання з учителем використовується для задач, таких як класифікація (прогнозування категорій) та регресія (прогнозування числових значень). Наприклад, ви можете використовувати навчання з учителем для створення моделі, яка класифікує електронні листи як спам або не спам, або для прогнозування ціни на нерухомість на основі різних факторів.

Навчання без учителя (Unsupervised Learning)

Навчання без учителя - це тип машинного навчання, в якому система аналізує та виявляє структури та закономірності в не позначених даних без відомих правильних відповідей. Основна мета полягає в тому, щоб виявити приховані закономірності та структури у наборі даних, такі як кластери, асоціації та відмінності.

Процес навчання без учителя включає наступні кроки:

1. Підготовка даних – обробка та підготовка даних для аналізу, така як масштабування ознак, видалення відмінностей, відсіювання непотрібних ознак тощо.
2. Вибір методу – вибір алгоритму чи методу для аналізу даних, який найкраще відповідає конкретній задачі. Це може бути алгоритм кластеризації, знаходження асоціативних правил, виявлення відмінностей тощо.
3. Аналіз даних – застосування вибраного методу до даних для виявлення структур та закономірностей. Це може включати розділення даних на кластери, виявлення асоціацій між ознаками, виявлення аномалій тощо.
4. Оцінка результатів – оцінка та інтерпретація отриманих результатів аналізу даних. Це дозволяє зрозуміти, які структури або закономірності були виявлені та як вони можуть бути використані для подальшого аналізу або прийняття рішень.
5. Використання результатів – використання виявлених структур та закономірностей для вирішення конкретної задачі або для вдосконалення процесів в організації. Наприклад, ви можете використовувати результати кластеризації для сегментації аудиторії вашого продукту або послуги.

Навчання без учителя широко застосовується для виявлення складних взаємозв'язків та шаблонів у даних, де важко або неможливо визначити правильні відповіді.

Навчання з підкріпленням (Reinforcement Learning)

Навчання з підкріпленням – це тип машинного навчання, в якому система навчається шляхом взаємодії з середовищем і отримання винагород або покарань за свої дії. Основна ідея полягає в тому, щоб максимізувати кумулятивну винагороду, яку система отримує за свої дії в середовищі.

Процес навчання з підкріпленням включає наступні кроки:

1. Взаємодія з середовищем – система взаємодіє з визначеним середовищем, виконуючи дії та спостерігаючи наслідки цих дій.
2. Отримання винагороди – система отримує винагороду або покарання в залежності від результату своїх дій. Мета полягає в тому, щоб максимізувати кумулятивну винагороду протягом тривалого часу.
3. Вивчення стратегії – система вивчає оптимальну стратегію, яка дозволяє досягти максимальної винагороди в середовищі. Це може включати вивчення оптимальних дій для різних станів середовища та адаптацію стратегії на основі отриманих даних.
4. Адаптація до змін – система адаптується до змін в середовищі та вдосконалює свою стратегію на основі нових навчальних даних та отриманої винагороди.

Навчання з підкріпленням широко застосовується в робототехніці, автоматизованих системах, іграх та інших сценаріях, де агент повинен приймати рішення в реальному часі на основі отриманої відповіді на свої дії.

Список популярних алгоритмів машинного навчання:

1. Лінійна регресія

Лінійна регресія — це простий алгоритм, який використовується для відображення лінійного зв'язку між вхідними характеристиками та постійною цільовою змінною. Він працює, підбираючи лінію до даних, а потім використовуючи лінію для прогнозування нових значень.

2. Логістична регресія

Логістична регресія є розширенням лінійної регресії, яка використовується для завдань класифікації для оцінки ймовірності того, що примірник належить до певного класу.

3. SVM (машина опорних векторів)

SVM — це керовані алгоритми навчання, які можуть виконувати завдання класифікації та регресії. Він знаходить гіперплощину, яка найкраще розділяє класи в просторі ознак.

4. KNN (K-найближчий сусід)

KNN — це непараметричний метод, який можна використовувати як для класифікації, так і для регресії. Він працює шляхом ідентифікації k найбільш подібних точок даних до нової точки даних, а потім прогнозування мітки нової точки даних за допомогою міток цих точок даних.

5. Дерево рішень

Дерева рішень — це тип методики навчання під наглядом, який можна використовувати як для класифікації, так і для регресії. Він працює, сегментуючи дані на все менші групи, поки кожен групу не можна буде класифікувати або передбачити з високим ступенем точності.

6. Random Forest

Random Forest — це тип методу ансамблевого навчання, який використовує набір дерев рішень для прогнозування шляхом агрегування прогнозів з окремих дерев. Це покращує точність і стійкість окремих дерев рішень. Його можна використовувати як для завдань класифікації, так і для регресії.

7. Naïve Bayes

Naïve Bayes — це ймовірнісний класифікатор, заснований на теоремі Байеса, який використовується для класифікаційних завдань. Він працює, припускаючи, що характеристики точки даних не залежать одна від одної.

8. PCA (аналіз основних компонентів)

РСА — це техніка зменшення розмірності, яка використовується для перетворення даних у простір меншої розмірності, зберігаючи при цьому якомога більше дисперсії. Він працює, знаходячи напрямки в даних, які містять найбільше варіацій, а потім проектує дані на ці напрямки.

9. Априорні алгоритми

Алгоритм Аpriori — це традиційний метод інтелектуального аналізу даних для аналізу правил асоціації в транзакційних базах даних або наборах даних. Він призначений для виявлення зв'язків і закономірностей між речами, які регулярно зустрічаються в транзакціях. Аpriori виявляє часті набори елементів, які є групами елементів, які з'являються разом у транзакціях із заданим мінімальним рівнем підтримки.

10. Кластеризація K-Means

Кластеризація K-Means — це підхід до неконтрольованого навчання, який можна використовувати для групування точок даних. Він працює, знаходячи k кластерів у даних, щоб точки даних у кожному кластері були якомога схожішими одна на одну, залишаючись якомога відміннішими від точок даних в інших кластерах.

2.2. Глибинне навчання та архітектури

Глибинне навчання є підгалуззю машинного навчання, яке зосереджується на використанні штучних нейронних мереж, натхнених біологічною структурою людського мозку. Ці мережі складаються з багатьох шарів взаємопов'язаних вузлів (нейронів), які здатні навчатися на вхідних даних та виявляти складні закономірності та абстракції [4].

Основною відмінністю глибинного навчання від традиційного машинного навчання є здатність автоматично вилучати та навчатися на

ієрархічних ознаках з сирих даних. Наприклад, у задачі розпізнавання зображень традиційні алгоритми машинного навчання потребують ретельно розроблених ознак, таких як контури, текстури та кольори, які повинні бути вручну визначені експертами. З іншого боку, глибокі нейронні мережі можуть автоматично навчитися виявляти ці ознаки з сирих пікселів зображення.

Популярними архітектурами глибоких нейронних мереж є згорткові нейронні мережі (CNN), рекурентні нейронні мережі (RNN), довгі короткочасні пам'яті (LSTM), мережі з увагою та генеративно-змагальні мережі (GAN). Ці архітектури успішно застосовуються в багатьох галузях, таких як комп'ютерний зір, обробка природної мови, розпізнавання мовлення, рекомендаційні системи та багато інших.

Згорткові нейронні мережі (CNN)

Згорткові нейронні мережі (Convolutional Neural Networks, CNN) є спеціалізованим видом глибоких нейронних мереж, які широко використовуються в задачах комп'ютерного зору та обробки зображень. Вони натхненні біологічною структурою зорової кори тварин і людей [5].

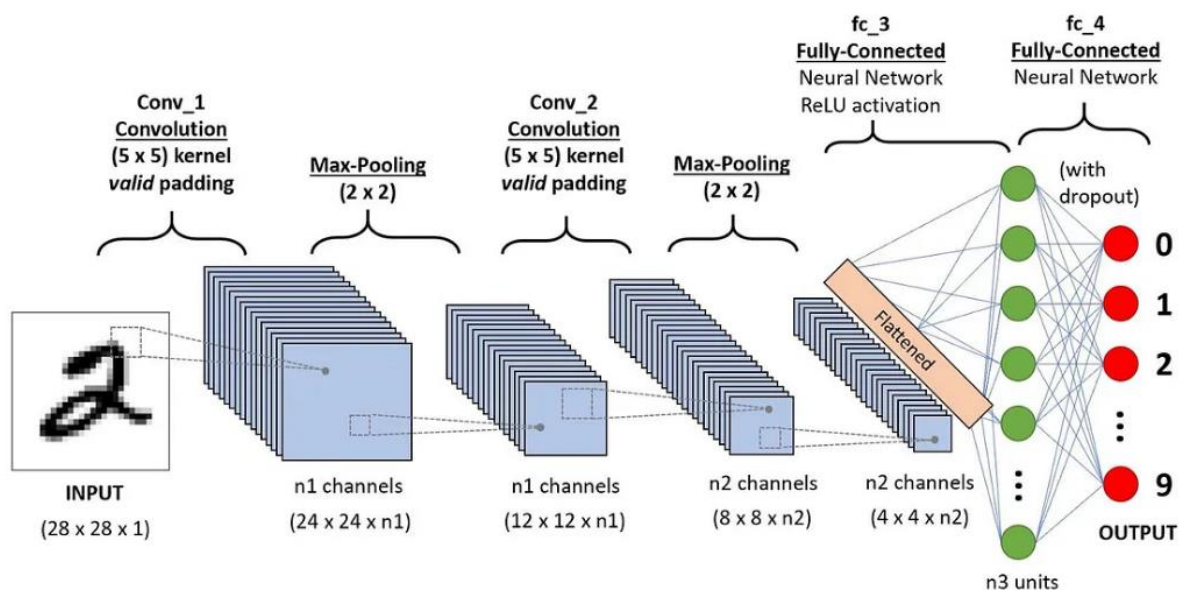


Рис.2.2. CNN архітектура

CNN складаються з кількох шарів, кожен з яких виконує специфічні операції на вхідних даних. Основними компонентами CNN є:

1. Згорткові шари (Convolutional Layers) – ці шари застосовують операцію згортки до вхідних даних (зазвичай зображень) за допомогою набору ядер (фільтрів). Ядра рухаються по всьому зображенню, виконуючи елементарні операції (наприклад, виявлення країв або текстур) та створюючи карти ознак.

2. Пулінгові шари (Pooling Layers) – ці шари виконують операції підвибірки або агрегації даних з карт ознак, створених згортковими шарами. Це допомагає зменшити просторові розміри даних і робить мережу більш стійкою до невеликих зсувів та спотворень.

3. Повнозв'язні шари (Fully Connected Layers) – наприкінці CNN знаходяться повнозв'язні шари, подібні до традиційних нейронних мереж, які об'єднують інформацію з попередніх шарів для класифікації або регресії.

CNN добре підходять для завдань комп'ютерного зору, оскільки вони можуть ефективно виявляти складні візуальні ознаки, такі як контури, текстури та форми, з сирих пікселів зображень. Вони також інваріантні до зсувів, масштабувань та деяких спотворень зображень.

Рекурентні нейронні мережі (RNN)

Рекурентні нейронні мережі (Recurrent Neural Networks, RNN) є спеціальним видом нейронних мереж, розроблених для обробки послідовностей даних, таких як текст, аудіо або часові ряди. На відміну від традиційних нейронних мереж, які обробляють вхідні дані одноразово, RNN можуть зберігати внутрішній стан, що дозволяє їм обробляти послідовності даних крок за кроком.

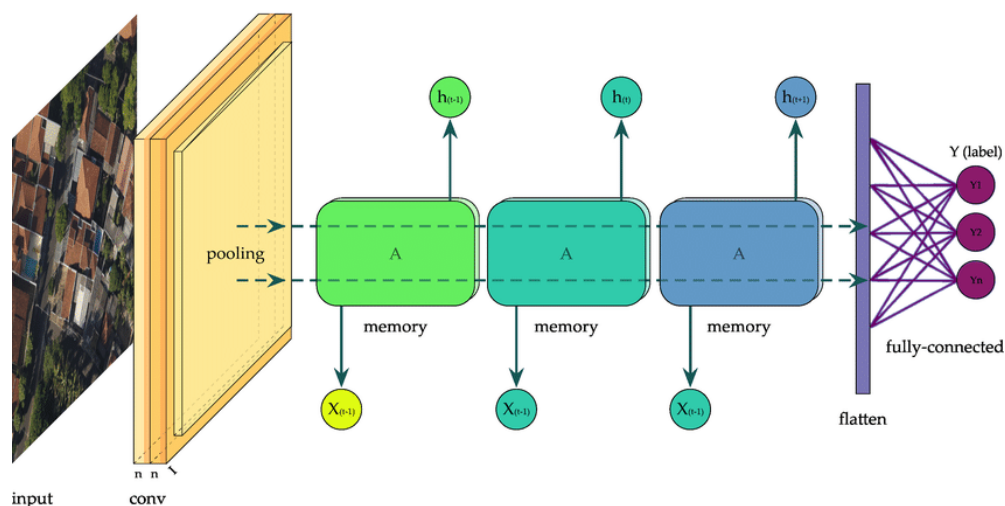


Рис. 2.3. RNN архітектура

RNN мають циклічну структуру, де вихід на певному кроці залежить не тільки від поточного вхідного сигналу, але й від попереднього стану мережі. Це дозволяє мережі «запам'ятовувати» попередній контекст, що особливо корисно для завдань, де важливо зберігати контекст, наприклад, в обробці природної мови або розпізнаванні мовлення.

Однак класичні RNN страждають від проблеми «зникаючого/вибухового градієнта», що ускладнює ефективне навчання на довгих послідовностях даних. Для подолання цієї проблеми були розроблені спеціалізовані архітектури, такі як LSTM (Long Short-Term Memory) та GRU (Gated Recurrent Unit).

Довгі короткочасні пам'яті (LSTM)

Довгі короткочасні пам'яті (Long Short-Term Memory, LSTM) є особливим видом рекурентних нейронних мереж, розробленим для ефективного навчання на довгих послідовностях даних. LSTM мають спеціальну структуру, яка дозволяє їм зберігати та регулювати інформацію протягом тривалого часу, запобігаючи проблемі зникаючого/вибухового градієнта.

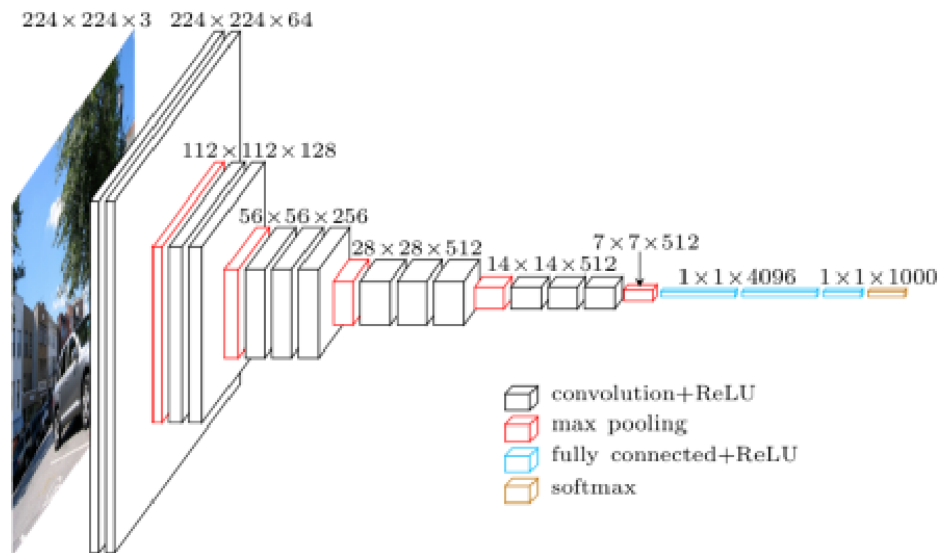


Рис. 2.4. LSTM архітектура

Основними компонентами LSTM є:

1. Forget Gate – визначає, яку інформацію з попереднього стану потрібно зберегти або забути.
2. Input Gate – вирішує, яку нову інформацію з поточного вхідного сигналу та попередньої пам'яті потрібно додати до стану комірки.
3. Cell State – акумулює та передає релевантну інформацію через послідовність.
4. Output Gate – вирішує, яку частину оновленого стану комірки потрібно використати як вихід LSTM [6].

За допомогою цих компонентів LSTM можуть ефективно навчатися на довгих послідовностях даних, зберігаючи релевантну інформацію протягом тривалого часу та ігноруючи незначущі деталі. Це робить їх корисними для завдань, таких як машинний переклад, розпізнавання мовлення та обробка природної мови.

Мережі з увагою (Attention Networks)

Мережі з увагою (Attention Networks) є розширенням рекурентних нейронних мереж, розробленим для покращення їхньої здатності

зосереджуватися на найбільш релевантних частинах вхідних даних. Замість рівномірного розподілу уваги на всіх елементах послідовності, мережі з увагою динамічно присвоюють більшу вагу (або увагу) тим елементам, які є найбільш важливими для поточного завдання.

Механізм «уваги» складається з двох основних компонентів:

1. Attention Function – ця функція обчислює вагові коефіцієнти (скори уваги) для кожного елемента вхідної послідовності, визначаючи їхню релевантність для поточного кроку обробки.

2.Context Vector – це зважена сума вхідних векторів, де вагами є скори уваги. Контекстний вектор містить найбільш релевантну інформацію з вхідної послідовності для поточного кроку обробки [7].

Механізми уваги можуть бути інтегровані в різні архітектури нейронних мереж, такі як рекурентні нейронні мережі (RNN), згорткові нейронні мережі (CNN) та трансформери. Вони показали значне покращення продуктивності в задачах машинного перекладу, обробки природної мови, розпізнавання мовлення та комп'ютерного зору.

Генеративно-змагальні мережі (GAN)

Генеративно-змагальні мережі (Generative Adversarial Networks, GAN) є особливим видом глибоких нейронних мереж, які використовуються для генерації нових, реалістичних даних, таких як зображення, звуки або текст. Вони складаються з двох основних компонентів: генератора (Generator) та дискримінатора (Discriminator), які змагаються один з одним у процесі навчання.

Generator – це нейронна мережа, яка намагається генерувати реалістичні зразки даних (наприклад, зображення) з випадкового шумового вектора [8].

Discriminator – інша нейронна мережа, яка намагається відрізнити справжні дані від згенерованих генератором [8].

Ці дві мережі навчаються одночасно в процесі, відомому як змагальне навчання:

1. Генератор намагається генерувати зразки, які будуть переконливими для дискримінатора, тобто дискримінатор не зможе відрізнити їх від справжніх даних.

2. Дискримінатор намагається покращити свою здатність розрізняти справжні дані від згенерованих генератором.

Під час навчання ці дві мережі постійно вдосконалюються, змушуючи одна одну ставати кращими. Зрештою, генератор повинен навчитися генерувати зразки, які неможливо відрізнити від реальних даних.

GAN успішно застосовуються в різноманітних завданнях, таких як генерація зображень, збільшення роздільної здатності зображень, перенесення стилю зображень, синтез мовлення та багато інших. Вони також є перспективним напрямком досліджень в галузі штучного інтелекту та машинного навчання.

Однією з ключових переваг глибинного навчання є його здатність ефективно працювати зі складними, неструктурованими даними, такими як зображення, відео, аудіо та текст. Глибинні моделі можуть виявляти складні закономірності та абстракції, які важко виразити явно за допомогою традиційних алгоритмів. Проте варто зазначити, що глибинне навчання вимагає величезних обсягів даних для ефективного навчання, потужних обчислювальних ресурсів та ретельного налаштування гіперпараметрів. Крім того, глибинні моделі часто розглядаються як «чорні скриньки», оскільки важко зрозуміти та інтерпретувати процеси, що відбуваються всередині складних нейронних мереж.

2.3. Алгоритми на основі нейронних мереж

YOLO (You Only Look Once)

YOLO є одноетапним детектором об'єктів, що базується на згорткових нейронних мережах. Ключова особливість YOLO полягає в тому, що він розглядає процес виявлення об'єктів як задачу регресії, обробляючи ціле зображення одночасно, замість генерування безлічі пропозицій регіонів та їх подальшої класифікації. Алгоритм розбиває зображення на сітку комірок та для кожної комірки передбачає обмежувальні рамки і ймовірності приналежності до класів об'єктів, що знаходяться в цій комірці.

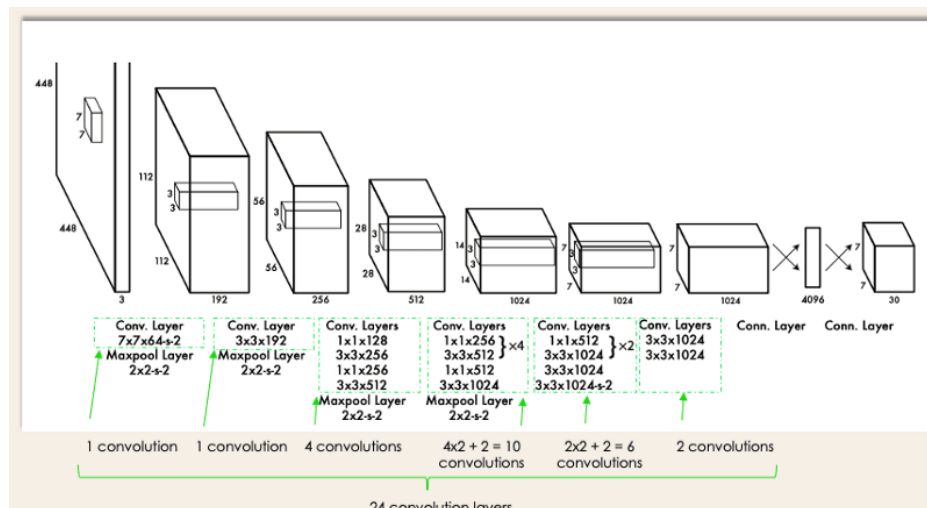


Рис. 2.5. Архітектура YOLO [9]

Загалом алгоритм виявлення об'єктів YOLO складається з:

- Залишкові блоки;
- Регресія обмежувальної області;
- Перетин над об'єднаннями (IOU);
- Немаксимальне придушення.

Залишкові блоки (Residual Blocks): Залишкові блоки є ключовою архітектурною особливістю в моделях YOLO. Вони дозволяють будувати глибокі нейронні мережі, усуваючи проблему зникаючого градієнта під час навчання. У залишкових блоках вихід одного шару додається до виходу

декількох наступних згорткових шарів, створюючи так звану "залишкову" структуру. Це полегшує потік градієнтів через мережу, що дозволяє ефективніше навчатися на більш глибоких архітектурах.

Регресія обмежувальної області (Bounding Box Regression): YOLO розглядає виявлення об'єктів як задачу регресії, де для кожної комірки сітки мережа передбачає чотири значення, що визначають обмежувальну рамку потенційного об'єкта: x , y (координати центру), ширину та висоту. Під час навчання, мережа коригує ці значення, щоб вони якнайкраще відповідали наявним обмежувальним рамкам у навчальних даних.

Перетин над об'єднаннями (Intersection over Union, IoU): IoU є метрикою, що використовується для оцінки якості передбачених обмежувальних рамок порівняно з істинними обмежувальними рамками в навчальних даних. Вона обчислюється як відношення площі перетину передбаченої та істинної обмежувальних рамок до площі їх об'єднання. Під час навчання YOLO максимізує значення IoU для точних передбачень обмежувальних рамок.

Немаксимальне придушення (Non-Maximum Suppression, NMS): NMS - це постобробка, що застосовується до результатів YOLO для видалення дублікатних виявлень одного і того ж об'єкта. Алгоритм сортує передбачені обмежувальні рамки за ймовірністю та видаляє рамки з перекриттям вище заданого порогу IoU. Таким чином, залишаються лише найкращі виявлення для кожного об'єкта на зображенні.

Переваги YOLO:

- + Висока швидкість обробки, що дозволяє виявляти об'єкти в режимі реального часу;
- + Краща узагальнююча здатність порівняно з двоетапними алгоритмами;
- + Можливість виявляти об'єкти з різними формами та розмірами.

Недоліки YOLO:

- Відносно нижча точність порівняно з двоетапними алгоритмами, особливо для дрібних об'єктів.
- Проблеми з виявленням перекриваючих об'єктів;
- Вимагає більше обчислювальних ресурсів під час навчання.

SSD (Single Shot MultiBox Detector)

SSD є ще одним одноетапним детектором об'єктів на основі згорткової нейронної мережі. На відміну від YOLO, SSD використовує техніку anchor boxes для генерації пропозицій регіонів з різними співвідношеннями сторін та розмірами. Це дозволяє SSD ефективно виявляти об'єкти різної форми та розміру на зображенні.

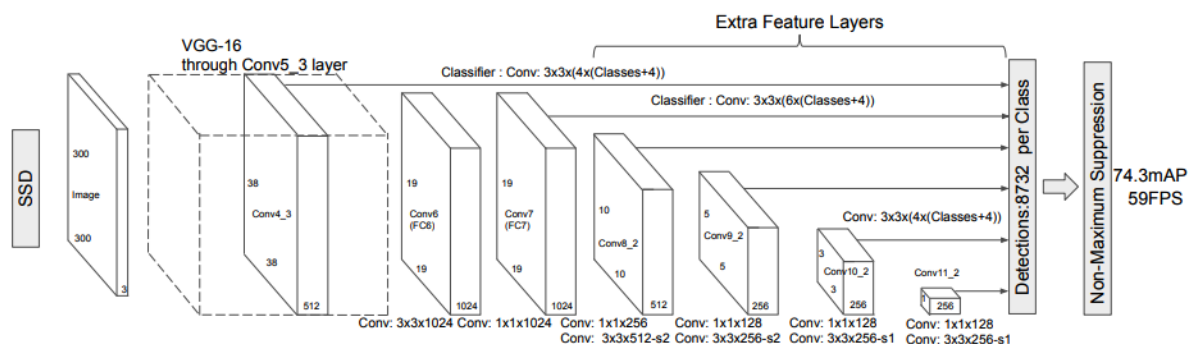


Рис. 2.6. SSD архітектура [10]

Архітектура SSD складається з двох основних компонентів: базової мережі та додаткових конволюційних шарів. Базова мережа використовується для екстракції ознак із вхідного зображення. Ця мережа може бути попередньо навченою з використанням великих наборів даних, таких як ImageNet, або ж може бути створена спеціально для конкретного завдання.

Після вилучення ознак з базової мережі, SSD використовує додаткові конволюційні шари для виявлення об'єктів та їх класифікації. Ці шари зазвичай мають різну роздільну здатність, що дозволяє аналізувати

зображення на різних масштабах. Крім того, SSD використовує ідею «мультибоксів», що означає використання фіксованих пропозицій областей об'єктів на різних шарах мережі. Це дозволяє моделі виявляти об'єкти різних розмірів та аспектів без необхідності генерації пропозицій областей, що робить алгоритм SSD швидшим та ефективнішим порівняно з іншими методами об'єктного виявлення.

Переваги SSD:

- + Висока точність виявлення об'єктів, особливо для дрібних та складних об'єктів;
- + Можливість ефективно виявляти об'єкти різних форм та розмірів завдяки використанню anchor boxes;
- + Висока швидкість обробки, що робить його придатним для застосувань у реальному часі.

Недоліки SSD:

- Нижча швидкість порівняно з YOLO для великих вхідних зображень;
- Можливі проблеми з виявленням перекриваючих об'єктів;
- Вимагає ретельного підбору налаштувань anchor boxes для оптимальної продуктивності.

R-CNN (Regions with CNN features)

R-CNN є двоетапним підходом до виявлення об'єктів. На першому етапі алгоритм генерує безліч пропозицій регіонів за допомогою спеціального методу, наприклад, вибіркового пошуку або алгоритмом EdgeBoxes. Потім для кожної запропонованої області обчислюються ознаки згорткової нейронної мережі, які використовуються для класифікації об'єктів та уточнення обмежувальних рамок.

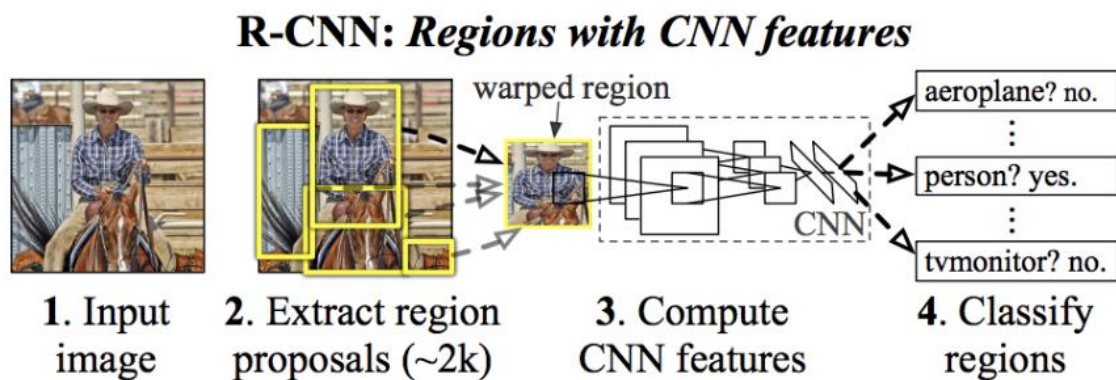


Рис. 2.7. Архітектура R-CNN [1]

Цей метод був вперше представлений в 2014 році роботою «Rich feature hierarchies for accurate object detection and semantic segmentation» авторами Girshick, Donahue, Darrell та Malik [1].

Основна ідея R-CNN полягає в тому, щоб спочатку виявити пропозиції областей, які містять потенційні об'єкти на зображенні, а потім використовувати глибоку нейронну мережу (зазвичай CNN) для витягування ознак з кожної пропозиції та подальшої класифікації цих ознак на конкретні класи об'єктів.

Для виявлення пропозицій областей R-CNN використовує методи сегментації та відстеження об'єктів, такі як Selective Search, який видає кілька тисяч пропозицій областей для кожного вхідного зображення. Кожна пропозиція області потім обробляється незалежно, витягуючи ознаки з CNN для кожної області. Ці ознаки подаються на вхід до класифікатора (наприклад, SVM або softmax-класифікатор), який визначає, чи належить кожна пропозиція області до конкретного класу об'єктів або ні.

Після класифікації пропозицій областей, R-CNN виконує пост-обробку для об'єднання перекриваючихся пропозицій та відкидання низькоімовірних детекцій. Цей підхід дозволяє R-CNN досягати високої точності виявлення об'єктів, але в той же час він виявляється досить повільним у порівнянні зі

своїми наступниками через необхідність обробки кожної пропозиції області окремо.

Переваги R-CNN:

- + Висока точність виявлення об'єктів, особливо для складних сцен з перекриваючими об'єктами;
- + Можливість виявляти об'єкти різних форм та розмірів завдяки використанню пропозицій регіонів;
- + Гнучкість у виборі методу генерації пропозицій регіонів.

Недоліки R-CNN:

- Низька швидкість обробки, оскільки кожна область обробляється окремо;
- Вимагає значних обчислювальних ресурсів під час навчання та інференції;
- Проблеми з виявленням дрібних об'єктів через недосконалість пропозицій регіонів.

Пізніші розробки, такі як Fast R-CNN та Faster R-CNN, були запропоновані для подолання деяких недоліків R-CNN, покращуючи швидкість та ефективність алгоритму.

У порівняльному аналізі алгоритмів YOLO, SSD і R-CNN для їхнього використання у системі автоматизованих розумних дверей враховувалися ряд ключових критеріїв.

По-перше, у системах реального часу, як от системи автоматизованих розумних дверей, великого значення набуває швидкодія алгоритмів. YOLO і SSD відзначаються високою швидкодією, бо вони використовуються як алгоритми одноетапного виявлення об'єктів. Вони здатні до детекції об'єктів за один прохід через мережу. Наприклад, SSD базується на використанні

«мультибоксів», що спрощує процес виявлення об'єктів. На відміну від цього, R-CNN потребує додаткового етапу передбачення регіонів, що відзначається відчутним зниженням швидкодії.

Другий критерій - точність. SSD і YOLO, хоча менш точні в порівнянні з R-CNN, забезпечують прийнятну точність для більшості застосувань. Однак R-CNN зазвичай досягає найвищого рівня точності, так як він використовує більш складні архітектури та детальнішу обробку областей об'єктів, проте, якщо розглядати сучасні версії алгоритмів YOLO, то вони навіть опереджають R-CNN за даним критерієм.

Третій критерій - вимоги до ресурсів. У більшості випадків SSD і YOLO вимагають менших обчислювальних ресурсів у порівнянні з R-CNN, особливо на етапі навчання та перевірки. Це може бути важливим у виборі алгоритму для систем, які працюють у обмежених умовах обчислювальних ресурсів, наприклад, у системи автоматизованих розумних дверей. Проте, нові версії YOLO можуть бути більш ресурсо-затратними, тому варто приділити увагу правильному вибору версії алгоритму YOLO. В залежності від бюджету системи.

Отже, у контексті використання у системі автоматизованих розумних дверей було обрано YOLOv3 для розпізнавання облич. Він забезпечує непогану точність, швидкодію та вимагає менше ресурсів порівняно з R-CNN або більш новіших версій YOLO, що робить його ідеальним вибором для таких застосувань.

2.4. Мета та задачі дипломного проєкту

На сьогодні питання безпеки та контролю доступу до приміщень є надзвичайно важливими як для державних установ, так і для приватних організацій. Надійна система контролю доступу дозволяє забезпечити належний рівень безпеки, обмеживши несанкціонований вхід сторонніх осіб.

Традиційні методи, такі як використання охоронців або пропускних систем на основі карток чи біометричних даних, мають певні недоліки: високі експлуатаційні витрати, людський фактор та можливість обману систем. Завдяки застосуванню нейронних мереж та методів глибинного навчання можливо досягти високої точності автоматизованого розпізнавання обличь, виявлення підозрілої поведінки чи виявлення інших небажаних ситуацій.

Метою даної роботи є створення автоматизованої системи контролю безпеки з використанням нейронних мереж для розпізнавання обличь та надання доступу лише авторизованим особам, підвищуючи тим самим рівень безпеки без додаткового залучення людських ресурсів.

Для досягнення поставленої мети необхідно виконати:

1. Дослідити особливості та вимоги до систем контролю доступу в різних середовищах (офісні приміщення, виробничі об'єкти, житлові будинки тощо);
2. Проаналізувати існуючі технічні рішення для розпізнавання обличь та контролю доступу;
3. Дослідити можливості нейронних мереж;
4. Обрати оптимальне апаратне та програмне забезпечення для побудови системи контролю безпеки, включаючи камери відеоспостереження, обчислювальні пристрої та бібліотеки машинного навчання;
5. Розробити алгоритми розпізнавання обличь та надання доступу в режимі реального часу на основі алгоритмів YOLO;
6. Провести навчання нейронної мережі на відповідному наборі даних з обличчями авторизованих осіб;
7. Інтегрувати розроблену систему з механізмами контролю доступу (електронні замки, турнікети тощо).

Розділ 3. АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ КОНТРОЛЮ БЕЗПЕКИ З ВИКОРИСТАННЯМ НЕЙРОНИХ МЕРЕЖ

3.1. Вибір апаратної бази

Вибір правильної камери є вирішальним кроком у розробці систем комп'ютерного зору, особливо для виявлення людей. Адже саме камера надає вхідні візуальні дані, які визначають точність та ефективність роботи всієї системи. Основними характеристиками, які слід враховувати при виборі камери для такої системи, є:

1. **Роздільна здатність камери.** Чим вища кількість мегапікселів, тим більше деталей зможе зафіксувати камера, що забезпечить підвищення точності детектування плодів. Однак, слід зважати на те, що надто висока роздільна здатність може негативно позначитися на швидкодії та ресурсоємності системи;
2. **Чутливість до кольору.** Для коректного розпізнавання обличч людей критично важливим є точне відтворення кольорів. Тому камера повинна мати широкий динамічний діапазон та високу колірну чутливість, щоб забезпечити адекватне відтворення відтінків;
3. **Частота кадрів.** Ця характеристика впливає на швидкодію та затримки системи. Для забезпечення плавної роботи в режимі реального часу потрібна достатньо висока частота кадрів, однак її надмірне збільшення може призвести до перевантаження системи;
4. **Вартість.** При створенні комерційного рішення необхідно знайти оптимальне співвідношення між ціною камери та її характеристиками, адже надто дорога камера може значно збільшити вартість кінцевого продукту.
5. **Умови експлуатації.** Слід враховувати специфіку середовища, в якому буде працювати система детектування людей: освітленість, температуру, вологість тощо. Відповідно, камера повинна бути пристосована до таких умов.

Розглянемо кілька прикладів камер, які можуть бути використані в системах детектування людей, з їхніми перевагами та недоліками.

Basler acA2500-14gc

Basler acA2500-14gc – високопродуктивна промислова камера з роздільною здатністю 5 МП, частотою кадрів до 14 Гц та широким динамічним діапазоном [12].



Рис. 3.1. Камера Basler acA2500-14gc

Технічні характеристики:

- Роздільна здатність: 2592 x 1944 пікселів (5 мегапікселів);
- Швидкість зйомки: До 14 кадрів в секунду при максимальному розширенні;
- Сенсор зображення: Монохромний CMOS-сенсор;
- Розмір пікселя: 3.45 мкм × 3.45 мкм;
- Інтерфейс: Gigabit Ethernet;
- Динамічний діапазон: Широкий динамічний діапазон для забезпечення відмінної якості зображення в різних умовах освітлення;
- О'єктив: Підтримується різна лінзова оптика з різними фокусними відстанями;
- Робоча температура: Від 0°C до +50°C.

Переваги:

- + Висока роздільна здатність;

- + Висока швидкість зйомки;
- + Завдяки Gigabit Ethernet інтерфейсу забезпечується швидкий та надійний зв'язок з іншими промисловими системами;
- + Динамічний діапазон дозволяє отримувати зображення з високою якістю в різних умовах освітлення.

Недоліки:

- Висока ціна;
- Потреба в високоякісній обробці зображення;
- Обмежений діапазон робочої температури.

IDS UI-3370CP-C-HQ Rev.2

IDS UI-3370CP-C-HQ Rev.2 – це промислова USB 3.0 камера від IDS Imaging Development Systems GmbH, яка відома своїми високоякісними продуктами для застосувань у сфері машинного зору та промислової автоматизації.



Рис. 3.2. Камера IDS UI-3370CP-C-HQ Rev.2

Технічні характеристики:

- Роздільна здатність: Залежно від моделі, але зазвичай у камери висока роздільна здатність від 2 до 20 мегапікселів;
- Швидкість зйомки: Зазвичай до 60 кадрів в секунду або більше, залежно від роздільної здатності;
- Сенсор зображення: CMOS-сенсор;

- Розмір пікселя: Відносно малий розмір пікселя, що дозволяє отримувати високоякісні зображення з деталізацією;
- Інтерфейс: USB 3.0, що забезпечує високу швидкість передачі даних та простоту підключення;
- Об'єктив: Підтримується використання різних об'єктивів в залежності від потреб застосування;
- Робоча температура: Зазвичай від 0°C до +60°C або більше, що дозволяє використовувати камеру в різних умовах.

Переваги:

- + Висока роздільна здатність;
- + Широкий діапазон моделей;
- + USB 3.0 інтерфейс забезпечує високу швидкість передачі даних;
- + Надійність та якість виконання.

Недоліки:

- Потребується вибір та придбання додаткового об'єктива;
- Вартість;
- Потреба в програмному забезпеченні для обробки зображень.

Raspberry Pi Camera Module V2

Raspberry Pi Camera Module V2 – це високоякісна камера, розроблена спеціально для використання з одноплатними комп'ютерами Raspberry Pi.



Рис. 3.3. Камера Raspberry Pi Camera Module V2

Технічні характеристики:

- Роздільна здатність: 8 мегапікселів (3280 x 2464 пікселів);
- Сенсор зображення: Sony IMX219PQ CMOS;
- Розмір пікселя: 1.12 мкм;
- Діафрагма (F-число): F/2.0;
- Інтерфейс: MIPI CSI-2 (Camera Serial Interface);
- Швидкість зйомки: До 30 кадрів в секунду при повній роздільності;
- Об'єктив: Фіксований об'єктив без можливості зміни фокусної відстані;
- Робоча температура: Від -20°C до +60°C [13].

Переваги:

- + Широкий спектр застосувань;
- + Простота використання;
- + Висока роздільна здатність та якість зображення;
- + Доступність та вартість.

Недоліки:

- Фіксований об'єктив;
- Відсутність захисту від вологи та пилу.

Також непоганими варіантами камер для детектування є FLIR Blackfly S BFS-U3-51S5P-C та бюджетний варіант Arducam 16MP 1/2.3" AR1638. Камеру потрібно обирати в залежності від кінцевої вартості системи. Адже пропонована система може бути інтегрована в різних цінових межах.

Комп'ютер

В залежності від місця встановлення системи, її можна реалізувати з використанням стаціонарного комп'ютера/ноутбука або ж з використанням одноплатного комп'ютери та екрана. Систему за простою можна реалізувати на

таких одноплатних комп'ютерах як: Raspberry Pi, Nvidia Jetson Nano, Intel Neural Compute Stick та інші. Для перевірки та тестування системи в житті гарним варіантом є використання Raspberry Pi 4.

Raspberry Pi 4 – це одноплатний комп'ютер, що розроблений фондом Raspberry Pi Foundation. Ця платформа представляє собою невелику плату, але при цьому має потужність і можливості традиційного персонального комп'ютера [14]. Raspberry Pi 4 (рис. 3.4.) відомий своєю високою продуктивністю, різноманітними можливостями взаємодії з різними пристроями та доступною вартістю. Він має вбудований процесор ARM, різні роз'єми для підключення периферійних пристроїв (наприклад, HDMI, USB, Ethernet), а також можливість встановлення різноманітного програмного забезпечення, включаючи операційні системи, програми для роботи з даними, веб-сервери та ігри. Raspberry Pi 4 широко використовується як у навчальних закладах для вивчення програмування та комп'ютерних технологій, так і в промисловості для створення прототипів, IoT-проектів та інших застосувань вбудованих систем.



Рис. 3.4. Одноплатний комп'ютер Raspberry Pi 4

Технічні характеристики:

- Процесор -Broadcom BCM2711, чотирьохядерний ARM Cortex-A72 (64-біт), тактова частота до 1,5 ГГц;

- Пам'ять: варіанти з пам'яттю від 2 ГБ, 4 ГБ або 8 ГБ LPDDR4-2400 SDRAM;
- Вбудований графічний процесор VideoCore VI, який підтримує OpenGL ES 3.x та 4K відеовідтворення;
- Порти і роз'єми:
 - 2х порти HDMI 2.0, підтримка відеовиводу до 4K при 60 Гц;
 - 2х порти USB 3.0;
 - 2х порти USB 2.0;
 - Гігабітний Ethernet порт (RJ45);
 - 3,5-міліметровий аудіороз'єм для навушників або динаміків;
 - Мікрофонний вхід (3,5 мм);
 - Слот для карт пам'яті MicroSD;
 - 40-пінний GPIO роз'єм (General Purpose Input/Output);
 - Інтерфейс камери (CSI);
 - Інтерфейс дисплея (DSI);
- Вбудований модуль Wi-Fi 802.11ac та Bluetooth 5.0;
- Може живитися від стандартного мікро-USB або від USB-C (залежно від версії);
- Підтримується багато операційних систем, включаючи Raspbian (базова операційна система для Raspberry Pi), Ubuntu, Windows 10 IoT та інші;
- Розміри: 85,6 мм × 56,5 мм × 17 мм.

3.2. Алгоритм та структурна схеми системи контролю безпеки

Алгоритм роботи системи розумних дверей складається з таких основних етапів:

1. Ініціалізація та підключення всіх необхідних бібліотек та компонентів;
2. Далі потрібно отримати відеопотік з камери;
3. Наступний крок – попередня обробка зображення.
4. Після чого зображення подається на вхід моделі YOLOv3, яка розділяє зображення на сітку і прогнозує прямокутники, що містять об'єкти різних класів, у нашому випадку обличчя.
5. Отримані координати прямокутників, що містять обличчя, можуть бути фільтровані для відсівання хибних детекцій або дублікатів. Це може включати використання алгоритмів порогової фільтрації, які враховують достовірності моделі в прогнозі.
6. Якщо обличчя розпізнано і підтверджено, можна виконати відповідні дії – відкривання дверей.

На рис. 3.5. зображено детальний алгоритм роботи пропонованої системи за використанням алгоритмів YOLO, одноплатного комп'ютера Raspberry Pi та соленоїдного замку.

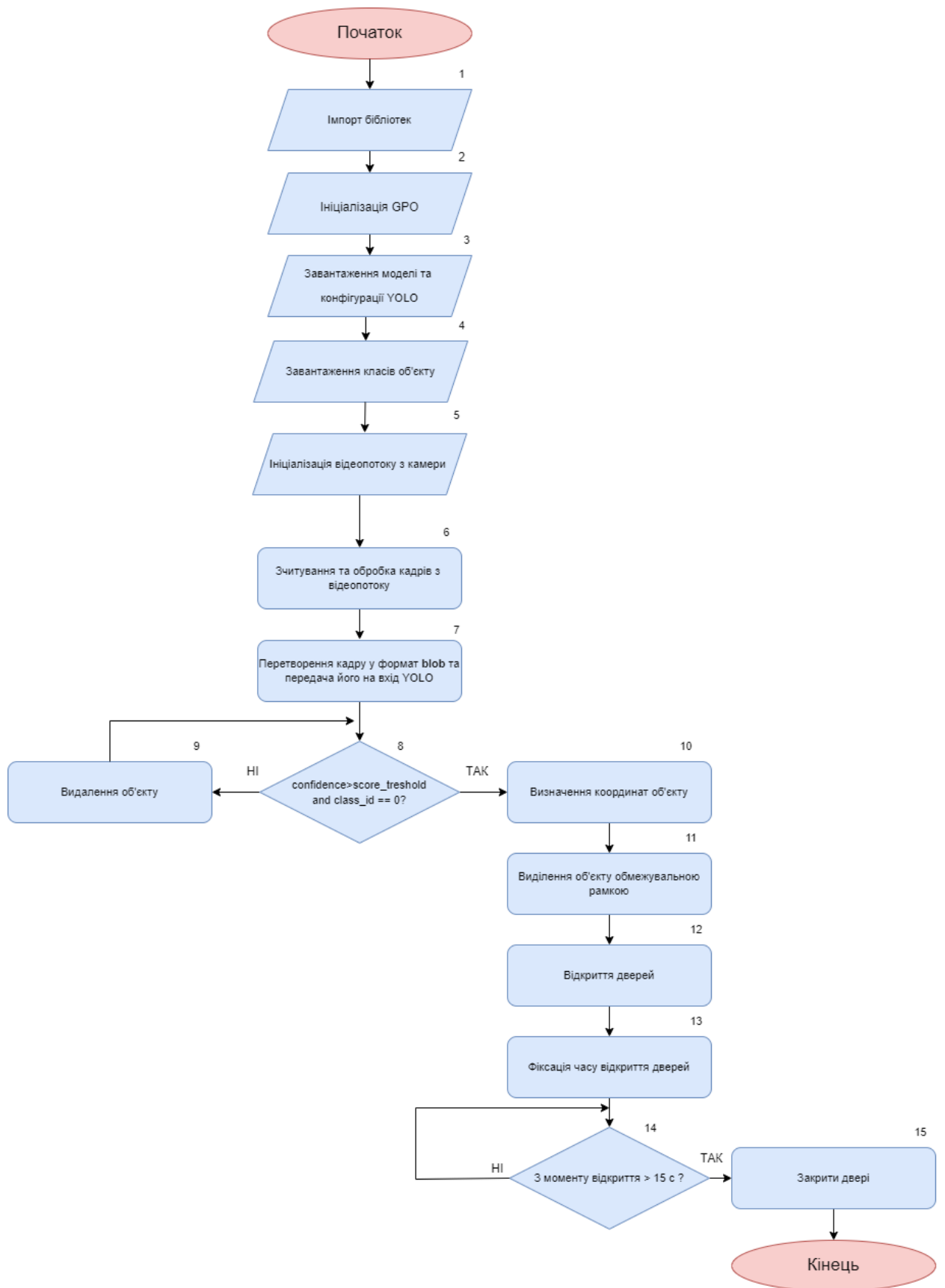


Рис. 3.5. Алгоритм роботи системи

Систему можливо інтегрувати в різних варіаціях, в даній дипломній роботі пропонується реалізація на основі одноплатного комп'ютера Raspberry Pi.

Структурна схема системи наведена на рис. 3.6.

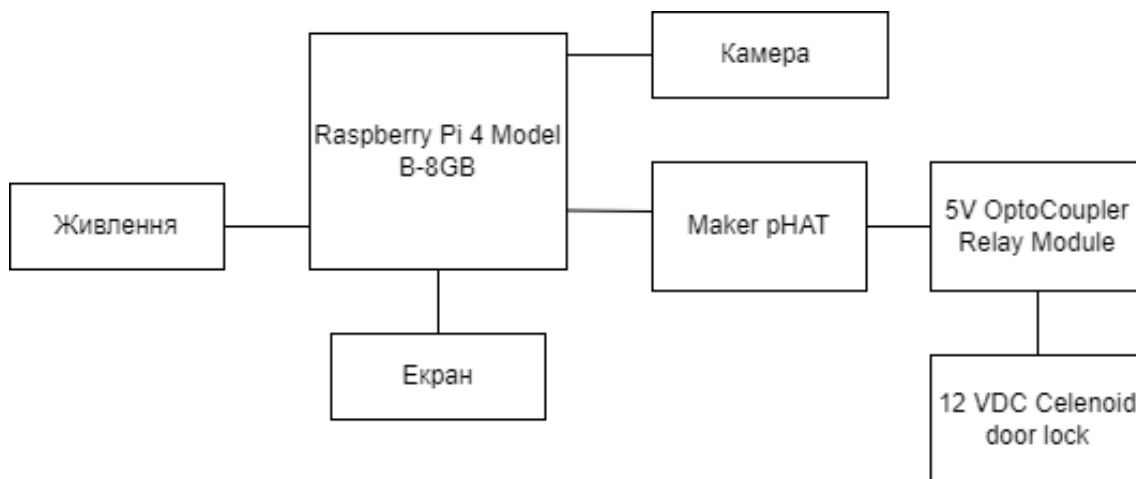


Рис. 3.6. Структурна схема системи

Розділ 4. ПРОГРАМНА РЕАЛІЗАЦІЯ

4.1. Створення власного набору даних

Розпізнавання обличь на основі глибокого навчання є актуальною задачею у галузі комп'ютерного зору та штучного інтелекту. Для досягнення високої точності цього завдання необхідна наявність великого обсягу анотованих зображень. Анотація полягає у розміченні зображень, ідентифікуючи та позначаючи області, що містять обличчя. Цей процес є трудомістким і часто вимагає залучення людських ресурсів. Використання потужних обчислювальних ресурсів, таких як графічні процесори (GPU), є критичним для успішного тренування моделей глибокого навчання на великих наборах даних у розумні терміни.

У цьому проєкті розглядається процес тренування моделі YOLO, яка є одним з провідних підходів для розпізнавання об'єктів та обличь. Було використано можливості безкоштовного сервісу Google Colab, який надає доступ до обчислювальних ресурсів GPU, що дозволить прискорити процес тренування моделі.

Для забезпечення ефективного тренування моделі розпізнавання обличь рекомендується використовувати набір даних, що містить значну кількість різноманітних зображень із зображеннями обличь. Оптимальним є наявність не менше 100 зображень різних типів, які охоплюють різні вирази обличчя, положення та умови освітлення. Різноманітність і репрезентативність зображень є ключовими факторами для забезпечення узагальнюючої здатності моделі та її ефективності у реальних умовах (рис.4.1).

Під час створення власного набору даних для тренування моделі розпізнавання обличь важливо враховувати наступні аспекти:

1. Різноманітність зображень – включення зображень із різними фонами, освітленням, кутами зйомки та умовами, що імітують реальні ситуації, в яких буде застосовуватися модель.

2. Анотація даних – ретельна анотація зображень, позначаючи області, що містять обличчя, є критичним етапом для забезпечення якісного тренування моделі.

3. Розділення даних - поділ набору даних на тренувальну, валідаційну та тестову вибірки для забезпечення належного оцінювання продуктивності моделі.



Рис. 4.1. Вибірка зображень для набору даних

Анотація зображень є критичним етапом у процесі підготовки даних для тренування моделей глибокого навчання для розпізнавання об'єктів та обличч. Цей процес полягає у ретельному розмічуванні зображень, ідентифікуючи та позначаючи області, що містять об'єкти інтересу, в даному випадку – обличчя.

Анотація зображень зазвичай виконується вручну, залучаючи людські ресурси. Цей процес може бути трудомістким і вимагати значних зусиль, особливо для великих наборів даних. В загальному інструментів для анотації зображень дуже багато. Є безкоштовні версії та дороговартісні, але всі вони виконують одну і ту саму задачу – виділення об'єкта та присвоєння йому певного класу для подальшого навчання зображення.

В даному дипломному проєктуванні було використано безкоштовний онлайн ресурс MakeSense, який підходить для анотування зображень під алгоритми YOLO (Рис. 4.2.).

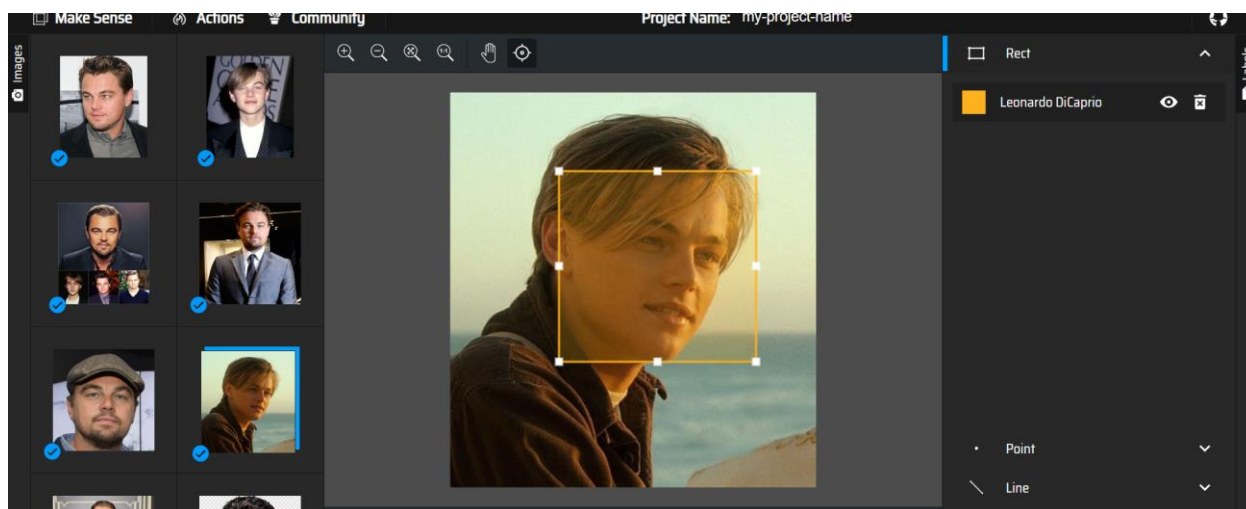


Рис. 4.2. Процес анотації зображення

Имя	Дата изменения	Тип	Размер
10.txt	29.03.2024 13:45	Текстовый документ	1 КБ
071_a052296d.txt	29.03.2024 13:45	Текстовый документ	1 КБ
072_c887ec5c.txt	29.03.2024 13:45	Текстовый документ	1 КБ
073_42e32f65.txt	29.03.2024 13:45	Текстовый документ	1 КБ
074_2b2a84b6.txt	29.03.2024 13:45	Текстовый документ	1 КБ
078_a7fd9be1.txt	29.03.2024 13:45	Текстовый документ	1 КБ
079_6ec91919.txt	29.03.2024 13:45	Текстовый документ	1 КБ
080_2bffa0b.txt	29.03.2024 13:45	Текстовый документ	1 КБ
081_a32678db.txt	29.03.2024 13:45	Текстовый документ	1 КБ
085_a292bf25.txt	29.03.2024 13:45	Текстовый документ	1 КБ
086_a1441427.txt	29.03.2024 13:45	Текстовый документ	1 КБ
090_b316be20.txt	29.03.2024 13:45	Текстовый документ	1 КБ
091_e39f525d.txt	29.03.2024 13:45	Текстовый документ	1 КБ
092_cbc6d525.txt	29.03.2024 13:45	Текстовый документ	1 КБ
093_811dc474.txt	29.03.2024 13:45	Текстовый документ	1 КБ
094_78e0fb7b.txt	29.03.2024 13:45	Текстовый документ	1 КБ
096_c30a9446.txt	29.03.2024 13:45	Текстовый документ	1 КБ
097_1837d89e.txt	29.03.2024 13:45	Текстовый документ	1 КБ
098_f0bc37c7.txt	29.03.2024 13:45	Текстовый документ	1 КБ
099_c25b7b04.txt	29.03.2024 13:45	Текстовый документ	1 КБ
100_ec9fe97f.txt	29.03.2024 13:45	Текстовый документ	1 КБ

Рис. 4.3. Анотовані зображення

Анотовані файли містять символічні значення, які представляють інформацію про розташування об'єктів інтересу на зображеннях. Ця інформація зазвичай зберігається у форматі, зручному для зчитування та обробки моделями глибокого навчання. У файлі наведено набір цифр «15 0.392578 0.504395 0.566406 0.791992», що містить в собі інформацію про ідентифікатор класу та нормалізовані координати обмежувального прямокутника.

Створення власної бази даних буде відбуватися в середовищі GoogleColab, за допомогою блокнотів навчання, що є у відкритому доступі.

В блокноті Google Colab підключаємо GPU, що буде спрощувати час навчання нашої бази даних.

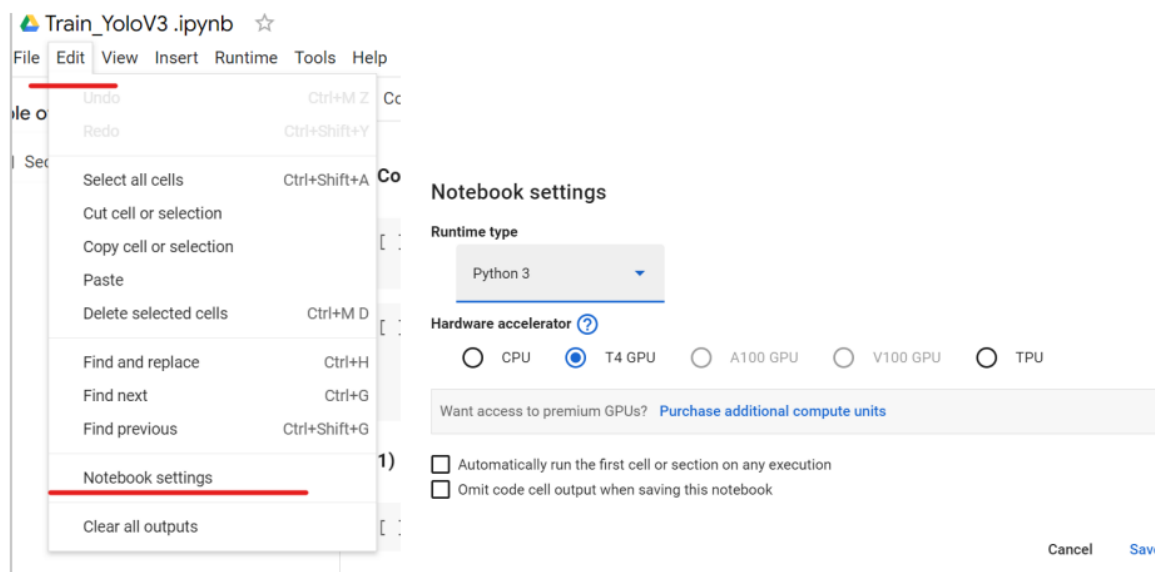


Рис. 4.4. Підключення GPU

Також важливим етапом є встановлення всіх конфігурацій, бібліотек та фреймворку Darknet. Також важливо налаштувати правильний формат архітектури YOLO та додати шлях до анотованих файлів, що збережені у форматі zip. Останнім кроком є запуск процесу навчання (рис. 4.5.)

```
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 82 Avg (IOU: 0.187677), count: 4, class_lo
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 94 Avg (IOU: 0.000000), count: 1, class_lo
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 106 Avg (IOU: 0.000000), count: 1, class_l
total_bbox = 36, rewritten_bbox = 0.000000 %
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 82 Avg (IOU: 0.346730), count: 4, class_lo
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 94 Avg (IOU: 0.000000), count: 1, class_lo
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 106 Avg (IOU: 0.000000), count: 1, class_l
total_bbox = 40, rewritten_bbox = 0.000000 %
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 82 Avg (IOU: 0.256218), count: 4, class_lo
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 94 Avg (IOU: 0.000000), count: 1, class_lo
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 106 Avg (IOU: 0.000000), count: 1, class_l
total_bbox = 44, rewritten_bbox = 0.000000 %
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 82 Avg (IOU: 0.414854), count: 4, class_lo
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 94 Avg (IOU: 0.000000), count: 1, class_lo
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 106 Avg (IOU: 0.000000), count: 1, class_l
total_bbox = 48, rewritten_bbox = 0.000000 %
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 82 Avg (IOU: 0.309240), count: 4, class_lo
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 94 Avg (IOU: 0.000000), count: 1, class_lo
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 106 Avg (IOU: 0.000000), count: 1, class_l
total_bbox = 52, rewritten_bbox = 0.000000 %
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 82 Avg (IOU: 0.220097), count: 4, class_lo
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 94 Avg (IOU: 0.000000), count: 1, class_lo
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 106 Avg (IOU: 0.000000), count: 1, class_l
```

Рис. 4.5. Процес навчання

На виході отримуємо файли вагів та навчання «last.weights», які є нашою навченою базою даних (рис.4.6.).

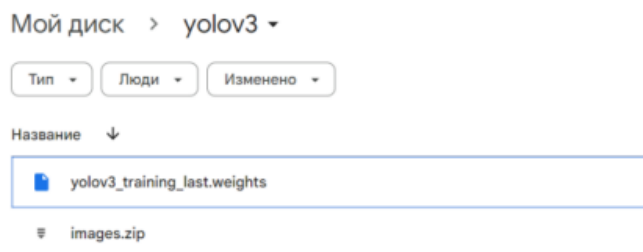


Рис. 4.6. Отриманий навчений набір даних

4.2. Результати виконання програми

Роботу отриманої моделі було перевірено на зображеннях (рис 4.7.). Підтвердивши працездатність набору, стає можливим його інтеграція будь-якої програми.

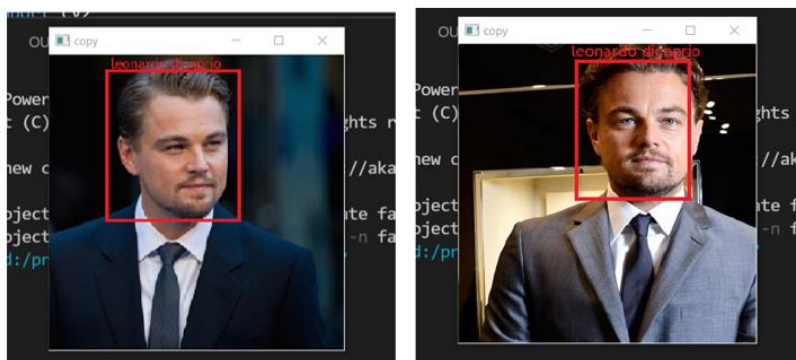


Рис. 4.7. Перевірка отриманої навченої моделі

Написане програмне забезпечення здійснює обробку вхідного сигналу, який подається на його вхід. Особлива увага приділяється процесу виявлення людей у вхідних даних, який здійснюється з використанням навченого набору даних. При детектуванні присутності людини на вхідному зображенні, програмне забезпечення виконує процедуру аналізу та прийняття рішення щодо відкриття дверей.

Програмна реалізація наведена в додатку А.

Основні етапи роботи програми:

1. Імпорт бібліотек: важливим етапом є підключення необхідних бібліотек для працездатності майбутньої системи, а саме бібліотеки для роботи з зображеннями та управління GPIO на Raspberry Pi. Відповідно, cv2 - OpenCV, numpy – бібліотека для обробки числових даних, imutils – набір утиліт для спрощення роботи з зображеннями, time – для роботи з часом, а RPi.GPIO – для керування GPIO на Raspberry Pi.

2. Ініціалізація GPIO для електромагнітного замка: задається пін, на якому підключено реле електромагнітного замка (у нашому випадку пін 17). Потім GPIO налаштовується на вихідний режим, а потім реле встановлюється у низький стан, щоб заблокувати двері.

3. Завантаження моделей та конфігурацій YOLO.

4. Ініціалізація відеопотоку з камери.

5. Зчитування та обробка кадрів з відеопотоку (цикл while True): у даному фрагменті виконується безкінечний цикл, який читає кадри з відеопотоку (vs.read()). Якщо кадр не було зчитано (ret дорівнює False), то цикл завершується. Кожен кадр зменшується за розміром до ширини 500 пікселів за допомогою функції imutils.resize. Запам'ятовує отриманий розмір кадра за допомогою frame.shape.

6. Детектування облич за допомогою YOLO: тобто кожен кадр перетворюється в blob-об'єкт, що використовується як вхід для моделі YOLO. Виклик net.forward виконує передачу кадру через мережу та отримання вихідних значень з останніх шарів мережі.

7. Обробка результатів детектування (цикл for): у цьому циклі перебираються всі об'єкти, виявлені моделлю YOLO. Для кожного об'єкту обчислюються координати та розміри прямокутника, який обведе об'єкт.

8. Визначення та відображення облич на кадрі.

9. Відкриття та закриття дверей: якщо обличчя було розпізнано та визначено, що це людина, то відбувається відкриття дверей. GPIO

піднімається до високого рівня для активації реле, яке відкриває двері. Також фіксується час відкриття дверей (prevTime) та змінна doorUnlock встановлюється в True, щоб запобігти безкінечному відкриттю дверей. Якщо двері були відкриті (doorUnlock = True) і час з моменту відкриття перевищує встановлений час, то закриваємо двері. GPIO встановлюється в низький рівень, щоб вимкнути реле, яке утримує двері відчиненими.

ВИСНОВКИ

Так як питання безпеки та контролю доступу до приміщень є надзвичайно важливими як для державних установ, так і для приватних організацій. Надійна система контролю доступу дозволяє забезпечити належний рівень безпеки, обмеживши несанкціонований вхід сторонніх осіб. Традиційні методи, такі як використання охоронців або пропускних систем на основі карток чи біометричних даних, мають певні недоліки: високі експлуатаційні витрати, людський фактор та можливість обману систем. Тому було виконано розробку власної системи безпеки.

Створена база даних, побудована на основі алгоритмів YOLO забезпечуючи необхідний обсяг та репрезентативність даних. Надалі, на основі отриманих результатів, було розроблено та реалізовано систему, яка використовує Raspberry Pi у поєднанні з алгоритмами YOLO та селеноїдним замком. Ця система, функціонуючи за принципом розпізнавання обличчя та автоматичного відкривання дверей при виявленні людини, демонструє практичну застосовність розроблених алгоритмів та моделей.

Таким чином, отримані результати свідчать про можливість успішної інтеграції систем розпізнавання обличчя на основі алгоритмів YOLO у реальних умовах, що відкриває широкі перспективи для подальших досліджень та застосувань у сфері автоматизованої безпеки та контролю доступу.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Типи контролю безпеки [Електронний ресурс]. – Режим доступу: <https://www.ibm.com/topics/security-controls>
2. Певні аспекти ідентифікації [Електронний ресурс]. – Режим доступу: <https://training.gatestlab.com/blog/technical-articles/authentication-authorization-and-identification/>
3. Машинне навчання [Електронний ресурс]. – Режим доступу: <https://www.ibm.com/topics/machine-learning-algorithms>
4. Глибинне навчання [Електронний ресурс]. – Режим доступу: <https://medium.com/@sophiekhodol/%D0%BA%D0%BE%D1%80%D0%BE%D1%82%D0%BA%D0%BE-%D0%BF%D1%80%D0%BE-%D0%B3%D0%BB%D0%B8%D0%B1%D0%B8%D0%BD%D0%BD%D0%B5-%D0%BD%D0%B0%D0%B2%D1%87%D0%B0%D0%BD%D0%BD%D1%8F-4c441d556f7c>
5. CNN архітектура [Електронний ресурс]. – Режим доступу: <https://medium.com/@draj0718/convolutional-neural-networks-cnn-architectures-explained-716fb197b243>
6. LSTM архітектура [Електронний ресурс]. – Режим доступу: <https://www.semanticscholar.org/paper/CNN-LSTM-Architecture-for-Action-Recognition-in-Orozco-Buemi/df8beecc6c0d16e9b75675c46b99aee80aaa83d5>
7. Attention Networks [Електронний ресурс]. – Режим доступу: <https://machinelearningmastery.com/a-tour-of-attention-based-architectures/>
8. GAN архітектура [Електронний ресурс]. – Режим доступу: <https://aws.amazon.com/what-is/gan/>
9. YOLO [Електронний ресурс]. – Режим доступу: <https://arxiv.org/pdf/1506.02640.pdf>

10. SSD [Электронный ресурс]. – Режим доступа:
<https://towardsdatascience.com/review-ssd-single-shot-detector-object-detection-851a94607d11>
11. R-CNN [Электронный ресурс]. – Режим доступа:
<https://blog.roboflow.com/what-is-r-cnn/>
12. Камера Basler acA2500-14gc [Электронный ресурс]. – Режим доступа:
<https://www.sodavision.com/product/basler-aca2500-14gc/>
13. Камера Raspberry Pi Camera Module V2 [Электронный ресурс]. – Режим доступа: <https://evo.net.ua/raspberry-pi-camera-board/>
14. Raspberry Pi 4 [Электронный ресурс]. – Режим доступа:
<https://evo.net.ua/mikrokomputer-raspberry-pi-4-model-b-8gb/>

ДОДАТКИ

Програмне забезпечення розробленої системи

```
import cv2
import numpy as np
import imutils
import time
import RPi.GPIO as GPIO

RELAY = 17

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(RELAY, GPIO.OUT)
GPIO.output(RELAY, GPIO.LOW)

net = cv2.dnn.readNet("yolov3.weights", "yolov3.cfg")
classes = []

with open("coco.names", "r") as f:
    classes = [line.strip() for line in f.readlines()]

vs = cv2.VideoCapture(0)
time.sleep(2.0)
prevTime = 0
doorUnlock = False

while True:
    ret, frame = vs.read()
    if not ret:
        break
    frame = imutils.resize(frame, width=500)
    height, width, channels = frame.shape
    blob = cv2.dnn.blobFromImage(frame, 0.00392, (416, 416), (0, 0, 0), True, crop=False)
    net.setInput(blob)
    outs = net.forward(net.getUnconnectedOutLayersNames())
    conf_threshold = 0.5
    nms_threshold = 0.4
```

```

for out in outs:
    for detection in out:
        scores = detection[5:]
        class_id = np.argmax(scores)
        confidence = scores[class_id]
        if confidence > conf_threshold and class_id == 0: # 0 represents person class in COCO
dataset
            center_x = int(detection[0] * width)
            center_y = int(detection[1] * height)
            w = int(detection[2] * width)
            h = int(detection[3] * height)
            x = int(center_x - w / 2)
            y = int(center_y - h / 2)
            cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 2)
            GPIO.output(RELAY, GPIO.HIGH)
            prevTime = time.time()
            doorUnlock = True
            print("Door unlocked")
        if doorUnlock and time.time() - prevTime > 15:
            doorUnlock = False
            GPIO.output(RELAY, GPIO.LOW)
            print("Door locked")
        cv2.imshow(" Recognition", frame)
        key = cv2.waitKey(1) & 0xFF
        if key == ord("q"):
            break
    GPIO.output(RELAY, GPIO.LOW)
    GPIO.cleanup()
    vs.release()
    cv2.destroyAllWindows()

```