Міністерство освіти і науки України
Криворізький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерних систем та мереж

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

за спеціальністю 123 «Комп'ютерна інженерія»

Тема наукової роботи: МОДЕЛЬ ІНФОРМАЦІЙНОЇ ОХОРОННОЇ СИСТЕМИ НА БАЗІ БІОМЕТРИЧНОЇ ІДЕНТИФІКАЦІЇ

Виконав _____ Т. Р. Бухало

Керівник роботи _____ І. О. Музика

Нормоконтроль _____ Д. І. Кузнєцов

Завідувач кафедри _____ А. І. Купін

Кривий Ріг
2022

Криворізький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерних систем та мереж

Ступінь вищої освіти магістр
Спеціальність 123 «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ
Завідувач кафедри, голова циклової комісії

_____ А. І. Купін

"____" _____ 20__ року

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

_____
(прізвище, ім'я, по батькові)

1. Тема роботи _____
_____
_____

керівник роботи _____,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від "___"_____ 20__року №__

2. Строк подання студентом роботи _____

3. Вихідні дані до роботи _____
_____
_____
_____
_____
_____

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

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

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання_____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Строк виконання етапів роботи	Примітка

Студент _____ _____
(підпис) (прізвище та ініціали)

Керівник роботи _____ _____
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка: 81 сторінка, 26 рисунків, 3 таблиці, 1 додаток, 20 використаних джерел.

Об’єкт дослідження – процес інтеграції біометричної автентифікації в охоронних системах.

Робота складається з трьох розділів.

Перший розділ присвячено аналізу інформаційних систем та автентифікації на основі біометричних даних. Зроблено огляд сучасних, існуючих підходів до автентифікації людини. Висвітлено питання актуальності використання новітніх охоронних систем з використанням біометрики.

Другий розділ розкриває поняття штучного інтелекту. Проведено моделювання, розроблено математичну модель алгоритму-класифікатора SVM та Haar Cascade які використовуються для знаходження та розпізнавання обличчя людини.

У третьому розділі спроектовано охоронну систему на базі штучного інтелекту з використанням біометричних даних, реалізовано програмну частину серверного компоненту та апаратну і програмну частини контролеру.

Ключові слова: ШТУЧНИЙ ІНТЕЛЕКТ, МАШИННЕ НАВЧАННЯ, КЛАСИФІКАТОР, КОНТРОЛЕР, БІОМЕТРИКА, АВТЕНТИФІКАЦІЯ.

Explanatory note: 81 pages, 26 figures, 3 tables, 1 appendix, 20 used sources.

The object of research is the process of biometric authentication integration in security systems.

The work consists of three sections.

The first section is devoted to the analysis of information systems and authentication based on biometric data. An overview of modern, existing approaches to human authentication is made. The issue of the relevance of using the latest security systems using biometrics is highlighted.

The second chapter reveals the concept of artificial intelligence. Modeling was carried out, a mathematical model of the SVM and Haar Cascade classifier algorithm was developed, which are used to find and recognize a person's face.

In the third section, a security system based on artificial intelligence with the use of biometric data is designed, the software part of the server component and the hardware and software parts of the controller are implemented.

Keywords: ARTIFICIAL INTELLIGENCE, MACHINE LEARNING, CLASSIFIER, CONTROLLER, BIOMETRICS, AUTHENTICATION

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

СІТУ – система інженерно-технічного укріплення;
СКУД – система контролю та управління доступом;
СОВН – система охоронного відеоспостереження;
СОТС – система охорони та тривожної сигналізації;
ШІ – штучний інтелект;
ШНМ – штучні нейронні мережі;
SVM – Support Vector Machine;

ВСТУП

У сучасному суспільстві невпинного розвитку технологій, забезпечення безпеки стало ключовою задачею. Будь-хто турбується про свою безпеку, чи то пересічна людина, чи керівник компанії. Проте вимоги до методів та рівнів забезпечення такої безпеки досить різні. Фізичний захист об’єктів займає провідну ланку їх функціонування, адже це убезпечить від несанкціонованих вторгнень, крадіжок матеріальних цінностей та конфіденційних даних та зумовить надійний захист в цілому.

Таким чином, система безпеки – це автоматизована система для забезпечення охорони об'єктів різних типів. Основною функцією такої системи є унеможливлення неправомірного проникнення та гарантоване оперативне сповіщення власників та охоронних служб про спробу вчинення протиправних дій. В той же час важливим є не тільки виявлення, а й розпізнавання зловмисників на основі вбудованих систем ідентифікації.

Ефективний розв’язок такої задачі можливий за рахунок розумного оснащення охоронного об’єкта високонадійними засобами техніки, які не просто можна зламати. Також варто зауважити, що надійність системи охорони суттєво залежать від вартості технічних та програмних засобів, на що слід звертати увагу при виборі методів та засобів її розробки.

Основою програмно-технічних засобів захисту є процедури ідентифікації (користувач називає себе) та автентифікації (перевірки автентичності) користувачів. Процес автентифікації користувача можливий після пред’явлення чогось (пароль, особистий номер), володіння чимось (картка, ключ), частини самого користувача (біометричні характеристики) [1].

Досить часто підприємства для системи розпізнавання людей використовують електронні турнікети зі зчитувачами карт або відбитків пальців, проте такі системи характеризуються низькою надійністю. Це полягає у тому, що будь-хто може заволодіти карткою особи, чи то шляхом крадіжки чи перехоплення і проникнути на об’єкт. В той же час людина-працівник підприємства добровільно може передати картку іншій особі. Ідентифікація по відбитках пальців часто унеможливлює розпізнати людину, оскільки відбитки є нечіткі, погано зчитуються [2].

Однак розвиток науки в галузі цифрових технологій не стоїть на місці й суттєво впливає на всі сфери життєдіяльності людини, охоплюючи і змінюючи уявлення про зміст персональних даних, як до способу ідентифікації певної особи.

Найпоширенішим та часто використовуваним способом ідентифікації людини є зображення, проте такі біометричні характеристики, як відбитки пальців, сітківки ока, голос, а також васкулярна автентифікація як спосіб ідентифікації не менш відомі [3].

Цифрові технології уможливлюють збір біометричних характеристик, перетворення їх у певний код та використання як новий вид персональних даних. Однак процес впровадження та використання нових складових компонентів персональних даних пов'язаний із ризиками їх крадіжки та незаконного використання. Таким чином, біометричні характеристики як частина персональних даних виконують функції: ідентифікації людини та захист інших видів персональних даних.

Варто зауважити, що система безпеки це загальний термін, який характеризує типи систем, в саме: пожежної та охоронної сигналізації, контролю доступу, відеоспостереження, охоронних систем активної дії. Чимало підприємств почали використовувати як засіб охорони – систему контролю та управління доступом (СКУД). Це біометричні системи ідентифікації, що здатні суттєво підвищити безпеку підприємства та його працівників за рахунок зручності, підвищеної точності та якості [4; 5].

Таким чином, метою кваліфікаційної роботи є розробка моделі інформаційної охоронної системи на основі біометричної ідентифікації з метою підвищення надійності на основі аналізу вразливостей та оцінки загроз.

Для досягнення поставленої мети потрібно виконати такі задачі:

- охарактеризувати поняття біометричних даних та біометричної автентифікації;
- проаналізувати методи біометричної автентифікації ;
- провести аналіз вимог до сучасних систем охорони;
- проаналізувати існуючі системи охорони;
- провести аналіз вимог до сучасних систем охорони;
- розробити модель інформаційної охоронної системи на основі біометричної автентифікації;
- дослідити надійність розробленої моделі охоронної системи та запропонувати технології підвищення рівня безпеки

Об’єкт дослідження. Процес інтеграції біометричної автентифікації в охоронних системах.

Предмет дослідження. Модель системи охорони на основі біометричної ідентифікації.

Методи дослідження. Порівняльного аналізу при дослідженні існуючих охоронних систем, аналізу та синтезу при розробці моделі охоронної системи, комп’ютерного моделювання при дослідженні системи.

Матеріал дослідження. Базується на нормативних документах, книгах, статтях, матеріалах конференцій, інтернет-джерелах.

Наукова новизна. Розроблено модель інформаційної охоронної системи, яка, на відміну від існуючих, ґрунтується на процедурі біометричної ідентифікації з використанням машинного навчання з алгоритмом SVM, що підвищує точність розпізнавання та надійність системи в цілому.

Практичне значення. Система управління допуском на основі біометричної ідентифікації.

Апробація дослідження. Основні результати кваліфікаційної роботи будуть використані у розробці самостійної домашньої охоронної системи.

1 АНАЛІЗ ІНФОРМАЦІЙНИХ ОХОРОННИХ СИСТЕМ

1.1 Аналіз автентифікації на основі біометричних даних

Біометрія – це технічний термін, що означає фізичні чи поведінкові особливості людини. Біометрична автентифікація – це певна концепція безпеки даних [8]. Програмні системи для біометричної автентифікації розробляють модель на основі даних, яка представляє людину. На основі такої моделі та біометричної інформації системи безпеки можуть автентифікувати доступ до програмних додатків та інших мережевих ресурсів. Варто зазначити, що біометрична автентифікація стає популярним компонентом стратегій багатофакторної автентифікації, оскільки поєднує строгу автентифікацію із зручним інтерфейсом користувача.

Довшим часом імена користувачів та паролі були основним засобом безпеки. Проте зловмисні зломи у великих фінансових та комерційних компаніях призвели до крадіжки мільйонів комбінацій імені користувача та пароля. Варто поєднати це із тенденцією повторювання паролів для кількох облікових записів і масштаб вразливості стане очевидним.

Системи біометричної автентифікації менш схильні до зазначеної вразливості, оскільки біометричні дані користувача унікальні. Зловмиснику дуже складно обманним шляхом відтворити сканування відбитка пальця або особи людини, якщо вони виконуються надійними засобами із жорстким виявленням підробки, проте для автентифікації відповідного користувача потрібна лише одна секунда. Зважаючи на це, біометрію вважають зручнішою та безпечнішою, ніж паролі.

Біометрична автентифікація не залежить від таємності біометричних характеристик, а покладається на складність уособлення живої людини, що представляє цю характеристику, пристрою захоплення.

Зважаючи на вищезазначене, біометрична автентифікація або автентифікація на основі біометричних характеристик має такі переваги:

- зручно використовувати завдяки відсутності контакту;
- неможливо втратити або забути, тому що біометричні характеристики є частиною користувача, що ідентифікується;
- суттєво утруднені передача та підробка біометричних характеристик.

Методи біометричної автентифікації поділяють на статичні та динамічні [1]. До статичних відносять автентифікацію: по відбитку пальця, по райдужній оболонці ока, п сітківці ока, по геометрії руки, по геометр обличчя, по термограмі обличчя. Одним із поширених методів є розпізнавання обличчя – відома форма біометричної автентифікації, популярна у багатьох шпигунських драмах та науково-фантастичних казках, у популярних засобах масової інформації. Використовують розпізнавання осіб у повсякденному житті, щоб дізнаватися про друзів та родичів, відрізняти незнайомців. При автентифікації принципи такого процесу оцифровуються, щоб дозволяє, наприклад, мобільному пристрою розпізнавати обличчя.

Програмне забезпечення для розпізнавання облич аналізує геометрію обличчя, включаючи відстань між очима, відстань між підборіддям і носом і т. д., щоб створити зашифровану цифрову модель для особистих даних. При автентифікації інструмент розпізнавання облич відскановує особу в реальному часі і порівнює модель із тією, що зберігається в системі. Слід зауважити, що зазначений метод перебуває на стадії розвитку та характеризується певними ризиками.

Розпізнавання облич – один із найбільш зручних способів біометричної автентифікації. Погляд у камеру пристрою вимагає меншого тертя, ніж сканування відбитка пальця або коду автентифікації, що є його перевагою. Проте не всі системи розпізнавання осіб розроблено однаково, що збільшує ризик підробки.

Системи розпізнавання осіб із «активним визначенням життєздатності» вимагають, щоб користувач рухав головою, моргав чи виконував інші дії під час перевірки запиту, що створює певні незручності для користувача. Такий процес зловмисник може проаналізувати й обійти. «Пасивне виявлення живучості» відбувається за лаштунками, що не заважає користувачу і його складніше ідентифікувати.

Розпізнавання відбитків пальців – це один із методів біометричної автентифікації. Офіцери правоохоронних органів роками використали відбитки пальців як форму ідентифікації. Сканер відбитків пальців працює за тими ж принципами, але весь процес оцифровується. Відбитки пальців кожної людини є унікальними. Таким чином, аналізуючи гребені та візерунок відбитка, сканери відбитків пальців створюють цифрову модель, яка порівнюється з майбутніми спробами автентифікації. Продуктивність системи і таким методом автентифікації може знизитися через якість відбитка пальця або поточні умови, наприклад, мокрі або брудні пальці.

Автентифікація на основі розпізнавання очей базується на скануванні ока. Сканування використовує розпізнавання райдужної оболонки чи сітківки ока для ідентифікації користувачів. При скануванні сітківки автентифікатор на короткий

час спрямовує світло в око, щоб висвітлити унікальний візерунок кровоносних судин в оці. Порівнюючи цей шаблон, інструмент розпізнавання очей може порівнювати очі користувача з очима оригіналу. Сканування райдужної оболонки ока працює аналогічним чином, проте аналізують кольорові кільця, виявлені в райдужній оболонці [9].

Розпізнавання очей може бути швидким і точним, як розпізнавання осіб, але менш зручним. При сонячному світлі буває складно отримати зразок для порівняння (зіниці звужуються). Реалізація таки систем часто вимагає спеціалізованого обладнання.

В ході автентифікації за геометрією руки використовується сканування форми кисті руки (вигини, довжина, товщина пальців, ширина та товщина тильного боку руки, відстань між суглобами та структура кістки). Цей метод не є безпомилковим через чутливість сканерів до проявів різних захворювань суглобів [8].

Автентифікація з геометрії обличчя відрізняється найбільш складною технічною реалізацією. Метод ґрунтується на побудові тривимірної моделі особи, для цього виділяються контури різних елементів обличчя, обчислюється відстань між ними. Для визначення унікального шаблону необхідно від 12 до 40 характерних елементів. Унікальний шаблон має враховувати безліч варіацій зображення.

Автентифікація за термограмою особи – метод, який сьогодні не є широко поширеним. Найбільш стійким серед біометричних характеристик обличчя є зображення кровоносних судин. Метод ґрунтується на створенні температурної карти (термограми) шляхом сканування обличчя в інфрачервоному світлі. Дослідженнями доведено, що термограма обличчя є унікальною біометричною характеристикою. Незмінність та відкритість біометричних характеристик, що використовуються у статичних методах, допускають підробку біометричного ключа.

Динамічні методи ґрунтуються на поведінкових характеристиках людей [6]. В теперішній час широко використовується автентифікація голосу. Унікальний голос кожної людини визначається довжиною його голосового тракту та формою носа, рота та гортані. Сам процес обробки голосу розбивають на такі етапи: передопрацювання сигналу, виділення критеріїв, розпізнавання [12]. Основними перевагами даного методу є простота у використанні та реалізації, загальнодоступність та наявність різних способів побудови шаблону. Проте часто фоновий шум, настрій, застуда, бронхіт або інші поширені захворювання можуть спотворювати голос та перешкоджати автентифікації. Тож у громадських місцях людині може бути незручно говорити вголос, що є недоліком такого методу автентифікації.

Автентифікація за рукописним почерком ґрунтується на специфіці рухів руки під час підписання документів. Виділяють два способи обробки даних про підпис: аналіз самого підпису, аналіз динамічних характеристик написання. Ведуться також дослідження автентифікації користувача за клавіатурним почерком, що, безперечно, має перспективи подальшого розвитку.

Традиційно біометрична автентифікація з геометрії особи відноситься до класу, що ґрунтується на статичних методах. Прийнято також вважати, що геометрія обличчя має низьку унікальність, що згодом призводить до значних помилок першого та другого роду. Слід зазначити, що навіть якщо первинна автентифікація пройшла без помилок і система сприйняла «свого» за «свого», це не дасть 100% захисту інформації [8].

Перспективним методом біометричної автентифікації є оцінка емоційного стану та міміки людини шляхом постійного прихованого моніторингу за допомогою web-камери [1]. Емоційний стан та міміка людини постійно змінюються залежно від зовнішніх та внутрішніх факторів. При цьому вираження емоцій – це неусвідомлений процес (людина не несе відповідальності за свій емоційний стан), що практично виключає можливість приховування та заміни цієї біометричної характеристики.

Зловмисник виражатиме певні емоції, які відрізнятимуться від повсякденних. З огляду на це обґрунтованим є вивчення зміни емоційного стану та міміки користувачів для зниження ймовірності помилок першого та другого роду, а також підвищення захисту при спробі заподіяння шкоди співробітниками, які успішно пройшли процедури ідентифікації та автентифікації.

Біометричні системи автентифікації є досить зручними для користувачів. На відміну від парольних або ключових систем автентифікації, біометричні використовують параметри, які неможливо забути або втратити. Проблеми безпеки автентифікаційних даних немає. Біометричний метод автентифікації за голосом простий у застосуванні. Він не вимагає спеціальної апаратури, достатньо лише мікрофона та звукової плати. Нині технологія розвивається, оскільки цей метод широко використовується у сучасних бізнес-центрах. Основним недоліком методу голосової автентифікації є низька точність. Голос людини може змінюватися залежно від стану здоров'я, настрою, віку тощо. Крім того, не варто забувати про сторонні шуми. Через високу ймовірність помилок другого роду його застосовують в основному в приміщеннях середнього рівня безпеки [1].

Біометрична автентифікація використовується в багатьох додатках у багатьох галузях для підвищення безпеки та ефективності існуючих процесів. Наприклад, деякі авіакомпанії та аеропорти пропонують своїм пасажирам можливість зареєструватись на рейс за допомогою розпізнавання осіб. Так само готелі та готельні компанії запроваджують самостійну реєстрацію за допомогою біометричної автентифікації.

Безпека та автентифікація для мобільного банкінгу. Фінансові установи використовують біометричну автентифікацію як частину стратегії двофакторної автентифікації або багатофакторної автентифікації для захисту банку та його клієнтів від атак із захопленням даних з рахунків.

Біометрична автентифікація у вигляді сканерів відбитків пальців, сканерів райдужної оболонки та розпізнавання осіб допомагає лікарням підтвердити особистість пацієнта, гарантувати, що особи, які здійснюють огляд, мають доступ до потрібної медичної інформації та багато іншого.

Найбільш очевидною перевагою біометричної автентифікації для користувачів є зручність. При використанні традиційних методів перевірки через SMS або електронну пошту користувачам доводиться перевіряти вхідні повідомлення та вручну вводити код. Ці способи автентифікації створюють додаткові проблеми, якщо користувач втрачає доступ до свого номера телефону або електронної пошти. Для відновлення доступу може знадобитися звернення до служби підтримки.

Біометрія робить автентифікацію набагато зручнішою, замінюючи традиційні процеси простим поглядом чи дотиком до екрану. За даними Apple, ймовірність того, що випадкова людина, з одягненою маскою або без неї, подивиться в камеру вашого пристрою і увійде до системи за допомогою Face ID, становить менше 1 до 1 000000 [7].

Біометрія забезпечує унікальний безпековий рівень. На відміну від кодів та паролів, ваше обличчя та відбитки пальців завжди з вами. Це дві речі, які досить складно вкрасти або використовувати в шахрайських цілях, таких як фішинг. Наприклад, шахрай, який одержав доступ до електронної пошти користувача, може легко перехопити коди перевірки 2FA. Однак під час спроби входу він не зможе використовувати особу або відбитки пальців користувача.

Підробка біометричної інформації – це складний процес, набагато складніший, ніж крадіжка паролів або PIN-кодів. З погляду користувачів, така система перевірки знижує ризики та дає впевненість у тому, що їхні засоби не потраплять у чужі руки.

З точки зору організацій, біометрична автентифікація дозволяє зрозуміти, чи справді вхід виконується власником облікового запису. Звичайне сканування особи може стати тим вирішальним фактором, який не дозволить зловмиснику зламати обліковий запис і завдати непоправної шкоди. За допомогою цих інструментів користувачі захищають від шахраїв свої активи та екосистему загалом.

Із вищепроведеного аналізу впливає, що біометрична ідентифікація є досить ефективною з точки зору точності та надійності при розробці систем безпеки. У наступному підрозділі охарактеризуємо існуючі системи охорони.

1.2 Характеристика систем охорони

Надійний спосіб забезпечити безпеку будь-якого – встановлення охоронної системи. Якщо для одного об’єкта застосовують лише один тип охорони – захист від проникнення, то для іншого система безпеки може містити різні підсистеми: спостереження за підходами та під'їздами до будівлі, охорона периметра, охорона внутрішніх приміщень.

Таким чином існує чотири основних види охоронних систем [2, 12]:

- СОТС – система охорони та тривожної сигналізації. При проникненні в приміщення спрацьовують тривожні датчики, і на пульт охоронної служби надходить сигнал. Це найпоширеніший вид охорони квартир та офісів у багатоповерхових спорудах.
- СКУД – система контролю та управління доступом. З її допомогою можна керувати замками, приводами зовнішніх воріт, здійснювати контроль доступу до приміщень, куди вхід для сторонніх осіб не бажаний.
- СОВН – система охоронного відеоспостереження, що здійснюється за допомогою спеціальних камер.
- СІТУ – система інженерно-технічного укріплення. До неї відноситься висота паркану, міцність вікон і воріт, надійність дверних замків, наявність бункера або потаємних укріплених кімнат, тобто все, що може ускладнити зловмисникам доступ до приміщення.

«Розумний дім» – технологія управління всіма охоронними системами за допомогою одного пристрою.

За допомогою системи охорони та тривожної сигналізації можна здійснювати цілодобовий контроль за будь-яким об'єктом. Таку систему поділяють на два типи:

Пасивна система охоронної сигналізації. При несанкціонованому вторгненні на територію подається світловий і звуковий сигнал тривоги. При цьому ні на пульт охорони, ні до власника тривожний сигнал не надходить. Система розрахована лише на психологічний вплив на злочинців, що знижує її ефективність.

При активній СОТС об'єкт обладнаний датчиками, що посилають тривожний сигнал на контрольну панель, а звідти на пульт спеціалізованої охоронної організації. Такий пульт здатний приймати сигнали з кількох об'єктів. Після отримання сигналу охоронна структура в найкоротші терміни прибуває на місце передбачуваного проникнення. У системі пультової охорони є і так звана кнопка тривоги, при натисканні на яку сигнал також надходить на пульт управління. Такі кнопки встановлюють у банках, магазинах для захисту від розбійних нападів.

Типова СОТС складається з кількох елементів:

- центральний приймально-контрольний пристрій, що приймає сигнали від датчиків;
- датчики, або оповіщувачі;
- пристрої оповіщення – радіо- та GSM-передавачі, світлозвукові пристрої;
- пульт керування;
- пристрій контролю стану оповіщувачів та кабельних ліній ;
- джерело живлення – блоки та перетворювачі, що забезпечують безперебійну роботу системи навіть при відключенні від основної системи електроживлення.

Одним із основних елементів СОТС є датчики. Саме вони відповідають за оперативне реагування системи на вторгнення, тому зупинимося на їх аналізі детальніше. Інфрачервоні датчики руху реагують на будь-які переміщення в межах об'єкта контролю. Деякі моделі здатні надсилати сигнал на включення камер спостереження. Датчик складається із спеціального інфрачервоного передавача та приймача, який реагує на зникнення сигналу в тому випадку, якщо певна оптична вісь перекривається порушником. До речі, якщо в будинку є домашні тварини, то рекомендовано встановити спеціальний датчик, інакше охорона приїжджатиме щоразу, як тільки вихованець потрапить у його поле дії.

Ємнісний датчик створює в об'єкті контролю поле з певною ємністю. При попаданні у поле будь-якого предмета ємність поля змінюється, викликаючи спрацьовування сигналізації. Приблизно так само діють дротовохвильові датчики, що генерують електромагнітне поле між двома зонами. Прилади спрацьовують під час перетину цього поля. На рух реагують також радіопроменеві, радіохвильові та пасивні ІЧ-датчики.

Вібраційні датчики посилають сигнал на пульт керування у разі виникнення вібрації: наприклад, при спробі проламати стіну або розкрити сейф, при розбиванні вікна. На звук розбитого скла, як і на будь-який інший гучний звук, реагує акустичний датчик. Магнітоконтактні датчики спрацьовують при спробі відчинити вікна або двері. Зазвичай, СОТС включає відразу кілька типів охоронних оповіщувачів, щоб забезпечити надійний захист від усіх типів проникнення.

Після ідентифікації з контролера подається команда виконавчого пристрою – дозволити допуск чи ні. Виконавчі пристрої – це замки, турнікети, приводи воріт, шлагбауми, тобто все те, що є фізичною перешкодою для несанкціонованого доступу.

Система охоронного відеоспостереження передбачає спостереження за об’єктом за допомогою відеокамер. Спостерігати одночасно можна за кількома пунктами: наприклад, головним та додатковим входом до будівлі, брамою, периметром, окремими приміщеннями. Головне завдання СОВ – забезпечення

безпеки об’єкта саме шляхом візуального спостереження та отримання наочної інформації про обстановку на об’єкті. Крім актуальної інформації СОВ уможливлює аналіз подій, що вже відбулися, так як вся інформація з камер, зазвичай, записується і її можна переглянути в будь-який час.

Сьогодні існує кілька типів відеоспостереження – домашнє, вуличне, приховане. За способом обробки сигналу системи діляться на аналогові, комбіновані та цифрові. Найбільш досконалими та затребуваними є цифрові, оскільки забезпечують високу якість зображення, високу швидкість доступу до архіву, можливість віддаленого перегляду та інтеграцію з іншими охоронними системами.

За способом передачі відеосигналу СОВ можуть бути провідними та бездротовими. Бездротові системи можна встановлювати в важкодоступних місцях, зате провідні надійніші та дешевші.

Камери СОВ також поділяють на види в залежності від різних технічних характеристик. Цифрові або IP-камери використовують цифровий сигнал, їх досить просто інтегрувати в мережу, тому цифрове відеоспостереження при необхідності легко можна масштабувати.

Роздільна здатність в цифрових камерах значно вища, ніж в аналогових, але іноді у високій роздільній здатності просто немає необхідності, тим більше, що вона вимагає ємних систем зберігання даних.

Окрім того, камери СОВ можуть бути зовнішніми та внутрішніми залежно від місця їх встановлення. За особливостями конструкції камери поділяють на розділити на корпусні, модульні, купольні, мініатюрні та відеовічка.

Система інженерно-технічної укріпленості будь-якого об’єкту – це захист від несанкціонованого проникнення шляхом посилення його конструктивних елементів. В ідеалі кожна система повинна мати 5 зон контролю:

- периметр території;
- периметр будівлі;
- приміщення, доступні відвідувачам;
- приміщення, доступні лише для власників будинку;
- приміщення з максимально обмеженим доступом.

Як засоби захисту периметра ділянки використовують паркани, ворота, пропускні пункти, шлюзи для перевірки та пропуску транспортних засобів. Для захисту будівлі – віконні ґрати та ґрати для посилення стін, тамбури, оснащені засобами безпеки, укріплені двері, замки.

Охоронна система «Розумний дім» – дозволяє стежити за станом будівлі практично звідусіль – потрібен лише будь-який пристрій із виходом в інтернет: планшет, смартфон чи ноутбук. Контроль у межах системи ведеться цілодобово, а сигнал надходить власнику і на пульт охорони відразу ж, як тільки відбувається будь-яка позаштатна ситуація. До речі, перевірити, чи увімкнена охоронна

сигналізація, також можна віддалено, за необхідності її можна ввімкнути дистанційно.

Крім охоронної функції, система розумного будинку за допомогою спеціальних датчиків стежить за станом протипожежної, газової та водопровідної систем, фіксує позаштатні ситуації та сповіщає власника. Також залежно від запрограмованих можливостей охоронної системи загалом існує можливість знеструмлення приміщення чи відключення газу.

За ситуацією в будівлі можна стежити з телефона або планшета в режимі он-лайн за допомогою охоронних відеокамер. Записи з них дублюються в «хмару», тому підробити або видалити їх неможливо. «Розумний дім» може імітувати присутність господаря, періодично включаючи світло в різних кімнатах, телевізор або радіо. За допомогою «розумної» системи безпеки власник можуть відповідати на дзвінки домофону, відчиняти ворота чи двері.

Таким чином, концепція інтелектуального захисту досить перспективна, тому охоронні системи в її межах удосконалюватимуться.

Сучасні автоматизовані СКУД призначені для захисту від несанкціонованого доступу до приміщень або певних зон та для ідентифікації осіб, які мають право доступу. Найбільш відомим пристроєм такого типу є домофон, що обмежує право доступу в під'їзд багатоквартирного будинку тим, хто не має електронного ключа типу Touch Memory [3].

Існують такі типи систем контролю доступу на основі біометрики:

- автономні (локальні) – призначені для невеликих офісів;
- мережеві – для великих компаній із кількома приміщення.

Автономна система передбачає керування однією або декількома точками доступу, не використовує комп’ютер керування, передачі на центральний пульт, контроль. Мережева СКУД включає можливість управління великою кількістю точок доступу та з’єднання всіх контролерів із керуючим комп'ютером.

СКУД також класифікують за кількістю точок контролю:

- малої місткості (до 16 точок доступу);
- середньої місткості (від 16 до 64 точок доступу);
- великої місткості (понад 64 точок доступу).

СКУД складається з декількох основних елементів: контролера, зчитувача, ідентифікатора та виконавчого пристрою. У контролері зберігається вся інформація про конфігурацію, режим роботи системи, права доступу і т.д. Зчитувач отримує інформацію, записану на ідентифікаторі, та передає її в контролер на обробку. Ідентифікатором може бути електронний ключ, карта доступу, а також біометричні термінали СКУД, що ідентифікують людину за відбитками пальців, райдужною оболонкою ока чи зображенням обличчя. Після ідентифікації з контролера подається команда виконавчого пристрою – дозволити

допуск чи ні. Виконавчі пристрої – це замки, турнікети, приводи воріт, шлагбауми, тобто все те, що є фізичною перешкодою для несанкціонованого доступу.

Таким чином, системи контролю доступу можуть включати такі елементи:

- пристрої перегородження, зокрема, електромагнітні та електромеханічні замки, шлагбауми, ворота, турнікети;
- ідентифікатори: картки чи брелоки, що використовують для отримання доступу, код, який потрібно ввести на клавіатурі, біометричні дані;
- контролери – ядро СКУД, пристрій, що визначає, чи надати доступ власнику ідентифікатора (рис. 1.1);

Рисунок 1.1 – Мережевий контролер доступу

- зчитувачі, які «зчитують» дані з ідентифікатора та передають їх контролеру (рис. 1.2);

Рисунок 1.2 – Безконтактний зчитувач

– термінали – це багатофункціональні механізми, що поєднують зчитувач і контролер в одному корпусі (рис. 1.3);

Рисунок 1.3 – Біометричний Wi-Fi термінал

- програмне забезпечення – спеціальний софт, необхідний для обробки даних, отриманих зчитувачем, формування звітів, початкового програмування і управління системою.

З проблемою побудови надійної та недорогої СКУД стикаються чимало організацій. Оскільки надійність СКУД залежить від ідентифікації, то постає питання про використання методів надійних та недорогих ідентифікацій. Атрибутні ідентифікатори (паролі, магнітні картки, електронні ключі), що відчужуються, мають серйозні недоліки – їх можна забути, втратити або дублювати.

Вирішенням цих проблем є використання біометричних методів ідентифікації. Ідентифікація за фізіологічними параметрами – структурою ока, параметрами пальців – надійна, має малу кількість хибно-позитивних допусків, проте потребує наявності спеціалізованого обладнання (сканерів відбитків пальців, райдужної оболонки ока, тощо), що в умовах малих організацій не завжди є економічно доцільним [3].

Ідентифікація за психологічними (поведінковими) параметрами дешевша та зручніша для користувачів – голос, клавіатурний почерк. Вимоги до апаратної частини при таких системах – це наявність мікрофона та звукової плати. На першому етапі аналоговий сигнал проходить дискретизацію і квантування, на другому – аудіодані піддаються фільтрації від шумів, оскільки система ідентифікації заснована на порівнянні чистих спектрограм. Після цього записані зразки аудіоданих користувачів підлягають швидкому перетворенні Фур'є.

Для обчислення міри подібності порівнюють масиви, що містять значення потужності фрагментів спектрограм, за максимальної міри подібності та перевищення встановленого порогу – виноситься позитивне рішення ідентифікації користувача. Поріг ідентифікації можна підлаштувати під умови конкретної організації залежно від її вимог до хибно-позитивних спрацьовувань. Хоч система є дешевою точки зору апаратного забезпечення, проте людський голос піддається змінам при застудах дихальної системи, його легко можна підробити.

Також варто зауважити, що ринок СКУД постійно розвивається. Наприклад, контролери СКУД характеризуються більшим об’ємом пам’яті, вищою продуктивністю, новою якісною логікою роботи, більшим рівнем захищеності. Загалом, зазначені контролери можна назвати спеціалізованими комп'ютерами, які оснащені свою операційною системою, що уможливлює задання сценаріїв роботи СКУД різної складності на апаратному рівні і гнучко реагувати на нові загрози шляхом зміни прошивки інтегрованої програмної оболонки.

Реалізація критично-важливих функцій охоронної системи не лише на програмному, а й на апаратному рівні суттєво підвищує надійність. За таких умов

навіть при відсутності зв’язку із сервером система буде працювати, зберігаючи найважливіший функціонал.

Варто відзначити наявність серед контролерів СКУД захищеного OSDP (Open Supervised Device Protocol)-інтерфейсу між зчитувачем і контролером. Протокол OSDP другого покоління із алгоритмом шифрування AES-128 забезпечує захист передачі даних між пристроями. Це уможливлює для користувача отримання зворотного зв’язку від зчитувача, своєчасно виявляти його несправність чи відключення. В той же час слід зауважити, що OSDP пристрої, які підключають до контролерів СКУД, можуть бути не лише зчитувачі, але й релейні модулі, електрозамки ,кнопки виходу, інша периферія [2, 3].

Розглянуті вище типи охоронних системах автоматизації процесів. В той же час варто зауважити, що існують також неавтоматизовані систем охорони. Такі системи передбачають залучення охоронця для здійснення контролю території та об’єкту, аналізу камер відеоспостереження та з умов небезпеки натискання на тривожну кнопку. Слід зауважити що, які б розумні технології не впроваджували, вони не замінять людини, яка здатна накопичувати досвід, що полегшує прийняття рішення у різних ситуаціях, що суттєво знижує івен хибних тривог. Проте такі системи не є надійними, оскільки існує людський фактор, також вимагають постійної перевірки та контролю якості охорони.

Таким чином, із вище проведеного дослідження випливає, що й ефективні сучасному світі є СКУД-системи, вимоги до яких розглянемо у наступному підрозділі.

1.3 Аналіз вимог до охоронних систем

Перш, ніж перейти до опису загасни вимог до розробки охоронних систем, розглянемо принципи побудови систем безпеки об’єктів, що лягають в основу формування вимог до охоронних систем в цілому, а також до складових її технічних засобів.

При розробці систем захисту об’єктів потрібно враховувати такі принципи [10]:

- адекватність розробленим моделям загроз;
- принцип зональності – охоронні системи повинні передбачати організацію та створення контрольних зон із обмеженим доступом, що забезпечує «багаторівневий» захист об'єктів та їх критичних елементів.
- рівень безпеки для всіх типів порушників та способів заподіяння шкоди;
- адаптивність – системи забезпечення безпеки не повинна перешкоджати функціонуванню об’єкта та бути адаптованою до технологічних особливостей його роботи.

Дотримання вищевказаних принципів при розробці систем безпеки уможливлює ефективний захист об’єктів через здатність технічних підсистем протидіяти нештатним ситуаціям на об'єкті, враховуючи виявлені загрози та моделі порушників.

Як показують дослідження, зважаючи на загальну ситуацію щодо порушень, фахівці служби безпеки висувають вимоги до систем охорони, зокрема:

- використання сучасних елементів в структурі системи;
- необхідність включення етапу проектування життєвий цикл побудови системи, як елемент технології;
- оптимізація структури та функціональних характеристик комплексу засобів захисту.

Таким чином, обґрунтування структури і функціональних характеристик системи на основі критерію оптимізації, формування вимог і рекомендацій до її складових є основою при розробці та функціонуванні систем забезпечення безпеки. В той же час, слід враховувати особливості об’єкта охорони. Зважаючи на це, можна сформувати загальні етапи розробки охоронних систем:

- категоризація об’єкта;
- вибір критерію ефективності системи;
- аналіз вразливості об’єкта для оцінки рівня функціональності системи;
- формування вимог до структури та функціональних характеристик засобів підсистем;
- розробка концепту системи;
- оптимізація структури системи;
- внесення змін до концепції розробки системи, враховуючи зміну загроз і умов функціонування об’єкта.

Категоризацію проводять з метою визначення рівня небезпеки об’єкта в умовах реалізації прийнятої моделі загроз та вирішують експертними методами шляхом визначення об’ємів збитків, який можуть бути у разі реалізації загроз. Результатом цього етапу є включення об’єкта до певної категорії, що визначає загальні вимоги до системи охорони та уможливлює задання кількісних критеріїв ефективності системи.

Для реалізації наступного кроку проводять аналіз вразливості об'єкта на основі експертних методів чи методів імітаційного моделювання. Такий аналіз включає:

- комплексну оцінку ефективності існуючої охоронної системи (якщо така наявна) при при прийнятій моделі загроз та пріоритетах необхідності захисту, порівнюючи отримані дані з заданими критеріями;
- розробку заходів щодо досягненню встановлених критеріїв;
- оцінка ефективності таких заходів.

Міри досягнення встановлених критеріїв ефективності є, так би мовити, вимогами до архітектури та функціональних характеристик системи охорони й завершують етап розробки концепту побудови системи. Результатами зазначеного етапу є:

- рекомендації щодо вибору структури системи та змісту організаційно-регламентуючих документів про забезпечення охорони об’єкта;
- проєкт організаційно-штатної структури та рекомендації з приводу організації охорони об’єкта;
- вимоги до інженерно-технічних засобів і підсистем, що є основою для розробки технічного завдання на проектування системи.

Такий аналіз потрібно проводити із залученням спеціальних організацій із кваліфікованими фахівцями та спеціалізованим програмно-методичним забезпеченням. За таких умов забезпечується об'єктивність т відповідна якість аналізу вразливості об’єкта та розробити ефективну охоронну систему, яка буде економічно доцільною для об'єкта захисту.

Існуючі способи оптимізації структури охоронної системи використовують різні методи та із залученням достатньо кваліфікованих аналітиків уможливлюють отримання прийнятних результатів. Зважаючи на зазначене, науково-методичний забезпечення розробки охоронної системи дозволяє отримати:

- адекватність системи моделі загроз;
- гнучкість щодо змін загроз та умов функціонування об’єктів;
- оптимальну архітектуру системи на основі критерію «ефективність-вартість»;
- оптимізацію структури охоронної системи в цілому.

Варто зауважити, що науково-методичний забезпечення обґрунтування і розробки основних вимог до охоронних систем забезпечення об’єктів є обов'язковим при розробці систем безпеки.

Тепер опишемо основні технічні вимоги до охоронних систем. Вони включають:

- функціональні вимоги до підсистем;
- вимоги стійкості та зовнішніх чинників;
- вимоги електричної безпеки;
- вимоги до електропостачання.

Окрім вищезазначеного, повинен бути встановлений порядок організації тестування, приймання охоронного комплексу в експлуатацію, організації дослідної експлуатації, проведення технічного нагляду.

Таким чином, для забезпечення повної безпеки в організаціях встановлюються системи контролю доступу та сканери безпеки, системи сигналізації та відеоспостереження, системи протипожежного захисту та інші методи захисту.

Створення сучасних систем охорони різних об'єктів, зазвичай, вимагає використання інтегрованих систем, до складу яких входять такі основні системи [3]:

- охоронно-тривожної сигналізації;
- пожежної сигналізації;
- контролю та управління доступом;
- охоронного телебачення;
- збору, обробки та відображення інформації;
- пожежогасіння та димовиведення;
- оповіщення та управління евакуацією;
- оперативний зв'язок;
- гарантованого електроживлення.

Структура класичної системи охорони виконується за класичною схемою та, як правило, складається з таких елементів:

- сервер або головний комп'ютер, де зберігаються та обробляються всі бази даних системи;
- робочі станції окремих систем (за потреби), які здійснюють обмін даними та командами з периферійними пристроями своїх підсистем та виконують попередню обробку одержуваної інформації;
- периферійні пристрої (контролери, розширювачі, пульти управління тощо), які безпосередньо на апаратному рівні взаємодіють зі своїми оповіщувачами, датчиками або виконавчими пристроями, а на інформаційному рівні зв'язують їх по локальному інтерфейсу з робочими станціями або з головним комп'ютером;
- оповіщувачі охоронної, тривожної, пожежної сигналізації, зчитувачі, клавіатури, телекамери, світлові та звукові оповіщувачі тощо;
- локальна мережа, що інформаційно зв'язує в єдиний інтегрований комплекс окремі системи, кожна з яких може функціонувати окремо та незалежно;
- мережеве, системне та прикладне програмне забезпечення сервера та робочих станцій, а також мікропрограмне забезпечення системних контролерів, контрольних панелей та модулів;
- система гарантованого електроживлення, яка включає:
  - електрощитову системи, підключену до мережі 220 В першої категорії та містить всі необхідні вхідні та вихідні силові автомати;
  - джерела безперебійного живлення, що забезпечують безперервне та якісне електроживлення всієї апаратури протягом заданого часу;
  - розведену по всьому об'єкту окрему мережу живлення з розміщенням при необхідності окремих джерел безперебійного живлення у

спеціально виділених приміщеннях, нішах чи шафах, що знаходяться під охороною.

З огляду на вищезазначене, випливає висновок, що сучасні системи охорони, що ґрунтуються на системах контролю управління доступом повинні працювати комплексно. Зважаючи на це, сформуємо формальні (основні) та неформальні вимоги до розробки архітектури охоронної системи.

Охоронна система має передбачати під'єднання до мережі на основі розумного центрального комп'ютера – хаба (Hub). Такий пристрій приймає сигнали від датчиків, аналізує їх та приймає рішення щодо сповіщення віддаленого сервера про стан об'єкта охорони. Таким чином, хабом може бути будь-який апаратний засіб, який має функцію безпровідного підключення, та включати програмне забезпечення, яке керуватиме роботою датчиків, опрацьовує дані із них та відправляє сповіщення про можливу тривогу на віддалений сервер.

Основними превагами зазначеного засобу є:

- працездатність системи навіть при неякісному зв'язку: достатня швидкість GPRS 0,5 кбіт / сек;
- постійне тестування і просте налаштування за рахунок двостороннього зв'язку з пристроями;
- автоматичне оновлення прошивки і програмного забезпечення;
- керування брелоком, через браузер, із застосуванням додатків для смартфонів (iOS/Android);
- підтримка платформ для мобільних додатків: iOS 7.1 і вище, Android 4.1 і вище;
- збереження історії всіх подій, зазначених системою;
- перегляд даних з відеокамер у реальному часі;
- всі роз'єми і кнопки заховані в корпус;
- ввімкнення тривоги при відключенні зовнішнього електроживлення;
- працездатність на резервному живленні сягає 15 годин;
- пінгувальна перевірка відгуку датчиків частотою від 12 с.;
- наявність підсистем виявлення та запобігання глушіння, шифрування каналів, автентифікації з метою захисту від підробки пристроїв;
- зв'язок із кожним пристроєм мережі на основі вільної частоти та функція перебудування за умов збігу;
- модуль GSM продукує резервний канал зв'язку за умов відключення інтернету.

Система має включати механізм накопичення даних з датчиків, включаючи аналіз, опрацювання та формування звітностей (базу даних). Такий модуль накопичення даних повинен зберігатися на відділеному сервері у безпечному місці. З метою забезпечення безпеки усі потоки даних мають бути

зашифрованими, а датчики промаркованими. Датчики повинні включати можливість комунікації із хабом про стан та зібрані дані, що вимагає їх навчання.

Підсистема аналізу та опрацювання даних є ключовою у архітектурі охоронної системи, яка розміщується на центральному комп'ютері. Вона повинна включати механізм приймання даних із датчиків, аналізу на основі бази знань, прийняття рішень щодо стану об'єкта та комунікації із віддаленим сервером.

На основі аналізу вимог до сучасних охоронних систем можна перейти до розробки моделі інформаційної охоронної системи на основі процедури біометричної автентифікації.

1.4 Поняття штучного інтелекту

Інтегрування біометрики в охоронні системи можна назвати відносно новою галуззю та не досить розповсюдженою в охоронних системах "середнього" рівня - житлові будинки, бізнес приміщення, школи, садки тощо.

Біометрика включає досить велику кількість фізичних і поведінкових рис, які використовуються для ідентифікації людини: тривимірна фотографія обличчя (або тіла), зразок голосу, відбитки пальців, малюнок вен руки, група крові, фото рогівки ока тощо.

Приклади, де можна використовувати технологію розпізнавання обличчя:

- Фінанси: такі банківські гіганти, як HSBC і Chase, уже використовують FaceID від Apple, щоб дозволити клієнтам входити у свої мобільні банківські програми. Інші фінансові установи тестують розпізнавання облич, щоб дозволити клієнтам використовувати камери свого телефону для схвалення онлайн-покупок.
- Страхування: Cigna дозволяє клієнтам у Китаї подавати заяви про медичне страхування, використовуючи свої фотографії замість письмового підпису. Страхова компанія каже, що цей спосіб скоротив страхове шахрайство.
- Водіння: автовиробники тестують технологію розпізнавання облич, щоб допомогти зменшити кількість крадіжок автомобілів. Розглянемо проект Mobil: Ford і Intel тестують проект, у якому камера приладової панелі використовує розпізнавання обличчя для ідентифікації основного водія автомобіля та, можливо, інших авторизованих водіїв. Технологія може запобігти запуску автомобіля, якщо за кермом сидить хтось інший, ніж законний водій.
- Подорожі: British Airways використовує розпізнавання облич, щоб пасажирам із США було легше сісти на рейс. Пасажири можуть просканувати обличчя за допомогою камери, щоб підтвердити їхню

особу. Таким чином, вони можуть сідати на свої рейси без пред'явлення паспорта чи посадкового талона.

Як і кожна технологія, вона має як переваги, так і недоліки (табл. 1.1).

Таблиця 1.1 - Переваги та недоліки технології розпізнавання обличчя.

Переваги	**Недоліки**
Пошук зниклих людей: за допомогою розпізнавання облич правоохоронні органи можуть розшукувати зниклих дітей, іноді навіть після того, як вони зникли протягом багатьох років.	*Загроза конфіденційності*: ви хочете, щоб ваше обличчя було збережено в базі даних, до якої можуть звертатися правоохоронні органи? Ви хочете, щоб роздрібні продавці зберегли зображення вашого обличчя? Якщо ні, ви не самотні. Багато критиків стурбовані тим, що розпізнавання обличчя є ще одним порушенням особистої приватності.
Ідентифікація злочинців: правоохоронні органи також можуть використовувати розпізнавання облич для ідентифікації злочинців або підозрюваних у злочинах.	*Помилка ідентифікації:* розпізнавання обличчя не ідеальне. Що робити, якщо правоохоронний орган помилково ідентифікує вас як підозрюваного у вчиненні злочину, коли ви реєструєтесь на літак?
Безпечні польоти: аеропорти по всьому світу використовують розпізнавання облич, щоб ідентифікувати злочинців і потенційних загроз, коли вони входять в аеропорти або намагаються сісти на літак.	*Можна обманути*: злочинці можуть обдурити розпізнавання обличчя, одягнувши маски або замаскувавши обличчя. Це може знизити ефективність цієї технології.
Більш ефективні покупки: Роздрібні торговці можуть використовувати розпізнавання облич, щоб споживачам було легше розраховуватися. Замість того, щоб змушувати клієнтів платити готівкою або кредитом,	*Старіння знижує ефективність*: Дослідження показали, що в міру того, як люди старіють і їхні риси обличчя змінюються, розпізнаванню облич стає все важче їх ідентифікувати. Інші дослідження показали, що

роздрібні торговці можуть використовувати розпізнавання обличчя, щоб негайно стягувати плату за покупки з їхніх рахунків.	розпізнавання облич є менш ефективним у ідентифікації кольорових людей і жінок.

У створюваній охоронній системі в рамках цього проекту буде задіяна саме фотографія обличчя з подальшим його розпізнаванням, що є не найточнішим способом ідентифікації, але принаймні не потребує від людини ніяких додаткових дій. Такая ідентифікація називається системою розпізнавання облич.

Система розпізнавання облич — це технологія, що здатна зіставляти людське обличчя з цифрового зображення чи відеокадру з базою даних облич. Така система зазвичай використовується для автентифікації користувачів за допомогою служб верифікації ідентифікаторів і працює шляхом точного визначення та вимірювання рис обличчя на певному зображенні.

Системи розпізнавання обличчя сьогодні використовуються в усьому світі урядами та приватними компаніями. Їх ефективність різна, і деякі системи раніше були скасовані через їх неефективність. Використання систем розпізнавання облич також викликало суперечки, стверджуючи, що системи порушують конфіденційність громадян, зазвичай роблять неправильну ідентифікацію, заохочують гендерні норми та расове профілювання та не захищають важливі біометричні дані. Поява синтетичних засобів масової інформації, таких як deepfakes, також викликала занепокоєння щодо їх безпеки. Ці заяви призвели до заборони систем розпізнавання обличчя в кількох містах США. Через зростаючу стурбованість суспільства Meta оголосила, що планує закрити систему розпізнавання облич Facebook, видаливши дані сканування облич понад одного мільярда користувачів. Ця зміна стане одним із найбільших змін у використанні розпізнавання обличчя в історії технології.

Аналіз обличчя також виконується навіть на простих процесорах, а вся система може обійтися у невелику ціну.

Розпізнавання обличчя складається із трьох етапів:

- Є фотографія або відео на якому знаходиться обличчя. Воно може бути одне або у натовпі. Обличчя може дивитися прямо або повернуто у профіль.
- Програма розпізнавання обличчя зчитує геометрію обличчя. Ключові фактори включають відстань між очима та відстань від чола до підборіддя. Програмне забезпечення визначає орієнтири обличчя — одна система визначає 68 із них — які є ключовими для розрізнення вашого обличчя. Результат: цифровий підпис обличчя.

– Підпис обличчя — математична формула — порівнюється з базою даних відомих облич. У нашому випадку це обличчя власника системи.

Другий та третій етапи покладаються на плечі штучного інтелекту, а саме - Computer Vision (“Комп’ютерний зір” або “Комп’ютерне візуальне сприйняття”).

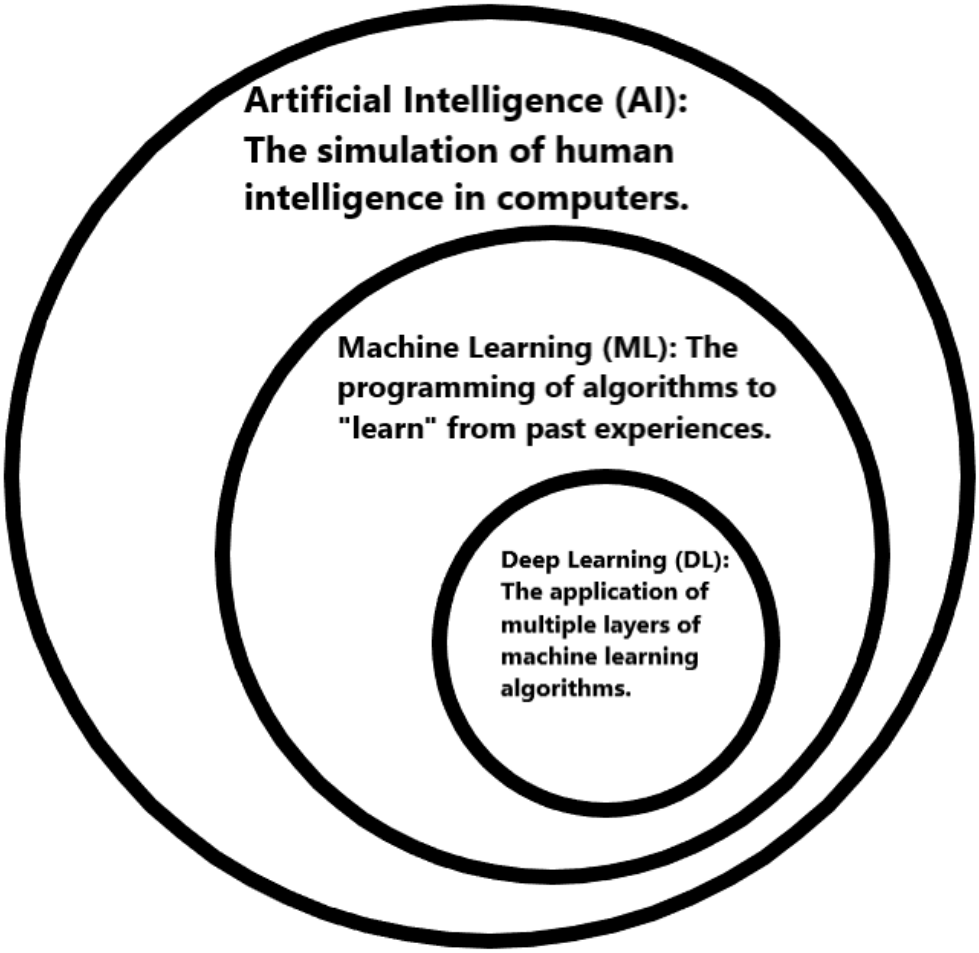


Рисунок 1.4 - Галузі штучного інтелекту.

Штучний інтелект — це інтелект, що сприймає, синтезує та виводить інформацію завдяки техніці, на відміну від інтелекту тварин і людей. Теорія та розробка комп’ютерних систем, здатних виконувати завдання, які зазвичай вимагають людського інтелекту, такі як візуальне сприйняття, розпізнавання мови, прийняття рішень та переклад між мовами. До поняття “штучного інтелекту” також відносять такі поняття як Machine Learning, Deep Learning та Computer Vision (рис. 1.4).

Машинне навчання — це галузь досліджень, присвячена розумінню та розробці методів, які «навчаються», тобто методів, які використовують дані для підвищення ефективності результатів певного набору завдань. Його розглядають як частину штучного інтелекту. Алгоритми машинного навчання створюють модель на основі вибіркових даних, відомих як навчальні дані, щоб робити прогнози чи приймати рішення без явного програмування для цього. Алгоритми

машинного навчання використовуються в широкому спектрі програм, таких як медицина, фільтрація електронної пошти, розпізнавання мовлення, сільське господарство та комп'ютерний зір, де важко або неможливо розробити звичайні алгоритми для виконання необхідних завдань. Підмножина машинного навчання тісно пов'язана з обчислювальною статистикою, яка зосереджується на прогнозуванні за допомогою комп'ютерів, але не все машинне навчання є статистичним навчанням. Вивчення математичної оптимізації надає методи, теорію та області застосування в галузі машинного навчання. Інтелектуальний аналіз даних є спорідненою галуззю дослідження, яка зосереджена на дослідницькому аналізі даних за допомогою неконтрольованого навчання. Деякі реалізації машинного навчання використовують дані та нейронні мережі таким чином, щоб імітувати роботу біологічного мозку. Застосовуючи в бізнес-проблемах, машинне навчання також називають прогнозною аналітикою.

Глибоке навчання (також відоме як глибоке структуроване навчання) є частиною ширшого сімейства методів машинного навчання, заснованих на штучних нейронних мережах. Навчання може бути контрольованим, напів контрольованим і неконтрольованим.

Архітектури глибокого навчання, такі як глибокі нейронні мережі, мережі глибоких переконань, глибоке навчання з підкріпленням, рекурентні нейронні мережі, згорточні нейронні мережі та трансформатори, застосовувалися в таких областях, як комп'ютерне бачення, розпізнавання мови, обробка природної мови, машинний переклад, біоінформатика, дизайн ліків , аналіз медичних зображень, кліматологія, інспекція матеріалів і програми настільних ігор, де вони дали результати, які можна порівняти з результатами експертів, а в деяких випадках навіть перевершити їх.

Штучні нейронні мережі (ШНМ) були натхненні обробкою інформації та розподіленими комунікаційними вузлами в біологічних системах. ШНМ мають різні відмінності від біологічного мозку. Зокрема, штучні нейронні мережі мають тенденцію бути статичними та символічними, тоді як біологічний мозок більшості живих організмів є динамічним та аналоговим [10].

Прикметник «глибокий» у глибокому навчанні відноситься до використання кількох рівнів у мережі. Рання робота показала, що лінійний перцептрон не може бути універсальним класифікатором, але мережа з неполіноміальною функцією активації з одним прихованим шаром необмеженої ширини може. Глибоке навчання — це сучасний варіант, який стосується необмеженої кількості шарів обмеженого розміру, що дозволяє практичне застосування та оптимізоване впровадження, зберігаючи теоретичну універсальність у помірних умовах. У глибокому навчанні шарам також дозволяється бути неоднорідними та значно відхилятися від біологічно інформованих коннекціоністських моделей заради

ефективності, можливості навчання та розуміння, отже, «структурованої» частини.

Комп'ютерне бачення — це міждисциплінарна наукова галузь, яка займається тим, як комп'ютери можуть отримати розуміння високого рівня з цифрових зображень або відео. З точки зору інженерії, він прагне зрозуміти та автоматизувати завдання, які може виконувати зорова система людини.

Завдання комп'ютерного зору включають методи отримання, обробки, аналізу та розуміння цифрових зображень, а також вилучення даних великої розмірності з реального світу для отримання числової або символьної інформації, напр. у формах рішень. Розуміння в цьому контексті означає перетворення візуальних образів (введення сітківки ока) в описи світу, які мають сенс для мисленнєвих процесів і можуть викликати відповідні дії. Таке розуміння зображення можна розглядати як відокремлення символічної інформації від даних зображення за допомогою моделей, побудованих за допомогою геометрії, фізики, статистики та теорії навчання.

Наукова дисципліна комп'ютерного зору займається теорією створення штучних систем, які витягують інформацію із зображень. Дані зображення можуть приймати різні форми, наприклад відеоряди, зображення з кількох камер, багатовимірні дані з 3D-сканера або медичних скануючих пристроїв. Технологічна дисципліна комп'ютерного бачення прагне застосувати свої теорії та моделі до побудови систем комп'ютерного бачення.

Підобласті комп'ютерного зору включають реконструкцію сцени, виявлення об'єктів, виявлення подій, відстеження відео, розпізнавання об'єктів, оцінку 3D пози, навчання, індексування, оцінку руху, візуальне обслуговування, моделювання 3D сцени та відновлення зображення.

1.5 Ретроспективний аналіз

Розквіт технології розпізнавання облич - 1960-ті роки. Найпершими піонерами розпізнавання облич були Вуді Бледсо, Хелен Чан Вулф і Чарльз Біссон. У 1964 і 1965 роках Бледсо разом з Вольфом і Біссоном почали роботу з використанням комп'ютерів для розпізнавання людського обличчя. Через те, що проект фінансувало неназване розвідувальне агентство, значна частина їхньої роботи ніколи не була опублікована. Однак пізніше з'ясувалося, що їх початкова робота включала ручне позначення різних «орієнтирів» на обличчі, таких як центри очей, рот тощо. Потім вони математично оберталися комп'ютером, щоб компенсувати варіацію пози. Відстані між орієнтирами також автоматично обчислювалися та порівнювалися між зображеннями для визначення ідентичності. Ці найперші кроки до розпізнавання облич Бледсоу, Вольфа та Біссона були

серйозно ускладнені технологіями тієї епохи, але це залишається важливим першим кроком у доведенні того, що розпізнавання облич було біометричним.

Підвищення точності розпізнавання облич - 1970-ті роки. Продовжуючи початкову роботу Бледсо, естафету підхопили в 1970-х роках Голдштейн, Хармон і Леск, які розширили роботу, включивши 21 конкретний суб'єктивний маркер, включаючи колір волосся та товщину губ, щоб автоматизувати розпізнавання. Хоча точність підвищилася, вимірювання та розташування все ще потрібно було обчислювати вручну, що виявилося надзвичайно трудомістким, але все ще є прогресом у порівнянні з попередніми напрацюваннями.

Використання лінійної алгебри для розпізнавання обличчя – 1980-90-ті роки. Лише наприкінці 1980-х років ми побачили подальший прогрес у розробці програмного забезпечення для розпізнавання облич як життєздатного засобу для бізнесу. У 1988 році Сирович і Кірбі почали застосовувати лінійну алгебру до проблеми розпізнавання обличчя. Система, яка стала відома як Eigenface, показала, що аналіз функцій на колекції зображень обличчя може сформувати набір основних ознак (рис. 1.2). Вони також змогли показати, що для точного кодування нормалізованого зображення обличчя потрібно менше ста значень. У 1991 році Турк і Пентланд продовжили роботу Сировича і Кірбі, відкривши, як виявляти обличчя на зображенні, що призвело до перших випадків автоматичного розпізнавання обличчя. Цьому значному прориву завадили технологічні та екологічні фактори, однак він проклав шлях для майбутніх розробок у технології розпізнавання облич [11].

Рисунок 1.5 - Система Eigenface.

Програма FERET – 1990-mi/2000-mi роки. Агентство передових оборонних дослідницьких проектів (DARPA) і Національний інститут стандартів і технологій (NIST) запровадили програму технології розпізнавання облич (FERET) на початку 1990-х років, щоб заохотити комерційний ринок розпізнавання облич. Проект передбачав створення бази даних зображень обличчя. До тестового набору було включено 2413 нерухомих зображень обличчя, які представляли 856 людей. Сподівалися, що велика база даних тестових зображень для розпізнавання облич надихне на інновації та може призвести до більш потужної технології розпізнавання облич.

Промислові випробування розпізнавання облич – 2000-mi. На початку 2000-х років Національний інститут стандартів і технологій (NIST) розпочав промислове тестування розпізнавання облич (FRVT). Спираючись на FERET, FRVT були розроблені, щоб забезпечити незалежні державні оцінки систем розпізнавання обличчя, які були комерційно доступними, а також прототипів технологій. Ці оцінки були розроблені, щоб надати правоохоронним органам і уряду США інформацію, необхідну для визначення найкращих способів розгортання технології розпізнавання облич.

Конкурс на покращення технології розпізнавання облич - 2006. Започаткований у 2006 році головною метою програми Face Recognition Grand Challenge (FRGC) було просування та просування технології розпізнавання облич, призначеної для підтримки існуючих заходів з розпізнавання облич в уряді США. FRGC оцінив найновіші доступні алгоритми розпізнавання облич. У тестах використовувалися зображення обличчя високої роздільної здатності, 3D-сканування обличчя та зображення райдужної оболонки ока. Результати показали, що нові алгоритми були в 10 разів точнішими, ніж алгоритми розпізнавання облич 2002 року, і в 100 разів точнішими, ніж алгоритми 1995 року, що свідчить про прогрес у технології розпізнавання облич за останнє десятиліття.

Соціальні мережі - 2010 - поточний час. Ще в 2010 році Facebook почав впроваджувати функцію розпізнавання облич, яка допомагала ідентифікувати людей, чиї обличчя можуть бути зображені на фотографіях, які користувачі Facebook оновлюють щодня. Ця функція миттєво викликала суперечки в ЗМІ, викликавши низку статей про конфіденційність. Однак користувачі Facebook загалом не заперечували. Не маючи явного негативного впливу на використання або популярність веб-сайту, понад 350 мільйонів фотографій завантажуються та позначаються тегами за допомогою розпізнавання облич щодня.

iPhone X – 2017. Технологія розпізнавання облич стрімко розвивалася з 2010 року, і 12 вересня 2017 року став ще одним значним проривом для інтеграції розпізнавання облич у наше повсякденне життя. Це була дата, коли Apple випустила iPhone X — перші користувачі iPhone могли розблокувати за допомогою FaceID — маркетинговий термін Apple для розпізнавання обличчя.

NEC і розпізнавання обличчя. Прикордонний контроль, авіакомпанії, аеропорти, транспортні вузли, стадіони, мега-події, концерти та конференції. Біометрія відіграє все більшу роль не лише в поліцейському контролі в режимі реального часу та забезпеченні безпеки у дедалі більш людних і різноманітних місцях у всьому світі, але й у забезпеченні безперебійного та приємного досвіду для громадян, які їх відвідують. Будучи давнім піонером біометричних досліджень і рішень, NEC розробила мультимодальні технології, включаючи розпізнавання обличчя, райдужної оболонки ока та голосу, ідентифікацію за відбитками пальців і долонь, а також акустичну автентифікацію вуха, і доповнила їх штучним інтелектом і аналітикою даних для підвищення обізнаності про ситуацію. і сприяти ефективним діям у режимі реального часу або після події як у правоохоронних органах, так і в сферах, орієнтованих на споживачів. Розпізнавання обличчя часто може виявитися одним із найкращих біометричних показників, оскільки зображення можна робити, не торкаючись і не взаємодіючі з особою, яку ідентифікують, і ці зображення записуються та миттєво перевіряються за існуючими базами даних. Система розпізнавання обличчя NEC пропонує високоефективні масштабовані рішення для найвибагливіших вимог у

режимі реального часу чи після події. Завдяки функціям спостереження за особами, пошуку, ідентифікації та верифікації – усе це на одній платформі, його можна легко інтегрувати в існуючі системи спостереження, щоб отримувати обличчя в режимі реального часу, зіставляти їх із наявною базою даних або списком спостереження та створювати сповіщення в режимі реального часу, щоб допомогти зменшити ризики громадської безпеки. Завдяки здатності обробляти та аналізувати кілька каналів камери та тисячі облич за хвилину, потужна система розпізнавання облич NEC здатна вирішувати найбільші та найскладніші проблеми безпеки з неперевершеною ефективністю, чутливістю та сприйняттям.

Розпізнавання обличчя NEC у дії. У відповідь на бурхливий попит на різноманітні біометричні дані NEC розширила свій традиційний набір програм для розпізнавання облич від правоохоронних органів і забезпечення безпеки до нових сфер, і тепер може похвалитися понад 1000 активними системами в більш ніж 70 країнах і регіонах, включаючи поліцію, служби імміграційного контролю, національне посвідчення особи, банківська справа, розваги, стадіон, системи для проведення конференцій та багато іншого. Наразі багато технологічних рішень для розпізнавання облич прокладають шлях для першого використання в нових областях і місцях по всьому світу, як-от наскрізний досвід подорожей до аеропорту в США, спостереження за мега-подіями в Японії та на саміті ЄС. безпеки в Європі. У 2021 році NEC отримала нагороду Frost Radar™ Award 2020 Global Biometrics in Security Market Growth Innovation & Leadership Excellence Award від Frost & Sullivan. Ця нагорода відображає орієнтацію NEC на біометричні рішення, які надають компанії значні можливості для розвитку, за допомогою яких можна розробляти нові, широкомасштабні рішення ідентифікації та сценарії використання за межами традиційних біометричних ринків. Така зосередженість на рішеннях ідентифікації на основі біометрії дозволила NEC вийти за рамки традиційних державних або корпоративних клієнтів і перейти до стратегії B2C за допомогою індивідуальних вертикальних рішень.

Майбутнє технології розпізнавання облич. З наближенням 2022 року технологія розпізнавання облич продовжує розвиватися швидкими темпами, і використання цієї технології стає все більш поширеним. Від Pay by Face до вирішення проблем, спричинених збільшенням використання масок у світлі пандемії COVID-19. За правильного використання технологія розпізнавання облич може значно змінити наше життя та те, як ми відчуваємо багато речей.

1.6 Основні задачі дослідження

Основними задача дослідження є:

- ознайомлення з машинним інтелектом та його типами, а також з нейронними мережами та глибоким навчанням;
- знаходження математичної моделі алгоритмів розпізнавання та ідентифікації облич та їх програмна реалізація як компоненти серверного додатку;
- підготовка навчального набору зображень обличь для підвищення точності алгоритму ідентифікації обличчя та мінімізації відсотка хибних спрацювань;

Висновки за розділом

У даному розділі був проведений аналіз автентифікації на основі біометричних даних, де було розглянути методи такої автентифікації, її переваги та недоліки. Виконали характеристику кожного з видів систем охорони, а також навели конкретні вимоги до кожного з них.

Проведений аналіз технологій розпізнавання обличчя з урахуванням їх переваг та недоліків.

Ретроспективний аналіз показав значну складність процесу розпізнавання обличчя, урахування різних біометричних характеристик, тому задача вдосконалення існуючих систем є актуальною на сьогодні. Таким чином, в основу розробки проекту доцільно покласти методи машинного навчання, глибокого навчання та комп’ютерного бачення.

.

2 МАТЕМАТИЧНА МОДЕЛЬ РОЗПІЗНАВАННЯ ОБЛИЧЧЯ

2.1 Типи машинних навчань

Перед вивченням математичної моделі розпізнавання обличчя необхідно ознайомитися з основами машинного навчання.

Машинне навчання може бути з учителем, без учителя та з частковим залученням вчителя. Існують і інші види навчання, але ознайомимося лише з основними.

2.1.1 Навчання з учителем

У навчанні з учителем набір даних організований як колекція розмічених зразків $\{(x_i, y_i)\}_{i=1}^{N}$. Кожен елемент x_i із N називається вектором ознак. Вектор ознак - це вектор, у якому кожен вимір j = 1, ..., D містить значення, що описує деяку характеристику зразка. Це значення називається ознакою та позначається як $x^{(j)}$. Наприклад, якщо кожен зразок x у нашій колекції представляє людину, тоді перша ознака, $x^{(1)}$, могла б описувати її зростання в сантиметрах, друга ознака, $x^{(2)}$, могла б описувати її вагу в кілограмах,$x^{(3)}$ — стать і т. д. Для всіх даних у наборі ознака позиції j у векторі ознак завжди описує ту саму характеристику. Це означає, що якщо $x_i^{(2)}$ описує вагу деякого зразка x_i у кілограмах, тоді $x_k^{(2)}$ також описуватиме вагу в кілограмах кожного зразка x_k, k = 1, ..., N. Мітка y_i може бути елементом кінцевої множини класів {1, 2, ..., C}, речовим числом або складнішою структурою, такий як вектор, матриця, дерево або граф. Клас можна представити як категорію, якій належить зразок. Наприклад, якщо роль даних відіграють електронні листи і ви вирішуєте завдання визначення спаму, тоді ви могли б визначити два класи: {спам, не_спам}. Мета алгоритму навчання з учителем — на основі набору даних створити модель, яка приймає вектор ознак x на вході та повертає інформацію, яка дозволяє визначити мітку для цього вектора ознак. Наприклад, модель, створена з використанням набору даних людей, могла б приймати вектор ознак, що описують людину, і повертати ймовірність, що ця людина хвора на рак [13].

Процес навчання з учителем починається зі збору даних. Дані для цього виду навчання — це колекція пар (вхідні, вихідні). Вхідними даними може бути все, що завгодно, наприклад електронні листи, зображення або виміри, отримані від датчика. Вихідними даними зазвичай є дійсні числа або мітки (наприклад, "спам", "не_спам", "кішка", "собака", "миша" і т. д.). У деяких випадках вихідні дані можуть бути представлені векторами (наприклад, чотирма координатами кутів прямокутника навколо людини на зображенні), послідовностями (наприклад, [«прикметник», «прикметник», «сущий») для входу «велика гарна машина») або мати якусь іншу структуру.

Припустимо, ми будемо використовувати навчання з учителем для вирішення задачі визначення спаму. Збираємо дані, наприклад, 10 000 електронних листів, кожен з яких забезпечується міткою «спам» або «не_спам». Потім необхідно перетворити кожен електронний лист у вектор ознак.

На основі свого досвіду фахівець із аналізу даних вирішує, як перетворити сутність реального світу, таку як електронний лист, у вектор ознак. Часто для перетворення тексту на вектор ознак використовується метод, що називається мішком слів. Його суть полягає в тому, щоб взяти словник (припустимо, що він містить 20 000 слів, відсортованих за алфавітом) і домовитися, що у векторі ознак:

- перша ознака дорівнює 1, якщо електронний лист містить слово "а" (союз), і 0 в іншому випадку;
- друга ознака дорівнює 1, якщо електронний лист містить слово «аарон» (ім'я), та 0 в іншому випадку;
- …;
- ознака у позиції 20 000 дорівнює 1, якщо електронний лист містить слово «ящур» (хвороба), та 0 в іншому випадку.

Необхідно повторити описану процедуру для кожного електронного листа у колекції та отримати 10 000 векторів ознак (кожний вектор має розмірність 20 000) та міток («спам»/«не_спам»).

Тепер маємо вхідні дані, але вихідні мітки, як і раніше, мають вигляд простого тексту. Деякі алгоритми навчання вимагають перетворення міток у числа. Наприклад, деякі алгоритми потребують використовувати числа 0 (для представлення мітки “не_спам”) і 1 (для представлення мітки “спам”). Модель машинного навчання, яка використовується у цьому прикладі навчання з учителем, називається методом опорних векторів (Support Vector Machine, SVM). Вона вимагає, щоб позитивна мітка (в даному випадку “спам”) мала числове значення +1 (один), а негативна мітка (“не_спам”) мала значення -1 (мінус один).

Тепер маємо *набір даних* та *алгоритм навчання* і можемо застосувати алгоритм навчання до набору даних, щоб отримати *модель*.

SVM розглядає кожен вектор ознак як точку в багатовимірному просторі (у цьому випадку простір має 20 000 вимірів). Алгоритм поміщає всі вектори ознак

на уявний 20 000-мірний графік і малює уявну 19 999-мірну лінію (гіперплощина), яка відокремлює дані з позитивними мітками від даних з негативними мітками. Кордон, що розділяє дані різних класів, у машинному навчанні називається *межею ухвалення рішення.*

Рівняння гіперплощини визначається двома параметрами: речовим вектором w тієї ж розмірності, що і вхідний вектор ознак x, і дійсним числом b (2.1):

$$wx - b = 0, \tag{2.1}$$

де вираз wx означає $w^{(1)}x^{(1)} + w^{(2)}x^{(2)} + \ldots + w^{(D)}x^{(D)}$, а D - число вимірів у векторі ознак x.

Тепер прогнозовану мітку для деякого вхідного вектора ознак x можна виразити формулою (2.2).

$$y = \text{sign}(wx - b) \tag{2.2}$$

де sign - це математичний оператор, що приймає довільне значення і повертає +1, якщо вхідне значення є позитивним числом, -1, якщо вхідне значення є негативним числом.

Мета алгоритму навчання, у разі SVM, використовуючи набір даних, знайти оптимальні значення w^* і b^* для параметрів w і b. Після того як алгоритм навчання знайде ці оптимальні значення, модель $f(x)$ буде виглядати як (2.3):

$$f(x) = \text{sign}(w^* x - b^*) \tag{2.3}$$

Щоб за допомогою моделі SVM передбачити, чи є електронний лист спамом чи ні, необхідно взяти текст листа, перетворити його на вектор ознак, потім помножити цей вектор на w*, відняти b* і взяти знак результату. Це дасть нам прогноз (+1 означає "спам", а -1 означає "не_спам").

Для того, щоб знайти w* і b* машина вирішує завдання оптимізації. Вона добре справляється з оптимізацією функцій за умов обмежень.

Насамперед, модель повинна правильно передбачати мітки наявних 10 000 даних. Обмеження виглядають наступним чином (2.4):

$$\begin{aligned} wx_i - b &\geqslant +1, \text{ якщо } y_i = +1, \\ wx_i - b &\leqslant -1, \text{ якщо } y_i = -1 \end{aligned} \tag{2.4}$$

Також бажано, щоб гіперплощина відокремлювала позитивні дані від негативних з максимальним зазором. Зазор — це відстань між найближчими зразками двох класів, які відокремлюються межею ухвалення рішення. Великий зазор сприяє кращому узагальнення, тобто тому, наскільки добре модель класифікуватиме нові дані. Для максимізації зазору необхідно мінімізувати евклідову норму w, яка позначається як ||w|| і визначається виразом $\sqrt{\Sigma_{j=1}^{D}\left(w^{(j)}\right)^2}$.

Машина повинна вирішити задачу оптимізації, яка формулюється так:

Мінімізувати ||w|| з урахуванням $y_i(wx_i - b) \geqslant 1$ для i = 1, ..., N.

Розв'язання цієї задачі оптимізації, позначеної параметрами w* та b*, називається *статистичною моделлю*, або просто *моделлю*. Процес побудови моделі називається *навчанням*.

Для двовимірних векторів завдання та рішення можна подати візуально, як показано на рис. 2.1. Сині та помаранчеві точки становлять позитивні та від'ємні зразки відповідно, а лінія, задана рівнянням $wx - b = 0$, є межею прийняття рішення.

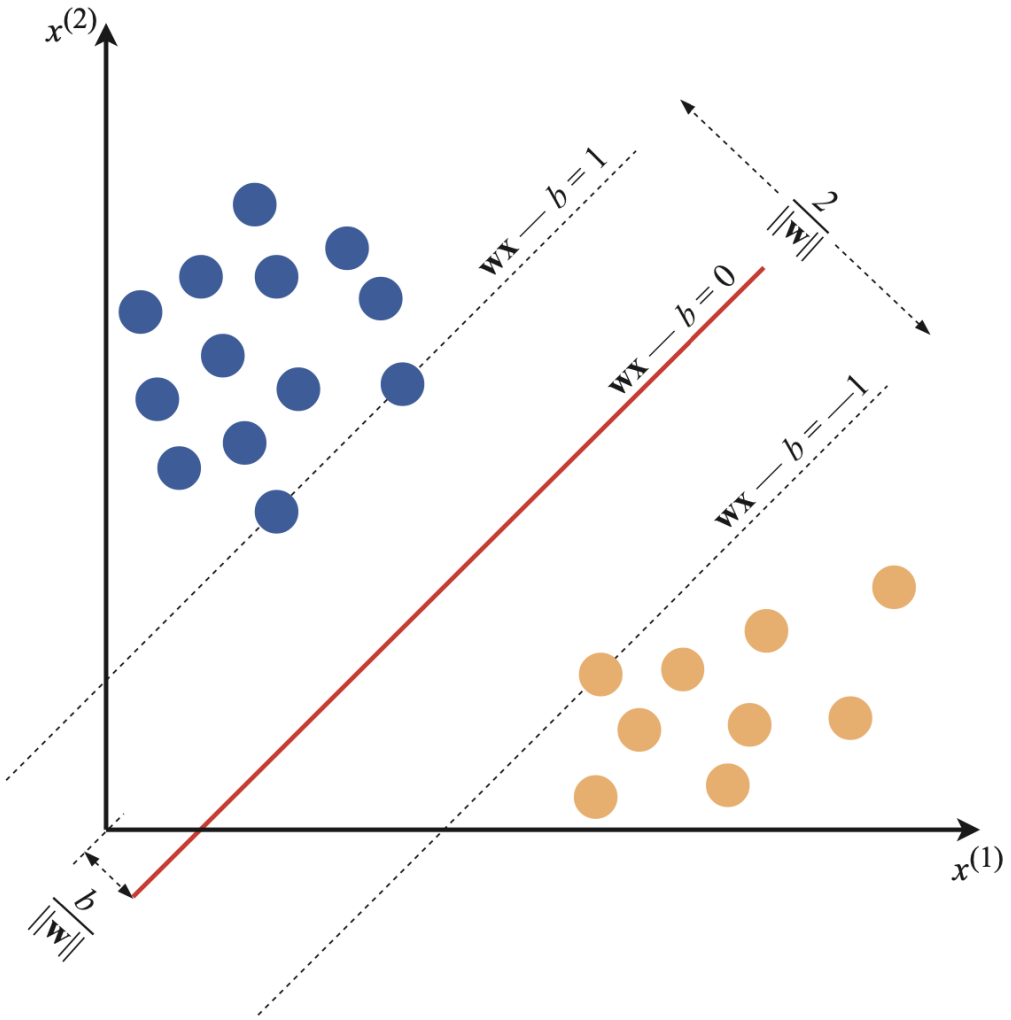


Рисунок 2.1 - Приклад моделі SVM для двовимірних векторів ознак.

Геометрично рівняння $wx - b = 1$ і $wx - b = -1$ визначають дві паралельні гіперплощини, як показано на рис. 2.1. Відстань між цими гіперплощинами однакове, тому що чим менше норма ||w||, тим більше відстань між цими двома гіперплощинами. Так працює метод опорних векторів (SVM). Ця конкретна версія алгоритму будує так звану лінійну модель. Вона називається лінійною, тому що межа ухвалення рішення — це пряма лінія (або площина, або гіперплощина). Метод опорних векторів також може містити ядра, здатні зробити межу рішення довільно нелінійною. У деяких випадках неможливо повністю розділити дві групи точок через шум у даних, помилок розмітки або аномалій (даних, які сильно відрізняються від «типового» зразка в наборі даних). Для таких випадків є версія алгоритму SVM, здатна включати *гіперпараметр штрафу* за неправильну класифікацію конкретних класів, що навчають даних.

Гіперпараметр — це властивість алгоритму навчання, що зазвичай (але не завжди) має чисельне значення. Це значення впливає роботу алгоритму. Значення гіперпараметрів не визначаються самим алгоритмом даних. Вони мають задаватися аналітиком перед запуском алгоритму [14].

Будь-який алгоритм розв'язання задачі класифікації при побудові моделі явно чи неявно створює межу розв'язання. Кордон рішення може бути прямий, зігнутим, мати складну форму або утворюватися накладенням деяких геометричних фігур. Форма межі рішення визначає точність моделі (тобто частку зразків, мітки яких передбачаються моделлю правильно). Форма кордону рішення і те, як вона алгоритмічно чи математично обчислюється з урахуванням навчальних даних, відрізняють один алгоритм навчання з іншого.

На практиці необхідно враховувати також дві інші важливі характеристики алгоритму навчання: швидкість побудови моделі та час отримання прогнозу. У багатьох практичних ситуаціях перевага надається більш швидким алгоритмам, які будують менш точні моделі. Крім того, іноді краще мати менш точну модель, але швидко обчислює прогнози [15].

2.1.2 Навчання без вчителя

У навчанні без вчителя набір даних представлений колекцією нерозмічених зразків $\{x_i\}_{i=1}^{N}$. І знову, x — це вектор ознак, а мета алгоритму навчання без вчителя — створити модель, яка приймає вектор ознак x на вході і перетворює його на інший вектор або значення, яке можна використовувати для вирішення практичного завдання. Наприклад, у задачах кластеризації модель повертає ідентифікатор кластера для кожного вектора ознак набору даних. У задачах зменшення розмірності модель повертає вектор ознак, що має менше елементів,

ніж вхідний вектор x. У задачах виявлення аномалій повертається дійсне число, яке вказує, наскільки x відрізняється від «типового» зразка набору даних [15].

2.1.3 Навчання з частковим залученням вчителя

У навчанні з частковим залученням вчителя (semi-supervised learning) набір даних містить як розмічені, і нерозмічені зразки. Зазвичай нерозмічених зразків набагато більше, ніж розмічених. Алгоритм навчання із частковим залученням вчителя переслідує таку ж мету, як і алгоритм навчання з учителем. В даному випадку передбачається, що використання множини нерозмічених даних допоможе алгоритму навчання знайти (можна також сказати "виконати" або "обчислити") кращу модель. Припущення про виграш від додавання більшої кількості нерозмічених даних може бути нелогічним. На перший погляд, здається, що ми, навпаки, додаємо більше невизначеності. Однак, додаючи нерозмічені дані, ви вносите більше інформації про завдання: більша вибірка краще відображає розподіл ймовірностей у даних, звідки взято розмічені зразки. Теоретично алгоритм навчання повинен вміти скористатися цією додатковою інформацією.

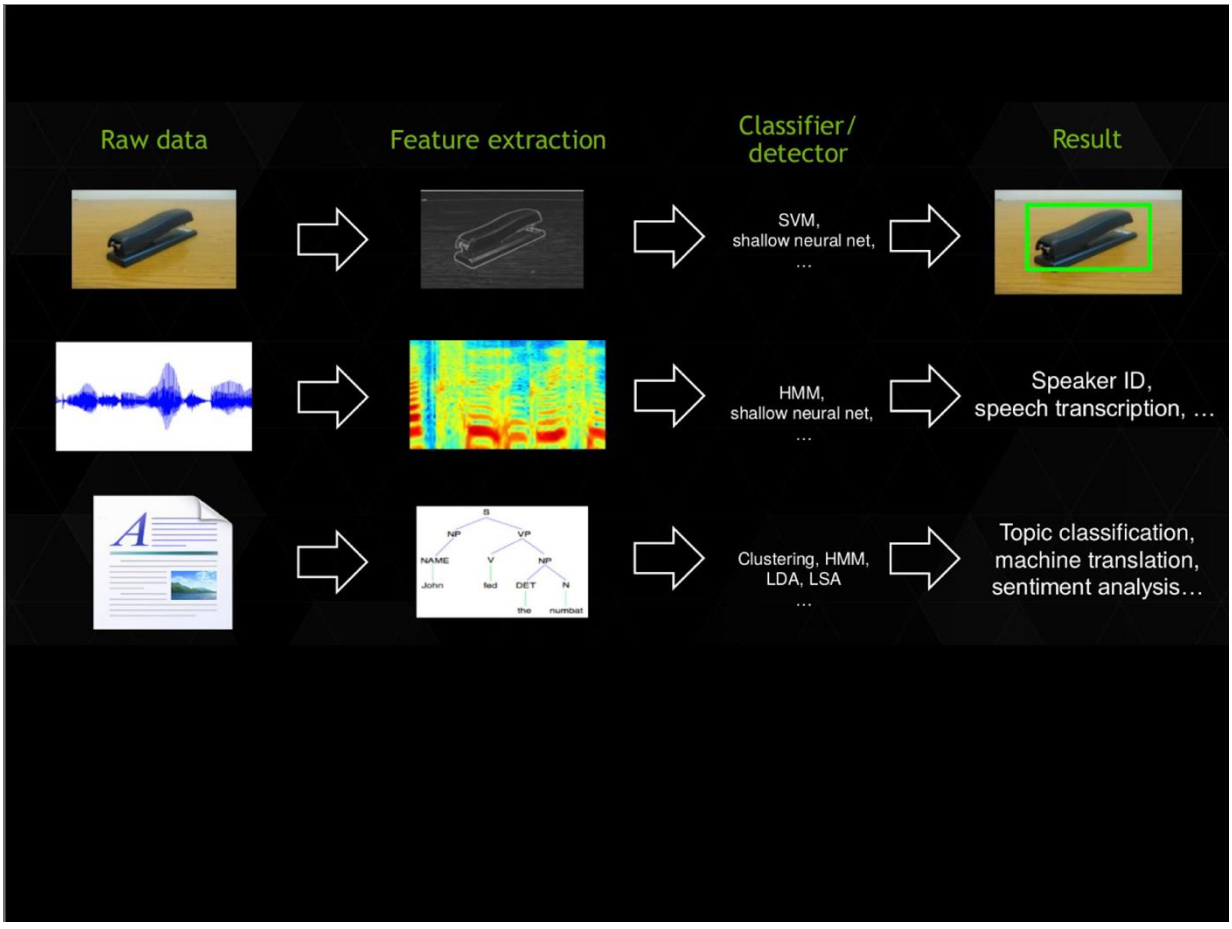


Рисунок 2.2 - Приклад витягування особливостей з фото, аудіо та текстових файлів.

2.2 Нейронні мережі

Нейронна мережа (neural network, NN), як і модель регресії чи SVM, — це лише математична функція (2.5):

$$y = f_{NN}(x) \tag{2.5}$$

Функція f_{NN} має особливу форму: це вкладена функція (на зразок матрьошки). Для тришарової нейронної мережі, що повертає скаляр, f_{NN} виглядає так (2.6):

$$y = f_{NN}(x) = f_3(f_2(f_1(x))) \tag{2.6}$$

f_1 і f_2 у рівнянні вище - це векторні функції, які визначаються як (2.7):

$$f_l(z) = g_l(W_l z + b_l) \tag{2.7}$$

де l це індекс шару і може охоплювати від 1 до будь-якої кількості шарів.

Функція g_l називається функцією активації. Це фіксована, зазвичай нелінійна функція, обрана аналітиком на початку навчання. Параметри W_l (матриця) і b_l (вектор) визначаються для кожного шару в ході навчання, з використанням градієнтного спуску для оптимізації, залежно від задачі, конкретної функції вартості (такої як середньоквадратична помилка — MSE). Функція f_3 є скалярною функцією для задачі регресії, але може бути векторною функцією, залежно від конкретної задачі [16].

2.2.1 Приклад багатошарового перцептрона

Розглянемо одну конкретну конфігурацію нейронних мереж, так звану нейронну мережу прямого поширення (Feedforward Neural Network, FFNN), і ще конкретніше — архітектуру, названу багатошаровим перцептроном (Multilayer Perceptron, MLP). Як ілюстрацію розглянемо MLP із трьома шарами. Наша мережа

приймає на вхід двовимірний вектор ознак та виводить число. Ця FFNN може бути моделлю регресії чи класифікації, залежно від функції активації, що у третьому вихідному шарі [14].

Багатошаровий перцептрон зображено на рис. 2.2. Графічно нейронну мережу можна подати у вигляді комбінації зв'язаних вузлів, логічно організованих в один або кілька шарів. Кожен вузол представлений кружком чи прямокутником. Вхідна стрілка представляє вхід вузла і вказує, звідки надходять дані цей вхід. Вихідна стрілка є вихід вузла.

Вихід кожного вузла є результатом математичної операції, яка виконується всередині прямокутника. Вузли, позначені кружками, нічого не роблять із вхідними даними; вони просто пересилають їх у свій вихід.

Ось що відбувається у кожному вузлі, позначеному прямокутником. Спочатку всі входи вузла об'єднуються, і їх формується вхідний вектор. Потім вузол змінює лінійне перетворення до вхідного вектора, так само, як це робить модель лінійної регресії зі своїм вектором вхідних ознак. І насамкінець застосовує функцію активації g до результату лінійного перетворення, отримуючи вихідне значення — дійсне число. У класичній нейронної мережі прямого поширення (FFNN) вихідне значення вузла в певному шарі стає вхідним значенням для всіх вузлів у наступному шарі [17].

На рис. 2.2 функція активації g_l має один індекс: *l* - індекс шару, якому належить вузол. Зазвичай всі вузли в одному шарі рівня використовують ту саму функцію активації, але це не обов'язково. Кожен шар може мати різну кількість вузлів. Кожен вузол має параметри $w_{l,u}$ і $b_{l.u}$, де *u* — індекс вузла, а *l* — індекс шару. Вектор y_{l-1} у кожному вузлі визначається як:

$$\left[y_{l-1}^{(1)},\ y_{l-1}^{(2)},\ y_{l-1}^{(3)},\ y_{l-1}^{(4)}\right] \tag{2.8}$$

Вектор **x** у першому шарі визначається як:

$$\left[x^{(1)},\ \ldots,\ x^{(D)}\right] \tag{2.9}$$

Як можна побачити на рис. 2.2 у багатошаровому перцептроні всі виходи одного шару підключені до кожного входу наступного шару. Така архітектура називається пов'язаною. Нейронна мережа також може містити пов'язані шари, тобто шари, вузли яких отримують на вході вихідні дані від кожного з вузлів у попередньому шарі.

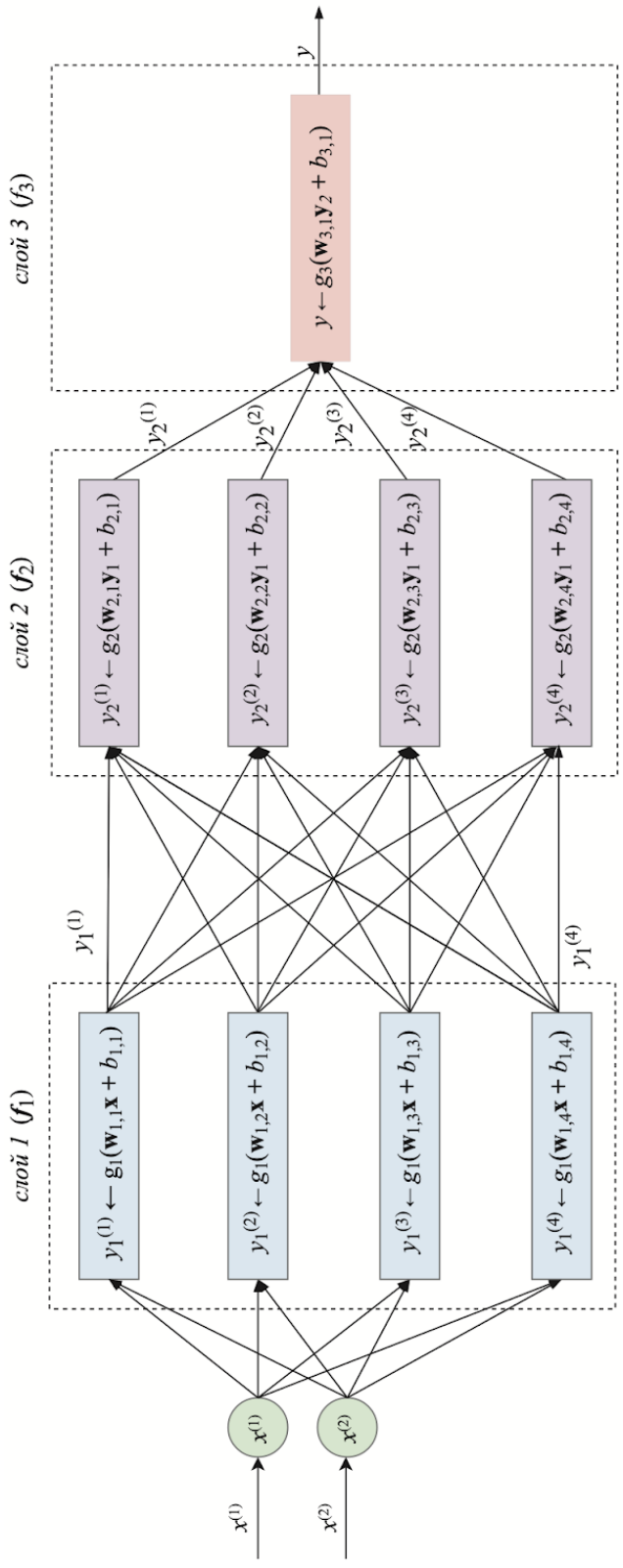


Рисунок 2.3 - Багатошаровий перцептрон з двовимірним входом, двома шарами по чотири вузли в кожному та вихідним шаром з єдиним вузлом.

2.3 Глибоке навчання

Під глибоким навчанням мається на увазі навчання нейронних мереж, що мають більше двох невихідних верств. У минулому навчанні таких мереж ускладнювалося дедалі більше зі зростанням кількості шарів. Серед найбільших проблем називалися вибухове зростання градієнта і загасання градієнта, оскільки визначення параметрів мережі використовувався градієнтний спуск.

Якщо проблема вибухового зростання градієнта вирішувалася відносно просто, із застосуванням таких простих методів, як обмеження градієнта та L1- або L2- регуляризація, то проблема згасання градієнта залишалася нерозв'язною протягом десятиліть. Для оновлення значень параметрів у нейронних мережах зазвичай використовується алгоритм, що має назву зворотним поширенням. Зворотне поширення – це ефективний алгоритм обчислення градієнтів у нейронних мережах із використанням правила диференціювання складних функцій. У кожній ітерації навчання, у процесі градієнтного спуску, параметри нейронної мережі оновлюються пропорційно до приватної похідної функції вартості для поточного параметра. Проблема в тому, що іноді градієнт виявляється зникаючим малим, що фактично заважає зміні значень деяких параметрів. У найгіршому випадку це може повністю зупинити навчання нейронної мережі.

Традиційні функції активації, такі як функція гіперболічного тангенсу, мають градієнти в діапазоні (0, 1), при цьому градієнти обчислюються на етапі зворотного розповсюдження за правилом диференціювання складних функцій. В результаті для обчислення градієнтів попередніх шарів (розташованих ліворуч) у *n*-шаровій мережі проводиться перемноження *n* цих невеликих чисел, через що градієнт експоненційно зменшується зі збільшенням *n*. Це призводить до того, що ранні верстви навчаються набагато повільніше, якщо взагалі навчаються [18].

2.4 Виявлення обличчя використовуючи Haar cascade classifier

Перш ніж перейти до ідентифікації обличчя, важливо обговорити іншу пов’язану техніку: виявлення обличчя. Як випливає з назви, розпізнавання обличчя – це техніка, яка ідентифікує людські обличчя на цифровому зображенні, але не виконує ніякі порівняння з іншими обличчями, тобто відсутня *ідентифікація.* Розпізнавання обличчя є відносно давньою технологією. Ще у старих цифрових камерах, дивлячись у об’єктив можна було побачити прямокутники наверного облич людей.

Існує один з найвідоміших алгоритмів виявлення обличчя під назвою Віола-Джонса, або каскади Хаара (Haar cascade). Каскади Хаара з’явилися задовго до

того, як глибоке навчання стало популярним, і є одним із найбільш часто використовуваних методів виявлення облич.

У таблиці 2.1 наведено рівень виявлення різної кількості помилкових виявлень для Haar Cascade системи, а також для інших опублікованих систем.

Таблиця 2.1 - Коефіцієнти виявлення різної кількості хибних спрацювань у тестовому наборі з іншими алгоритмами, який містить 130 зображень і 507 облич.

Алгоритм\ Помилкові виявлення	**10**	**31**	**50**	**65**	**78**	**95**	**167**
Viola-Jones	76.1%	88.4%	91.4%	92.0%	92.1%	92.9%	93.9%
Rowley-Baluja-Kanade	83.2%	86.0%	-	-	-	89.2%	90.1%
Schneiderman-Kanade	-	-	-	94.4%	-	-	-
Roth-Yang-Ahuja	-	-	-		(94.8%)	-	-

2.4.1 Алгоритм роботи Haar Cascade

Каскадний класифікатор використовується для виявлення об'єкта, для якого його було натреновано. Тобто це не обов’язково можуть бути обличчя.

Для того, аби навчити алгоритм виявляти обличчя, необхідно підготувати набір даних - множину фотографій облич (позитивні дані) та множину фотографій з обличчями (негативні дані). Потім витягуються особливості облич із зображень (рис. 2.3). Прямокутники A та B визначають кути маркерів, C - лінії та D – прямокутники [12].

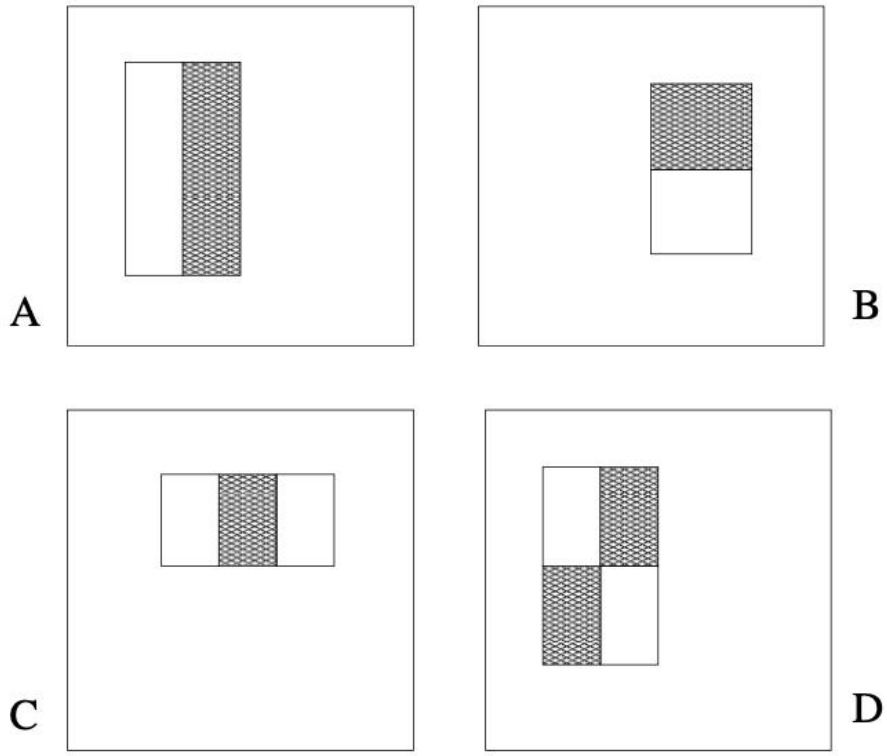


Рисунок 2.4 - Приклад прямокутних елементів, показаних відносно охопленого вікна виявлення. Сума пікселів у білих прямокутниках віднімається від суми пікселів у сірих прямокутниках. Два прямокутники показані на (A) і (B). На малюнку (C) показано три прямокутника, а (D) чотири прямокутника.

Щоб розпізнати обличчя на зображенні, алгоритм шукає наявність різних особливостей, які зазвичай зустрічаються на людських обличчях, наприклад брови, де область над бровою світліша за область під нею.

Якщо зображення містить комбінацію всіх цих ознак, воно вважається таким, що містить обличчя (рис. 2.4).

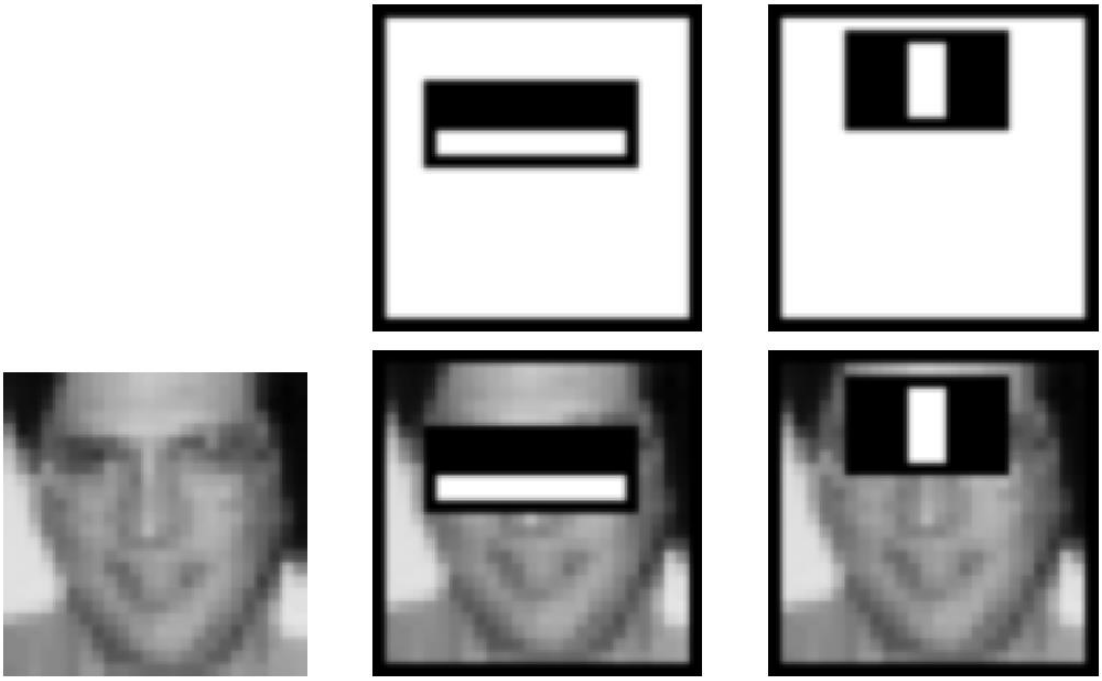


Рисунок 2.5 - Приклад пошуку особливостей (ознак/маркерів) обличчя на фото.

Тепер усі можливі розміри та місце розташування кожного прямокутника використовуються для заходження необхідних міток облич. Навіть вікно 24x24 має до понад 160 000 міток. Для кожного обчислення мітки нам потрібно знайти суму пікселів під білим і чорним прямокутниками. Щоб вирішити цю проблему, розробники алгоритму ввели інтегроване зображення. Яким би великим не було ваше зображення, воно зменшує кількість обчислень до чотирьох пікселів. Це робить роботу надзвичайно швидкою.

Але серед усіх цих міток, більшість із них не мають ознаки обличчя. На рис 2.4 верхній рядок показує дві хороші мітки. Перша, здається, вибрана тому, що область очей часто темніша за область носа та щік. Друга вибрана ознака базується на тому, що очі темніші за перенісся. Але ті ж самі мітки, нанесені на щоки або будь-яке інше місце, не мають значення. Тому, постає питання, яким способом можливо найефективніше обрати саме ті мітки, які нам потрібні. Це досягається за допомогою алгоритма під назвою Adaboost [15].

2.4.2 Математична модель алгоритму Haar Cascade

AdaBoost алгоритм використовується для навчання класифікатора [20].

1. Маємо фото як навчальний набір даних $\{(x_i, y_i)\}^N$, де y_i = 0, 1 для позитивного вектору ознак (фото обличчя) та негативного вектору ознак (фото з обличчями).
2. Ініціалізація $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ ваги для y_i = 0, 1 відповідно, де *m* та *l* є числами позитивних та негативних вектору ознак відповідно.
3. Для $t = 1, \ldots, T$:
    1. Нормалізуємо ваги (2.10)

$$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}, \quad (2.10)$$

так, що w_t показує розподіл ймовірностей.

б. Для кожної мітки j навчаємо класифікатор h_j, який обмежений використовувати лише одну мітку. Помилку оцінюємо як $w_t, e_j = \sum_i w_i \left| h_j(x_i) - y_i \right|$

в. Обираємо класифікатор h_t з найменшою помилкою e_j

г. Обновляємо ваги (2.11):

$$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}, \quad (2.11)$$

де e_i = 0, де вектор ознак x_i класифікований правильно і e_i = 1 навпаки і $\beta_t = \frac{e_t}{1 - e_t}$

4. Остаточна формула класифікатора виглядає як (2.12):

$$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geqslant \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & otherwise \end{cases}, \quad (2.12)$$

де $\alpha_t = \log \frac{1}{\beta_t}$

2.4.3 Haar Cascade алгоритм на практиці

Спробуємо створити програму на мові Python3 з використанням бібліотеки OpenCV у якій вже реалізований алгоритм Haar Cascade Classifier.

Спочатку необхідно встановити бібліотеку командою:

```
$ pip3 install opencv-python
```

На щастя, розробники бібліотеки OpenCV опублікували у вільному доступі на GitHub список каскадів, що дозволяє пропустити етап навчання алгоритму на тысячах фото. Закачуємо файл *haarcascade_frontalface_default.xml.*

Лістинг 2.1 показує код простої програми, яка з веб-камери комп'ютера у реальному часі виконує пошук обличчя та малює навколо нього жовтий квадрат.

```
import cv2

face_cascade =
cv2.CascadeClassifier('haarcascade_frontalface_default.xml
')

screen_width = 1280
screen_height = 720

stream = cv2.VideoCapture(0)

while(True):
    (grabbed, frame) = stream.read()
    rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

 faces = face_cascade.detectMultiScale(rgb,
scaleFactor=1.3, minNeighbors=5)

    for (x, y, w, h) in faces:
        color = (0, 255, 255)
        stroke = 5
       cv2.rectangle(frame, (x, y), (x + w, y + h),
color, stroke)

    cv2.imshow("Image", frame)
    key = cv2.waitKey(1) & 0xFF
    if key == ord("q"):
        break
```

```
stream.release()
cv2.waitKey(1)
cv2.destroyAllWindows()
cv2.waitKey(1)
```

Лістинг 2.1 - Програма для виявлення обличчя за алгоритмом Haar Cascade Classifiers.

Рисунок 2.6 демонструє результат виявлення обличчя за допомогою цієї програми.

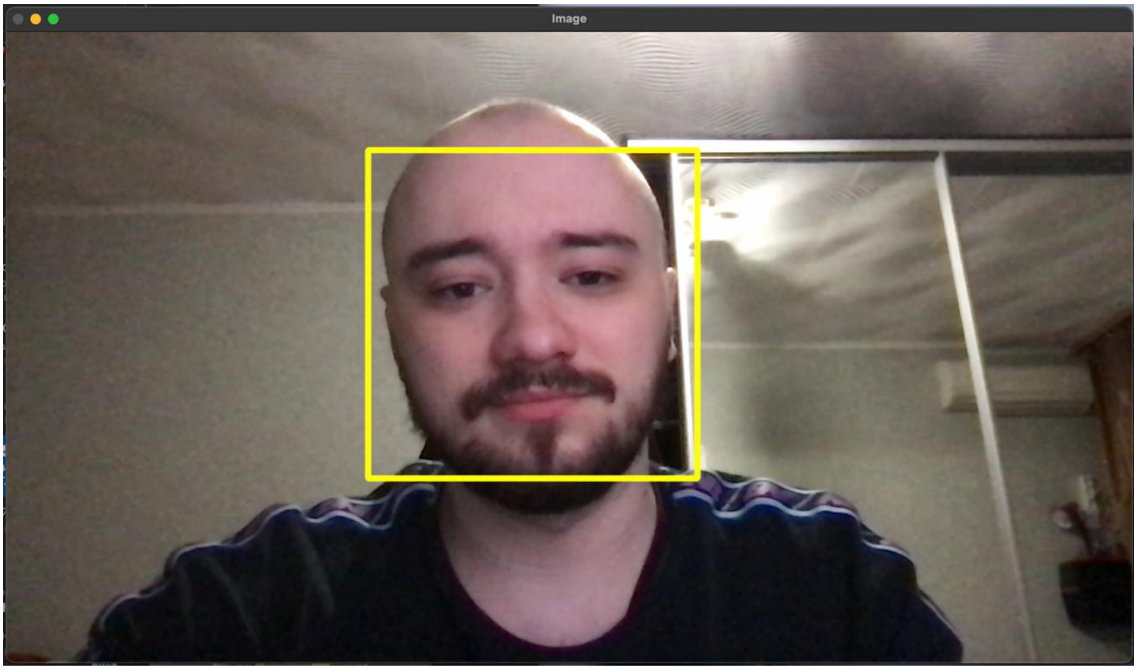


Рисунок 2.6 - Тестування програми виявлення обличчя за допомогою алгоритма Haar Cascade.

Висновок до розділу

Розглянуто базову теорію машинного навчання. Описано 3 види навчання: з учителем, без учителя та з частковим залученням вчителя. Детальну увагу приділено моделі ШНМ на базі багатошарового перцептрона. та поняттям глибокого навчання.

Обраний алгоритм Viola-Jones (Haar Cascade) дозволяє отримати точність більше 90% вже при наявності 50 навчальних зображень обличь завдяки алгоритму Adaboost, який значно підвищує точність та швидкість навчання класифікатора.

На практиці реалізували алгоритм Haar Cascade Classifier для виявлення обличчя на мові програмування Python3 використовуючи бібліотеку OpenCV.

3 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ СИСТЕМИ ОХОРОНИ

3.1 Проектування охоронної системи

У рамках проекту планується створити прототип охоронної системи на базі ESP32 CAM контролеру (рис. 3.1), що буде виконувати роль відеокамери, яка виконує зйомку у реальному часі та надсилає кадри до серверу, або як ще його можна назвати - “хаб.”. У разі успішної ідентифікації особи, контролер зможе відчинити двері та пустити людину до приміщення.

Характеристики контролеру ESP32 CAM:

- Контролер: ESP32, 2 ядра, 32-біт
- Робоча частота процесора: 240 МГц / 600 DMIPS
- Оперативна пам'ять: Вбудована: 520 КБ / Зовнішня: 4 МБайт
- Підтримувані інтерфейси: UART/SPI/iec/PWM/ADC/DAC
- Вбудований Lwip і FreeRTOS
- Підтримка STA/AP/STA + AP Робочий режим
- Підтримка Smart Config/AirKiss розподіленої мережі
- Напруга живлення: 5В
- Модуль камери: OV2640 (в комплекті)
- Роздільна здатність: 2 Мп
- Підтримка камер: OV2640 і OV7670
- Зберігання даних: micro-SD
- Розмір: 27 х 39 мм
- Вага: 10 г

Рисунок 3.1 - Розташування та призначення контактів.

Таблиця 3.1 - Розташування та призначення контактів.

CAM	ESP32	SD	ESP32
D0	PIN5	CLK	PIN14
D1	PIN18	CMD	PIN15
D2	PIN19	DATA0	PIN2
D3	PIN21	DATA1/flash	PIN4
D4	PIN36	DATA2	PIN12
D5	PIN39	DATA3	PIN13
D6	PIN34		
D7	PIN35		
XCLK	PIN0		
PCLK	PIN22		
VSYNC	PIN25		
HREF	PIN23		
SDA	PIN26		
SCL	PIN27		
POWER PIN	PIN32		

Рисунок 3.2 - ESP32 CAM контролер.

Вбудований Wi-Fi модуль дозволить виконати під’єднання до хабу через технологію Websocket у якості клієнта, що дозволяє підтримувати TCP підключення між двома пристроями стільки, скільки потрібно та надсилати повідомлення в обидні сторони (дуплексний зв’язок).

У свою чергу хаб буде хостити веб додаток, що буде складатися з Websocket серверу та модуля із натернованим алгоритмом штучного інтелекту. Websocket дозволить тримати підключення з контролерами системи охорони. У рамках проекту система складається лише з одного контролеру, але у теорії їх може бути більше і всі вони будуть під’єднанні до одного хабу.

У якості алгоритму штучного інтелекту буде використовуватися SVM, математична модель якого вже була описана у розділі 2 (2.1.1 Навчання з учителем).

Таким чином, взаємодія компонентів системи буде виглядати наступним чином (рис. 3.3): ESP32 CAM контролер надсилає відео кадри у режимі реального часу по Websocket дуплексному каналу до хабу. Останній віддає кадри тренованому алгоритму і той дає відповідь, чи знаходиться у кадрі довірена особа.

Довірена особа - це особа, яка має права доступу в системі охорони і у разі її успішної ідентифікації алгоритмом буде допущена до приміщення, що охороняється.

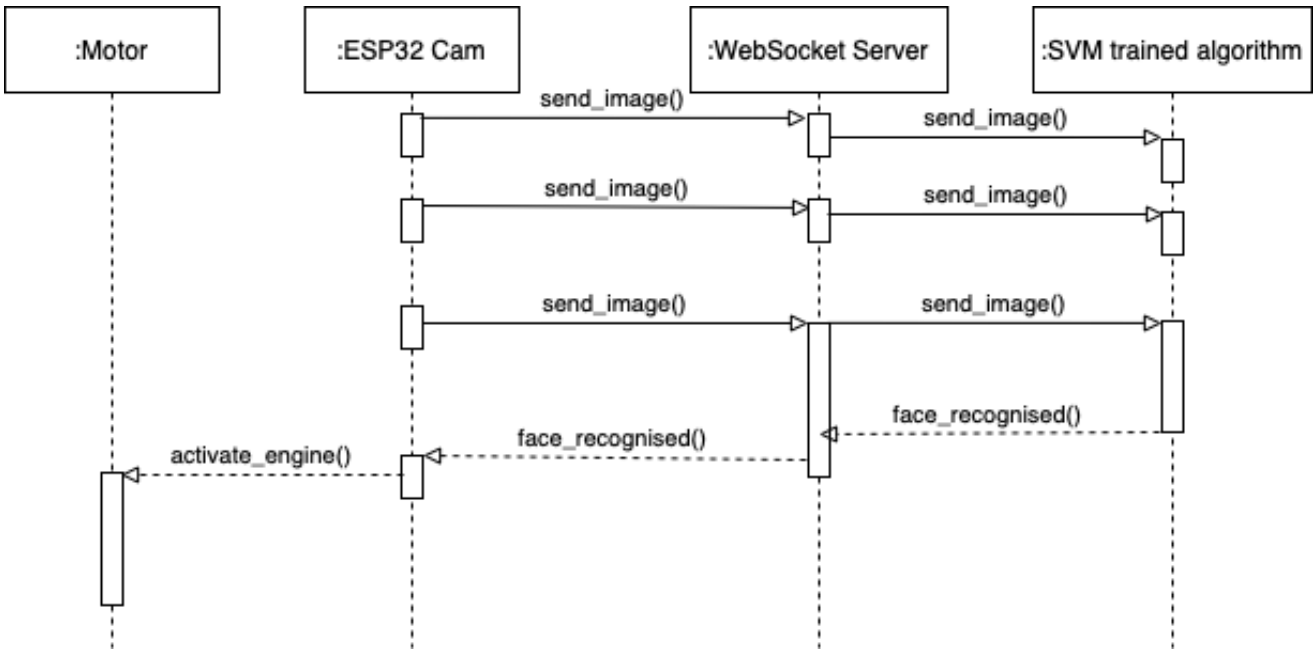


Рисунок 3.3 - Діаграма взаємодії компонентів охоронної системи

3.2 Реалізація SVM алгоритму розпізнавання обличчя

SVM — це набір контрольованих методів навчання, які можна використовувати для класифікації. SVM знаходить межу, яка розділяє класи якомога більшою лінією. Основна ідея SVM полягає в тому, щоб найкращим чином провести межу між двома чи більше класами (рис. 3.3).

Після того, як лінія поділила об’єкти на класи, можна використовувати її для прогнозування майбутніх даних. Наприклад, враховуючи довжину обличчя та геометрію вуха нової невідомої тварини, можна використовувати роздільну лінію як класифікатор, щоб передбачити, чи є тварина собакою чи котом.

У проекті будемо використовувати алгоритм SVM для розпізнавання обличчя на основі набору тестових зображень обличчя умовної довіреної особи та знаменитості.

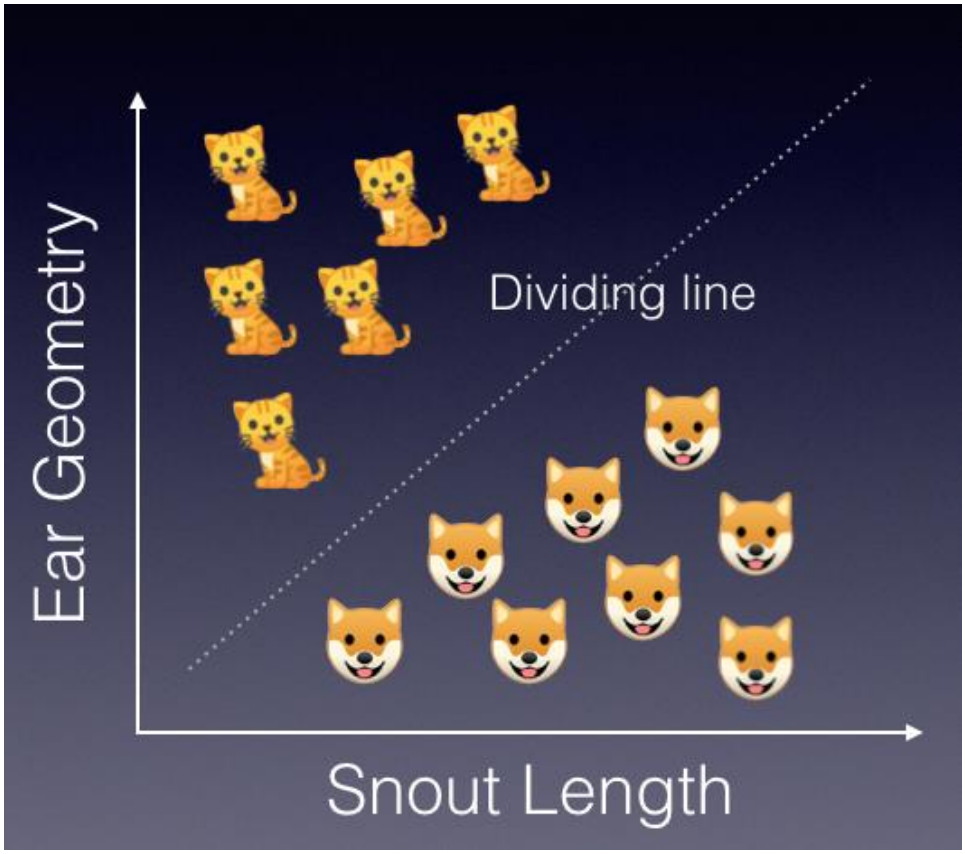


Рисунок 3.4 - SVM алгоритм націлений на максимальне розділення між об’єктами класів.

Спочатку необхідно підготувати не менш ніж по 10 фотографій облич кожної особи яку плануємо розпізнавати (рис. 3.6). Помістимо фото в умовну директорію під назвою *Headshots* (рис. 3.5). Однак ці фото необхідно буде опрацювати таким чином, аби всі вони мали однаковий розмір та маску RGB (рис. 3.7).

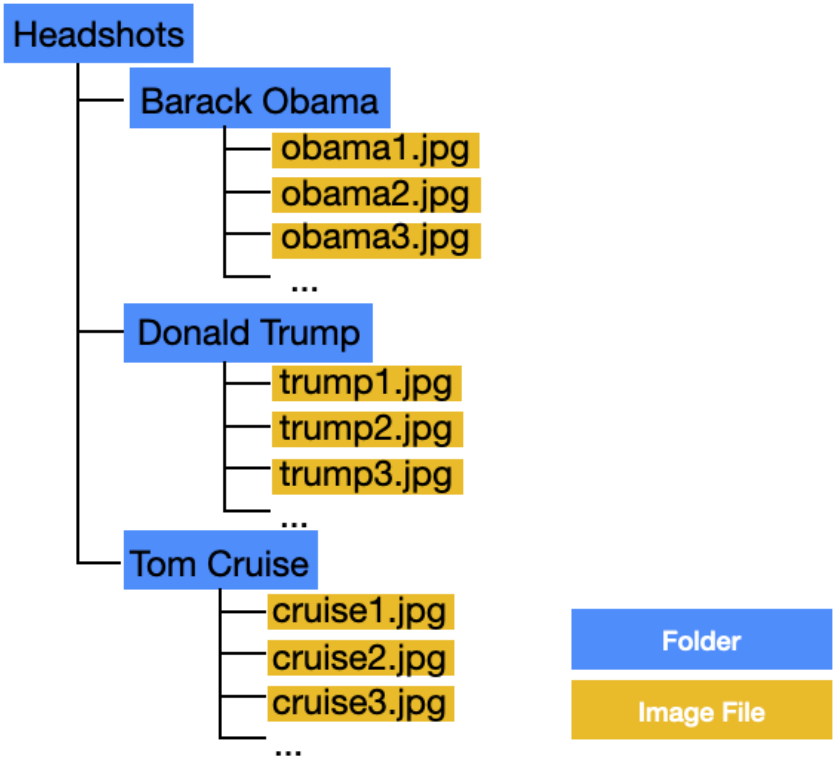


Рисунок 3.5 - Папки із зображеннями людей, яких необхідно розпізнавати.

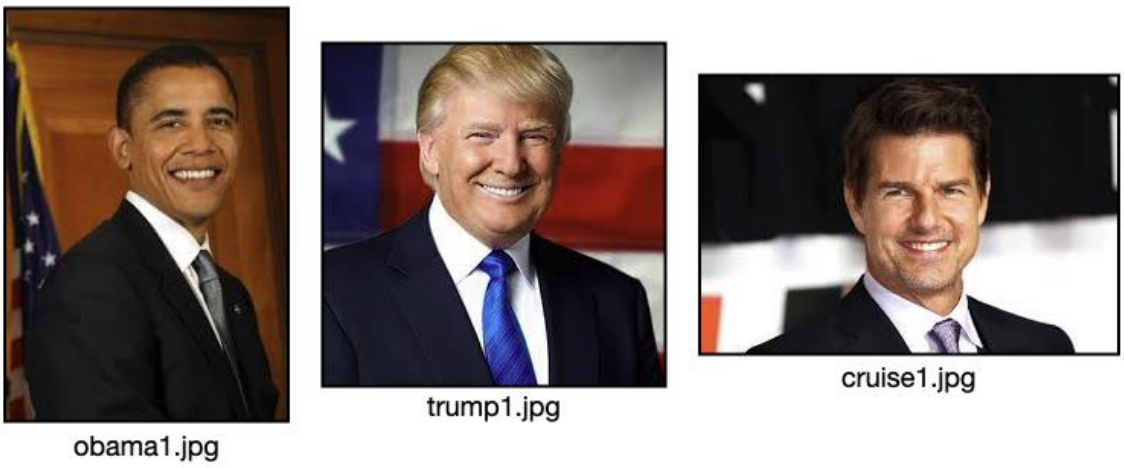


Рисунок 3.6 - Деякі фото для навчання алгоритму.

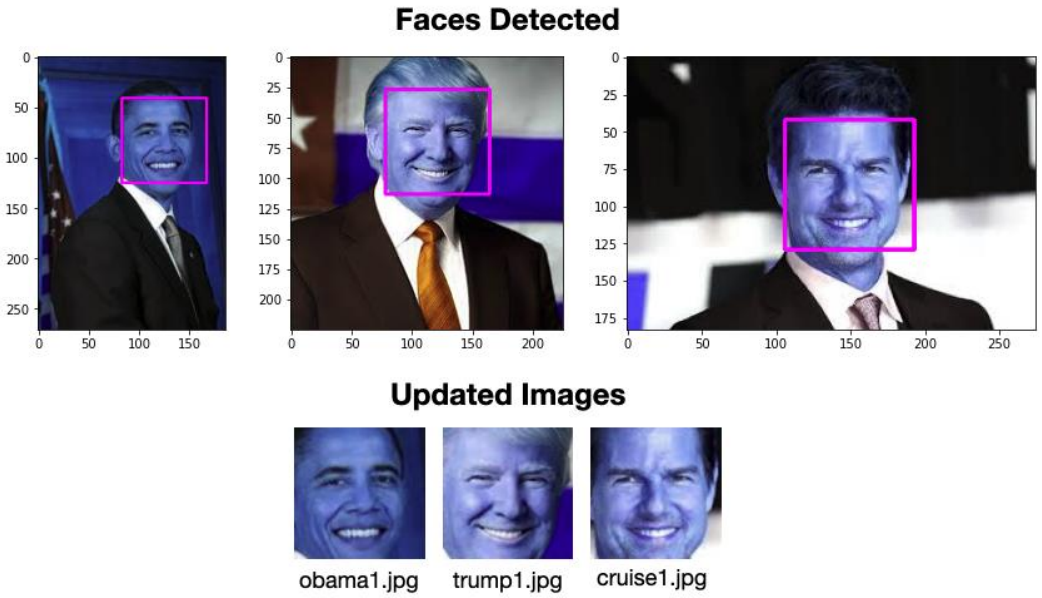


Рисунок 3.7 - Виявлення облич на зображенні та оновлення зображень за допомогою виявлених облич.

```
import os
import imageio
import matplotlib.pyplot as plt

new_image_size = (150,150,3)

images_dir = './Headshots'

current_id = 0

label_ids = {}

images = []
labels = []

for root, _, files in os.walk(images_dir):
    for file in files:
        if file.endswith(('png','jpg','jpeg')):
            path = os.path.join(root, file)
            label = os.path.basename(root).replace(
                " ", ".").lower()
```

```
                if not label in label_ids:
                    label_ids[label] = current_id

                    current_id += 1

                labels.append(current_id-1)

                image = imread(path)
                image = resize(image, new_image_size)
                images.append(image.flatten())

    categories = list(label_ids.keys())

    pickle.dump(categories,  open("faces_labels.pk",  "wb"
))

    df = pd.DataFrame(np.array(images))
    df['Target'] = np.array(labels)
```

Лістинг 3.1 - Завантаження фотографій для тренування моделі.

Кожне зображення завантажується, а потім змінюється до розміру 150x150 з глибиною кольору трьох каналів (RGB). Зображення порівнюються, а потім зберігається – кожен піксель зберігається як стовпець у фреймі даних (рис. 3.7) (отже, у фреймі даних є стовпці розміром 150x150x3, що представляють кожне зображення). Останній стовпець у кадрі даних визначає, кому належить обличчя.

	0	1	2	3	...	67491	67492	67493	67494	67495	67496	67497	67498	67499	Target
0	0.037281	0.085307	0.122588	0.028523	...	0.000915	0.196993	0.467582	0.000000	0.195059	0.465647	0.000000	0.192157	0.462745	0
1	0.371873	0.421886	0.484631	0.366693	...	0.364480	0.396078	0.461604	0.346928	0.382222	0.448889	0.330088	0.365382	0.432049	0
2	0.005618	0.009539	0.000000	0.012931	...	0.003922	0.003922	0.003922	0.003922	0.003922	0.003922	0.003922	0.003922	0.003922	0
3	0.421833	0.475477	0.508837	0.425464	...	0.318301	0.374859	0.408323	0.293046	0.355791	0.399232	0.286513	0.349258	0.397046	0
4	0.792157	0.792157	0.792157	0.792157	...	0.856559	0.856559	0.856559	0.859075	0.859075	0.859075	0.859791	0.859791	0.859791	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
64	0.109908	0.102065	0.114069	0.074069	...	0.532209	0.355791	0.240235	0.521464	0.351670	0.245840	0.513435	0.350954	0.248026	2
65	0.132075	0.108546	0.108546	0.154771	...	0.754405	0.625167	0.515137	0.717186	0.583853	0.474049	0.696729	0.563395	0.453591	2
66	0.067634	0.067634	0.067634	0.073490	...	0.290441	0.322491	0.383624	0.320137	0.348304	0.386448	0.322297	0.349748	0.376180	2

Рисунок 3.8 - Дата фрейм.

Після завантаження зображень у фрейм даних потрібно розділити фрейм даних на набір для навчання та тестування:

```
 x = df.iloc[:,:-1]
 y = df.iloc[:,-1]

 x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.10, random_state=77, stratify = y)
```

Лістинг 3.2 - Розділення фрейму даних на набори для навчання та тестування

Перш ніж розпочати навчання моделі за допомогою SVM, потрібно точно налаштувати гіперпараметри моделі, щоб вона добре працювала з набором даних, який у нас є. Для цього можна скористатися функцією GridSearchCV у модулі sklearn.

GridSearchCV — це функція, яка є в пакеті *model_selection* бібліотеки *sklearn*. Це дозволяє вказати різні значення для кожного гіперпараметра та спробувати всі можливі комбінації під час підгонки вашої моделі. Він виконує навчання та тестування за допомогою перехресної перевірки набору даних, звідси акронім CV у GridSearchCV. Кінцевим результатом GridSearchCV є набір гіперпараметрів, які найкраще відповідають вашим даним відповідно до показника скорингу, за яким необхідно оптимізувати свою модель.

Спробуємо знайти три гіперпараметри *C, Gamma* і *kernel*:

```
param_grid = {
    'C': [0.1, 1, 10, 100],
    'gamma' : [0.0001, 0.001, 0.1,1],
    'kernel' : ['rbf', 'poly']
}

svc = svm.SVC(probability=True)
print("Starting training, please wait ...")

model = GridSearchCV(svc,param_grid)
model.fit(x_train, y_train)

# print the parameters for the best performing model
print(model.best_params_)
```

Лістинг 3.3 - Пошук оптимальних гіперпараметрів для нашого набору даних.

Навчену модель можна протестувати на точність використовуючи тестовий набір даних (лістинг 3.4):

```
    y_pred = model.predict(x_test)
    print(f"The model is {accuracy_score(y_pred,y_test) *
100}% accurate")
```

Лістинг 3.4 - Знаходимо точність моделі.

Та зберігаємо навчену модель для подальшого використання без попереднього навчання:

```
    pickle.dump(model, open('faces_model.pickle','wb'))
```

Лістинг 3.5 - Зберігаємо навчану модель у файл.

Теперь можна використовуючи збережену модель виконати прогноз будь-якої фотографії обличчя:

```
    import cv2
    import pickle

    model = pickle.load(open('faces_model.pickle','rb'))
    categories    =    pickle.load(open('faces_labels.pk',
'rb'))

    facecascade = cv2.CascadeClassifier(
        'haarcascade_frontalface_default.xml')

    for i in range(1,40):
        test_image_filename = f'./facetest/face{i}.jpg'

    imgtest       =       cv2.imread(test_image_filename,
cv2.IMREAD_COLOR)
    image_array = np.array(imgtest, "uint8")

    faces = facecascade.detectMultiScale(imgtest,
        scaleFactor = 1.1, minNeighbors = 5)

    if len(faces) != 1:
        print(f'---We need exactly 1 face; photo skipped--
-\n')
        continue

    for (x_, y_, w, h) in faces:
        face_detect = cv2.rectangle(
            imgtest, (x_, y_), (x_+w, y_+h),
```

```
            (255, 0, 255), 2)
        plt.imshow(face_detect)
        plt.show()

        roi = image_array[y_: y_ + h, x_: x_ + w]
        img_resize = resize(roi, new_image_size)
        flattened_image = [img_resize.flatten()]

        probability = model.predict_proba(flattened_image)

        for i, val in enumerate(categories):
            print(f'{val}={probability[0][i] * 100}%')
            print(f"{categories[model.predict(flattened_i
mage)[0]]}")
```

Лістинг 3.6 - Виконання прогнозу для списка файлів з обличчями.

Рисунок 3.8 демонструє правильно виконане розпізнавання облич.

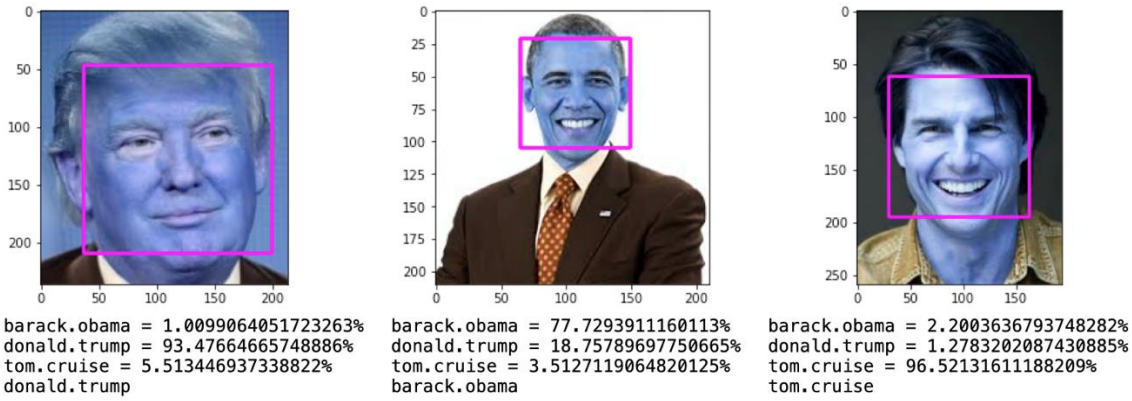
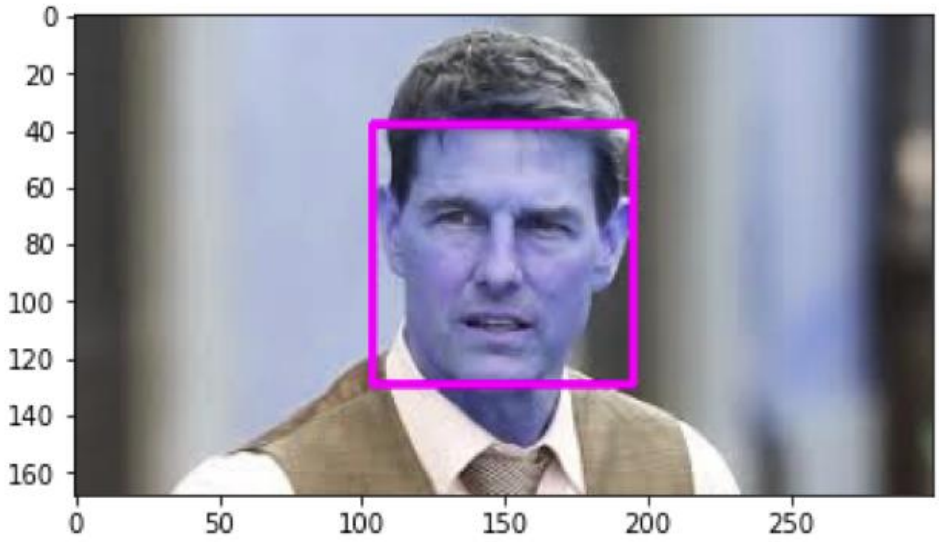


Рисунок 3.8 - Розпізнавання облич виконано правильно.

Але не завжди вдається точно виконати розпізнавання, трапляються помилкові ідентифікації (рис. 3.9).




```
barack.obama = 20.595171210647905%
donald.trump = 54.61251653603215%
tom.cruise = 24.792312253319913%
donald.trump
```

Рисунок 3.9 - Помилкове прогнозування людини алгоритмом.

3.3 Програмна реалізація серверного додатку.

Серверний додаток буде приймати повідомлення від своїх клієнтів (контролерів) дані, опрацьовувати та приймати подальші рішення. Якщо говорити про контролер ESP32 CAM, то дані будут представлені потоком кадрів з відео у реальному часі. Опрацював кадри, алгоритм серверного додатку визначає, чи знаходиться у кадрі обличчя довіреної особи. У випадку, коли таке обличчя знайдене, сервер по тому самому каналу зв’язку відправить відповідну команду про відчинення дверей (рис. 3.10).

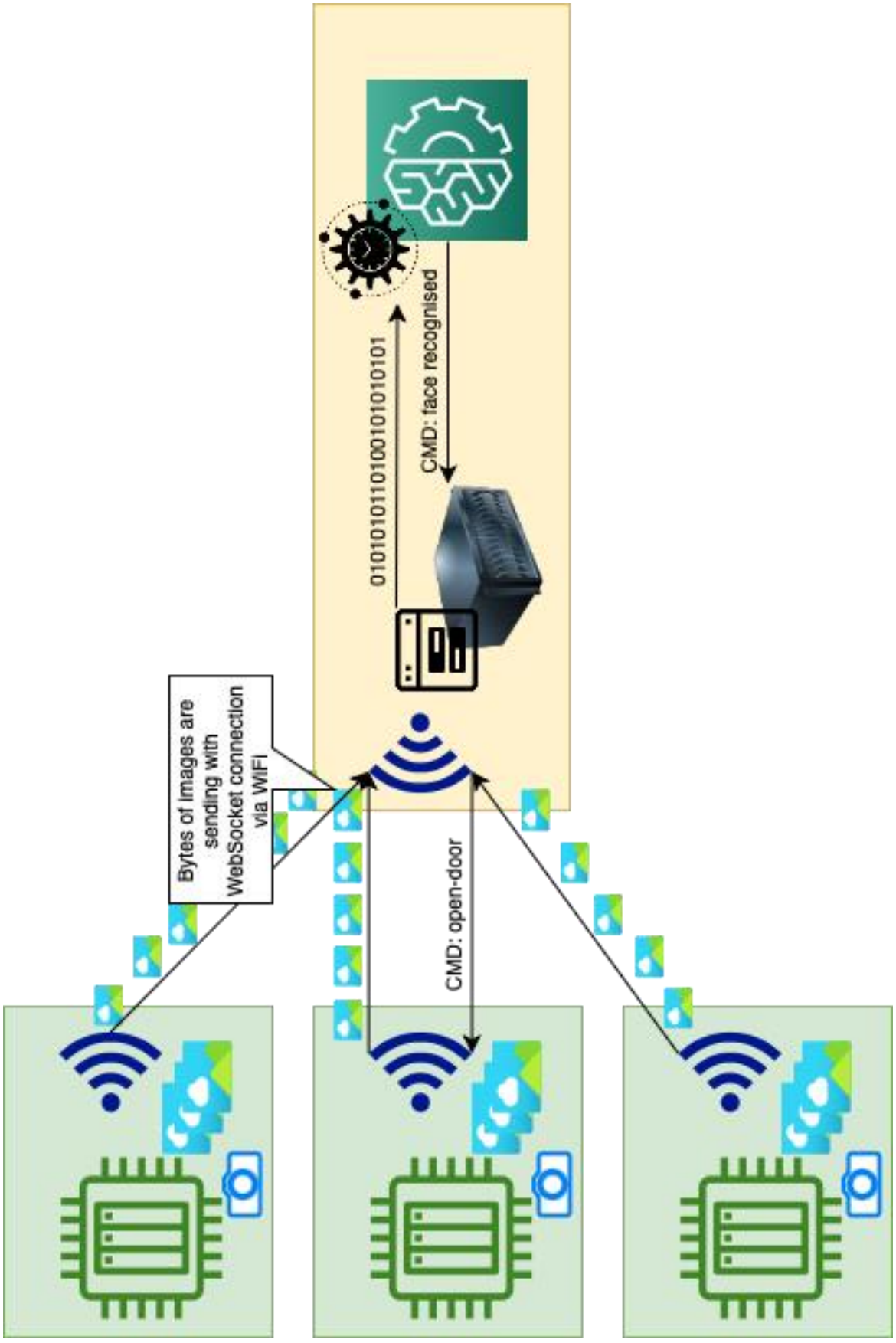


Рисунок 3.10 - Діаграма роботи охоронної системи.

Бажано, щоб серверний додаток міг працювати у багатопоточному режимі, тобто паралельно опрацьовував кожен запит клієнта, тим самим збільшуючи пропускну спроможність хабу у прямій залежності від максимальної кількості потоків у пулі процесу.

Однак, мова програмування Python та її найбільш поширений інтерпретатор CPython не може запускати два або більше потоки працювати паралельно через GIL - глобальний блокувальник інтерпретатора, який слідкує за тим, щоб лише один потік працював у будь-який момент часу. Інтерпретатор вказує процесору, на який саме потік необхідно витрачати процесорний час - зазвичай це 100 інструкцій на потік.

Якщо подумати глибше, то можна побачити, що багатопоточність хоч і могла б збільшити пропускну спроможність, але не так ефективно (з точки зору витрати процесорного часу) як асинхронне виконання програми.

Асинхронність у комп'ютерному програмуванні стосується виникнення подій, незалежних від основного потоку програми, і способів боротьби з такими подіями. Це можуть бути «зовнішні» події, такі як надходження сигналів, або дії, ініційовані програмою, які відбуваються одночасно з виконанням програми, без блокування програми для очікування результатів.

Асинхронне введення/виведення є прикладом останнього випадку асинхронності, і дозволяє програмам видавати команди до пристроїв зберігання чи мережевих пристроїв, які обслуговують ці запити, поки процесор продовжує виконувати програму. Це забезпечує певний ступінь паралелізму.

Таким чином, використовуючи асинхронне виконання програми, можна контролювати скільки процесорного часу виділяти на певні події і зводити до мінімуму "простої", коли процесор очікує результат виконання зовнішньої програми. У нашій системі це очікування кадрів від контролеру та очікування результату обчислення алгоритму розпізнавання облич.

На рис. 3.10 жовтим кольором зображений серверний додаток, що складається з двох частин: WebSocket сервер для отримання та надсилання даних та програма з навчаною моделлю для знаходження обличчя довіреної особи. Така архітектура, яка вміщує у собі декілька частин, що виконують різну бізнес-логіку, називається *монолітною архітектурою*, та стає все менш поширеною у сучасних програмних системах. Пов'язано це з розростанням программного коду, складності підтримки та оновлення систем з такою архітектурою.

Монолітна архітектура (рис. 3.11) — це традиційна модель програмного забезпечення, яка побудована як уніфікована одиниця, самодостатня та незалежна від інших програм. Слово «моноліт» часто приписують чомусь великому та льодовиковому, що не далеко від істини монолітної архітектури для розробки програмного забезпечення. Монолітна архітектура — це окрема велика обчислювальна мережа з єдиною кодовою базою, яка об'єднує всі бізнес-завдання. Щоб внести зміни в програму такого типу, потрібно оновити весь стек, отримавши

доступ до бази коду та створивши та розгорнувши оновлену версію інтерфейсу служби. Це робить оновлення обмежувальними та забирає багато часу.

Моноліти можуть бути зручними на ранніх стадіях життя проекту для полегшення керування кодом, когнітивних витрат і розгортання. Це дозволяє звільнити все в моноліті одночасно.

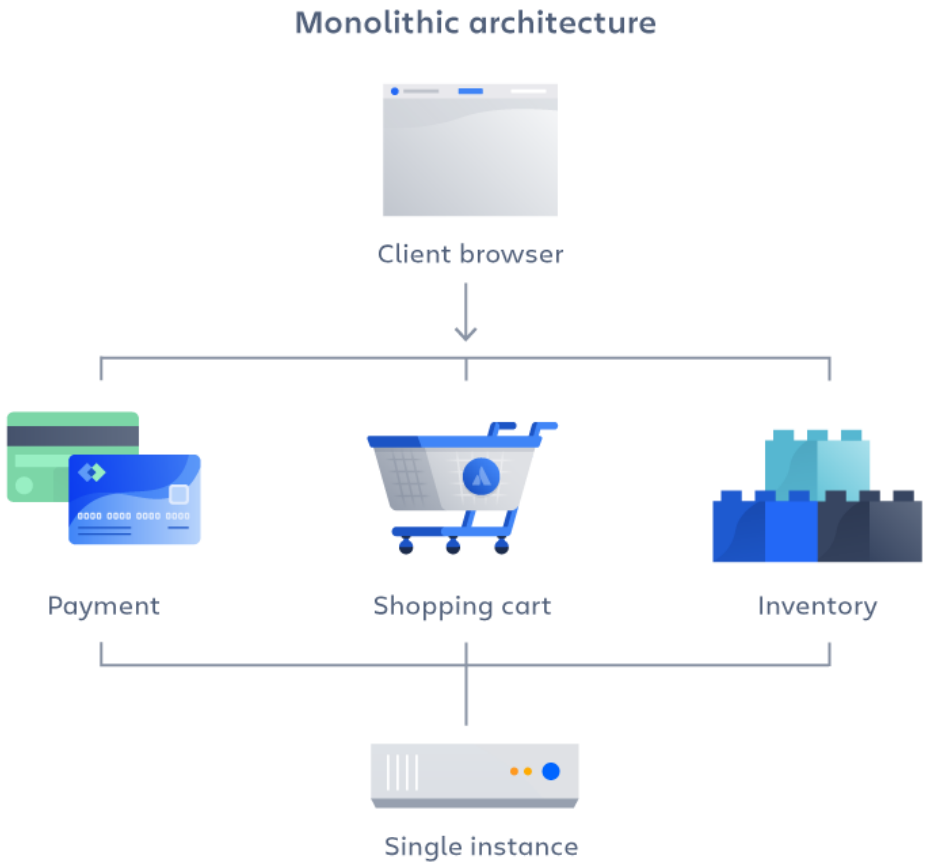


Рисунок 3.11 - Приклад монолітної архітектури.

Існує також мікросервісна архітектура, яка є протилежністю монолітній.

Архітектура мікросервісів (рис. 3.12), також відома просто як мікросервіси, — це архітектурний метод, який спирається на серію незалежно розгорнутих сервісів. Ці сервіси мають власну бізнес-логіку та базу даних із певною метою. Оновлення, тестування, розгортання та масштабування відбуваються в межах кожної служби. Мікросервіси відокремлюють основні бізнес-специфічні завдання на окремі незалежні бази коду. Мікросервіси не зменшують складність, але вони роблять будь-яку складність видимою та більш керованою, розділяючи завдання на менші процеси, які функціонують незалежно один від одного та роблять внесок у загальне ціле.

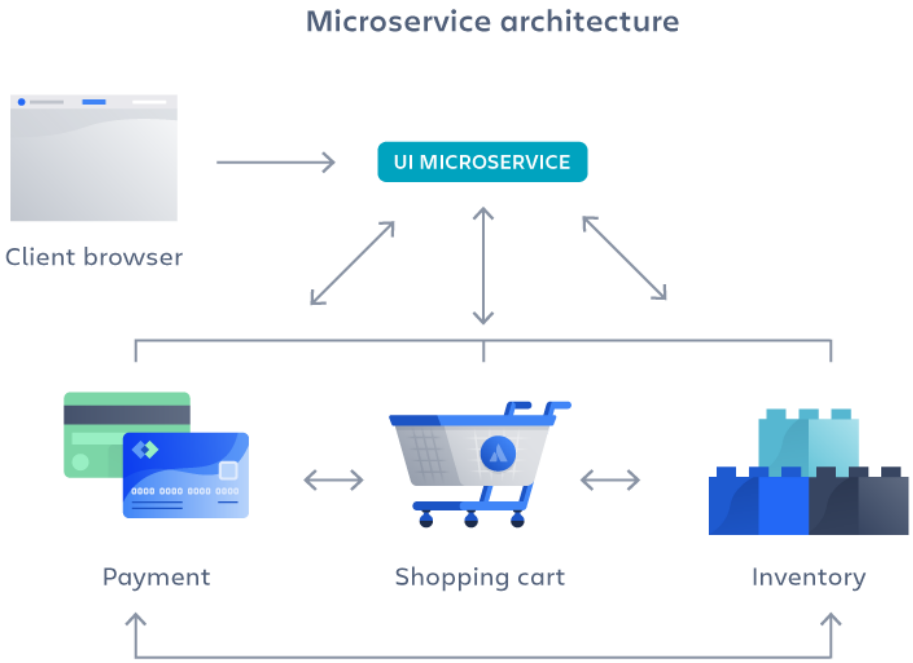


Рисунок 3.12 - Приклад мікросервісної архітектури.

3.3.1 Пересилання даних через Websocket

Спочатку реалізуємо комунікаційну частину серверного додатку - Websocket сервер, який буде по дуплексному підключенню отримувати байти кадрів з відео та повертати команди контролерам після виконання обчислень.

Використовуючи асинхронну бібліотеку *websockets* для Python3 можна у декількох строчок коду підняти Websocket сервер який може приймати та відправляти повідомлення контролерам (лістинг 3.7).

```
import asyncio
import websockets

async def echo(websocket):
    async for image_bytes in websocket:
        model.predicate(image_bytes)


async def main():
    async with websockets.serve(echo, "localhost",
8765):
        await asyncio.Future()
```

```
asyncio.run(main())
```

Лістинг 3.7 - Код серверного додатку для роботи Websocket серверу.

Оскільки нам потрібно, аби лише автентифіковані контролери могли надсилати повідомлення до хабу, необхідно реалізувати систему авторизації, використовуючи ID ключа та секретний ключ.

ID ключа дозволить серверу відрізняти повідомлення від різних клієнтів та знати, яким саме секретним ключем було зашифровано повідомлення.

У базовій версії продукту будемо зберігати пари ID ключа - секретний ключ у файлі з розширенням .yaml, але при подальшому розвитку продукту бажано використовувати бази даних, наприклад PostreSQL.

При встановленні TCP-підключення клієнту необхідно передати чотири додаткових Header:

1. Timestamp - час у unixtime коли був виконаний запит на встановлення підключення.
2. Content-Type - повинен бути *image/jpeg*, тому що сервер очікує отримувати лише фото.
3. Key-Id - ID ключа. Він необхідний, аби сервер міг зрозуміти, з використанням якого саме секретного ключа був згенерований підпис.
4. Signature - підпис, який згенеровано за наступним алгоритмом: *Signature = SHA512(x, y)*, де x - це рядок, який необхідно зашифрувати, в нашому випадку це Timestamp (unixtime), y - секретний ключ (сіль), який буде унікальним для кожного клієнта.

3.4 Налаштування контролера

Налаштування контролеру відбувається шляхом його прошивання. Для цього необхідно мати FTDI програматор, що під’єднаний до комп’ютера та дотримуватися принципової схеми (рис. 3.13)

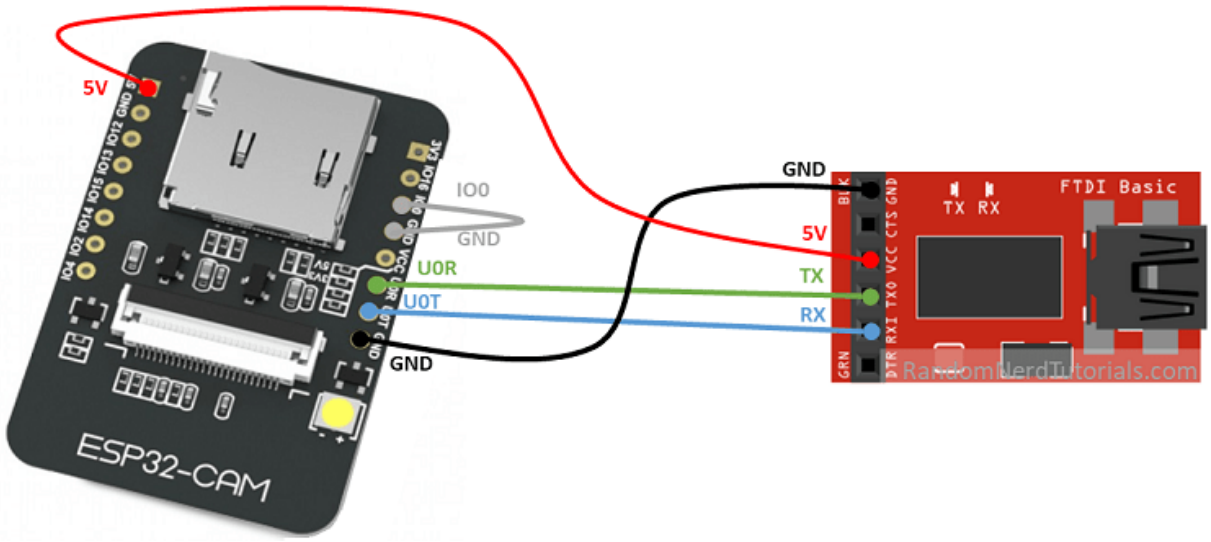


Рисунок 3.13 – Схема підключення програматора до контролера ESP32 CAM.

Використовуючи Arduino IDE та увімкнений flashing mode (режим перепрошивки), можна завантажити програму.

Спочатку необхідно встановити правильну конфігурацію контролера та розподілити порти:

```
camera_config_t config;
config.ledc_channel = LEDC_CHANNEL_0;
config.ledc_timer = LEDC_TIMER_0;
config.pin_d0 = Y2_GPIO_NUM;
config.pin_d1 = Y3_GPIO_NUM;
config.pin_d2 = Y4_GPIO_NUM;
config.pin_d3 = Y5_GPIO_NUM;
config.pin_d4 = Y6_GPIO_NUM;
config.pin_d5 = Y7_GPIO_NUM;
config.pin_d6 = Y8_GPIO_NUM;
config.pin_d7 = Y9_GPIO_NUM;
config.pin_xclk = XCLK_GPIO_NUM;
config.pin_pclk = PCLK_GPIO_NUM;
config.pin_vsync = VSYNC_GPIO_NUM;
config.pin_href = HREF_GPIO_NUM;
config.pin_sscb_sda = SIOD_GPIO_NUM;
config.pin_sscb_scl = SIOC_GPIO_NUM;
config.pin_pwdn = PWDN_GPIO_NUM;
config.pin_reset = RESET_GPIO_NUM;
config.xclk_freq_hz = 20000000;
config.pixel_format = PIXFORMAT_JPEG;
config.frame_size = FRAMESIZE_VGA;
config.jpeg_quality                =                10;
```

```
    config.fb_count = 2;
```

Лістинг 3.8 – Налаштування конфігурації контролера

Далі ініціюємо камеру та WiFi модуль і підключаємось до серверу:

```
     esp_err_t err = esp_camera_init(&config);
      if (err != ESP_OK) {
        Serial.printf("Camera init failed with error 0x%x",
err);
        return;
      }

      sensor_t *s = esp_camera_sensor_get();
      s->set_framesize(s, FRAMESIZE_SVGA);

      pinMode(12, INPUT);  // motion sensor
      pinMode(4, OUTPUT);  // white light
      pinMode(33, OUTPUT); // red light

      WiFi.begin(ssid, password);

      Serial.println("Connecting to WiFi: ");
      while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
      }
      Serial.println("");
      Serial.println("WiFi connected");

      // server address, port and URL
     webSocket.begin("192.168.0.100", 9999, "/");
     webSocket.onEvent(webSocketEvent);

      webSocket.sendTXT("Connected");
```

Лістинг 3.9 – Ініціалізація камера, WiFi модуля та підключення до серверу.

Життєвий цикл контролеру буде постійно очікувати появу людини завдяки вбудованому датчику руху. Як тільки но рух буде помічено, контролер почне стримінг відео по вебсокету до серверу та очікувати команди до відкриття дверей (Додаток А).

Як тільки написання програмного коду буде завершено, можна компілювати та завантажувати програму до контролеру, тобто виконувати прошивання (рис. 3.14).

Рисунок 3.14 – Виконання прошивання контролеру.

Висновок до розділу

Спроектований прототип охоронної системи на базі біометричної ідентифікації з використанням контролеру ESP32 CAM. Вбудований датчик руху дозволив не забивати мережевий трафік безкінечним потоком відео, що надсилається до серверу, оскільки дав можливість відслідковувати появу людини у зоні роботи контролера. Websocket протокол, TCP-підключення та шифрування з'єднання з алгоритмом SHA512 підтримують стабільний повно дуплексний захищений зв'язок між сервером та контролером. Програматор дозволяє лічені хвилини встановлювати прошивку на інші контролери ESP32 CAM, створюючи таким чином повноцінну охоронну мережу.

Серверний додаток у ролі центрального хаба виконує головну бізнес-логіку по знаходженню та ідентифікації обличчя довіреної особи за допомогою навчених моделей Haar Cascade та SVM. Безперешкодну умову. Віддаючи контролерам дистанційну команду про відкриття дверей у випадку успішної ідентифікації довіреної людини, система надає безперешкодні умови користувачам отримувати санкціоний доступ до приміщень без жодною взаємодії з системою.

ВИСНОВКИ

Розроблено модель інформаційної охоронної системи, яка, на відміну від існуючих, ґрунтується на процедурі біометричної ідентифікації з використанням машинного навчання з алгоритмами SVM та Haar Cascade, що підвищує точність розпізнавання та надійність системи в цілому.

Використання алгоритму Viola-Jones (Haar Cascade) дозволяє з мінімальної кількістю навчальних даних отримувати точність у понад 90% у пошуку обличчя на фото.

Безумовно, створена охоронна система є лише прототипом та потребує подальших доопрацювань і переоцінку доцільності використання SVM алгоритму для ідентифікації обличчя, оскільки він виявився недостатньо точним та потребує великої кількості навчальних даних для тренування моделі.

Однак, прототип вже використовує сучасні технології, наділений багатим функціоналом, здатен легко оновлюватися та має достатню точність розпізнавання людини за біометричними характеристиками аби бути використаним у некомерційних цілях.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1) Брагина Е. К., Соколов С. С. Современные методы биометрической аутентификации: обзор и определение перспектив развития // Вестник АГТУ. 2016. № 1 (61). С. 40-45.
2) Волхонський В.В. Системи охоронної сигналізації: 2-е вид. СПб.: Екополіс і культура, 2005. 204 с.
3) Ворона В.А., Тихонов В.А. Системи контролю і управління доступом. М.: Гаряча лінія. Телеком, 2010. 272 с.
4) Гінце О. СКУД – ключові тренди // Специальный выпуск журнала «Системы безопасности». 2018.
5) Гуреева О. Биометрическая идентификация по отпечаткам пальцев. Технология FingerChip // Компоненты и технологии. 2007. № 4 С. 176–180.
6) Динамические методы биометрической аутентификации личности: веб-сайт. URL: http://re.mipt.ru/infsec/ 2006/essay/2006_Dynamic_biometric_authentification__Cherkezov.pdf (дата обращения: 14.11.2022).
7) Евдокимов Н.О. Система контроля и управления доступом по аудиоданным пользователя // Психолого-педагогический журнал Гаудеамус. 2014. № 2 (24). С. 207-208.
8) Маркелов К. С., Нечаев В. В. Биометрические информационные технологии: актуальные и перспективные методы // Информационные и телекоммуникационные технологии. 2013. № 18. С. 24–42.
9) Методы аутентификации по сетчатке глаза: веб-сайт. URL: https://habrahabr.ru/post/261309/ (дата звернення 14.11.2022).
10) Омельянчук А.М. Формування системи комплексної безпеки. Частина 2. Підготовка техзавдання і проектування. Системи безпеки. СПб. 2009. 114-117 с.
11) Погребенник В.Д., Політило Р.В. Принципи побудови систем охоронної сигналізації // 2008. С. 93-99.
12) Тассов К. Л., Дятлов Р.А. Метод идентификации человека по голосу: веб-сайт. URL: HYPERLINK "http://engjournal.ru/catalog/it/biometric/1103.html" http://engjournal.ru/catalog/it/biometric/1103.html (дата зверненя 15.11.2022).
13) Bronstein, A.M., Bronstein, M.M., Gordon, E., and Kimmel. High-resolution structured light range scanner with automatic calibration. 2003. 430 с.
14) Carrihill, B. and Hummel. Experiments with the intensity ratio depth sensor. Computer Vision, Graphics and Image Processing. 1985. С. 337–358.

15) Cartoux, J.Y., La Preste, J.T., and Richetin. Face authentication or recognition by profile extraction from range images. 1989. С. 194–199.
16) Cox, I., Ghosn, J., and Yianilos. Feature-based face recognition using mixture distance. In Proc. 2006. С. 209–216.
17) Turk, M. and Pentland. Face recognition using eigenfaces.1991. С. 586–591.
18) Mavridis, N., Tsalakanidou, F., Pantazis, D., Malassiotis. The HISCORE face recognition application: Affordable desktop face recognition based on a novel 3D camera. 2001. С. 157–160.
19) Medioni, G. and Waupotitsch. Face recognition and modeling in 3D. 2003. С. 232–233.
20) Ortega-Garcia, J., Bigun, J., Reynolds, D., Gonzalez-Rodriguez. Authentication gets personal with biometrics. 2004. С. 50–62.

Додаток А
Життєвий цикл контролера

```
void loop() {
  webSocket.loop();

  val = digitalRead(12);
  if (val == HIGH) {
    if (motionState == false) {
      Serial.println("Motion detected!");
      motionState = true;
      digitalWrite(33, LOW);
      digitalWrite(4, HIGH);
    }
  } else {
    if (motionState == true) {
      Serial.println("Motion ended!");
      motionState = false;
      digitalWrite(33, HIGH);
      digitalWrite(4, LOW);
    }
  }
  if (motionState == true && webSocket.isConnected() ==
true) {
    streamCamera();
  }
}

void streamCamera(){
  camera_fb_t *fb = esp_camera_fb_get();
  if (!fb) {
      Serial.println("Frame buffer could not be acquired");
      return;
  }
  webSocket.sendBIN(fb->buf, fb->len, false);
  esp_camera_fb_return(fb);
}

WebSocketsClient webSocket;

void webSocketEvent(WStype_t type, uint8_t * payload,
size_t length) {

    switch(type) {
        case WStype_DISCONNECTED:
```

```
            Serial.printf("[WSc] Disconnected!\n");
            break;
        case WStype_CONNECTED:
            Serial.printf("[WSc] Connected to url: %s\n",
payload);
            break;
        case WStype_TEXT:
            Serial.printf("[WSc]    get    text:    %s\n",
payload);
            break;
        case WStype_BIN:
            Serial.printf("[WSc] get binary length: %u\n",
length);
            break;
        case WStype_ERROR:
        case WStype_FRAGMENT_TEXT_START:
        case WStype_FRAGMENT_BIN_START:
        case WStype_FRAGMENT:
        case WStype_FRAGMENT_FIN:
            break;
    }
}
```