МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ СИСТЕМ ЗАХИСТУ ІНФОРМАЦІЇ

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри Комп'ютеризованих систем захисту інформації

_____ Михайло СТЕПАНОВ
«____»_____ 2023 р.

На правах рукопису
УДК 004.42:004.62:004.8

КВАЛІФІКАЦІЙНА РОБОТА

ЗДОБУВАЧА ВИЩОЇ ОСВІТИ
ОСВІТНЬОГО СТУПЕНЯ «МАГІСТР»

Тема: Система моніторингу користувачів вебзастосунків з використанням технології SSO за геометрією обличчя

Виконавець: Наталія НОЩЕНКО
Керівник: к.т.н., Олена ВИСОЦЬКА
Консультант розділу «Охорона навколишнього середовища»: к.т.н., доцент Тетяна ДМИТРУХА
Нормоконтролер: к.т.н. Олена ВИСОЦЬКА

Київ 2023

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет: Кібербезпеки та програмної інженерії

Кафедра: Комп'ютеризованих систем захисту інформації

Освітній ступінь: Магістр

Спеціальність: 125 «Кібербезпека»

Освітньо-професійна програма: «Безпека інформаційних і комунікаційних систем»

ЗАТВЕРДЖУЮ
Завідувач кафедри Комп'ютеризованих систем захисту інформації

_____ Михайло СТЕПАНОВ
«___» _____ 2023 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи
здобувача вищої освіти Нощенко Наталії Володимирівни

1. Тема: Система моніторингу користувачів веб-застосунків з використанням технології SSO за геометрією обличчя
затверджена наказом ректора від «15» вересня 2023 р. № 1814/ст..
2. Термін виконання з 16.10.2023 р. по 31.12.2023 р.
3. Вихідні дані: проаналізувати технологію єдиного входу, методи авторизації, ідентифікації та моніторингу, а також існуючі методи біометричної ідентифікації та аналіз і обробку зображень; розробити системи моніторингу користувачів веб-застосунків, використовуючи технологію SSO, за геометрією обличчя; провести тестування розробленої системи моніторингу користувачів.
4. Зміст пояснювальної записки: аналіз технології єдиного входу, ідентифікації, автентифікації, моніторингу та біометричної ідентифікації; біометричні технології розпізнавання, аналіз та обробка зображень; розробка алгоритму та

програмної реалізації системи моніторингу та тестування та демонстрація роботи системи; охорона навколишнього середовища.

КАЛЕНДАРНИЙ ПЛАН

виконання кваліфікаційної роботи

№ з/п	Етапи виконання кваліфікаційної роботи	Термін виконання етапів	Примітка
1.	Уточнення постановки задачі	16.10.2023	*Виконано*
2.	Аналіз літературних джерел	17.10.2023-20.10.2023	*Виконано*
3.	Обґрунтування вибору рішення	21.10.2023	*Виконано*
4.	Збір інформації	22.10.2023-31.10.2023	*Виконано*
5.	Аналіз технології єдиного входу, ідентифікації, автентифікації, моніторингу та біометричної ідентифікації	01.11.2023-07.11.2023	*Виконано*
6.	Аналіз біометричних технологій розпізнавання. Аналіз та обробка зображень	08.11.2023-14.11.2023	*Виконано*
7.	Розробка алгоритму та програмної реалізації системи моніторингу. Тестування та демонстрація роботи системи	15.11.2023-04.12.2023	*Виконано*
8.	Апробація роботи на науково-практичній конференції «Живучість та резильєнтність – 2023»	19.10.2023	*Виконано*
9.	Оформлення і друк пояснювальної записки	05.12.2023-11.12.2023	*Виконано*
10.	Перевірка на антиплагіат	12.12.2023	*Виконано*
11.	Оформлення презентації	12.12.2023	*Виконано*
12.	Отримання рецензій	18.12.2023	*Виконано*

Консультанти з окремих розділів

Розділ	Консультант (посада, П.І.Б.)	Дата, підпис	
		Завдання видав	Завдання прийняв
Охорона навколишнього середовища	Дмитруха Т.І.		

Дата видачі завдання: «16» жовтня 2023 р.

Здобувач вищої освіти Наталія НОЩЕНКО

(підпис, дата)

Керівник кваліфікаційної роботи Олена ВИСОЦЬКА

(підпис, дата)

РЕФЕРАТ

Кваліфікаційна робота на тему «Система моніторингу користувачів веб-застосунків з використанням технології SSO за геометрією обличчя» складається зі вступу, основної частини, що містить 4 розділи, загального висновку та списку використаних джерел Загальний обсяг роботи – 110 сторінок. Робота містить 41 рисунок та 3 таблиці. Список використаних джерел включає 30 джерел.

Метою роботи є розробка системи моніторингу користувачів веб-застосунків, в якій завдяки поєднанню авторизації, з використанням технології SSO, та ідентифікації за геометрією обличчя відбувається контроль доступу користувачів до системи.

В роботі проаналізовано технологію єдиного входу, методи авторизації, ідентифікації та моніторингу, а також існуючі методи біометричної ідентифікації та аналіз і обробку зображень,

В роботі розроблено систему моніторингу користувачів веб-застосунків, з використанням технології SSO, в якій завдяки реалізації розпізнавання за геометрією обличчя, забезпечується контроль доступу користувачів до веб-застосунку

Розроблений метод та програмне забезпечення можуть бути використані на підприємствах для контролю доступу авторизованих користувачів.

Запропоновані методи пов’язані з метою забезпечення високого рівня безпеки та зручності для доступу до веб-систем.

Ключові слова: моніторинг, авторизація, ідентифікація, біометричні технології, технологія єдиного входу, ідентифікація за геометрією обличчя.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

2-D	2-dimensional
3-D	3-dimensional
IT	Інформаційні технології
API	Application Programming Interface
CBA	Context-Based Authentication
C-HOG	Compressed Histogram of Oriented Gradients
CNN	Convolutional Neural Network
CSS	Cascading Style Sheets
CV	Computer Vision
HOG	Histogram of Oriented Gradients
HTML	HyperText Markup Language
IAM	Identity and Access Management
ID	Identification
IDE	Integrated Development Environment
IdP	Identity Provider
IP	Internet Protocol
LBP	Local Binary Patterns
LDA	Linear Discriminant Analysis
MFA	Multi-Factor Authentication
ORM	Object-Relational Mapping
OTP	One-Time Password
PCA	Principal Component Analysis
REST	Representational State Transfer
RFID	Radio-Frequency Identification
RGB	Red Green Blue
SaaS	Software as a Service
SPOI	Single Point of Interaction

SQL	Structured Query Language
SSO	Single Sign-On
SURF	Speeded-Up Robust Features
SVM	Support Vector Machine
WSGI	Web Server Gateway Interface

ВСТУП

Актуальність. З розвитком технологій та збільшенням використання веб-застосунків у сучасному світі, виникає необхідність у забезпеченні безпеки, контролю та ефективного управління доступом користувачів до цих систем. Одним із важливих методів для цього є використання технології SSO (Single Sign-On), яка дозволяє користувачам отримувати доступ до різноманітних ресурсів за допомогою єдиного набору облікових даних.

Проте проблеми безпеки залишаються актуальними, адже важливо переконатися, що доступ надається саме правильним користувачам. У такому контексті технологія розпізнавання обличчя стає важливим інструментом, що забезпечує додатковий рівень безпеки та автентифікації користувачів. Н.Б. Шаховська, О.А. Басистюк у своїй статті досліджують алгоритми розпізнавання облич, що ґрунтуються на машинному навчанні, розглядаючи методи їх покращення, та пропонують використання гістограми орієнтованих градієнтів, L2-нормування та оцінки орієнтирів обличчя для отримання найкращого результату у розпізнаванні [1].

Цей напрямок досліджень поєднує в собі інноваційні методи автентифікації з використанням геометрії обличчя та технології SSO для створення систем, що не лише забезпечують зручний доступ до ресурсів, але й гарантують високий рівень захищеності та контролю за користувачами.

Системи моніторингу користувачів веб-застосунків стають невід'ємною складовою для багатьох компаній, що прагнуть забезпечити безпеку, ефективність та контроль у використанні цифрових ресурсів. Одним з важливих елементів таких систем є автентифікація - процес перевірки та підтвердження ідентичності користувача перед доступом до веб-застосунків. Використання технології SSO (Single Sign-On), що поєднує у собі декілька систем управління доступом до різних платформ під одним логіном, робить процес автентифікації більш зручним та безпечним.

Технологія SSO в поєднанні з методами розпізнавання за геометрією обличчя розширює можливості безпечного входу до систем, дозволяючи використовувати біометричні дані для ідентифікації. Цей підхід відкриває нові перспективи в області безпеки та контролю за доступом, забезпечуючи спрощений, але водночас надійний механізм перевірки особи.

Різні методи біометричної автентифікації, аналіз їх ефективності за різними показниками, та визначення найоптимальніших методів з найменшою кількістю помилок і відмов у роботі дослідили Г.Є. Ляшенко, А.А. Астраханцев [2].

Відомі підходи до вирішення поставленої задачі. До відомих підходів у розробці систем моніторингу користувачів веб-застосунків, що використовують технологію SSO (Single Sign-On) та геометрію обличчя, можна віднести інтеграцію системи автентифікації за допомогою SSO із модулем розпізнавання обличчя. Це дозволяє користувачам автоматично увійти до системи за допомогою одного облікового запису та забезпечити їм доступ до веб-застосунків, використовуючи геометрію їх обличчя для ідентифікації. Такий підхід сприяє покращенню безпеки, зручності використання та забезпечує ефективний механізм контролю доступу до системи.

Метою роботи є розробка системи моніторингу користувачів веб-застосунків, в якій завдяки поєднанню авторизації, з використанням технології SSO, та ідентифікації за геометрією обличчя відбувається контроль доступу користувачів до системи.

Для досягнення поставленої мети вирішуються такі **задачі:**

-Проаналізувати технологію єдиного входу, методи авторизації, ідентифікації та моніторингу, а також існуючі методи біометричної ідентифікації та аналіз і обробку зображень, та на основі проведеного аналізу обрати оптимальні методи для вирішення задачі ідентифікації користувачів.

-Розробити систему моніторингу користувачів веб-застосунків, з використанням технології SSO, в якій завдяки реалізації розпізнавання за

геометрією обличчя, забезпечується контроль доступу користувачів до вебзстосунку.

-Провести тестування розробленої системи моніторингу користувачів, що дасть змогу дослідити доцільність використання розробленої системи для вирішення поставленої задачі.

-Описати індивідуальну тему з охорони тему навколишнього середовища – «Біологічне забруднення довкілля».

Галузь застосування. Розроблений метод та програмне забезпечення можуть бути використані на підприємствах для контролю доступу авторизованих користувачів.

Об'єктом дослідження є процес моніторингу та автентифікації користувачів веб-застосунків з використанням технології SSO за геометрією обличчя.

Предметом дослідження. є технології та методи моніторингу користувачів, автентифікації на основі технології SSO, а також ідентифікації користувачів за геометрією обличчя.

Методи дослідження включають порівняльний аналіз різних методів ідентифікації користувачів, систем моніторингу, технологій біометрії, а також використання методів моделювання для розробки прототипу системи та оцінки її продуктивності.

Новизна одержаних результатів полягає в наступному: розроблено систему моніторингу користувачів, в якій завдяки використанню технології SSO та реалізації розпізнавання за геометрією обличчя, за допомогою методу гістограми напрямлених градієнтів, забезпечується контроль доступу користувачів до вебзастосунку.

Практична цінність отриманих результатів: розроблено систему моніторингу користувачів, з використанням технології SSO, за геометрією обличчя, що може бути використана компаніями, що прагнуть контролювати доступ авторизованих користувачів. Система була розроблена мовою програмування Python, з використанням бібліотек OpenCV, Dlib,

face_recognition, SQLAlchemy, Flask та для ідентифікації облич використовувався метод гістограми напрямлених градієнтів.

Апробація. Нощенко Н.В. Система моніторингу користувачів веб-застосунків з використанням технології SSO за геометрією обличчя.// Живучість та резильєнтність – 2023: міжнародна науково-практична конференція 19 жовтня 2023 р.: тези доповіді. – К., 2023. – С.149-152 [3].

РОЗДІЛ І. АНАЛІЗ ТЕХНОЛОГІЇ ЄДИНОГО ВХОДУ, ІДЕНТИФІКАЦІЇ, АВТЕНТИФІКАЦІЇ, МОНІТОРИНГУ ТА

1.1. Технологія SSO

В умовах швидкого розвитку веб-застосунків та зростаючого попиту на безпеку і зручність користування виникає необхідність у нових рішеннях для моніторингу користувачів та забезпечення їхньої безпеки. Технологія єдиного входу (SSO) стала важливою складовою сучасних систем ідентифікації та автентифікації, дозволяючи користувачам уникати необхідності повторно вводити облікові дані на різних веб-платформах.

У цьому контексті ідентифікація за геометрією обличчя стає об'єктом зростаючого інтересу для створення нових методів моніторингу користувачів. Використання геометрії обличчя в поєднанні з технологією SSO надає можливість створити системи, що поєднують надійність біометричної ідентифікації з зручністю SSO.

Single sign-on (SSO) – Технологія єдиного входу – служба автентифікації сеансу та користувача , що дозволяє користувачеві використовувати один набір облікових даних для входу (зазвичай ім’я користувача та пароль) для доступу до кількох програм, різних систем, додатків чи ресурсів [4].

Система єдиного входу забезпечує безперебійний доступ для користувачів під час використання ваших програм і служб. Замість того, щоб запам’ятовувати окремі набори облікових даних для кожної програми чи служби, користувачі можуть просто ввійти один раз і отримати доступ до повного набору програм.

Кожного разу, коли користувачі переходять до домену, який потребує автентифікації, вони переходять до домену автентифікації, де їх може попросити ввійти. Якщо користувач уже ввійшов у домен автентифікації, його можна негайно переспрямувати до початкового домену без входу знову.

Система єдиного входу часто використовується в бізнес-контексті, коли користувальницькі програми призначаються та керуються внутрішньою IT-групою. Віддалені працівники, які використовують програми SaaS, також виграють від використання SSO.

SSO є важливим аспектом багатьох рішень для управління ідентифікацією та доступом (IAM) або контролю доступу. Перевірка особи користувача має вирішальне значення, щоб знати, які дозволи повинен мати кожен користувач.

1.1.1 Ключові аспекти технології SSO.

Кілька ключових аспектів і можливостей цього підходу зображена на рис.1.1.:

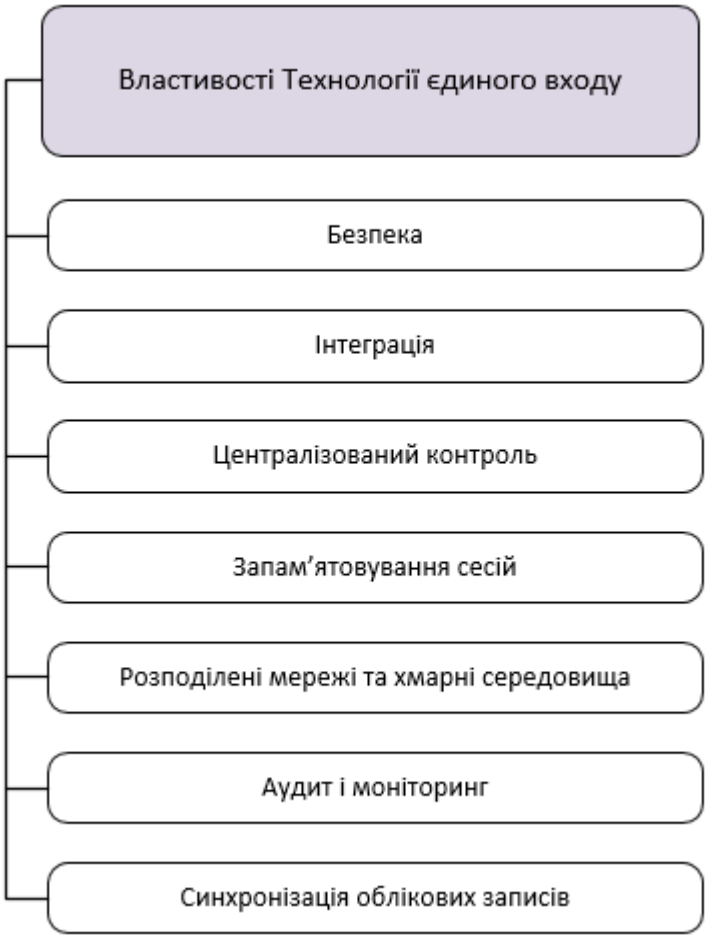


Рис. 1.1 Властивості Технології єдиного входу

1) Безпека: SSO може бути реалізований з додатковими шарами безпеки, такими як багатофакторна автентифікація (MFA) або біометрична ідентифікація, що забезпечує вищий рівень безпеки.

2) Інтеграція: SSO може легко інтегруватися з існуючими системами і додатками, дозволяючи користувачам однаково легко отримувати доступ до всіх ресурсів.

3) Централізований контроль: Адміністратори можуть керувати правами доступу та політиками безпеки в одному централізованому місці, що спрощує управління системами та додатками.

4) Запам'ятовування сесій: SSO може зберігати інформацію про сесію користувача, що дозволяє користувачам переходити від одного ресурсу до іншого без необхідності знову вводити облікові дані.

5) Розподілені мережі та хмарні середовища: SSO може бути особливо корисним у розподілених мережах та хмарних середовищах, де користувачі можуть потребувати доступу до різних хмарних ресурсів та додатків.

6) Аудит і моніторинг: SSO може спрощувати аудит та моніторинг доступу, дозволяючи відстежувати активність користувачів та виявляти потенційні загрози.

7) Синхронізація облікових записів: SSO також може включати можливість синхронізації облікових записів користувачів з різними системами, щоб забезпечити актуальність інформації та спростити процес управління користувачами.

SSO є важливим компонентом сучасної інформаційної безпеки та управління ідентичністю. Він дозволяє підприємствам та організаціям підвищити безпеку, зручність та ефективність доступу користувачів до різних ресурсів, що їм потрібні для роботи [5].

1.1.2. Переваги та недоліки технології SSO.

Переваги та недоліки технології SSO (рис. 1.2):

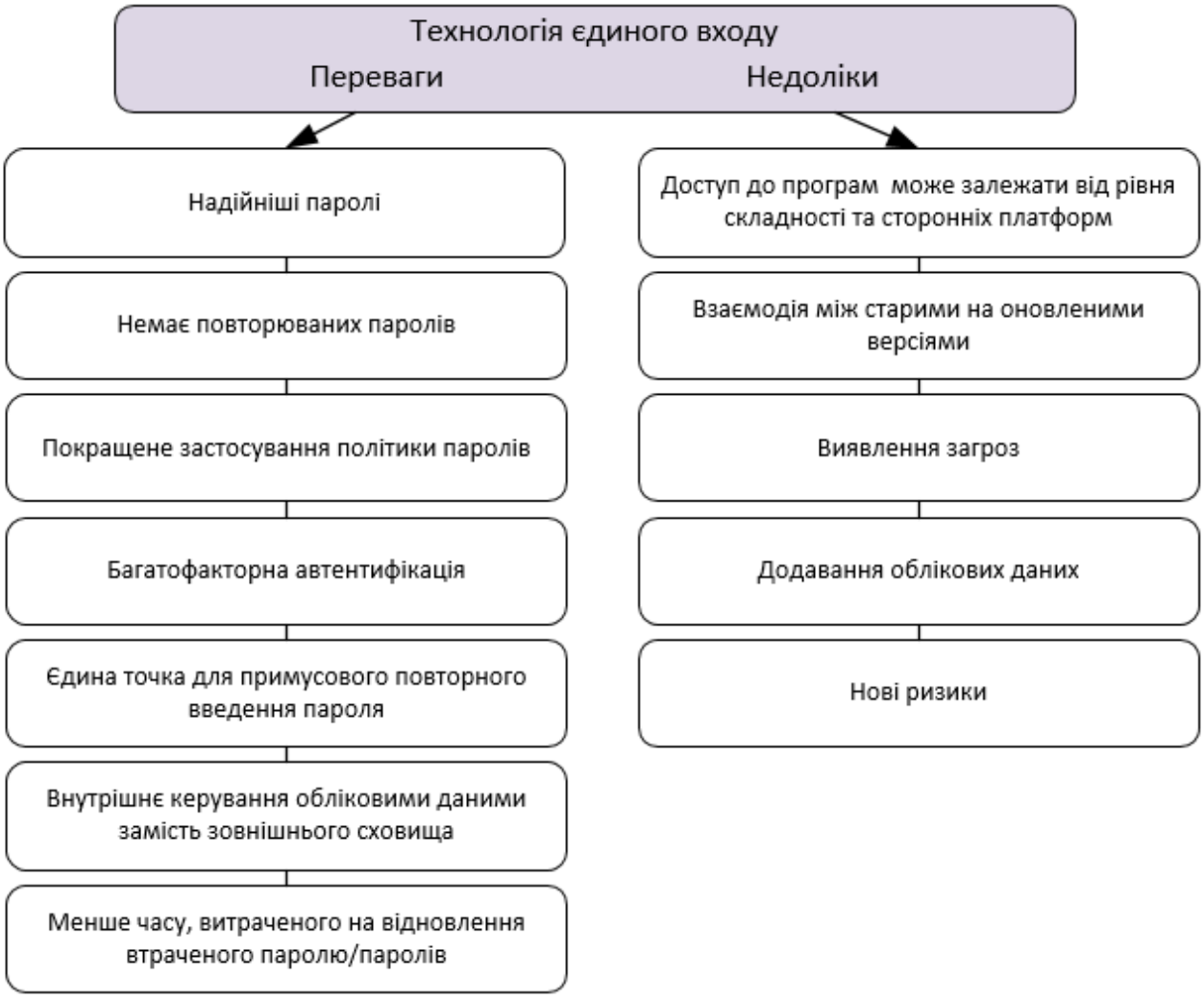


Рис. 1.2. Переваги та недоліки Технології єдиного входу

Крім того, що SSO є набагато простішим і зручнішим для користувачів, SSO вважається більш безпечним. Ось кілька переваг SSO:

- Надійніші паролі: оскільки користувачам потрібно використовувати лише один пароль, SSO полегшує створення, запам’ятовування та використання надійніших паролів. На практиці це зазвичай так: більшість користувачів використовують надійніші паролі з SSO.
- Немає повторних паролів: коли користувачам доводиться запам’ятовувати паролі для кількох різних програм і служб, може виникнути стан, відомий як «втома від пароля»: користувачі повторно використовуватимуть паролі для різних служб. Використання одного пароля для кількох служб становить величезний ризик для безпеки, оскільки це означає, що всі служби

настільки ж безпечні, як і служба з найслабшим паролем: якщо база даних паролів цієї служби скомпрометована, зловмисники можуть використати пароль, щоб зламати всі облікові записи користувача. а також інші послуги. SSO усуває цей сценарій, скорочуючи всі входи до одного входу.

- Покращене застосування політики паролів: єдине місце для введення пароля забезпечує IT-командам спосіб легко застосовувати правила безпеки паролів. Наприклад, деякі компанії вимагають від користувачів періодично скидати паролі. За допомогою системи єдиного входу скидання пароля легше реалізувати: замість постійного скидання пароля в різних програмах і службах користувачі мають лише один пароль для скидання. (Хоча цінність регулярного скидання пароля була поставлена під сумнів, деякі IT-команди досі вважають їх важливою частиною своєї стратегії безпеки).

- Багатофакторна автентифікація: багатофакторна автентифікація, або MFA, стосується використання більш ніж одного фактору ідентифікації для автентифікації користувача. Наприклад, окрім введення імені користувача та пароля, користувачеві може знадобитися підключити USB-пристрій або ввести код, який відображається на його смартфоні. Володіння цим фізичним об’єктом є другим «фактором», який підтверджує, що користувач є тим, ким він себе видає. MFA набагато безпечніше, ніж покладатися лише на пароль. SSO дає змогу активувати MFA в одній точці замість того, щоб активувати його для трьох, чотирьох або кількох десятків програм, що може бути неможливим.

- Єдина точка для примусового повторного введення пароля: адміністратори можуть примусово повторно вводити облікові дані через певний проміжок часу, щоб переконатися, що той самий користувач усе ще активний на пристрої, на якому виконано вхід. Завдяки системі єдиного входу вони мають центральне місце, з якого можна робити це для всіх внутрішніх програм, замість того, щоб застосовувати її в кількох різних програмах, які деякі програми можуть не підтримувати.

- Внутрішнє керування обліковими даними замість зовнішнього сховища: зазвичай паролі користувачів зберігаються віддалено некерованим

способом програмами та службами, які можуть дотримуватися або не дотримуватися найкращих практик безпеки. Однак за допомогою SSO вони зберігаються внутрішньо в середовищі, яке IT-команда має більше контролю.

- Менше часу, витраченого на відновлення пароля: окрім вищезазначених переваг безпеки, SSO також скорочує час, який витрачають внутрішні команди. IT-спеціалісти повинні витрачати менше часу на те, щоб допомогти користувачам відновити або скинути паролі для десятків програм, а користувачі витрачають менше часу на вхід у різні програми, щоб виконувати свою роботу. Це має потенціал для підвищення продуктивності бізнесу [6].

Однак важливо знати про можливі недоліки SSO:

- Доступ: впровадження нових рівнів складності та залежність від сторонніх платформ може вплинути на доступ до програм, якщо портал для входу стає недоступним або не працює.
- Взаємодія. Застарілі програми, які не підтримують систему єдиного входу, можуть створювати проблеми для IT-команд, яким може знадобитися створювати винятки та обхідні шляхи, щоб забезпечити продовження роботи цих програм.
- Виявлення загроз: надмірна залежність від одного механізму автентифікації може ускладнити виявлення підозрілої поведінки групами безпеки.
- Додавання облікових даних: система єдиного входу може підвищити ризик атак із додаванням облікових даних, коли хакери використовують скомпрометовані сеанси або викрадені облікові дані з однієї програми, щоб отримати доступ до іншої програми, підключеної до тієї самої системи єдиного входу.
- Нові ризики: система єдиного входу може створити нові ризики, наприклад посилити вплив зламаних облікових записів [7].

1.1.3. Загрози, пов’язані з використанням технології SSO

Використання SSO може покращити безпеку та зручність доступу користувачів до різних систем та ресурсів, але також вносить деякі потенційні загрози. Ось кілька головних загроз, пов'язаних із використанням SSO (рис. 1.3):

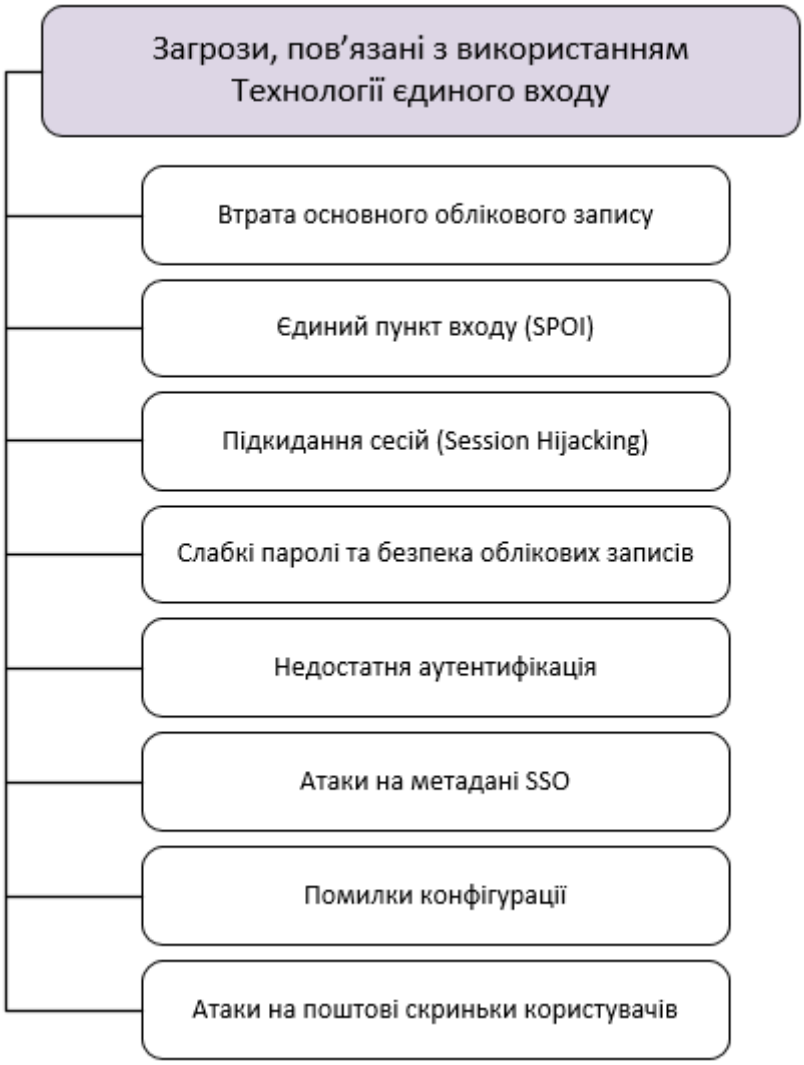


Рис. 1.3. Загрози, пов’язані з використанням Технології єдиного входу

Розглянемо більш детально:

1) Втрата основного облікового запису: Якщо основний обліковий запис SSO компрометований (наприклад, через слабкий пароль або фішинговий атаки), зловмисники можуть отримати доступ до всіх систем та ресурсів, підключених до SSO.

2) Сингл-пункт входу (SPOI): SSO створює одну точку входу, і якщо ця точка стає ціллю атаки, то успішна атака може негативно вплинути на доступ до всіх пов'язаних ресурсів.

3) Підкидання сесій (Session Hijacking): Зловмисники можуть спробувати викрасти або перехопити активну сесію користувача, отримуючи доступ до його облікового запису без необхідності знати пароль.

4) Слабкі паролі та безпека облікових записів: Якщо користувачі встановлюють слабкі паролі або не дотримуються інших безпечних практик, їхні облікові записи можуть бути вразливі до атак.

5) Недостатня автентифікація: Якщо система SSO не використовує додаткові методи автентифікації, такі як багатофакторна автентифікація (MFA), це може залишити облікові записи користувачів вразливими до атак.

6) Атаки на метадані SSO: Зловмисники можуть спробувати отримати доступ до метаданих SSO, таких як маркери або токени, що можуть бути використані для атак на ідентифікацію.

7) Помилки конфігурації: Неналежна конфігурація SSO може призвести до витоку даних або доступу до неавторизованих користувачів.

8) Атаки на поштові скриньки користувачів: Зловмисники можуть намагатися використовувати SSO для доступу до поштових скриньок користувачів, щоб виконати атаки фішингу або обману.

1.1.4. Принцип роботи технології SSO

SSO базується на взаємодії між додатком/провайдером послуг та системою керування доступу. Така взаємодія базується на обміні сертифікату між системою керування доступом та провайдером послуг. Такий сертифікат може використовуватись для позначення ідентифікаційної інформації, яка надсилається від системи керування доступом провайдеру послуг. В SSOідентифікаційні данні приймають форму токенів, що мають ідентифікаційні значення інформації по користувача такі, як emailта ім’я користувача.

Порядок авторизації зазвичай виглядає наступним чином (рис. 1.4):

1) Користувач заходить в додаток чи на сайт, доступ до якого він хоче отримати, тобто до провайдера послуг.

2) Провайдер послу надсилає токен, що містить інформацію про користувача (email та ім’я користувача) системі SSO (також, відомій як система керування доступами), як частину запиту на автентифікацію користувача.

3) В першу чергу, система керування доступом перевіряє, чи був користувач автентифікований до цього моменту. Якщо так, він надає користувачу доступ до додатку провайдера послуг, одразу переходячи до кроку 5.

4) Якщо користувач не авторизований, йому необхідно це зробити, надавши ідентифікаційні данні, що вимагає система керування доступами. Це може бути просто логін та пароль або інші вид автентифікації, наприклад одноразовий пароль (OTP – One Time Password).

5) Як тільки система керування доступом підтвердить ідентифікаційні данні, вона поверне токен провайдеру послуг, підтверджуючи успішну автентифікацію.

6) Цей токен проходить «крізь браузер» користувача провайдеру послуг.

7) Токен, отриманий провайдером послуг, підтверджується згідно з домовленостями, встановленими між провайдером послуг та системою керування доступами під час першої перевірки.

8) Користувачу надається доступ до провайдеру послуг [8].

Рис. 1.4. Схематичне зображення принципу роботи SSO

1.2. Ідентифікація

Ідентифікація – процес розпізнавання об’єкта чи суб’єкту, за допомогою певного ідентифікатора. Ідентифікація є першим кроком у підтвердженні особи людини і має відбутися перед автентифікацією та авторизацією. Користувачі також можуть надати додаткову інформацію, як-от державну фотографію, ідентифікаційний код або номер соціального страхування, щоб додатково ідентифікувати себе [9].

1.2.1. Види ідентифікації та типи ідентифікаторів

На сьогодні існує кілька способів ідентифікації користувачів, в кожного є як переваги, так і недоліки. В загальному існує три види ідентифікації:

1. Парольна ідентифікація. Донедавна парольна ідентифікація була чи не єдиним методом визначення особистості користувача. Кожен авторизований користувач отримує набір індивідуальних реквізитів (логін – пароль). Далі при

кожному вході необхідно вказати свою інформацію. Перевагою парольної ідентифікаціє є те, що вона проста в реалізації й використанні. До недоліків можна віднести залежність надійності ідентифікації безпосередньо від користувачів, а саме від складності паролів, що вони обрали, а також від їх зберігання.

2. Апаратна ідентифікація базується на визначені особистості користувача за певним предметом, ключі (карткові ключі або токени), що є в ексклюзивному використанні. Ідентифікація на основі магнітної картки – менш надійна, за рахунок того, що вони не захищені від копіювання та чутливі до механічних пошкоджень. Смарт-карти або токени можна вважати більш надійними, через можливість використання двофакторної ідентифікації, та такий метод має свої недоліки (ключ легко загубити, а також такий метод потребує більше витрат на його реалізацію).

Також до апаратної ідентифікації можна ще віднести два види : радіочастотну RFID та штрих-кодову ідентифікацію. Перша використовується при ідентифікації товарів у торгівлі, а також може використовуватись для ідентифікації особи (через картку з штрих-кодом, що їй видається). Проте такий метод має суттєвий недолік – його легко підробити, через те, що інформація знаходиться у відкритому вигляді. Метод радіочастотної ідентифікації полягає у використанні двох пристроїв – пристрої зчитування або базового блока та транспортера, чи RFID-позначки.

3. Біометрична ідентифікація. Цей метод ідентифікації особи є найбільш новим у галузі інформаційних технологій. Ідентифікація людини, в такому випадку, відбувається за унікальними біологічними ознаками, властивим тільки їй. До переваг такої ідентифікації можна віднести те, що ідентифікатори майже не можливо втратити чи забути, їх складно підробити, а також кожна людина володіє неповторними біометричними ознаками. До недоліків можна віднести грошову затратність та підробку таких ідентифікаторів як відбитки пальців, голо чи підпис [10].

Ідентифікація зазвичай відбувається шляхом надання імені користувача, адреси електронної пошти або номера телефону, але ідентифікація також може відбуватися за допомогою системи єдиного входу та інших методів. Ось кілька різних типів ідентифікаторів (рис. 1.5.):

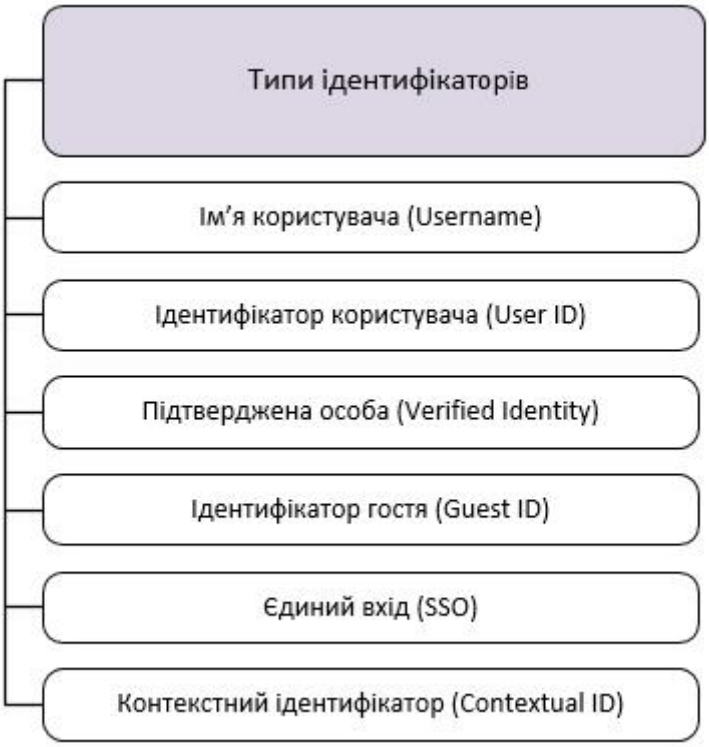


Рис. 1.5. Типи ідентифікаторів

Розглянемо детальніше:

- Ім’я користувача (Username): це ім’я за вашим вибором, яке ідентифікує ваш онлайн-акаунт.
- Ідентифікатор користувача (User ID): ідентифікатор користувача зазвичай надається сервером або адміністратором, тому це може бути випадковий алфавітно-цифровий шаблон, ряд цифр або частина вашого імені чи адреси електронної пошти.
- Підтверджена особа (Verified Identity): для «важливих» облікових записів (які також часто стають мішенню кіберзлочинців) вам може знадобитися надати рівні підтвердження для облікового запису. Це може включати

надсилання державного посвідчення особи з фотографією, як-от посвідчення водія або паспорт.

- Ідентифікатор гостя (Guest ID): це протилежність підтвердженому ідентифікатору, оскільки це одноразовий ідентифікатор сеансу, який, швидше за все, не містить додаткової інформації. Прикладом може бути придбання квитків на спортивну чи музичну подію; неважливо, яка особа, оскільки квиток є анонімним.
- Єдиний вхід (Single Sign-On): це форма ідентифікації, поширена серед підприємств, за допомогою якої довірений постачальник ідентифікаційної інформації (IdP) надає користувачеві ідентифікаційну інформацію, яку він може використовувати для входу в кілька облікових записів.
- Контекстний ідентифікатор (Contextual ID): коли ви пересуваєтеся в Інтернеті, різні служби створюють ваш профіль, часто навіть не знаючи вашого імені. Це використовуватиме файли cookie, збережені у вашому браузері, або контекстну інформацію, таку як ваша IP-адреса, тип пристрою чи географічне розташування, щоб ідентифікувати вас [11].

1.3. Автентифікація

Автентифікація — це те, як користувач доводить, що він є особою, за яку себе видав на етапі ідентифікації. Легко заявити про особу. Наприклад, ви можете ввести в систему будь-яке ім’я користувача, яке вам подобається. Ось чому автентифікація є критично важливим кроком безпеки. Онлайн-автентифікація вимагає одного або кількох із наведеного нижче [12].

- Паролі/щось, що ви знаєте (Passwords/Something You Know): оригінальний метод автентифікації, ще до появи онлайн-світу, полягає в тому, щоб особа отримувала доступ, надавши пароль або щось, що знають як особа,

так і воротар, нібито таємне. На жаль, секрети можуть бути розкриті або здогадані (зі швидкістю тисячі за секунду). Іноді користувача також можуть обманом передати його зловмиснику.

- Щось у вас є (Something You Have): хоча паролі, можливо, були початковою формою автентифікації й досі залишаються найпоширенішою формою автентифікації, для боротьби з атаками було введено додаткові рівні автентифікації. Тепер багатофакторна автентифікація (MFA) є рекомендованим мінімумом для процесів автентифікації. Це означає надання двох або більше з того, що ви знаєте, того, що у вас є, або того, чим ви є.
- Щось, що ви є (Something You Are): цей фактор автентифікації, який також називається притаманністю, базується на тому, що вам властиво, що не можна скопіювати чи вкрасти. Це включає такі функції, як сканування обличчя, сканування сітківки ока, розпізнавання відбитків пальців або голосу. Оскільки сканери для цих біометричних функцій тепер включені в більшість стандартних персональних інтелектуальних пристроїв, стало легше включити цей надійний метод автентифікації в стандартні процеси.

Представлення необхідних факторів автентифікації є лише одним аспектом процесу автентифікації, частиною, яку бачить користувач. Те, що відбувається на сервері для перевірки факторів автентифікації, означає різницю між безпечним, стійким до фішингу процесом автентифікації та процесом, який можна зламати або обійти. Наприклад, якщо особу підтверджено шляхом зіставлення відбитків пальців або сканування обличчя користувача з тими, що зберігаються в базі даних, тоді їх можна перехопити за допомогою таких методів атаки, як « людина посередині» , що робить його не набагато безпечнішим, ніж пароль. Перевірка автентифікації повинна проводитися за допомогою безпечних протоколів криптографічного обміну відкритим ключем.

1.3.1. Методи автентифікації

Методи автентифікації розглянуті нижче (рис. 1.6.):

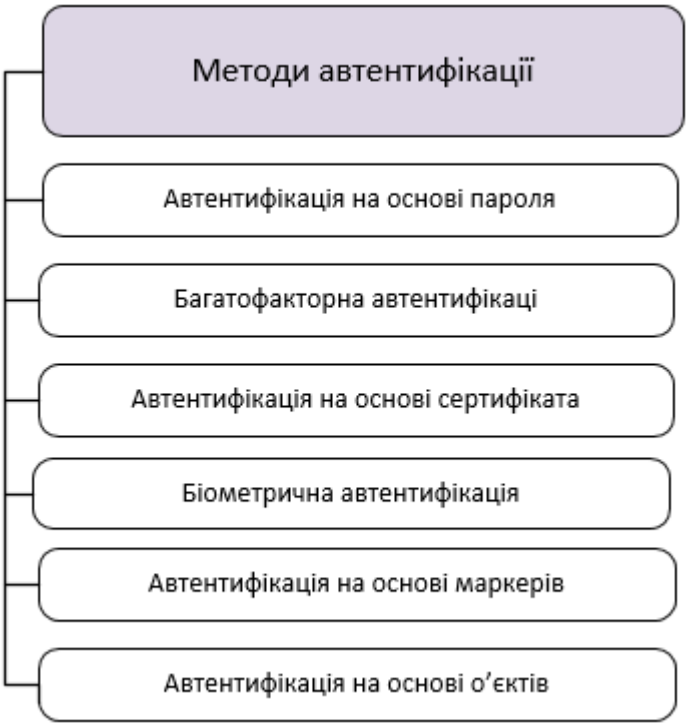


Рис. 1.6. Методи автентифікації

- Автентифікація на основі пароля (Password-based authentication). Паролі є найпоширенішим методом автентифікації. Оптимальний захист вимагає використання багатьох різноманітних паролів із різними рядками літер, цифр і символів. Однак багато людей використовують однакові або схожі паролі в облікових записах, що робить їх уразливими до фішингу та злому паролів. Зловмисники можуть легко обійти захист паролем, якщо вони мають доступ до облікового запису електронної пошти користувача або раніше використаного пароля. Коротше кажучи, одних лише паролів недостатньо для захисту облікового запису.
- Багатофакторна автентифікація (Multi-factor authentication). Більш безпечний метод автентифікації передбачає багатофакторну автентифікацію (MFA), яка вимагає використання кількох форм автентифікації, як-от запит Captcha або код безпеки, надісланий на вашу електронну пошту чи телефон у вигляді SMS-повідомлення. MFA мають свої недоліки, оскільки деякі користувачі можуть втратити доступ до попередньої електронної пошти чи

номера телефону, фактично блокуючи їх із свого облікового запису без втручання.

- Автентифікація на основі сертифіката (Certificate-based authentication). По суті, автентифікація на основі сертифіката (CBA) використовує цифровий сертифікат для ідентифікації користувача, пристрою чи машини перед наданням доступу до програми, мережі чи іншого ресурсу. Ця форма автентифікації більш безпечна, оскільки базується як на тому, що має користувач (цифровий сертифікат), так і на тому, що він знає (його пароль).
- Біометрична автентифікація (Biometric-based authentication). Автентифікація на основі біометричних даних покладається на унікальні біологічні характеристики людини для ідентифікації її особистості. Такі методи, як розпізнавання обличчя, сканування відбитків пальців або очей і розпізнавання голосу, забезпечують високий рівень безпеки з мінімальними збоями. У поєднанні з багатофакторною автентифікацією цей метод забезпечує додатковий рівень безпеки. Однак автентифікація на основі біометричних даних викликає занепокоєння щодо конфіденційності, а для деяких — етичні питання.
- Автентифікація на основі маркерів (Token-based authentication). Автентифікація на основі маркерів спрощує процес автентифікації для розпізнаних користувачів. Після введення імені користувача та пароля користувач може отримати доступ до захищених систем без повторного надання облікових даних.
- Автентифікація на основі об’єктів. Цей метод базується на фізичних об'єктах (ключах, картках, USB-пристроях) як засобах для підтвердження ідентичності користувача. Цей метод використовується для визначення доступу та автентифікації осіб до систем або приміщень за допомогою конкретних фізичних предметів.

1.4. Моніторинг

Моніторинг – процес безперервного збору інформації про стан параметрів об’єкту чи суб’єкту спостереження (моніторингу). Основною метою моніторингу є накопичення інформації для її подальшої обробки/аналізу та прийняття управлінського рішення, або постійне відслідковування стану об'єкту моніторингу без збереження попередньої інформації про об'єкт з метою своєчасного реагування (прийняття управлінського рішення) при певних кількісних або якісних змінах об'єкта.

Ось деякі важливі функції моніторингу в IT (рис. 1.7):

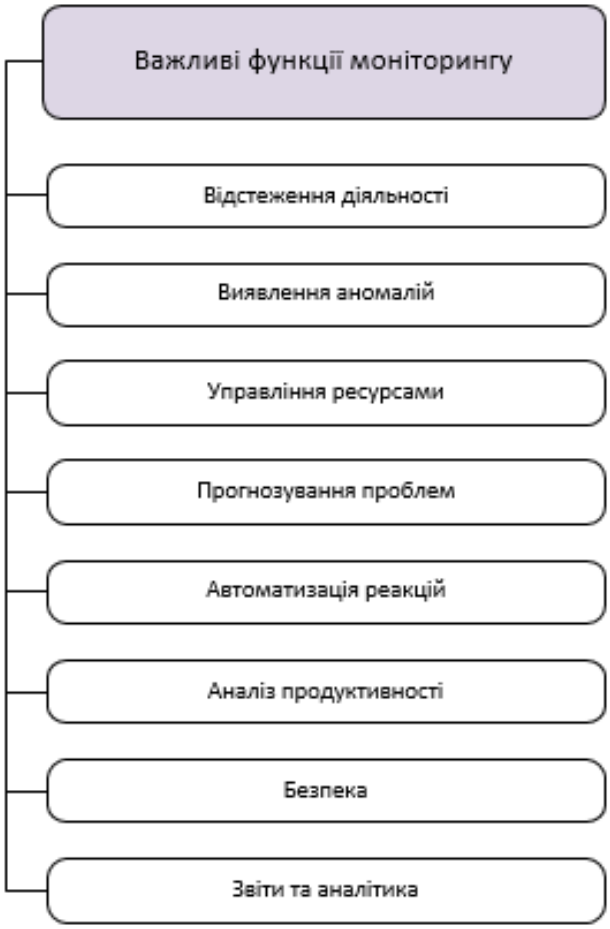


Рис. 1.7. Важливі функції моніторингу

Нижче наведено короткий опис функцій моніторингу:

- Відстеження діяльності: Моніторинг дозволяє відстежувати активність системи та мережі, включаючи ресурси, які використовуються, процеси, що виконуються, трафік мережі і використання обладнання.
- Виявлення аномалій: Системи моніторингу можуть виявляти аномальну активність або незвичайні події, які можуть вказувати на потенційні проблеми, включаючи загрози безпеці.
- Управління ресурсами: Моніторинг допомагає визначати, як ефективно використовуються ресурси, такі як потужність обчислювальних серверів, обсяги сховища даних і пропускна спроможність мережі.
- Прогнозування проблем: Аналіз даних моніторингу може допомогти прогнозувати проблеми, такі як вичерпання ресурсів або високий рівень навантаження, і вживати заходи для їх запобігання.
- Автоматизована реакція: В деяких випадках моніторинг може бути пов'язаним з автоматичними сценаріями реагування, які можуть виконувати дії на основі виявлених аномалій або стану системи.
- Аналіз продуктивності: Моніторинг допомагає вимірювати продуктивність системи і додатків, що дозволяє виявляти можливості для оптимізації.
- Безпека: Моніторинг також є важливим компонентом безпеки, оскільки він допомагає виявляти підозрілу активність та потенційні загрози для інформаційної безпеки.
- Звіти та аналітика: Дані, зібрані під час моніторингу, можуть бути використані для створення звітів, аналітики та прийняття рішень на основі даних.

Сучасні системи моніторингу можуть бути комплексними та автоматизованими, використовувати різні інструменти та технології, включаючи спеціалізовані програмні засоби, системи управління подіями та інші інструменти для забезпечення належного контролю і безпеки в інформаційних технологіях.

1.5. Різниця між ідентифікацією, автентифікацією та моніторингом

Ідентифікація, автентифікація і моніторинг – це три ключові аспекти управління ідентичністю та безпекою, і кожен з них виконує різні функції в процесі забезпечення безпеки інформації та доступу користувачів до систем та ресурсів. Давайте розглянемо їхню різницю:

Ідентифікація – це процес визначення, ким є особа, яка намагається отримати доступ до системи або ресурсу. Це визначення користувача на основі унікальної інформації, такої як ім'я користувача, адреса електронної пошти або інший ідентифікатор.

Автентифікація – це процес перевірки того, чи є особа, яка намагається отримати доступ, справжньою та авторизованою. Це вимагає введення вірного пароля, багатофакторної автентифікації (наприклад, введення пароля і використання мобільного підтвердження) або іншого методу перевірки.

Моніторинг – це процес слідкування за активністю користувачів та систем, щоб виявити підозрілу або незвичайну діяльність. Метою моніторингу є забезпечення безпеки системи, виявлення і реагування на потенційні загрози або інциденти безпеки.

Також узгоджена робота всіх трьох аспектів є важливою для забезпечення безпеки інформації та захисту доступу до систем. Ідентифікація і автентифікація встановлюють особу та її право на доступ, а моніторинг допомагає виявляти незвичайну активність і реагувати на потенційні загрози в реальному часі.

1.6. Висновки до розділу I

В першому розділі коротко описано про технологію SSO, ключові аспекти та властивості використання технології єдиного входу, а також наведено

переваги та недоліки. До переваг SSO віднесено: надійніші паролі, покращену політику застосування паролів, багатофакторну автентифікацію для більшого захисту, єдину точку для повторного введення паролю, внутрішнє керування обліковими даним замість зовнішнього сховища, менша затратність часу при відновлені втраченого паролю/паролів. Також проаналізовано загрози, які можуть бути пов’язані з використанням технології SSO та коротко описано принцип її роботи.

Розглянуто такі поняття як ідентифікація, автентифікація та моніторинг. Встановлено, що ідентифікація зазвичай поділяється на три види – парольну, апаратну та біометричну, також розглянуто типи ідентифікаторів за якими відбувається ідентифікація. Автентифікацію в свою чергу було описано за такими методами– автентифікація на основі пароля, багатофакторна автентифікація, автентифікації на основі сертифіката, біометрична автентифікація та автентифікація на основі маркерів. Також було описано основні аспекти моніторингу в інформаційних технологіях. Він допомагає визначити, як ефективно використовуються ресурси, що дозволяє оптимізувати їхнє використання.

Як результат подано різницю між ідентифікацією автентифікацією та моніторингом, вона полягає в тому, що ідентифікація та автентифікація відбувається один раз і за згоди користувача, моніторинг натомість відбувається постійно з певним проміжком часу та користувач не знає коли саме. Використання ідентифікації, автентифікації та моніторингу в сукупності дозволяє побудувати більш комплексну систему безпеки, яка захищає доступ до ресурсів та реагує на загрози.

В цілому використання SSO спрощує процес автентифікації для користувачів, забезпечуючи при цьому високий рівень безпеки. Це робить взаємодію з різними системами та ресурсами більш зручною та безпечною для користувачів. Також SSO дозволяє централізовано керувати ідентифікацією та доступом до ресурсів, що полегшує адміністрування та забезпечує послуги з безпеки для всіх користувачів.

З огляду на широке застосування біометричних даних у сучасному світі, які базовані на унікальних фізичних характеристиках особи, вони відкривають нові можливості в безпеці, впровадженні інновацій у сфері ідентифікації, доступу та автентифікації. Далі буде розглянуто аналіз зображень у контексті біометричних технологій надає можливість не лише розпізнавати особу, а й докладати зусиль для розвитку систем безпеки та індивідуального підходу до користувачів.

РОЗДІЛ ІІ. БІОМЕТРИЧНІ ТЕХНОЛОГІЇ РОЗПІЗНАВАННЯ. АНАЛІЗ ТА ОБРОБКА ЗОБРАЖЕНЬ

2.1. Біометричні технології розпізнавання

Біометрична ідентифікація – ідентифікація особи за властивими лише їй біологічними ознаками. Так як кожна людина має унікальні біометричн ознаки, властиві тільки їй, то є досить логічним рішення про їх використання в сфері інформаційної безпеки. Біометрична ідентифікая є складним та ефективним апаратом для ідентифікації людини, що на основі біометричних ознак дозволяє чітко ідентифіковувати людину.

Основна причина використання біометричних ідентифікаторів, полягає в тому, що кожна людина має біометричний ідентифікатор, який є унікальним для кожної людини. Переваги біометричної ідентифікації: по-перше, відмінно від інших методів, біометричні ідентифікатори неможливо загубити або забути; по-друге, біометричні ідентифікатори важко підробити, тому цей метод є найнадійнішим; по-третє, кожна людина має унікальні біометричні характеристики і тому точність біометричної ідентифікації наближається до 100 відсотків.

До недоліків можна віднести, по-перше, відносно великі інвестиції, необхідні для впровадження системи біометричної ідентифікації, і, по-друге, певні біометричні ідентифікатори можливо підробити, наприклад, відбитки пальців, голос чи підпис.

Однак, незважаючи на недоліки, ідентифікація за біометричним ознаками є найбільш надійними, а також її можливо застосувати у комплексній ідентифікацієї, тобто можливо використовувати одночасно з іншим методом ідентифікації (в більшості випадків додатково використовується ідентифікація за

допомогою пароля) чи проводити ідентифікацію за кількома біометричними ознаками одночасно, що в рази збільшує надійність ідентифікації [13].

2.1.1. Методи біометричної ідентифікації

На сьогодні існує багато методів біометричної автентифікації та ідентифікації, що можна поділити на дві групи: статичні та динамічні(рис. 2.1).

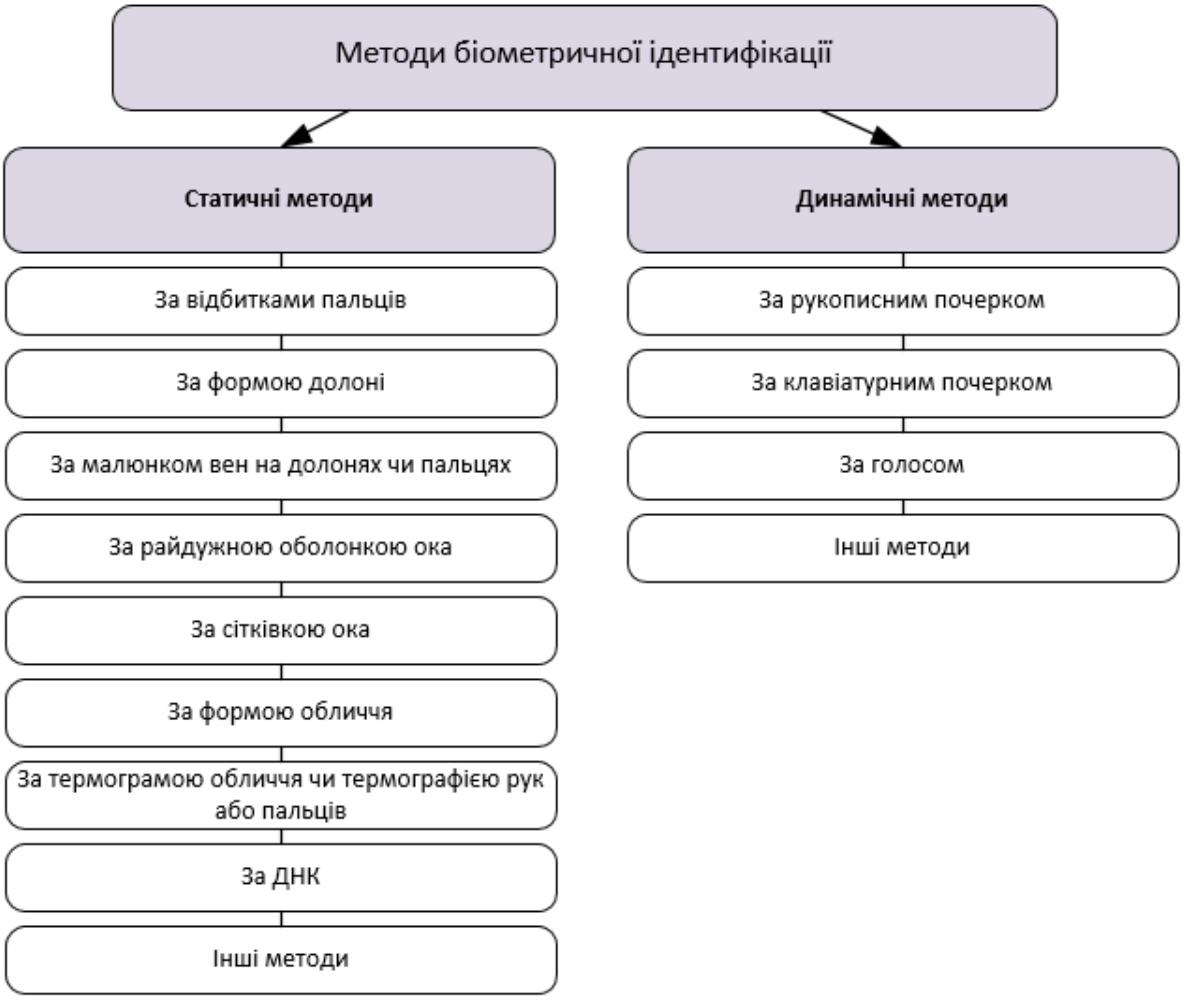


Рис. 2.1. Методи біометричної ідентифікації

Статичні методи біометричної автентифікації базуються на фізіологічних (статичних) характеристиках людини, тобто вроджених характеристиках, які даються людині при народженні, є невід'ємною частиною особистості і не змінюються з часом. Ці методи, базуються на розпізнаванні за:

- Відбитками пальців. Цей метод заснований на унікальності папілярного візерунка на пальцях кожної людини. Відбитки пальців, зібрані за допомогою спеціального сканера, перетворюються (згортаються) в цифровий

код і порівнюються з раніше отриманим і збереженим еталонним зразком. Апаратно-програмні технології, що використовують папілярні візерунки пальців, є найбільш поширеними порівняно з іншими методами біометричної автентифікації.

- Формою долоні (геометрією долоні, кисті чи пальця – використовується в досить вузькому сегменті). Цей метод базується на суто індивідуальній формі долоні, кисті або пальця. Використовуючи спеціальний пристрій, що складається з камери і кількох діодів підсвічування (які при послідовному включенні дають різні проекції предмета), створюється тривимірне зображення долоні (руки або пальця), яке використовується для формування згортки і розпізнавання осіб.
- Малюнками вен на долоні або пальці (ця технологія стає все більш популярною). Зчитування малюнка вен на долоні (руці) або передній стороні пальця відбувається за допомогою інфрачервоної камери, далі обробка отриманого зображення і створення відповідної цифрової згортки відповідно до малюнка вен.
- Райдужною оболонкою ока. Візерунок райдужної оболонки ока є унікальною особливістю людини, причому достатньо портативної камери для його сканування та спеціального програмного забезпечення, щоб відсканувати відповідну частину обличчя, виділити зображення ока, і відокремити з нього візерунок райдужної оболонки та сформувати відповідний цифровий ідентифікаційний код. Широке використання технології ідентифікації за райдужною оболонкою ока стримується патентними обмеженнями виробників та високою ціною необхідного обладнання.
- Ідентифікацією за сітківкою ока. Це метод ідентифікації, заснований на малюнку кровоносних очного дна. Щоб візуалізувати і зафіксувати цей візерунок, людині потрібно подивитися на віддалене джерело світла, пляму, а потім відсканувати освітлене очне дно за допомогою спеціальної камери. Наразі цей метод рідко використовується для ідентифікації з кількох причин.

- За формою обличчя. Цей метод ідентифікації дозволяє отримати дво- або тривимірне зображення обличчя людини. Він ідентифікує контури обличчя, такі як брови, очі, ніс і губи, і обчислює відстань між ними, формуючи не тільки зображення обличчя, але і ряд варіацій, таких як поворот і нахил обличчя і зміни виразу обличчя. Залежно від мети використання (автентифікація, зіставлення, віддалений пошук на великій відстані тощо), цей метод записує певну кількість зображень у базу даних. Цей статичний метод в даний час використовується в друкованих джерелах для ідентифікації форми і овалу обличчя (ці ідентифікатори використовуються в так званому 2-D методі, техніці розпізнавання двовимірних зображень обличчя з фотографій або відео) і форми і структури черепа (компанії, що працюють в цій галузі, розпізнають людей за допомогою тривимірних моделей обличчя, техніка, яка використовується в так званому 3-D методі), Слід зазначити, що існує два окремих методи (так званий 3-D метод вважається більш перспективним).
- Термограмами обличчя та термографії рук або пальців (ці технології, що базуються на ідентифікації, використовуються переважно в банківському секторі і поки що не набули широкого розповсюдження). Цей метод ідентифікації базується на унікальному розподілі артерій у кожній частині тіла людини, які постачають кров до вибраних ділянок шкіри і формують специфічний тепловий фон на ній. Термограму отримують за допомогою спеціальної інфрачервоної камери. Використовуючи цей метод можливо розрізнити навіть близнюків.
- ДНК. Переваги цього методу добре відомі, але оскільки сучасні методи отримання та обробки ДНК є дуже трудомісткими і тривалими (поки що немає можливості працювати в режимі реального часу), системи, що використовують цей метод, в основному застосовуються лише для спеціалізованих тестів.
- Використання інших методів Крім вищезгаданих методів, існують й інші унікальні методи автентифікації за підшкірним шаром, кількістю пальців, обраних для сканування, формою вух, запахом тіла та багатьма іншими

характеристиками. Однак головним недоліком цих менш поширених методів є те, що не створено автоматизованих систем або баз даних, які могли б масово розпізнавати осіб за цими ідентифікаторами. В основному це стосується автентифікації за запахом тіла та формою вух.

Динамічні методи автентифікації базуються на аналізі поведінкових характеристик людини, тобто характеристик, які є унікальними для кожної людини при виконанні будь-якої рухової активності. В основі лежать характеристики рухів, які виконуються несвідомо під час відтворення будь-якого руху. Динамічні методи зазвичай використовуються як допоміжні або додаткові, оскільки вони значно поступаються статичним методам за точністю та ефективністю.

Методи автентифікації в цій групі наступні

- Рукописний почерк. Як правило, цей вид автентифікації або персональної ідентифікації використовує підпис особи (іноді кодове слово). Цифрові ідентифікаційні коди генеруються залежно від потрібного ступеня захисту та наявності необхідного обладнання (наприклад, графічних планшетів, екранів кишенькових комп'ютерів).

Існує два типи ідентифікації почерку:

- Ідентифікація за ступенем конгруентності між двома зображеннями, тобто ідентифікація базується на ступені збігу між двома зображеннями.
- З метою ідентифікації формується цифрова згортка, яка містить інформацію про безпосередній часовий режим підпису, тобто часові характеристики пози підписувача та статистичні характеристики динаміки натискання на поверхню, на якому розміщено підпис.

- Клавіатурний почерк (динаміка набору тексту на клавіатурі). Цей метод в цілому схожий на попередній, але замість підпису використовують кодове слово (якщо для цього використовується особистий пароль користувача, така автентифікація називається двофакторною) і не вимагає спеціального обладнання, крім стандартної клавіатури, яка переробляється і дооснащується.

Основними характеристиками, що використовуються для формування згортки для ідентифікації, є динаміка набору кодових слів.

- За голосом. Цей метод є одним з найстаріших, але зараз його розвиток виходить на новий рівень, оскільки виникла необхідність в її більш широкому використанні. Існує багато способів генерації мовних ідентифікаційних кодів, але в принципі це різні комбінації частотних і статистичних властивостей мови.

- Використання інших методів Крім найбільш поширених динамічних методів, перерахованих вище, існують також унікальні методи, такі як ідентифікація за рухом губ, в процесі відтворення кодового слова, ідентифікація за ходою та ідентифікація за динамікою повороту ключа дверного замка [14].

Характеристики чотирьох найпоширеніших біометричних статичних методів ідентифікації представлена на таблиці 2.1.

Таблиця 2.1

Характеристики чотирьох найпоширеніших біометричних статичних методів ідентифікації

Система розпізнавання	Універсальність	Унікальність	Перманентність	Вимірність	Стійкість до навколишнього середовища	Стійкість до підробок	Соціальна прийнятність	Точність
За райдужною оболонкою	Д	В	В	П	Д	П	П	Д
За відбитком пальця	Д	Д	Д	П	П	П	П	Д
За обличчям (3D-зображення)	В	Д	Д	Д	Д	В	В	Д
За обличчям (2D-зображення	В	П	П	Д	П	П	В	П

У таблиці 2.1. наведено такі скорочення: В- відмінно, Д – добре, П – погано.

2.2. Ідентифікація за геометрією обличчя

Ідентифікація за геометрією обличчя має ряд переваг:

- Унікальність: кожне обличчя унікальне, і це дозволяє створювати індивідуальні шаблони для кожного користувача. Це важливо для точності ідентифікації.
- Невідтворюваність: геометрія обличчя складно відтворити, тому цей метод досить надійний. Він не піддатливий до шахрайства або обхідних методів.
- Біометрична точність: ідентифікація на основі геометрії обличчя має високий рівень точності порівняно з іншими біометричними методами.
- Використання в реальному часі: цей метод може бути швидким у реальному часі, що дозволяє швидку ідентифікацію користувачів.
- Зручність: ідентифікація за геометрією обличчя зазвичай не потребує додаткових пристроїв, оскільки багато сучасних пристроїв мають камери, які можна використовувати для цієї мети.

Враховуючи, ці переваги ідентифікація за геометрією обличчя може була вибрана для створення системи моніторингу користувачів у веб-застосунках. Далі розглянемо більш детально.

Ідентифікація за геометрією обличчя — це метод ідентифікації особи за його геометричними рисами: довжиною, шириною та пропорціями різних частин обличчя. Цей процес використовується в системах розпізнавання облич, де алгоритми аналізують геометричні особливості обличчя, такі як відстані між очима, розміри носа, ширина рота тощо.

Цей метод базується на використанні технологій комп'ютерного зору та штучного інтелекту для аналізу фотографій або відео та порівняння геометричних особливостей обличчя з базою даних. Він може бути використаний у багатьох сферах, таких як біометрична ідентифікація для

безпеки, розпізнавання облич у фотографіях або відео, автоматичний відбір облич у фотографіях для соціальних мереж тощо.

Переваги використання геометрії обличчя як біометричного ідентифікатора полягають у безконтактному зборі даних. Всупереч іншим технологіям біометричної ідентифікації, таким як відбитки пальців, райдужна оболонка ока або голос, система розпізнавання за рисами обличчя є пасивною і не потребує прямого контакту або специфічних дій з особою, що ідентифікується. Вона не потребує від громадян залишати відбитки пальців, дивитися в камеру або вимовляти слова. Для криміналістів важливо, що цей метод ідентифікації має широкий спектр джерел для отримання даних, таких як фотографії, відеоряди та дані відеоспостереження. Крім того, кількість можливих ідентифікаторів у цьому методі значно менша, ніж при ідентифікації за відбитками пальців – одне обличчя у кожної людини проти десяти відбитків пальців.

Під час ідентифікації біометрична система автоматично виокремлює та обробляє інформацію про найхарактерніші ділянки обличчя: контури носа, губ, брів, а також їх відстань одна від одної. Ці дані використовуються для створення цифрових моделей ідентифікаторів, що пізніше порівнюються між собою.

Технічна реалізація систем ідентифікації за обличчям є складнішою математично, ніж у випадку відбитків пальців, і вимагає дорожчого обладнання, такого як цифрова камера та спеціалізована електроніка. Проте у цьому методі є перевага: для зберігання даних одного ідентифікаційного шаблону (за характеристиками обличчя) потрібна мала кількість пам'яті. Обличчя може бути поділене на обмежену кількість ділянок, які є однаковими для всіх людей. Наприклад, для створення унікального шаблону конкретної людини потрібно обробити дані лише про 12–40 характерних ділянок обличчя.

Проте на практиці реалізація технології розпізнавання за обличчям не така проста. Ці технології надто чутливі до зовнішніх умов, таких як освітлення, положення голови тощо, і до змін у зовнішності особи, таких як зачіска, окуляри, макіяж. Це призводить до недостатньої точності розпізнавання та високого рівня

помилок, коли система помилково визначає одну людину як іншу, особливо при порівнянні з іншими біометричними технологіями, такими як відбитки пальців чи райдужна оболонка ока.

Також технології розпізнавання за рисами обличчя ефективні при порівнянні фотографій за умови високої якості знімків і відсутності спеціальних спроб змінити зовнішність [15].

2.3. Аналіз та обробка вхідного зображення

Обробка зображень — це метод перетворення зображення в цифрову форму та виконання над ним деяких операцій, щоб отримати покращене зображення або витягти з нього деяку корисну інформацію. Це тип розповсюдження сигналу, у якому входом є зображення, наприклад відеокадр чи фотографія, а виходом може бути зображення або характеристики, пов’язані з цим зображенням. Зазвичай система обробки зображень включає обробку зображень як двовимірних сигналів із застосуванням до них уже встановлених методів обробки сигналів. В основному вона включає три етапи: імпортування зображення за допомогою оптичного сканера або цифрової фотографії, аналіз і маніпулювання зображенням, що включає стиснення даних і покращення зображення та виявлення візерунків, невидимих для людського ока, як супутникові фотографії, а вихід є останнім етапом, на якому результат можна змінити зображення або звіт, який базується на аналізі зображення.

Комп’ютерне бачення (CV) – це комп’ютерне зображення, де програма не залучає людину до візуального циклу. Одними з головних тем в цій галузі комп’ютерного зору є аналіз зображень. Перший аналіз зображення передбачає перевірку даних зображення для полегшення вирішення проблеми зору. Другий аналіз включає дві інші теми, як виділення ознак, що є процесом отримання інформації про зображення вищого рівня, такої як інформація про форму чи

колір, а наступним є класифікація шаблонів, яка є актом отримання інформації вищого рівня та ідентифікації об’єктів на зображенні [16].

2.3.1. Процеси попередньої обробки зображень

Процеси попередньої обробки зображень в контексті аналізу обличчя включають кілька ключових етапів, що допомагають покращити якість зображень та підготувати їх для подальшого аналізу (рис. 2.2):

1. Очищення зображень – це процес видалення шуму та непотрібних елементів для покращення якості зображення. Це може включати:

- Фільтрацію шуму: Використання фільтрів для усунення артефактів та випадкових плям на зображенні, наприклад, застосування фільтрів Гауса або медіанних фільтрів.
- Корекцію освітлення: Виправлення яскравості та контрастності зображення для поліпшення видимості деталей.

2. Розділення на різні частини. Цей етап передбачає розділення обличчя на окремі частини або компоненти, що сприяє подальшому аналізу кожної частини окремо. Це може включати:

- Визначення ключових точок: Локалізацію ключових областей обличчя, таких як очі, ніс, губи, за допомогою алгоритмів визначення областей інтересу.
- Сегментацію областей: Розділення зображення на області з застосуванням методів, що використовують контури та різницю у текстурі.

3. Підготовка для подальшого аналізу. Цей етап включає підготовку даних для подальшого використання у моделях розпізнавання чи аналізу обличчя. Це може включати:

- Нормалізацію даних: Приведення розмірів частин обличчя до стандартних розмірів для однорідності у подальшому аналізі.
- Виділення ознак: Визначення характеристик або ознак (наприклад, текстурних, геометричних) для подальшої обробки або використання у моделях машинного навчання.

Ці процеси попередньої обробки зображень є важливими для ефективного аналізу обличчя, оскільки вони допомагають усунути шум, виділити ключові області та підготувати дані для подальшого використання в алгоритмах розпізнавання та класифікації облич [17].

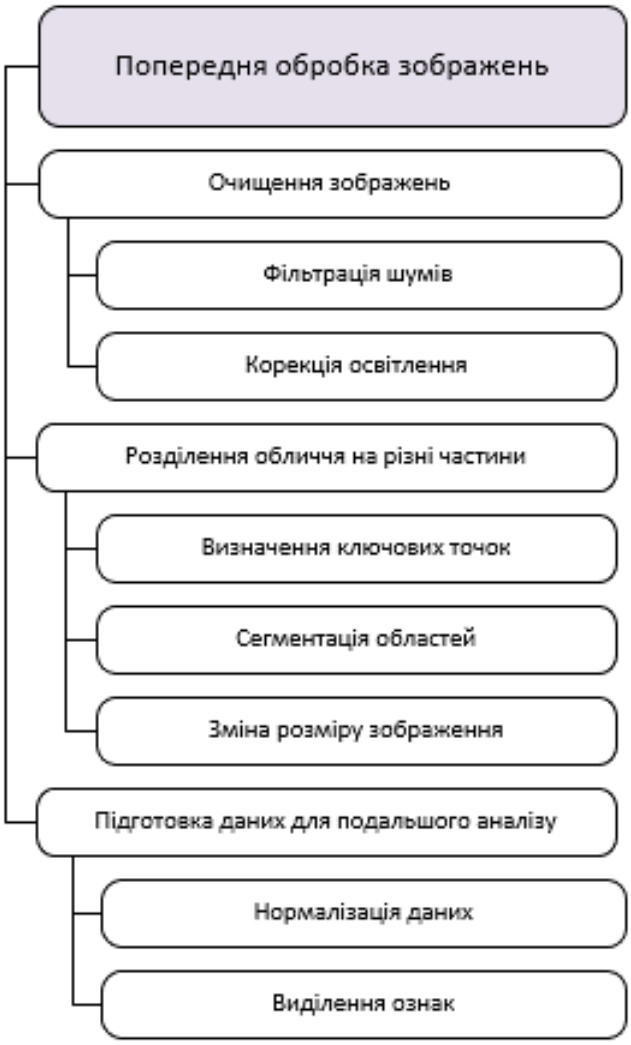


Рис.2.2. Попередня обробка зображень

2.3.2. Виділення основних рис обличчя

Визначення контурів, розпізнавання ключових точок (наприклад, очей, носа, губ) є ключовим етапом для подальшого аналізу та розпізнавання особи. Це включає кілька етапів:

1. Визначення контурів обличчя:

- Детектування контурів: Використання алгоритмів комп'ютерного зору для виявлення зовнішніх меж обличчя.

- Сегментація областей: Розділення зображення на області, що відповідають різним частинам обличчя.

2. Розпізнавання ключових точок обличчя:

- Локалізація ключових точок: Визначення та відстеження основних точок, таких як очі, ніс, губи тощо.
- Дескриптори ключових точок: Створення описів або характеристик для кожної ключової точки для подальшого порівняння.

3. Використання методів машинного навчання:

- Тренування моделей на основі ключових рис обличчя: Використання навчальних даних для розпізнавання та класифікації облич у зображеннях.

4. Використання геометричних та текстурних характеристик:

- Визначення геометричних параметрів: Вимірювання відстаней, кутів та інших геометричних характеристик обличчя.
- Аналіз текстурних особливостей: Використання текстурних шаблонів для розпізнавання особливостей обличчя.

На рис 2.3 показано визначення основних рис обличчя.

Ці етапи дозволяють точно виділити основні риси обличчя, такі як контури та ключові точки, що є важливими для подальшого аналізу та ідентифікації особи. Отримані дані про основні риси обличчя можуть бути використані для створення шаблонів та моделей для розпізнавання.

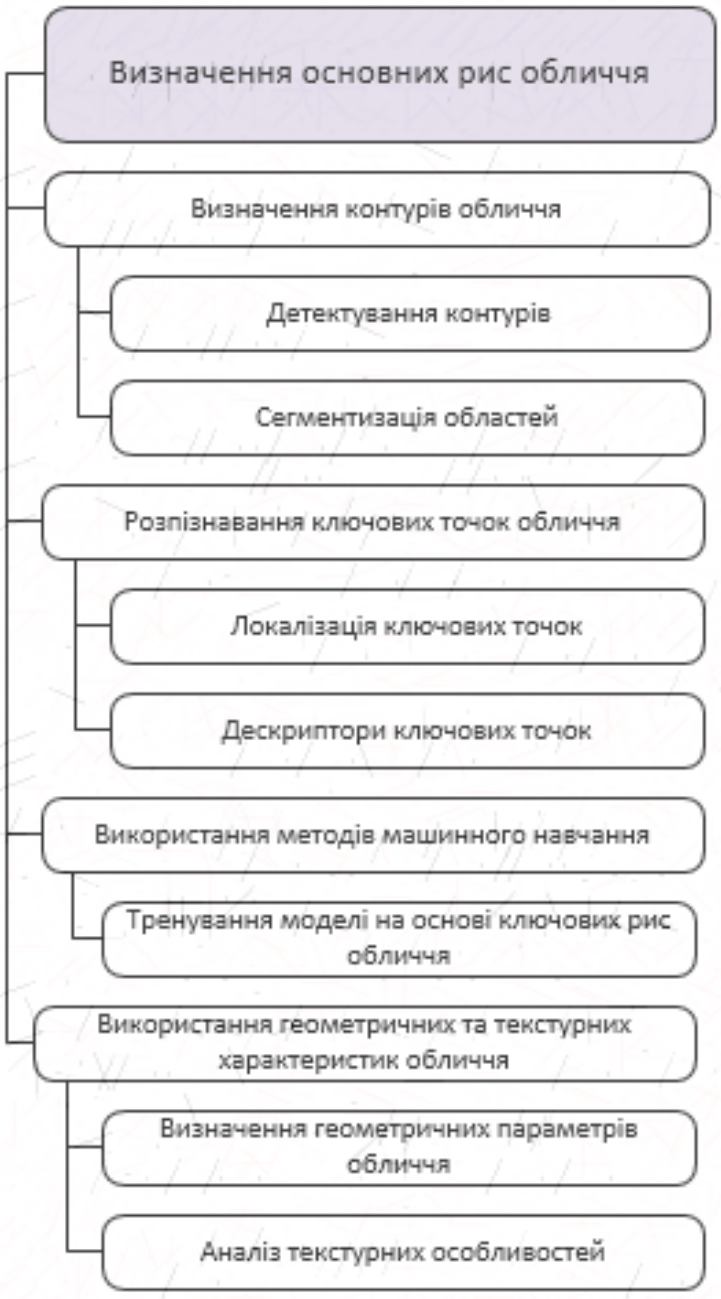


Рис.2.3. Визначення основних рис обличчя.

2.3.3. Фільтрація та покращення якості зображень

Фільтрація та покращення якості зображень грають ключову роль у підготовці образів для подальшого аналізу та обробки. Цей процес включає такі етапи:

1. Видалення шуму:

- Фільтри для підвищення якості: Застосування різних видів фільтрів, таких як медіанний, Гауссів, або фільтри, що базуються на хвильових перетвореннях, для зменшення або видалення шуму на зображенні.
- Адаптивні методи фільтрації: Використання алгоритмів, що адаптуються до рівня шуму для ефективного його усунення.

2. Виправлення освітлення:

- Корекція яскравості та контрастності: Застосування алгоритмів для виправлення яскравості та контрастності зображення, зокрема, для відновлення деталей у темних або надто освітлених областях.
- Нормалізація освітлення: Вирівнювання рівнів освітлення по всьому зображенню.

3. Усунення спотворень:

- Корекція геометрії та перспективи: Виправлення спотворень, що виникають внаслідок спотворення геометрії об'єктів на зображенні.
- Усунення дефектів та артефактів: Виявлення та виправлення дефектів, таких як плями, дрібні дефекти або артефакти зображення.

4. Використання алгоритмів машинного навчання:

- Удосконалення за допомогою нейронних мереж: Використання навчених моделей для автоматичного виправлення шуму, освітлення та інших артефактів на зображенні.

Ці етапи допомагають очистити та покращити якість зображень перед подальшим аналізом, забезпечуючи кращу точність та надійність алгоритмів розпізнавання та обробки зображень [16].

На рис. 2.4 схематично показано розподіл етапів фільтрації та покращення зображення.

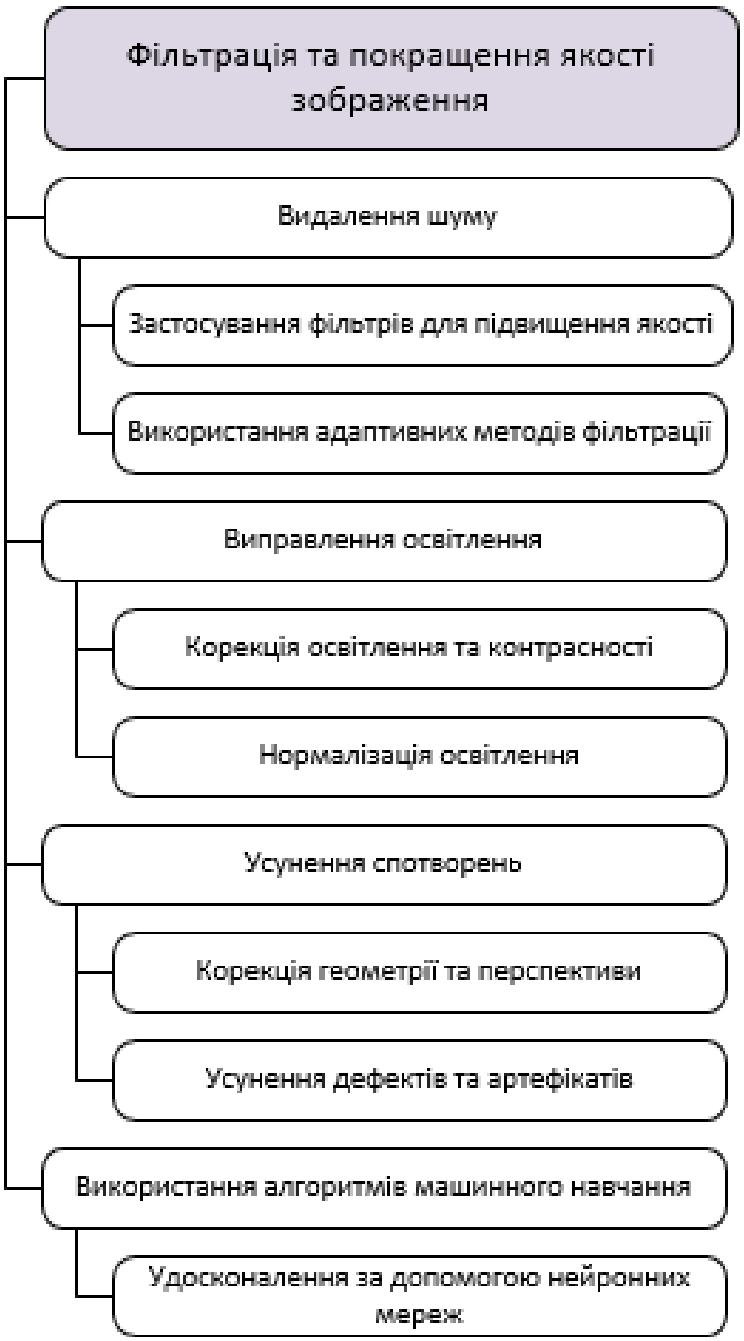


Рис.2.4. Фільтрація та покращення якості зображення.

2.4. Методи ідентифікації за геометрією обличчя

Ідентифікація за геометричними характеристиками включає в себе використання унікальних геометричних параметрів для розпізнавання та

ідентифікації об'єктів, зокрема, облич. Далі розглянуто кілька методів, які можуть бути використані для цієї цілі.

2.4.1. Аналіз геометричних характеристик особи

Аналіз геометричних характеристик особи базується на виділенні ключових точок чи областей обличчя та на визначенні ознак, які можуть бути різними відстанями між цими точками або відношеннями цих відстаней. Ці ознаки служать унікальними параметрами для подальшого порівняння.

Виділені ключові точки можуть представляти собою різні частини обличчя, такі як куточки очей, губи, носа або центр очей. Області, що включають у себе деякі частини обличчя, також можуть використовуватися як ключові зони для аналізу, наприклад, прямокутні області, що охоплюють очі, ніс та рот.

Ключові точки можуть включати куточки очей, губ, кінчик носа, центр очей тощо, тоді як ключові області можуть охоплювати прямокутні сегменти, що включають очі, ніс, рот тощо (рис.2.5).

У процесі розпізнавання порівнюються ознаки, визначені для невідомої особи, з ознаками, що зберігаються у базі даних.

Процес виявлення ключових точок є критично важливим для точності розпізнавання, оскільки правильність визначення цих точок на зображенні визначає успіх подальшого розпізнавання.

Отже, для успішного визначення ключових точок на зображенні обличчя особи, важливо, щоб обличчя було без перешкод, які можуть утруднити цей процес. Завади такого типу включають в себе наявність окулярів, борід, прикрас, елементів зачіски та макіяжу. Бажано мати однакове й рівномірне освітлення на всіх зображеннях, щоб покращити якість визначення контурів обличчя.

Також, оптимальною буде фотографія обличчя, зроблена з фронтального ракурсу, можливо з допустимими малими відхиленнями. Нейтральний вираз обличчя сприяє успішній роботі більшості методів, оскільки вони не передбачають моделей для урахування таких змін [13].

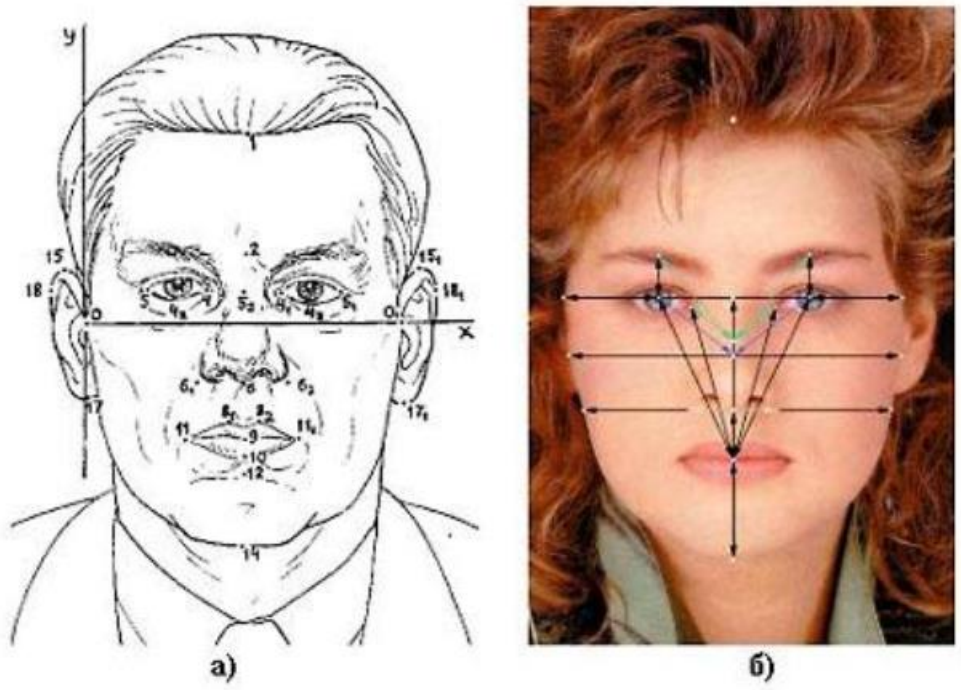


Рис 2.5. – Ідентифікаційні точки та відстані: а) використовувані при криміналістичній фотоекспертизі; б) найбільш часто використовувані при побудові автоматизованих систем ідентифікації

2.4.2. Порівняння еталонів

Метод порівняння еталонів (Template Matching) ґрунтується на виділенні областей особи на зображенні та подальшому порівнянні цих областей на двох різних зображеннях. Кожна область, яка збігається, збільшує ступінь схожості між зображеннями. Цей метод є одним з найперших у використанні для розпізнавання осіб за зображеннями обличчя. Для порівняння таких областей використовують прості алгоритми, наприклад, попіксельне порівняння.

Недолік цього підходу полягає у великому обсязі ресурсів, необхідних для зберігання та порівняння таких ділянок. Також, через використання простого алгоритму порівняння, зображення мають бути отримані в строгих умовах: без суттєвих змін у ракурсі, освітленні, виразі обличчя та інших факторах [18].

Приклад виконання методу порівняння зображень показано на рис. 2.6.

Рис .2.6. Приклад виконання методу порівняння зображень

2.4.3. Гнучкі контурні моделі особи

У цих методах розпізнавання базується на порівнянні контурів обличчя. Ці контури, як правило, охоплюють лінії голови, вух, носа, губ, брів і очей. Контур репрезентується ключовими позиціями, між якими розташовані точки на контурі, визначені за допомогою інтерполяції. Для локалізації контурів у застосованих методах використовується інформація, отримана як з апріорних даних, так і з аналізу тренувальних даних.

Спочатку ключові точки вручну визначалися на наборі тренувальних зображень. Пізніше отримувалася інформація про інтенсивність пікселів, що лежать на перпендикулярній лінії контуру для кожної ключової точки. Під час пошуку контурів нового обличчя застосовувався метод симуляції, який складався з двох компонент. Перша максимізувала відповідність інтенсивності пікселів, отриманих з тренувальних зображень, інтенсивностям пікселів перпендикулярної контуру. Друга компонента спрямовувалася на відповідність форми контуру зразкам тренувальних прикладів. Отже, вилучали не просто контур, а контур рис обличчя. Для порівняння зображень використовували значення головних компонент, отриманих з координат ключових точок. У цій роботі контурна модель використовувалася разом з напівтоновою моделлю, і їх спільне застосування підвищувало точність розпізнавання.

Головним завданням у розпізнаванні за контурами є правильне виявлення цих контурів. Це завдання за складністю порівнюється з розпізнаванням зображень. Крім того, саме застосування цього методу для завдання розпізнавання може бути недостатнім [14].

2.4.4. Лінійний дискримінантний аналіз

Лінійний дискримінантний аналіз (Linear Discriminant Analysis, LDA), також відомий як лінійний дискримінант Фішера, здійснює проекцію простору зображень на простір ознак таким чином, щоб мінімізувати внутрішньокласові відстані та максимізувати міжкласові відстані в цьому просторі. Ці методи передбачають, що класи розділяються лінійно.

Коли проектування на власний простір змішує класи, розпізнавання стає неможливим, тоді як лінійний дискримінантний аналіз вибирає проекцію на простір ознак так, щоб чітко розділити різні класи. Матриця W для проектування простору зображень на простір ознак вибирається з наступної умови:

$$W_{opt} = arg \max_{w} \left| \frac{W^T S_B W}{W^T S_W W} \right| \qquad (2.1)$$

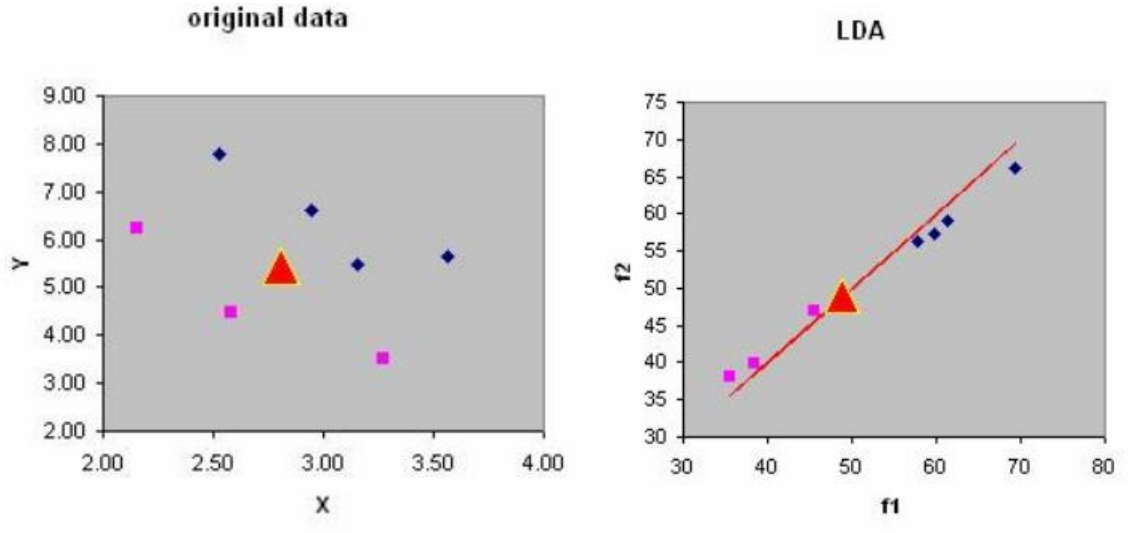
Де S_B – матриця міжкласової дисперсії, та S_W – матриця внутрішньокласової дисперсії

У просторі ознак може бути наявно до (c-1) векторів, що утворюють базис, де c – загальна кількість класів. Ці вектори використовуються для перетворення простору зображень у простір ознак. Оскільки робота з матрицею $S_W \in \Re^{n \times n}$ складною через її великі розміри, спочатку використовується зменшення розмірності за допомогою методу головних компонентів, і подальші обчислення проводяться в меншому розмірному просторі:

$$W_{fid} = arg \max_{w} \left| \frac{W^T W_{pca}^T S_B W_{pca} W}{W^T W_{pca}^T S_W W_{pca}^T W} \right| \qquad (2.2)$$

Де W_{pca} – матриця для перетворення до простору меншої розмірності (простір головних компонент). В даній роботі цей метод отримав назву "Fisherfaces". Так само, як і власні вектори, зображення базисних дискримінантних векторів мають [14].

Приклад застосування алгоритму лінійного дискримінантного аналізу показано на рис.2.7.



а) б)

Рис. 2.7. Приклад застосування алгоритму лінійного дискримінантного аналізу: а) вихідні дані, б) після застосування LDA

2.4.5. Метод Віоли-Джонса

Метод Віоли-Джонса – це популярний алгоритм для виявлення облич у зображеннях. Він базується на каскаді класифікаторів, що використовуються для швидкого та ефективного виявлення облич у реальному часі. Цей метод використовує ознаки, які називаються "адаптивні признаки зображення", такі як інтенсивність, текстура, форма тощо, для розпізнавання облич.

Необхідно відзначити, що цей детектор володіє вкрай низькою ймовірністю помилкового виявлення осіб. Метод успішно визначає риси обличчя навіть при спостереженні об'єкта під невеликим кутом, приблизно 30°. Однак при нахилі понад 30° можливість виявлення обличчя раптово зменшується. Ця особливість методу у стандартній реалізації ускладнює або навіть унеможливлює виявлення обличчя людини, що повернуте під довільним кутом, що значно ускладнює використання алгоритму в сучасних виробничих системах, з урахуванням їх різноманітних потреб.

Для визначення яскравості прямокутної ділянки зображення використовується інтегральне подання. Цей підхід широко використовується і в

інших методах, таких як вейвлет-перетворення, Speeded up robust feature (SURF), фільтри Хаара та інші алгоритми. Використання інтегрального представлення дозволяє ефективно обчислювати сумарну яскравість довільного прямокутника на зображенні, причому час розрахунку не залежить від площі прямокутника.

Інтегральне представлення зображення – це матриця, що збігається за розмірами з вихідним зображенням. Кожен елемент матриці містить суму інтенсивностей всіх пікселів, які розташовані лівіше і вище даного елемента. Елементи матриці обчислюються за формулою.

$$L(x_1 y) = \sum_{i=0,j=0}^{i\leq x, j\leq y} I(i;j) \tag{2.3}$$

де $I(x, y)$ представляє значення точки (x, y) у вигляді інтегрального зображення; а $i(x, y)$ – це значення інтенсивності вихідного зображення в точці (x, y). На основі інтегрального подання зображення, обчислення ознак з однаковою структурою, але з різними геометричними параметрами, відбувається за однаковий час.

Кожен піксель матриці I(x, y) є сумою всіх пікселів у прямокутній області від початку координат до точки (x, y) у вихідному зображенні. Іншими словами, значення кожного пікселя I(x, y) є результатом сумування значень усіх пікселів, розташованих лівіше і вище даного пікселя i(x, y). Обчислення цієї матриці вимагає часу, що залежить лінійно від кількості пікселів у зображенні і може бути проведене за такою формулою:

$$I(x,y) = i(x,y) - I(x-i, y-1) + I(x, y-1) + I(x, y-1) \tag{2.4}$$

Інтегральне подання має цікаву властивість – дозволяє швидко обчислити суму пікселів будь-якого прямокутника за інтегральною матрицею.

Хаар-подібні характеристики. Щодо необхідності використання простих алгоритмів отримання ознак, ефективним варіантом є використання Хаар-подібних характеристик, які визначаються через порівняння яскравостей двох прямокутних областей на зображенні.

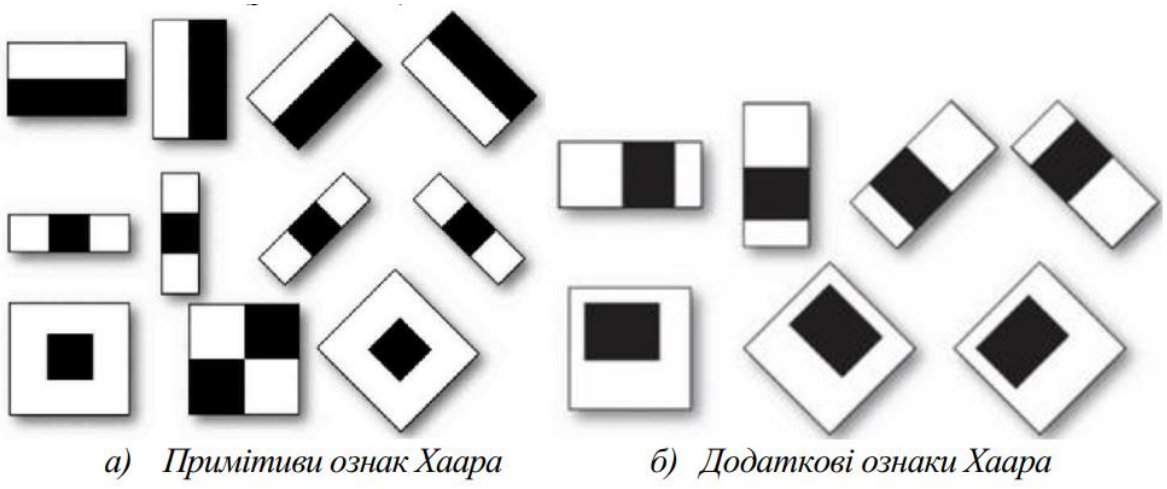
У стандартному методі Віоли-Джонса використовуються прямокутні ознаки (див. рис. 2.8, а), вони відомі як Хаар-примітиви. У розширеному методі

Віоли-Джонса, що представлений у бібліотеці OpenCV, використовуються додаткові типи ознак (див. рис. 2.8, б).

Значення цієї ознаки розраховується як:

$$F = U - V \tag{2.5}$$

Де U –сума яскравостей точок, що охоплюються світлою частиною ознаки, а V – сума яскравостей точок, які закриваються темною частиною ознаки. Для їх розрахунку використовується поняття інтегрального зображення. Хаар-подібні ознаки визначають величину зміни яскравості по осях X та Y на зображенні відповідно [19].



а) Примітиви ознак Хаара *б) Додаткові ознаки Хаара*

Рисунок 2.8. Примітиви Хаара

2.4.6. Метод гістограми напрямлених градієнтів

Гістограма напрямлених градієнтів (HOG) – це метод для виявлення об'єктів на зображенні шляхом оцінювання та подальшого використання градієнтів яскравості в різних частинах зображення. Зазвичай зображення розділяється на менші області, після чого для кожної області обчислюються градієнти (напрям та сила зміни яскравості).

Гістограма напрямлених градієнтів дозволяє оцінити частоту та напрямок зміни яскравості пікселів у кожній області зображення, і подальше складання цих інформаційних блоків для побудови опису зображення в цілому.

2.5. Алгоритм втілення методу гістограми напрямлених градієнтів

Для реалізації системи розпізнавання облич було обрано метод гістограми напрямлених градієнтів, через його здатність точно та ефективно визначати обличчя на зображеннях у реальному часі з високою стійкістю до різноманітних умов. Тому далі детальніше розглянемо його алгоритм роботи.

Реалізація методу гістограми напрямлених градієнтів включає в себе обчислення градієнтів, розподіл їх за гістограмами та налаштування параметрів, таких як розмір клітин, розмір блоків, кількість орієнтацій градієнтів та інші, для досягнення найкращих результатів розпізнавання об'єктів на зображеннях.

Втілення алгоритму відбувається за наступними кроками: обчислення градієнта, групування спрямувань, блоки описувача, унормовування блоків та розпізнавання об'єктів

2.5.1. Обчислення градієнта

В багатьох виявниках ознак першим етапом попередньої обробки зображення є нормалізація кольору та яскравості. Однак в описувачі HOG Далал і Тріггс вказують, що цей етап можна пропустити, оскільки наступна нормалізація описувача забезпечує схожі результати. Тому попередня обробка майже не впливає на ефективність. Замість цього першим кроком обчислення є визначення градієнтів. Зазвичай використовується одновимірне центроване ядро дискретної похідної у вертикальному або горизонтальному напрямку, таке як:

$[-1,0,1]$ та $[-1,0,1]^T$

Далал і Тріггс експериментували з іншими, більш складними ядрами, такими як Собель 3x3 або діагональні, проте вони показали гірші результати у виявленні людей на зображеннях. Також вони тестували гаусівське згладжування перед застосуванням ядра похідної, але виявили, що в практиці відсутність згладжування працює краще [20].

2.5.2. Групування спрямувань

Другим етапом обчислення є створення гістограм комірок. Кожен піксель у цих комірках вносить свої вагомість в залежності від напрямку градієнта, який був визначений під час розрахунку. Самі комірки можуть мати різні форми: прямокутні або радіальні, а канали гістограми розподілені рівномірно від 0 до 180 або 0 до 360 градусів, залежно від того, чи використовується "беззнаковий" або "знаковий" градієнт. Дослідники Далал і Тріггс показали, що у поєднанні з 9 каналами гістограми "беззнакові" градієнти приводять до найкращих результатів у виявленні людей, але знакові градієнти покращують розпізнавання інших об'єктів, наприклад, автомобілів чи мотоциклів. Щодо вагомості пікселів, вона може бути визначена або за значенням градієнта самого пікселя, або за певною функцією цього значення. В експериментах зазвичай кращі результати демонструє використання самого значення градієнта, хоча існують інші варіанти, такі як квадратний корінь або квадрат значення градієнта, або його обмежена версія [20].

2.5.3. Блоки описувача

Для врахування впливу змін в освітленні та контрасті потрібно локально нормалізувати виразність градієнту, що вимагає об'єднання комірок у більші, просторово зв'язані блоки. Таким чином, описувач HOG складається з об'єднаного вектора нормалізованих коміркових гістограм з усіх областей блоку. Ці блоки часто перекриваються, тобто кожна комірка сприяє утворенню кінцевого описувача більше одного разу. Блоки R-HOG часто мають квадратну форму, визначені трема параметрами: кількістю комірок у блоку, кількістю пікселів на комірку та кількістю каналів у гістограмі комірки. У дослідженні Далала та Тріггса оптимальними параметрами були чотири комірки 8x8 пікселів у блоку (16x16 пікселів у блоку) з 9 гістограмними каналами. Крім цього, вони встановили, що використання гаусового згладжування для зменшення ваги пікселів у краях блоків призводило до незначного покращення продуктивності.

Кругові блоки HOG (C-HOG) існують у двох форматах: з однією центральною коміркою та з центральною коміркою, розділеною на кутові

сектори. Дослідники виявили, що обидва варіанти мали однакову ефективність і що конфігурація з двома радіусними секторами і чотирма кутовими секторами, центральним радіусом 4 пікселі та коефіцієнтом розширення 2, показує найкращі результати у їх експериментах. Гаусівське згладжування виявилося неефективним при використанні з блоками C-HOG. Блоки C-HOG виглядають схожими на описи форм, але сильно відрізняються тим, що містять комірки із кількома каналами напряму, тоді як описи форм використовують лише один розрахунок присутності контуру [20].

2.5.4. Унормовування блоків

Далал та Тріггс перевірили чотири різні методи нормалізації блоків. Нехай v — це ненормований вектор, що включає всі гістограми у даному блоку, $\|v\|k$ — його k-норма для k = 1, 2, а e — деяка маленька константа (точне значення, ймовірно, не має великого значення). Отже, коефіцієнт нормалізації може бути одним із таких:

1. L2-норма: $f = \frac{v}{\sqrt{\|v\|_2^2+e^2}}$
2. L2-гіс: L2-норма з обмеженням максимальних значень v до 0.2 та подальшим перенормуванням.
3. L1-норма: $f = \frac{v}{(\|v\|_1+e)}$
4. L1-корінь: $f = \sqrt{\frac{v}{\|v\|_1+e}}$

У своїх експериментах вони виявили, що L2-гіс, L2-норма та L1-корінь мають схожу ефективність, тоді як L1-норма була трохи менш ефективною; проте усі чотири методи показали значне покращення порівняно з ненормованими даними.

2.5.5. Розпізнавання об'єктів

Описувачі HOG можуть служити ознаками для ідентифікації об'єктів, використовуючи їх як вхідні параметри для алгоритмів машинного навчання. Далал та Тріггс використовували описувачі HOG як функціональні особливості у методі опорних векторів (SVM), проте описувачі HOG не є прив'язаними до конкретного алгоритму машинного навчання [20].

2.6. Аналіз існуючих систем ідентифікації користувачів за геометрією обличчя

NEC NeoFace – це система ідентифікації за обличчям, розроблена компанією NEC Corporation. NeoFace використовує передові алгоритми обробки зображень для виявлення та аналізу особливостей обличчя. Вона враховує різноманітні аспекти, такі як форма обличчя, розміри, текстура шкіри, різноманітність в освітленні. Ця система може використовуватися у різних сферах, таких як безпека, контроль доступу, відправлення та отримання платежів, а також управління людськими ресурсами. NeoFace забезпечує високий рівень захисту особистих даних користувачів, оскільки вона використовує передові алгоритми шифрування та безпеки для зберігання та обробки даних.

Google Face Unlock – це технологія розблокування пристроїв, яка використовує розпізнавання обличчя для доступу до смартфонів чи інших пристроїв, що працюють на операційній системі Android. Ця технологія базується на використанні фронтальної камери пристрою для сканування та розпізнавання особливостей обличчя користувача.

Google Face Unlock працює шляхом створення біометричного шаблону обличчя, який зберігається у безпечному режимі на пристрої. Коли користувач хоче розблокувати пристрій, він просто спрямовує своє обличчя на фронтальну камеру, і система спробує впізнати його за цими особливостями. Якщо розпізнання вдале, пристрій розблоковується.

Apple Face ID – це біометрична технологія, яка використовується у смартфонах iPhone для ідентифікації користувача за його обличчям. Face ID від Apple – це технологія розпізнавання обличчя, розроблена для забезпечення безпечного та зручного способу розблокування пристроїв, авторизації платежів

та доступу до конфіденційної інформації. Вона використовує спеціальну камеру TrueDepth, яка сканує обличчя користувача, створюючи точну тривимірну модель. Ця модель порівнюється з збереженим шаблоном обличчя, який зберігається у зашифрованому вигляді на пристрої. Face ID має високу точність та безпеку завдяки використанню інфрачервоного світла та проєкції точок на обличчі для створення унікального шаблону.

DeepFace від Facebook – це система розпізнавання обличчя, розроблена для точного визначення осіб на зображеннях. Ця технологія використовує нейронні мережі для аналізу та порівняння облич з великою точністю.

DeepFace працює, створюючи векторні представлення облич, які потім порівнюються між собою для визначення, чи на зображеннях зображені одні й ті ж люди. Система навчається розпізнавати особи на зображеннях, але вона вимагає великої кількості даних для точності розпізнавання [21].

В таблиці 2.2 представлено аналіз характеристик систем ідентифікації

Таблиця 2.2

Основні характеристики систем ідентифікації за геометрією обличчя

Характеристики	**NEC NeoFace**	**Google Face Unlock**	**Apple Face ID**	**DeepFace (Facebook)**
Технологія зчитування зразка	Використання шаблонів облич	Використання фронтальної камери	Технологія TrueDepth для сканування облич	Зображення надає користувач
Спосіб розпізнавання	Використання шаблонів для порівняння облич	Використання алгоритмів машинного навчання	Тривимірне сканування облич	Використання нейронних мереж для створення векторних представлень облич для порівняння
Розблокування пристроїв	Так	Так	Так	Ні
Функція	Ідентифікація	Ідентифікація	Ідентифікація	Ідентифікація
Точність розпізнавання	Висока	Залежить від умов освітлення та якості камери	Висока	Висока
Використання нейронних мереж	Так	Ні	Так	Так

2.7. Висновки до розділу II

Застосування методів аналізу геометричних характеристик, порівняння еталонів, гнучких контурних моделей, лінійного дискримінантного аналізу, методу Віоли-Джонса та методу гістограми напрямлених градієнтів дозволяє отримувати точні та надійні результати у розпізнаванні об'єктів на зображеннях.

Процес обробки зображень для витягнення біометричних ознак включає у себе використання різноманітних алгоритмів, технік фільтрації, виявлення ознак, та класифікації, спрямованих на забезпечення точності ідентифікації та забезпечення невід'ємної безпеки систем.

Незважаючи на успішність біометричних технологій, продовжуються дослідження з метою поліпшення точності, швидкості та універсальності цих систем. Однак важливо також звертати увагу на питання конфіденційності та захисту персональних даних, щоб забезпечити баланс між високим рівнем безпеки та збереженням приватності користувачів.

Також у цьому розділу було розглянуто різновиди біометричних технологій ідентифікації, а також більше приділено уваги методам біометричної ідентифікації за геометрією обличчя.

Аналіз геометричних характеристик особи: Використовує геометричні параметри (такі як розмір, форма, пропорції) для аналізу особливостей обличчя або тіла.

Порівняння еталонів: Базується на порівнянні відомих "еталонів" (зразків) з особливостями, знайденими у зображеннях, для ідентифікації чи класифікації об'єктів.

Гнучкі контурні моделі особи: Використовують гнучкі моделі для відображення форми обличчя або областей тіла для розпізнавання.

Лінійний дискримінантний аналіз (LDA): Метод вибору проекції простору зображень для максимізації міжкласового розбіження і мінімізації внутрішньокласового розбіження у просторі ознак.

Метод Віоли-Джонса: Використовує каскадний класифікатор із зоровими ознаками для обличчя та інших об'єктів на зображеннях.

Метод гістограми напрямлених градієнтів: Описує області зображень шляхом обчислення та аналізу градієнтів яскравості, забезпечуючи інформацію про текстуру та форму об'єктів.

Ці методи займаються різними аспектами обробки зображень та розпізнавання об'єктів на них. Кожен з них має свої унікальні переваги та обмеження в залежності від конкретного застосування та умов задачі.

Враховуючи всі вищезазначені властивості, в ході роботи було прийнято рішення використовувати метод гістограми напрямлених градієнтів.

Проаналізувавши декілька систем ідентифікації користувачів, а саме NEC NeoFace, Google Face Unloc, Apple Face ID та DeepFace від Facebook, можна зробити висновок, що кожна з них має свої унікальні можливості та використовуються для ідентифікації користувачів за геометрією обличчя. Зважаючи на цю закономірність, створення системи моніторингу користувачів за геометрією обличчя, що корелює доступ авторизованих користувачів до системи, є актуальною.

РОЗДІЛ ІІІ. АЛГОРИТМ РЕАЛІЗАЦІЇ СИСТЕМИ МОНІТОРИНГУ. ТЕСТУВАННЯ ТА ДЕМОНСТРАЦІЯ СИСТЕМИ

3.1. Алгоритм реалізації системи моніторингу

Здійснення системи моніторингу користувачів веб-застосунків з використанням технології SSO за допомогою геометрії обличчя передбачає кілька етапів:

1. Автентифікація через SSO: Спершу користувач використовує систему SSO для входу. Він автентифікується на центральному сервері, отримуючи унікальний токен.

2. Отримання вихідного зображення геометрії обличчя: Після успішної автентифікації користувача система отримує доступ до веб-камери або іншого пристрою для захоплення зображення обличчя.

3. Аналіз обличчя: Отримане зображення обличчя аналізується для отримання ключових параметрів, таких як положення очей, рота, форма обличчя тощо.

4. Перевірка з існуючими даними: Отримані параметри обличчя порівнюються з раніше збереженими у системі. Ці дані можуть бути у вигляді шаблонів, векторів ознак або хешів, що відображають унікальні риси.

5. Прийняття рішення: На основі порівняння визначається, чи відповідає отримане обличчя будь-якому із зареєстрованих облич користувачів. Якщо так, то користувач автентифікується і моніторинг продовжується. Якщо ні, система повідомляє про помилку в автентифікації.

6. Продовження моніторингу: У разі успішної автентифікації система починає відслідковувати дії користувача в системі в реальному часі або за потреби записує ці дані для подальшого аналізу а також може надавати/обмежувати доступ користувача до певних функцій чи ресурсів веб-застосунку.

Блок-схема алгоритму зображена на рис.3.1

Рис.3.1. Блок-схема алгоритму системи моніторингу

3.1.1. Алгоритм автентифікації користувачів

SSO (Single Sign-On) – це метод автентифікації, який дозволяє користувачам отримувати доступ до кількох систем з одним набором облікових даних. Нижче наведений алгоритм авторизації з використанням SSO (див. рис.3.2):

- Ініціалізація запиту: Користувач намагається увійти в систему або послугу, яка підтримує SSO.
- Перенаправлення на постачальника SSO: Система перенаправляє користувача на сервер автентифікації (постачальника SSO).
- Автентифікація: Користувач вводить свої облікові дані на сторінці постачальника SSO. Постачальник SSO перевіряє ці дані.
- Створення токена доступу: Якщо облікові дані правильні, постачальник SSO генерує токен доступу.
- Повернення до системи: Постачальник SSO перенаправляє користувача назад до системи або сервісу з токеном доступу.
- Перевірка токену доступу: Система або сервіс перевіряє токен, щоб підтвердити автентифікацію. Якщо токен дійсний, користувач отримує доступ до системи без потреби вводити додаткові облікові дані.

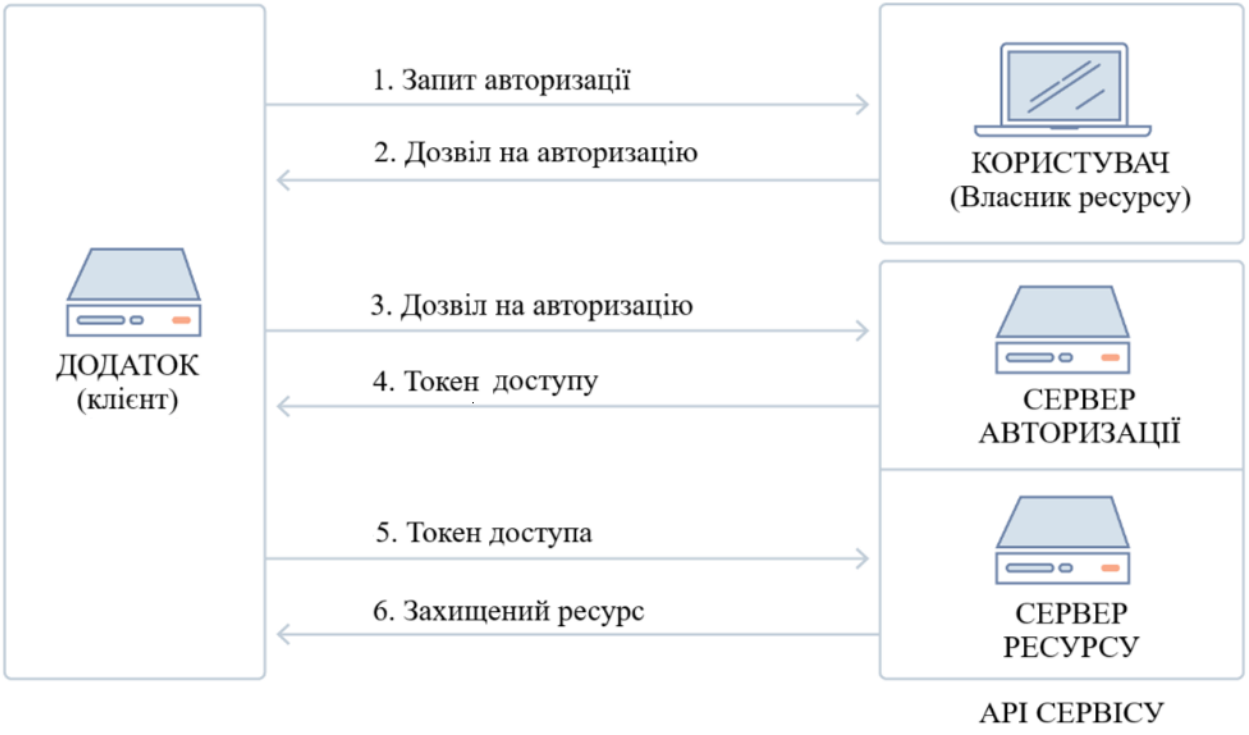


Рис.3.2.Схематичне зображення алгоритму авторизації

3.1.2. Алгоритм розпізнавання обличчя

Розпізнавання облич відомо як одна з технологій обробки зображень, яка може бути застосована в області систем безпеки. Розпізнавання обличчя – це одна з біометричних технологій, яка використовується для ідентифікації або розпізнавання особи на основі певних характеристик. Вважається, що розпізнавання обличчя має високий рівень безпеки, оскільки зображення обличчя важко імітувати, змінити або вкрасти порівняно з традиційними системами безпеки. Схематичне зображення алгоритму розпізнавання обличчя зображено на рис. 3.3.

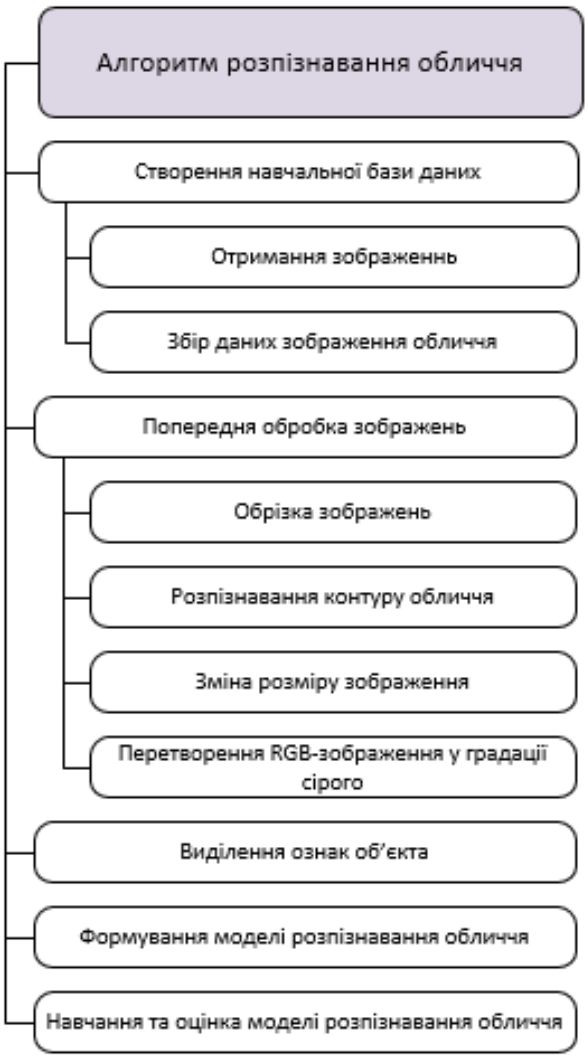


Рис. 3.3. Схематичне зображення алгоритму розпізнавання обличчя

У реалізації процесу розпізнавання обличчя є кілька етапів у процесі розпізнавання обличчя, включаючи навчання бази даних та індивідуальне зіставлення:

1. Навчальна база даних. Процес навчання зображень розпізнавання обличчя (ідентифікації обличчя) зазвичай включає кілька етапів, зокрема:

- *Отримання зображень*. Отримання зображення обличчя — це процес захоплення або сканування аналогового зображення для отримання цифрового зображення. Деякі фактори, які необхідно враховувати в процесі отримання зображення, включають тип пристрою отримання, роздільну здатність камери, методи освітлення, масштабування, відстань і кут захоплення зображення.

- *Збір даних зображення обличчя*. Наступним кроком у процесі навчання розпізнаванню обличчя є збір даних зображення обличчя, який використовуватиметься для навчання моделі. Цей набір даних має складатися із зображень обличчя, представлених векторами ознак або відповідними рисами обличчя, які можуть відрізняти одне обличчя від іншого.

2. Попередня обробка. Після збору даних зображення обличчя зазвичай обробляються, щоб видалити шум або нерелевантну інформацію. Попередня обробка — це один із етапів розпізнавання обличчя, на якому дані зображення обличчя проходитимуть через процес кадрування, визначення обличчя, зміни розміру та зміни формату RGB на градації сірого. Мета етапу попередньої обробки полягає в тому, щоб уможливити кращу обробку зображень обличчя та збільшити ймовірність системи швидкого успішного визначення облич.

- *Обрізка.* Кадрування — це техніка, яка використовується для визначення того, яка частина зображення обличчя містить область об’єкта, яка буде оброблена на наступному етапі, щоб її можна було вирізати та відокремити від небажаної області.

- *Розпізнавання контуру обличчя*. Розпізнавання контуру обличчя — це етап визначення лише частини обличчя особи, щоб отримати простіший вихід зображення. Розпізнавання обличчя в розпізнаванні обличчя зазвичай

використовує різні методи. Одним із найпоширеніших методів визначення обличчя є метод Віоли-Джонса. Цей метод спрямований на видалення частин, які не визначені як обличчя. Метою виявлення обличчя є покращення продуктивності системи під час зіставлення зображень обличчя.

- *Змінити розмір*. Зміна розміру — це процес зміни розміру зображення обличчя в пікселях. Кожне зображення, отримане в результаті розпізнавання обличчя, має різні розміри пікселів, тому необхідно стандартизувати розмір зображення, щоб полегшити системі розпізнавання осіб і створення бази даних зображень однакового розміру.
- *Від RGB до відтінків сірого*. Далі зображення RGB буде перетворено на зображення у градаціях сірого. Функція цього етапу полягає в тому, що зображення у градаціях сірого мають простішу структуру та полегшують процес обчислень. Є надія, що процес зіставлення обличчя буде швидшим і ефективнішим, ніж зображення RGB.

3. Виділення ознак – це процес фіксації характеристик об’єкта, який відрізняє його від інших зображень обличчя в наборі даних. Це досягається за допомогою використання таких алгоритмів виявлення облич, як локальні бінарні шаблони (LBP), аналіз основних компонентів (PCA), власне обличчя, гістограма орієнтованих градієнтів (HOG) або згорткові нейронні мережі (CNN).

На цьому етапі зображення обличчя представляється в системі як матриця. Потім значення матриці усереднюються для порівняння одного зображення з іншим. Ось приклад того, як вилучення ознак виконується за допомогою алгоритму Eigenfaces, який є результатом процесу PCA, який використовується для розпізнавання обличчя. Власні обличчя складаються з векторів ознак, які представляють загальні характеристики або шаблони облич у наборі даних зображень обличчя. На рис.3.3 наведено етапи виділення ознак за допомогою методу власних граней.

4. Формування моделі. Після вилучення рис обличчя з набору даних зображення обличчя модель розпізнавання обличчя може бути сформована за допомогою алгоритмів машинного навчання, таких як Support Vector Machine

(SVM), Random Forest або Convolutional Neural Network (CNN). На цьому етапі алгоритми машинного навчання застосовуються до навчальних даних і знаходить модель, яка підходить для даних. Ця модель генерується за допомогою спеціальних методів машинного навчання для пошуку кореляцій і шаблонів у даних, які були взяті на етапах попередньої обробки та вилучення ознак.

5. Навчання та оцінка моделі. Останнім кроком є навчання моделі за допомогою попередньо обробленого набору даних зображення обличчя та етапу формування моделі. Потім модель оцінюється за допомогою іншого набору даних тестування, щоб забезпечити оптимальну продуктивність і хороше узагальнення для нових даних обличчя.

Індивідуальне розпізнавання або зіставлення. Процес розпізнавання обличчя (ідентифікація обличчя) зазвичай включає кілька етапів, зокрема:

- Процес узгодження .На цьому етапі робиться спроба розпізнати нове зображення обличчя або тестове зображення. Після успішного виділення рис обличчя наступним кроком є порівняння цих рис із рисами обличчя, які вже зберігаються в базі даних. Загальноприйнятим методом є метод евклідової відстані або косинусної подібності.

Евклідова відстань — це метод вимірювання відстані по прямій між двома точками, наприклад, точкою X (X1, X2, …, Xn) і точкою Y (Y1, Y2, …, Yn). Функція евклідової відстані використовується для групування даних шляхом обчислення відстані між точками даних. Чим менше отримане значення відстані, тим подібніші дані можна класифікувати в одну групу (що мають схожість із зображеннями обличчя в наборі даних).

Евклідове значення використовується для обчислення відстані для кожного тренувального даних окремо, щоб знайти порогове значення. Порогове значення — це мінімальна відстань, яку повинні подолати дані тестування, щоб підтвердити, що дані тестування розпізнаються або присутні в базі даних. Якщо тестове зображення обличчя має значення, що перевищує встановлену мінімальну відстань, то тестове зображення обличчя не розпізнається або його немає в базі даних.

- Рішення. На завершальному етапі система прийме рішення на основі відповідних результатів попереднього етапу, наприклад, «розпізнано» або «не розпізнано». З цього рішення ви можете підключити його до інших результатів виводу, таких як надсилання двійкових даних до компонента мікроконтролера для блокування та розблокування систем тощо [22].

На рис.3.4 зображено блок-схему виділення ознак.

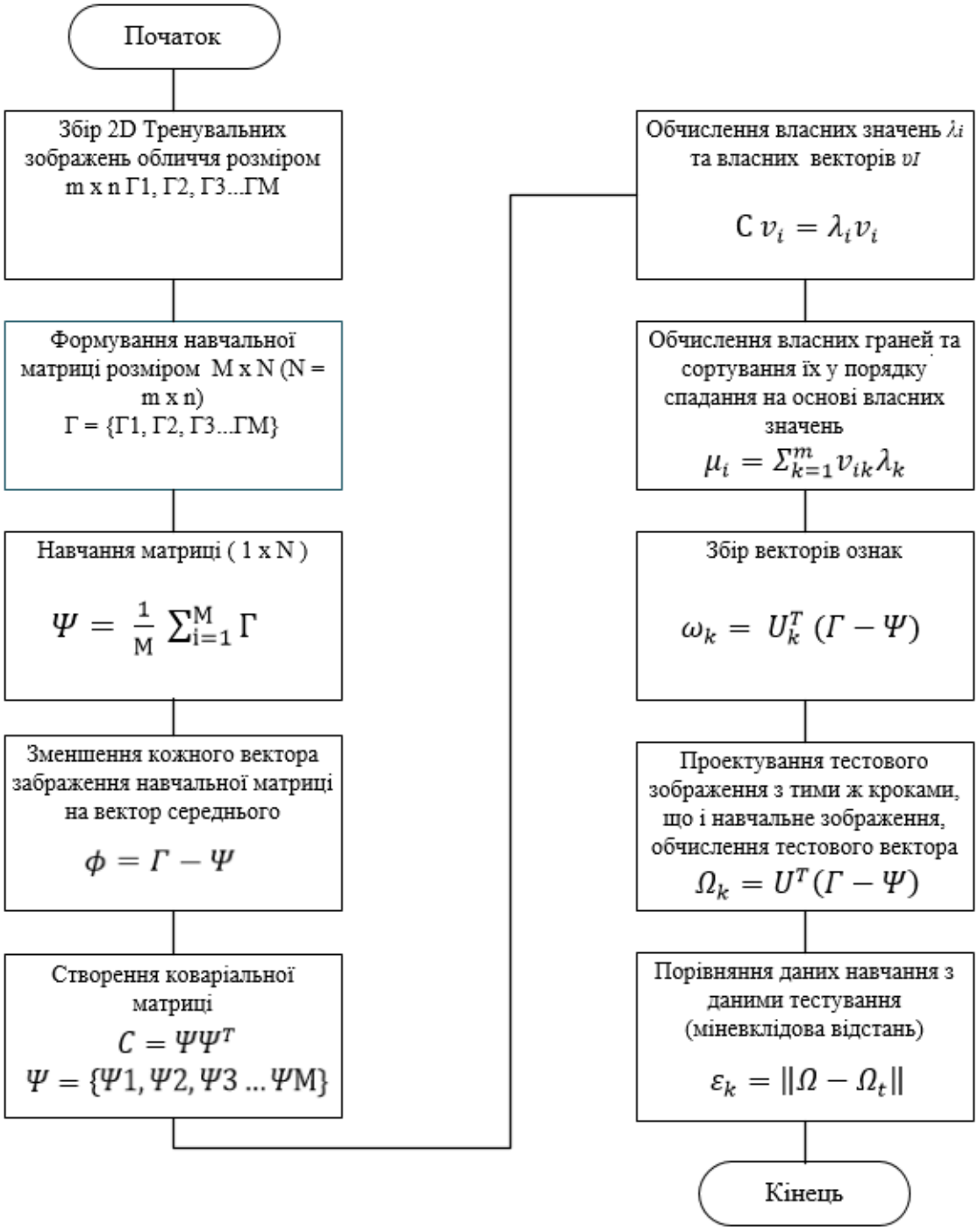


Рисунок 3.4. Етапи виділення ознак за допомогою методу власних граней

3.2. Середовище для розробки системи

Python – це високорівнева мова програмування, яка відома своєю простотою та читабельністю коду. Вона має широке застосування в різних галузях, включаючи веб-розробку, наукові обчислення, штучний інтелект, обробку даних, автоматизацію та багато іншого. Декілька особливостей Python:

- Простота читання та написання коду: Python має чітку та зрозумілу синтаксичну структуру, яка нагадує природну мову, що робить його дуже доступним для початківців.
- Широкий вибір бібліотек: Велика кількість бібліотек дозволяє використовувати Python у багатьох галузях. Наприклад, NumPy, Pandas, Matplotlib для наукових обчислень та обробки даних, Django та Flask для веб-розробки, TensorFlow та PyTorch для роботи зі штучним інтелектом і машинним навчанням.
- Платформонезалежність: Python працює на багатьох платформах, таких як Windows, macOS, Linux, тощо.
- Спільнота та підтримка: Python має велику активну спільноту розробників, яка активно розвиває мову, створює нові бібліотеки та допомагає новачкам.
- Відкритий код: Python є вільним програмним забезпеченням, що означає, що його можна вільно використовувати та змінювати [23].

Для розробки програми було обрано середовище PyCharm. PyCharm – це інтегроване середовище розробки (IDE) для мови програмування Python, розроблене компанією JetBrains. Воно спеціалізується на підтримці Python і надає розробникам широкі можливості для створення програм на цій мові. Ось деякі ключові особливості PyCharm:

- Підтримка Python: PyCharm має вбудовану підтримку для всіх версій Python, що дозволяє працювати з Python 2.x та Python 3.x.
- Інтелектуальне завершення коду: Ця функція пропонує автозаповнення коду, рекомендації по функціям, методам та змінним, що значно полегшує роботу.
- Аналіз коду та відлагодження: PyCharm має вбудовані інструменти для аналізу коду, що допомагають виявляти помилки, а також інструменти для відлагодження, такі як точки зупинки, панель відлагодження тощо.
- Підтримка веб-розробки: Навіть якщо PyCharm орієнтоване на Python, воно також має підтримку HTML, CSS, JavaScript, що робить його зручним для веб-розробки.
- Версії Community та Professional: PyCharm доступний у двох версіях – Community (безкоштовна) та Professional (платна). Версія Professional має більше функцій, таких як підтримка Django, SciPy, розширені інструменти для роботи з базами даних тощо.
- Спільнота та підтримка: Є активна спільнота користувачів PyCharm, а також офіційна документація, підтримка через форуми та інші ресурси [24].

Для візуалізації веб-застосунку було застосовано такі мови програмування як HTML та CSS.

HTML (HyperText Markup Language) - це мова розмітки, яка використовується для створення структури та вмісту веб-сторінок. Вона визначає елементи, такі як заголовки, абзаци, посилання, зображення та інші компоненти сторінки. HTML використовує теги для описування кожного елемента та його функціональності на веб-сторінці.

CSS (Cascading Style Sheets) - це мова стилів, яка використовується для оформлення веб-сторінок. CSS відповідає за зовнішній вигляд елементів HTML, таких як кольори, шрифти, розміри, відступи та інші стилістичні властивості. Вона дозволяє розділити структуру та вміст веб-сторінки від її представлення,

що спрощує роботу з дизайном та додає гнучкості при створенні веб-сторінок [25].

3.3. Основні необхідні бібліотеки

3.3.1. OpenCV

OpenCV – це відкрита бібліотека комп'ютерного зору та обробки зображень з широким функціоналом, доступним для використання у різних продуктах. Була успішно впроваджена в різні проекти, включаючи OCR Solutions – сервіс для розпізнавання техпаспортів, закордонних паспортів та інших документів. OpenCV – необхідний компонент нашого інструментарію для розробки систем штучного інтелекту та комп'ютерного зору. Ця бібліотека реалізована мовами C/C++, а також має підтримку для Python, Java, Ruby, Matlab, Lua та інших мов програмування.

OpenCV має ряд можливостей для роботи з зображенням, такі як: кадрування, зміна розміру, поворот зображення, зміна кольорового зображення на чорно-біле чи в градації сірого, згладжування або розмиття, малювання прямокутників і ліній, можливість написання тексту на зображенні, а також розпізнавання обличчя.

3.3.2. Dlib

Dlib – це сучасний набір інструментів на C++, що містить алгоритми машинного навчання та інструменти для створення складного програмного забезпечення на C++, призначеного для вирішення реальних проблем. Він використовується як в промисловості, так і у науці в широкому спектрі областей, включаючи робототехніку, вбудовані пристрої, мобільні телефони та великі високопродуктивні обчислювальні середовища. Ліцензування Dlib у відкритому доступі дозволяє використовувати його у будь-якій програмі безкоштовно.

Dlib містить широкий спектр інструментів для розв'язання різноманітних завдань: від розпізнавання облич до вирішення завдань комп'ютерного зору, включаючи робототехніку, вбудовані системи, мобільні додатки та багато іншого. Бібліотека поширюється з відкритою ліцензією BSD, що дозволяє безкоштовне використання у комерційних та некомерційних проектах. Dlib отримала популярність завдяки високій ефективності та розширеній функціональності, яка охоплює різні сфери роботи з зображеннями, обробкою даних та розв'язанням задач машинного навчання [26].

3.3.3. face_recognition

Бібліотека face_recognition – це інструмент у Python, призначений для розпізнавання облич. Вона стала досить популярною завдяки зручності та високому рівню абстракції для використання функцій розпізнавання облич.

Основні функції face_recognition включають у себе:

- Детекція облич: Можливість знаходження та виділення облич на зображенні або відео.
- Розпізнавання осіб: Ідентифікація або порівняння облич для встановлення того, чи на зображенні є конкретна особа.
- Екстракція облич: Отримання векторного представлення облич, яке можна використовувати для порівняння.

Ця бібліотека використовує модель глибокого навчання для виконання вищезазначених завдань. Вона доволі зручна для початківців і має добру документацію. Однак для високопродуктивних систем, можуть виникнути вимоги до оптимізації та удосконалення швидкості роботи.

face_recognition побудована на основі бібліотеки dlib, що забезпечує велику точність розпізнавання облич і високий рівень надійності [27].

3.3.4. Flask

Flask – це легкий та гнучкий веб-фреймворк для Python, створений з метою швидкої розробки веб-додатків. Основна ідея Flask полягає у простоті, гнучкості та легкості в освоєнні. Основні риси Flask:

- Легкість використання: Flask пропонує простий синтаксис та мінімальний набір компонентів, що дозволяє швидко розробляти веб-додатки.
- Модульність: Flask дозволяє використовувати лише ті функції, які потрібні для конкретного проекту, використовуючи різноманітні розширення та плагіни.
- Jinja2 шаблони: Шаблонний рушій Jinja2 у Flask дозволяє легко створювати HTML-сторінки з вбудованими підстановками та умовами.
- Розширюваність: Багато розширень та пакетів створено для розвитку функціональності Flask, які можна легко використовувати у ваших проектах.
- Werkzeug WSGI toolkit: Використання вбудованого WSGI toolkit (Web Server Gateway Interface) дозволяє створювати власні веб-сервери та обробники запитів.
- Розробка RESTful API: Flask підтримує легку розробку REST API через різноманітні розширення, що дозволяє створювати веб-сервіси.
- Спільнота та документація: Має широку спільноту користувачів та докладну документацію, що допомагає знайти відповіді на багато питань та завдань [28].

3.3.5. SQLAlchemy

SQLAlchemy – це бібліотека для роботи з базами даних у Python. Вона надає високорівневий, об'єктно-реляційний (ORM) інтерфейс для взаємодії з різними типами баз даних. Основні особливості SQLAlchemy:

- ORM-підхід: SQLAlchemy дозволяє працювати з базами даних у вигляді об'єктів Python, забезпечуючи зручний спосіб створення, читання, оновлення та видалення записів з бази даних.
- Підтримка різних баз даних: Підтримує багато типів баз даних, включаючи SQLite, PostgreSQL, MySQL та інші, дозволяючи вам працювати з різними системами без зміни коду.
- Функціональність SQL-виразів: Надає можливість використовувати SQL-вирази в коді Python, що дозволяє виконувати складні запити та операції з базою даних.

- Міграції баз даних: SQLAlchemy має інструменти для міграції схеми баз даних, дозволяючи зручно зберігати та розгортати зміни в схемі без втрати даних.
- Підтримка транзакцій: Дозволяє використовувати транзакції для забезпечення консистентності даних та управління виконанням операцій з базою даних.
- Спрощений доступ до даних: SQLAlchemy допомагає уникнути написання повторюваного SQL-коду, що спрощує доступ до даних та зменшує кількість помилок.
- Розширюваність та розширення: Має багато розширень, що розширюють його функціональність та додають додаткові можливості.

SQLAlchemy – це потужний інструмент для роботи з базами даних у Python, який надає зручний спосіб взаємодії з різними системами баз даних та полегшує роботу з даними в вашому додатку [29].

3.4. Створення підпрограм для розпізнавання користувачів та серверу SSO для авторизації

Для моніторингу користувачів ми використовуємо розпізнавання за геометрією обличчя, тобто ми отримуємо зображення з веб-камери.

Для розпізнавання користувачів за геометрією обличчя було створено відповідну програму. Щоб розпізнавання відбувалось коректно, необхідно провести тренування та валідацію даних.

Першим етапом було створення бібліотеки з можливими зображеннями у теці проєкта з ім’ям користувача. Для тренування даних створено теку training (рис.3.6). В ній розмістили зображення можливих користувачів у теках

nataliia_noshchenkp – 7 зображень (рис. 3.5), svitlana_repina – 5 зображень, olena_noshchenko – 6 зображень.

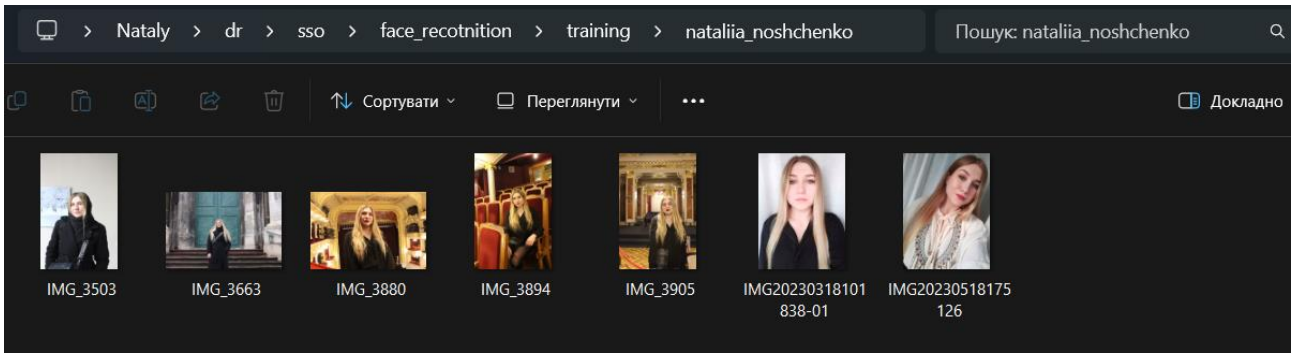


Рис.3.5. Вміст теки training/nataliia_noshchenko

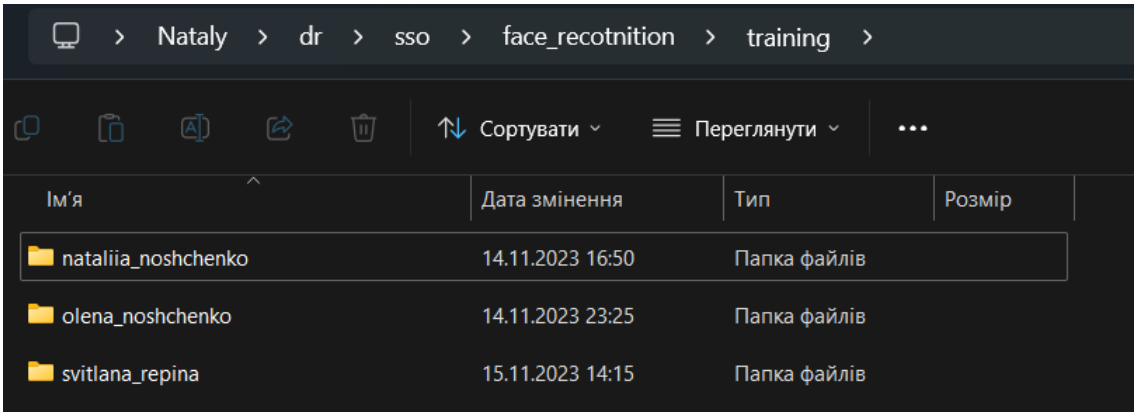


Рис.3.6. Вміст теки training

Для роботи зображеннями та методами ідентифікації було імпортовано бібліотеки cv2, face_recognition, picle (рис. 3.7)

```
import argparse
import pickle
from collections import Counter
from pathlib import Path

import cv2

import face_recognition
from PIL import Image, ImageDraw
```

Рис. 3.7. Фрагмент коду імпортованих бібліотек.

Для обробки тренувальних зображень обличчя створена функція encode_known_faces, ця функція витягає зображення обличчя з тренувальних даних, знаходить їх місцезнаходження та отримує векторні представлення

кожного знайденого обличчя. Потім ці дані (ім'я та векторне представлення) додаються до відповідних списків ‘names’ та ‘encodings’(рис. 3.8.).

```
def encode_known_faces(
    model: str = "hog", encodings_location: Path = ENCODINGS_PATH
) -> None:
    """
    Loads images in the training directory and builds a dictionary of their
    names and encodings.
    """
    names = []
    encodings = []

    for filepath in Path("training").glob("*/*"):
        name = filepath.parent.name
        image = face_recognition.load_image_file(filepath)

        face_locations = face_recognition.face_locations(image, model=model)
        face_encodings = face_recognition.face_encodings(image, face_locations)

        for encoding in face_encodings:
            names.append(name)
            encodings.append(encoding)

    name_encodings = {"names": names, "encodings": encodings}
    with encodings_location.open(mode="wb") as f:
        pickle.dump(name_encodings, f)
```

Рис. 3.8. Фрагмент коду для обробки тренувальних зображень обличчя

Для перевірки моделі на валідаційних даних створена функція validate призначена, які не були використані для тренування. Це дозволяє перевірити, як модель працює на даних, які вона раніше не бачила (рис. 3.9.).

```
def validate(model: str = "hog"):
    """
    Runs recognize_faces on a set of images with known faces to validate
    known encodings.
    """
    for filepath in Path("validation").rglob("*"):
        if filepath.is_file():
            recognize_faces(
                image_location=str(filepath.absolute()), model=model
            )
```

Рис. 3.9. Фрагмент коду валідації даних.

Після цього запускаємо скрипт detector.py для перевірки моделі на валідаційних даних. Для запуску програми відкриваємо термінал, переміщаємось до необхідної теки та виконуємо команди що відповідають за тренування та перевірку зображень, --train та –validate відповідно (рис. 3.10.).

```
C:\>cd Users\User\dr\sso\face_recotnition

C:\Users\User\dr\sso\face_recotnition>python detector.py --train

C:\Users\User\dr\sso\face_recotnition>python detector.py --validation
usage: detector.py [-h] [--train] [--validate] [--test] [-m {hog,cnn}] [-f F]
detector.py: error: unrecognized arguments: --validation

C:\Users\User\dr\sso\face_recotnition>python detector.py --validate
```

Рис.3.10. Результат запуску команд –train та --validate

В результаті тестування програми розпізнавання облич отримуємо зображення з підписом користувача. У випадку якщо, користувач не був авторизований, буде повернено “unknown” (рис. 3.11.).

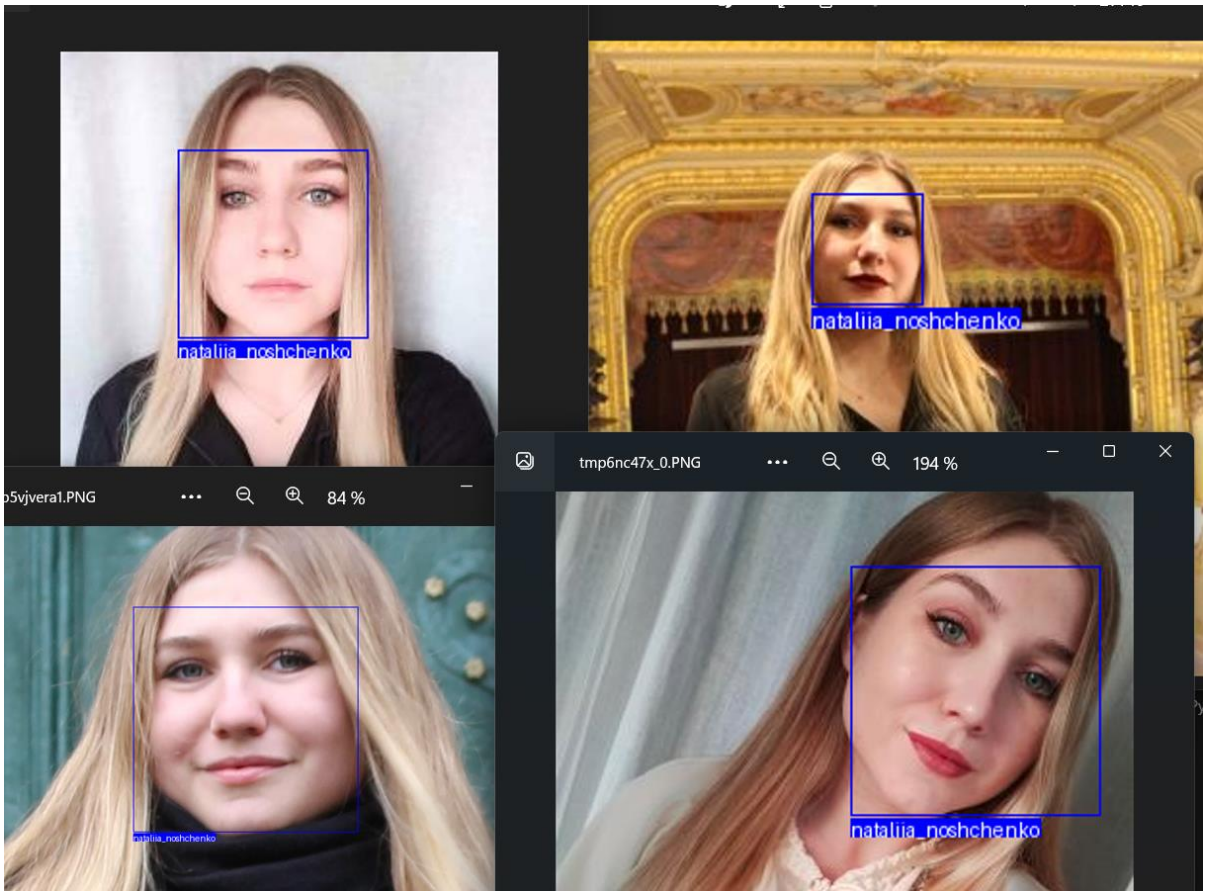


Рис.3.11. Результат виконання команди --validate

Далі для запуску розпізнавання користувачів було створено функцію capture_from_camera, ця функція відстежує та розпізнає обличчя у реальному часі за допомогою веб-камери на основі збережених закодованих даних облич (рис.3.12).

```python
def capture_from_camera(model: str = "hog", encodings_location: Path = ENCODINGS_PATH) -> None:
    """
    Captures frames from the video camera and recognizes faces in real-time.
    """
    cap = cv2.VideoCapture(0)  # 0 corresponds to the default camera, change it if needed

    
    with encodings_location.open(mode="rb") as f:
        loaded_encodings = pickle.load(f)
```

Рис.3.12. Фрагмент коду отримання даних з веб-камери

Для виконання розпізнавання обличчя в реальному часі за допомогою веб-камери, запускаємо скрипт командою python detector.py. Це спричинить виклик функції capture_from_camera() та розпочне потік відео з результатами розпізнавання обличчя (рис. 3.12.).

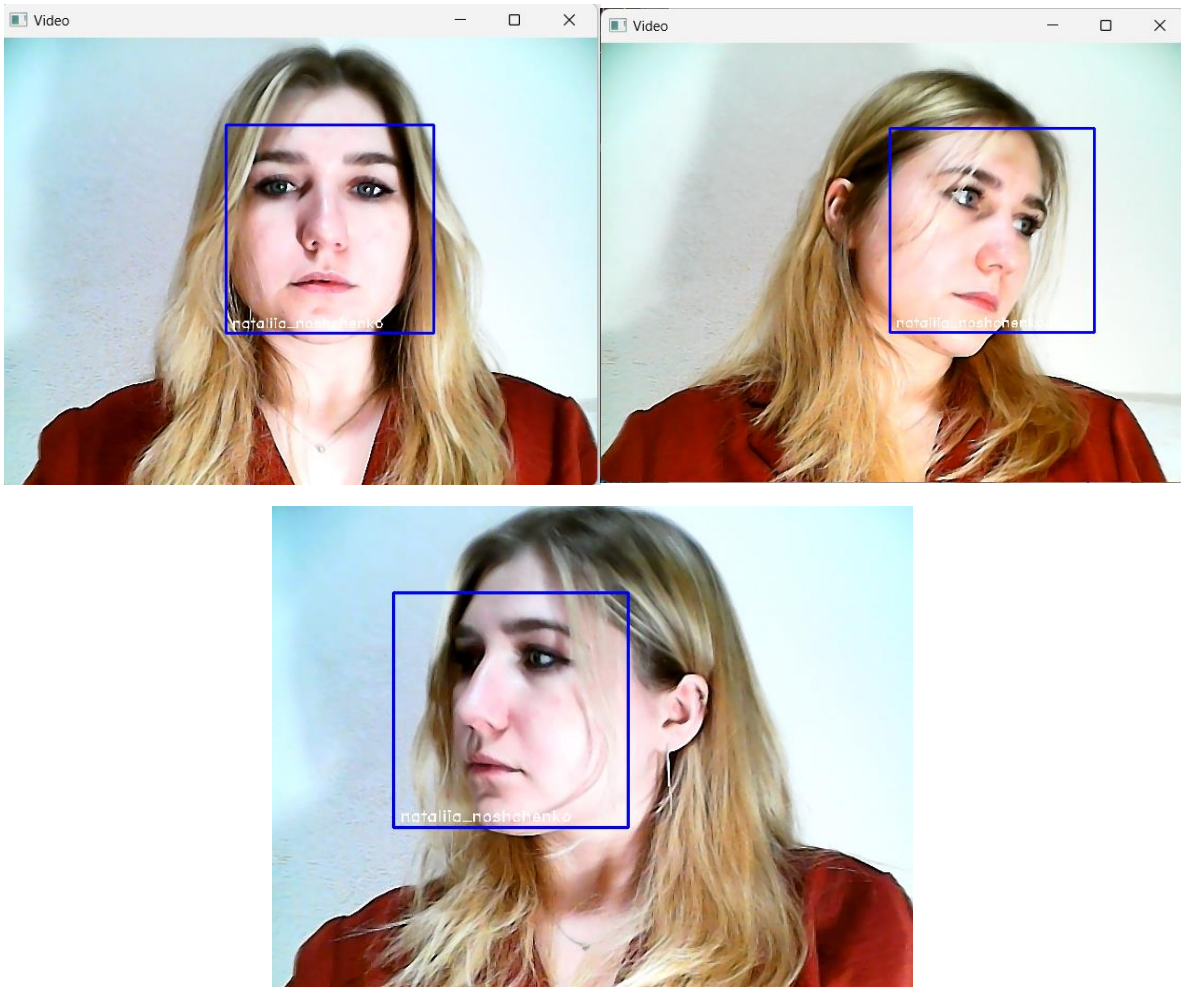


Рис. 3.12. Результат запуску підпрограми detector.py з різним нахилом та поворотом голови.

Для автентифікації та авторизації користувачів через Single Sign-On створено відповідно підпрограму.

Функція login обробляє запити на вхід користувачів. Вона перевіряє, чи відправлено запит на перенаправлення після успішного входу, та перевіряє, чи є це джерело допустимим для доступу до сервера SSO. Якщо користувач вже увійшов або якщо запит має цільовий URL без вказаного сервісу, він буде перенаправлений на головну сторінку. У разі, якщо користувач увійшов і запит має цільовий URL, створюється токен, який передається у цей URL для подальшого використання. Якщо ж запит ще не має користувача, відбувається перенаправлення на сторінку логіну (рис. 3.13.).

```
const login = (req, res, next) => {
  const { serviceURL } = req.query;
  if (serviceURL != null) {
    const url = new URL(serviceURL);
    if (alloweOrigin[url.origin] !== true) {
      return res
        .status(400)
        .json({ message: "Your are not allowed to access the sso-server" });
    }
  }
  if (req.session.user != null && serviceURL == null) {
    return res.redirect("/");
  }
  if (req.session.user != null && serviceURL != null) {
    const url = new URL(serviceURL);
    const intrmid = encodedId();
    storeApplicationInCache(url.origin, req.session.user, intrmid);
    runPythonScript();

    return res.redirect(`${serviceURL}?ssoToken=${intrmid}`);
  }
```

Рис. 3.13. Фрагмент коду функції login

Для перевірки та підтвердження коректності SSO токену створено функцію verifySsoToken. Якщо токени присутні, вони порівнюються з даними в кеші і базі даних для підтвердження їх валідності. Якщо токени не відповідають очікуваним значенням або не є валідними, повертається відповідь з помилкою Unauthorized (рис. 3.14.).

```
const verifySsoToken = async (req, res, next) => {
  const appToken = appTokenFromRequest(req);
  const { ssoToken } = req.query;

  if (
    appToken == null ||
    ssoToken == null ||
    intrmTokenCache[ssoToken] == null
  ) {
    return res.status(400).json({ message: "badRequest" });
  }

  const appName = intrmTokenCache[ssoToken][1];
  const globalSessionToken = intrmTokenCache[ssoToken][0];

  if (
    appToken !== appTokenDB[appName] ||
    sessionApp[globalSessionToken][appName] !== true
  ) {
    return res.status(403).json({ message: "Unauthorized" });
  }

  const payload = generatePayload(ssoToken);

  const token = await genJwtToken(payload);

  delete intrmTokenCache[ssoToken];
  return res.status(200).json({ token });
};
```

Рис. 3.14. Фрагмент коду перевірки та підтвердження токену

На рис.3.15 показано результат авторизації користувачів використовуючи SSO

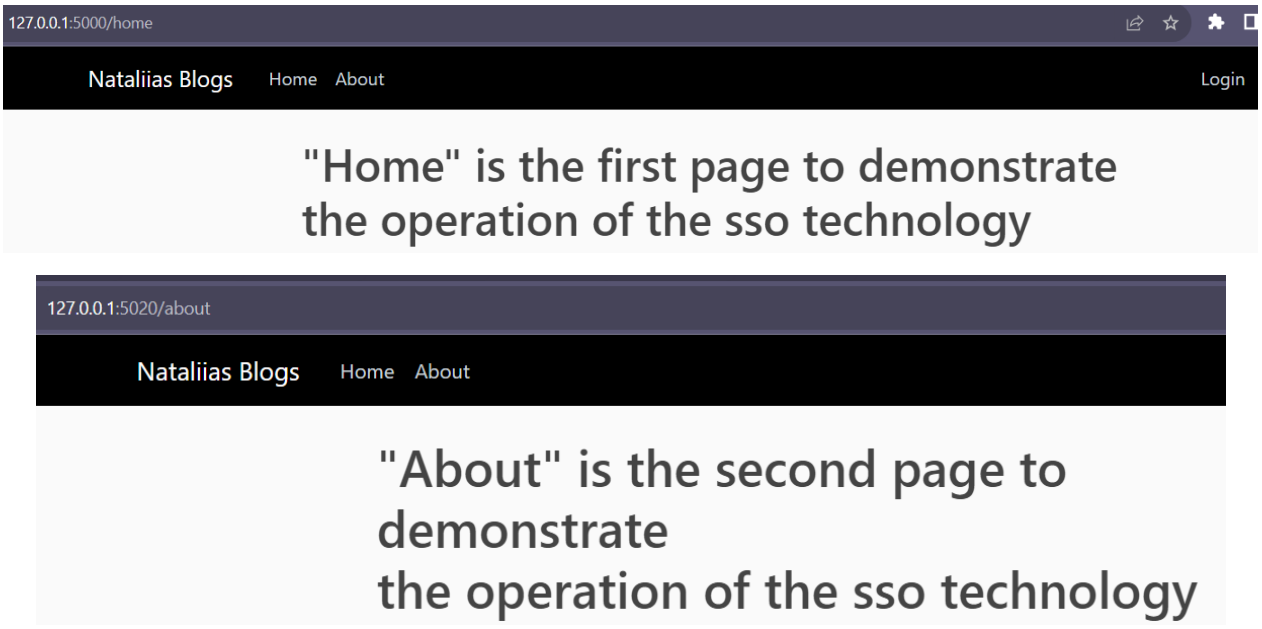


Рис.3.15. Демонстрація успішного входу на сайти з використанням SSO

3.5. Демонстрація та тестування системи моніторингу

Для початку роботи необхідно зареєструватись на сайті, при цьому вказавши свій логін користувача, email, пароль та повторити пароль, а також завантажити свою фотографію, для подальшої ідентифікації (рис. 3.16.).

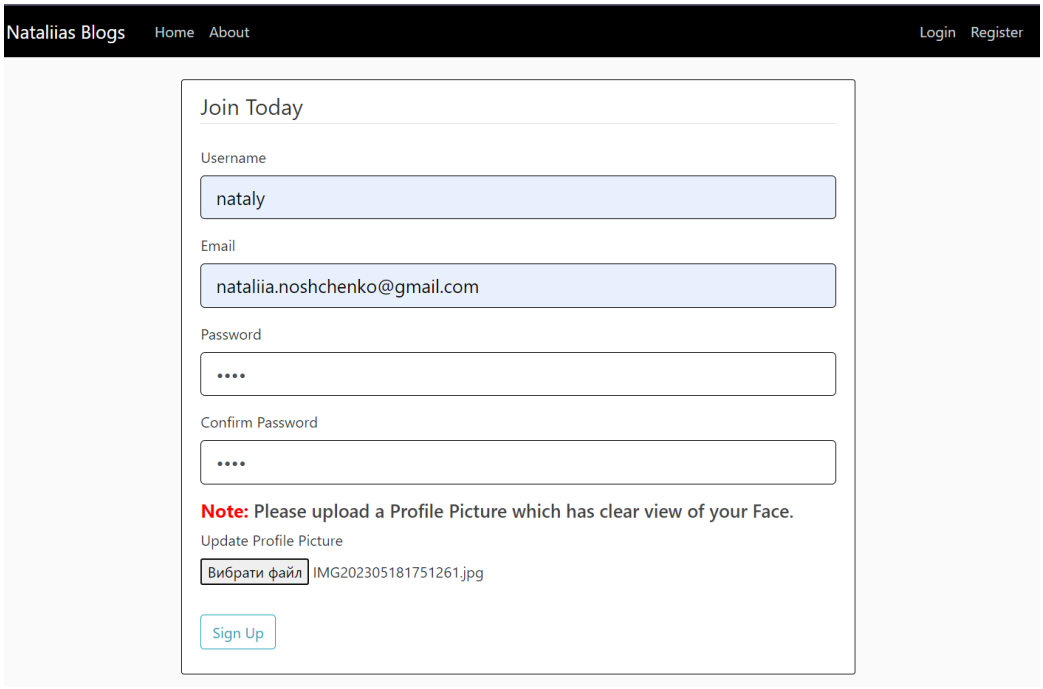


Рис.3.16. Форма реєстрації користувача

Нові фотографії зареєстрованих користувачів зберігаються у теці profile_pics, а логін, email пароль та вид доступу у файлі site.db (нові файли підсвічені зеленим кольором) (рис. 3.17.).

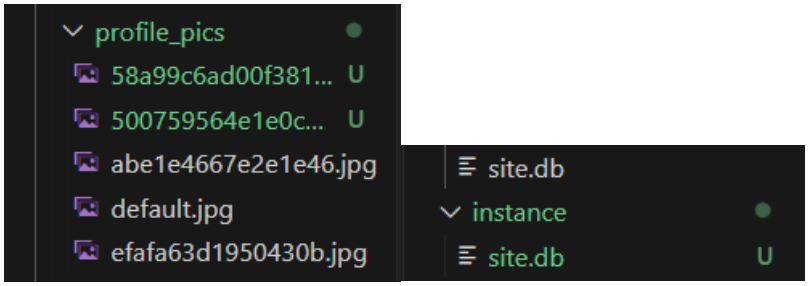


Рис.3.17 Місце збереження даних користувачів

Якщо реєстрація пройшла успішно, ви отримаєте повідомлення, що ваш акаунт було створено. Також одразу з’являється можливість зайти на сайт (рис. 3.18.).

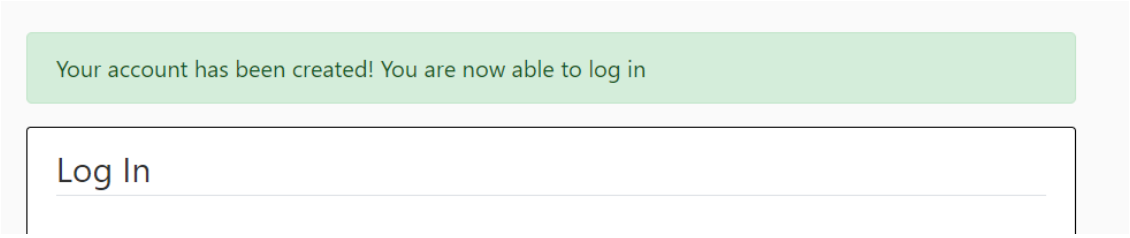


Рис.3.18. Результат успішної реєстрації на сайті

У випадку якщо, ім’я користувача чи email будуть вже зайняті, буде отримано помилку, і неможливо буде зареєструватись під зайнятим вже ім’ям (рис. 3.19.).

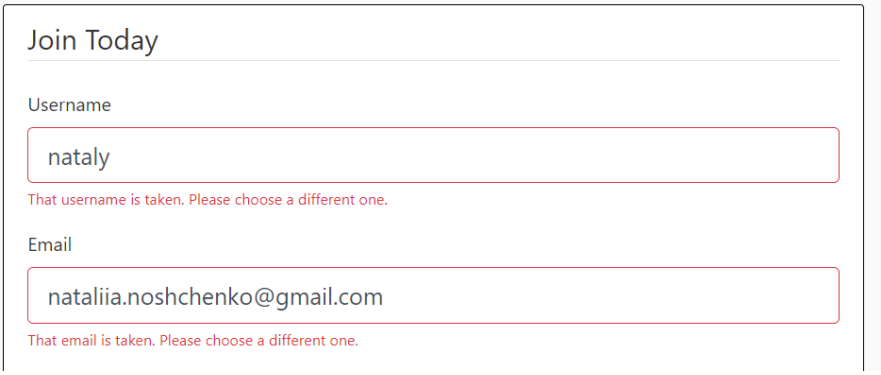


Рис.3.19. Демонстрація реєстрації логіну користувача, який вже зайнятий

Наступним кроком йде авторизація (рис. 3.20.).

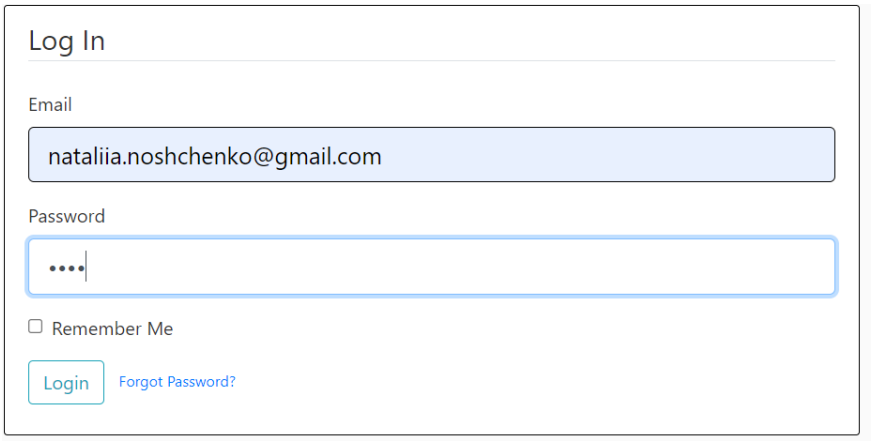


Рис.3.20.Процес авторизації користувача

Після успішної авторизації у фоновому режимі одразу запускається система моніторингу користувачів. Для демонстрації вікно з відображенням відеопотоку винесено на передній план Система кожні 30 секунд перевіряє чи за

комп’ютером є користувач і чи це та ж сама людина, що вводила свої дані при авторизації, якщо ні, то відбувається вихід з системи, і користувача повертає на початковий екран для авторизації. На рис.3.21, 3.22 показано результати моніторингу.

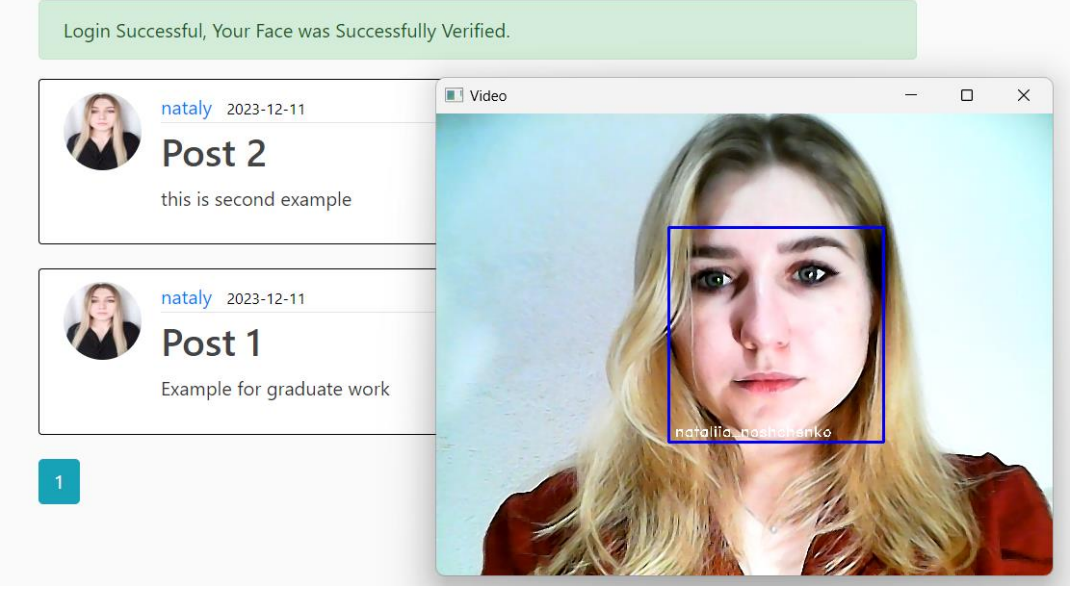


Рис 3.21. Результат успішної авторизації та розпізнаванням користувача

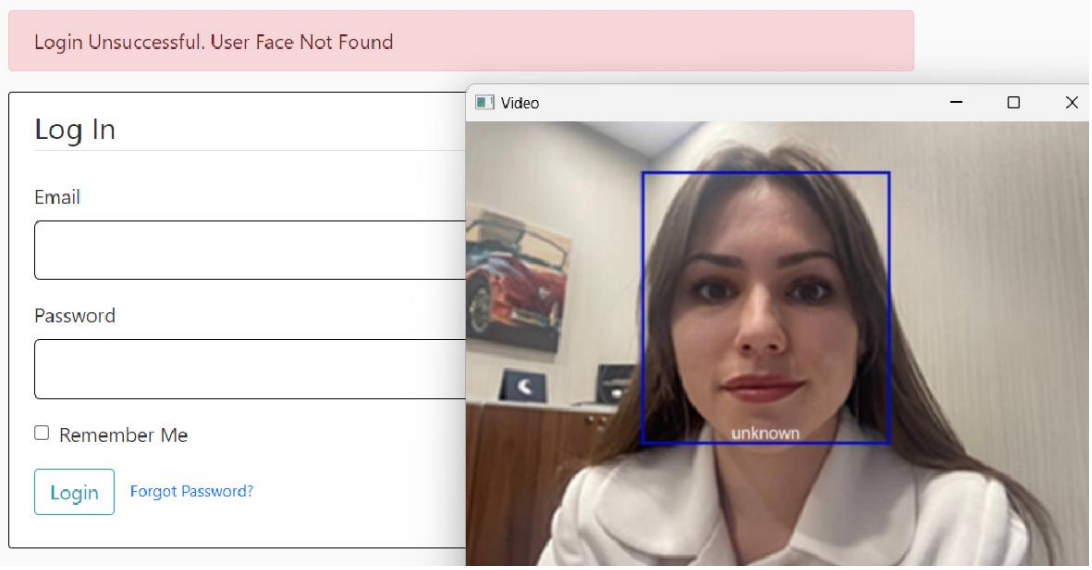


Рис 3.22. Результат моніторингу, коли за ПК невідомий користувач

Для авторизованих користувачів є можливість створювати, редагувати та видаляти пости (рис. 3.23.)

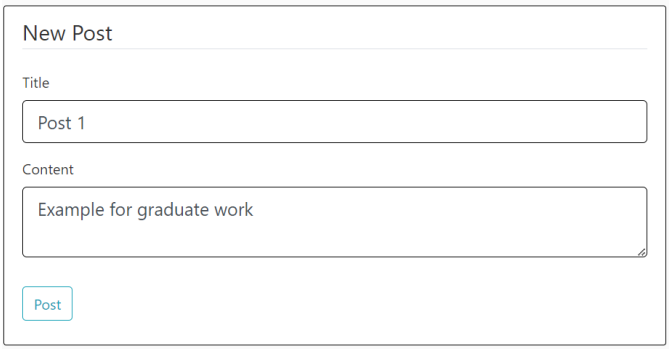


Рис.3.23. Демонстрація створення нового посту

Результат створення постів можна побачити зі вкладки Home (рис. 3.24.)

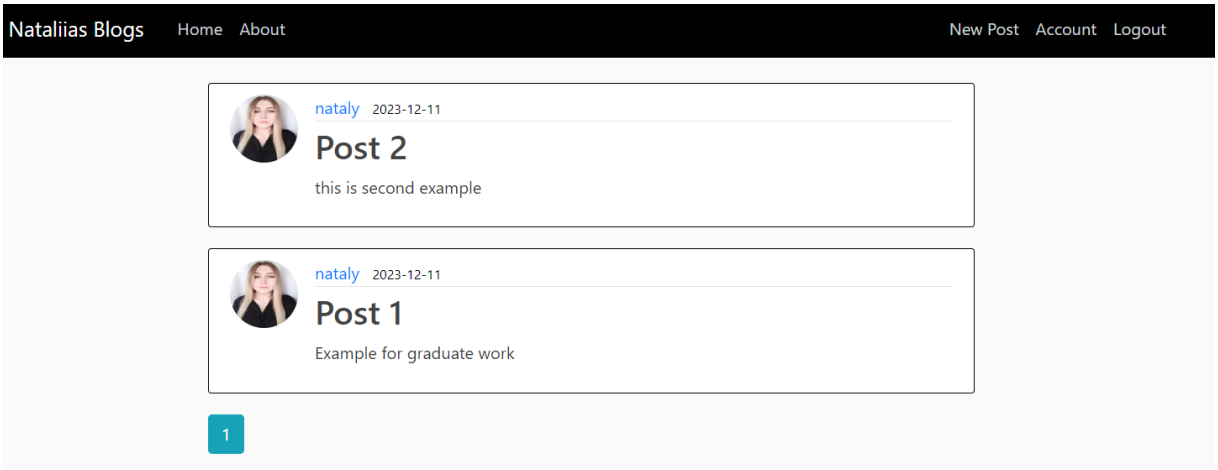


Рис.3.24. Результат створення кількох постів

На вкладці Account, можна побачити інформацію про авторизованого користувача (рис. 3.25.).

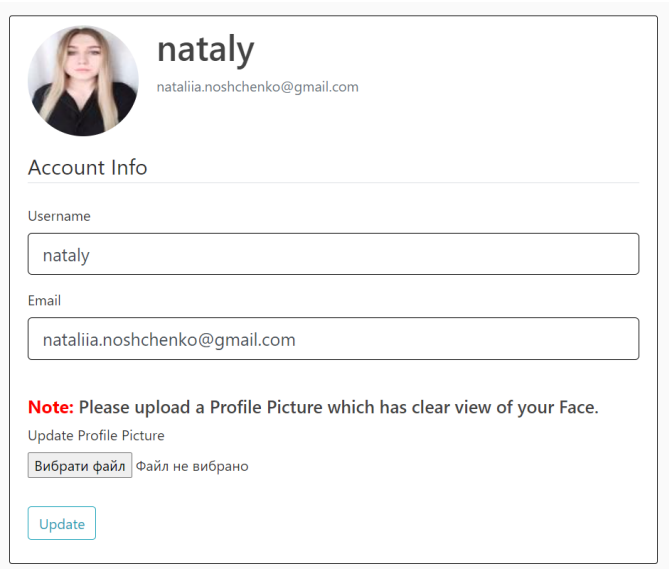


Рис.3.25. Зміст вкладки Account

Якщо користувач не авторизований йому буде не доступний перегляд та редагування наповнення сайтів (рис. 3.26.)

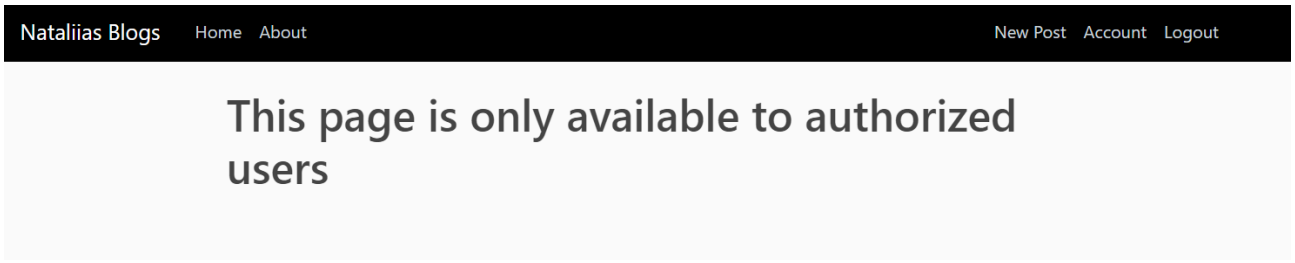


Рис. 3.26. Приклад сторінки неавторизованого користувача

3.6. Порівняння існуючих систем моніторингу з створеною

Порівняння основні характеристик систем ідентифікації за геометрією обличчя та створеної системи моніторингу, зображено в табл.3.1.

Таблиця 3.1

Порівняння основні характеристик систем ідентифікації за геометрією обличчя та створеної системи моніторингу

Характеристики	**NEC NeoFace**	**Google Face Unlock**	**Apple Face ID**	**DeepFace (Facebook)**	**Створена система**
Технологія зчитування зразка	Використання шаблонів облич	Використання фронтальної камери	Технологія TrueDepth для сканування облич	Зображення надає користувач	Використання зображення з веб-камери
Спосіб розпізнавання	Використання шаблонів для порівняння облич	Використання алгоритмів машинного навчання	Тривимірне сканування облич	Використання нейронних мереж для створення векторних представлень облич для порівняння	Метод гістограми напрямлених градієнтів
Розблокування пристроїв	Так	Так	Так	Ні	Ні

Функція	Ідентифікація	Ідентифікація	Ідентифікація	Ідентифікація	Моніторинг
Точність розпізнавання	Висока	Залежить від умов освітлення та якості камери	Висока	Висока	Висока
Використання нейронних мереж	Так	Ні	Так	Так	Так

3.7. Висновки до розділу III

В цьому розділі було розглянуто алгоритм системи моніторингу користувачів, який відбувається наступним чином. Користувач автентифікується один раз, отримує унікальний токен для доступу до різних систем. В системі здійснюється аналіз обличчя користувача для отримання унікальних характеристик, отриманого за допомогою відеопотоку. Отримані характеристики порівнюються зі збереженими даними користувачів для визначення ідентичності. На основі порівняння вирішується, чи належить обличчя до зареєстрованих користувачів. При успішній ідентифікації відбувається моніторинг дій користувача та надається/обмежується доступ до ресурсів веб-застосунку.

Для реалізації ідентифікації за геометрію обличчя було розглянуто відповідний алгоритм розпізнавання за геометрією обличчя, що включає в себе виявлення та вимірювання ключових параметрів обличчя, таких як відстань між очима, розмір рота, положення носа та вух, створення шаблону, де отримані параметри використовуються для створення унікального шаблону геометрії обличчя, який слугуватиме основою для подальшої ідентифікації, та порівняння та ідентифікація, де отриманий шаблон порівнюється з існуючими шаблонами в базі даних для визначення відповідності та ідентифікації особи.

Також окремо було розглянуто алгоритм авторизації з використанням SSO, коли користувач може увійти до системи з одним набором облікових даних (логін/пароль), інтегрованим з іншими системами, без необхідності повторно вводити дані, завдяки спільній авторизації між декількома веб-сайтами чи додатками. Це дає можливість забезпечити безпечний та зручний доступ до різних ресурсів з використанням одного набору облікових даних.

В якості мови програмування було обрано Python, через ряд своїх позитивних якостей, таких як простота та читабельність, універсальність та наявності розширених бібліотек. Як середовище розробки – PyCharm це інтегроване середовище розробки (IDE) для Python, яке має такі переваги: PyCharm пропонує розширені можливості для розробки, включаючи підтримку віртуальних середовищ, автодоповнення, рефакторинг, аналіз коду та інші корисні функції; IDE надає потужні інструменти для відлагодження програм, включаючи можливість встановлення точок зупинки, перевірку значень змінних та інші інструменти для пошуку та виправлення помилок; PyCharm підтримує різні операційні системи, має зручний інтерфейс та багато інших інструментів для підвищення продуктивності розробників.

Було доінстальовано ряд бібліотек, таких як OpenCV, face-recognizer, Dlib, SQLAlchemy та Flask.

Було розроблено систему моніторингу користувачів веб-застосунків, використовуючи технологію SSO, за геометрією обличчя, що дало змогу контролювати доступ авторизованих користувачів та проведено тестування розробленої системи моніторингу, що дало змогу дослідити доцільність використання розробленої системи для вирішення поставленої задачі. А також у вигляді таблиці подано порівняння основних функцій відомих систем моніторингу, та розробленої мною системи.

РОЗДІЛ IV. ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

У світі існує безліч форм забруднення, створених людьми через їхню економічну діяльність, що негативно впливає на навколишнє середовище. Одним з таких видів знецінення є біологічне забруднення. Це проявляється через населення живими організмами, які мають свій життєвий цикл та можуть поширюватися в середовищі, погіршуючи якість повітря, води, ґрунту та продуктів харчування.

Біологічне забруднення виникає в результаті присутності організмів з власним життєвим циклом, які пристосовані до життя в середовищі і можуть впливати на якість повітря, води, ґрунту та харчові продукти, що є загрозою для організмів. Це може призводити до інфекційних або паразитарних захворювань. Таким чином, коли ці організми заражають навколишнє середовище, відбувається біологічне забруднення, яке шкодить багатьом організмам, що залежать від цих ресурсів для свого життєвого циклу.

Існують різні види біологічного забруднення залежно від місця і типу організмів, що викликають зараження. Розглянемо основні:

- *Біологічне забруднення води:* викликане наявністю розкладених органічних речовин та патогенних мікроорганізмів у стічних водах, від сільськогосподарської діяльності або промислових викидів.
- *Біологічне забруднення повітря:* організми, які забруднюють повітря, можна знайти як у приміщеннях, так і на відкритому повітрі. Вони можуть бути вірусами та бактеріями, що впливають на людей та тварин. Погана вентиляція та вологість сприяють розвитку біологічних забруднень.
- *Біологічне забруднення ґрунту:* бактерії та віруси також можуть погіршувати якість ґрунту через побутове сміття, тваринництво та інші діяльності.
- *Біологічне забруднення в продуктах харчування:* організми можуть змінювати склад їжі, роблячи її непридатною для споживання.

Основні біологічні забруднювачі

Біологічне забруднення може бути викликане різними біологічними забруднювачами, які можна поділити на:

- *Бактерії:* Збудники таких захворювань, як пневмонія чи харчова отрута, наприклад, сальмонелою.
- *Найпростіші:* Одноклітинні мікроорганізми, які викликають хвороби у людей, такі як малярія, амебіаз та сонячний удар.
- *Віруси:* Інфекційні агенти, що розвиваються в клітинах інших організмів. Причини хвороб у людини, таких як СНІД, гепатит, віспа або кору.
- *Гельмінти:* Хробаки або паразити, які живуть у людини, викликаючи захворювання, такі як солітери, глисти або п’явки.
- *Гриби:* Паразитують в організмах, оскільки не можуть синтезувати поживні речовини. Можуть викликати поверхневі інфекції, такі як на шкірі чи нігтях.
- *Членистоногі:* Наприклад, кліщі, що викликають захворювання шкіри та можуть бути джерелом алергенів, наприклад, короста.

За індексом ризику зараження забруднювачі поділяються на чотири групи:

- *Група 1:* Малоймовірно, що викличуть хворобу у людини.
- *Група 2:* Можуть викликати захворювання, але є ефективні методи лікування та менша ймовірність поширення.
- *Група 3:* Здатні викликати серйозні захворювання, але зазвичай є методи лікування, наприклад, бактерії, що викликають туберкульоз або гепатит.
- *Група 4:* Найнебезпечніші збудники, легко поширюються і, зазвичай, не існує ефективного лікування.

Причини та наслідки

Біологічне забруднення виникає через викиди різноманітних забруднюючих речовин у твердому, рідкому або газоподібному стані. Ці речовини часто походять з таких джерел:

- Різні галузі промисловості.
- Лабораторії мікробіології.

- Виробництво продуктів харчування.
- Сільське господарство.
- Робота з гігієною, особливо у медичних установах.
- Видалення залишків.
- Очищення стічних вод.
- Будь-яка діяльність, що включає контакт з живими організмами.

Необхідно враховувати, що умови, такі як наявність поживних речовин, вологість та температура, сприяють активності біологічних забруднювачів.

Біологічне забруднення призводить до різноманітних хвороб у всіх формах життя. Медицина сьогодні може лікувати більшість захворювань, спричинених біологічними забруднювачами. Проте, виникають нові забруднювачі, з якими не завжди легко боротися або знайти спосіб запобігання чи лікування.

Тема варта роздумів про важливість мінімізації забруднення навколишнього середовища. Залежно від місця проживання та економічних можливостей, доступність лікування може відрізнятися.

Профілактика біологічного забруднення включає такі кроки:

- Регулярне прибирання та дезінфекція речей і простору, де ми перебуваємо.
- Відповідна обробка та утилізація відходів у будинках, офісах або на роботі для уникнення прямого контакту з ними.
- Використання спеціальних контейнерів для органічних відходів.
- Дотримання належної гігієни під час робочого дня, як в офісі, так і на місцях праці.
- Регулярні фізичні огляди для запобігання захворювань, що можуть вплинути на людей та тварин.
- Навчання та стимулювання працівників дотримуватися санітарних правил у робочому середовищі.
- Поширення знань про гігієну та профілактичні заходи серед дітей [30].

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи отримані наступні результати:

- Проаналізовано технологію єдиного входу, методи авторизації, ідентифікації та моніторингу, а також існуючі методи біометричної ідентифікації та аналіз і обробку зображень, та на основі проведеного аналізу обрано оптимальні методи для вирішення задачі ідентифікації користувачів.
- Розроблено систему моніторингу користувачів веб-застосунків, з використанням технології SSO, в якій завдяки реалізації розпізнавання за геометрією обличчя, забезпечується контроль доступу користувачів до вебзстосунку.
- Проведено тестування розробленої системи моніторингу користувачів, що дало змогу дослідити доцільність використання розробленої системи для вирішення поставленої задачі.
- Описано індивідуальну тему з охорони тему навколишнього середовища – «Біологічне забруднення довкілля».

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Розпізнавання обличчя за допомогою алгоритмів Машинного навчання./ Н.Б. Шаховська, О.А. Басистюк.// Національний університет «Львівська політехніка» - Режим доступа: World Wide Web. – URL http://dspace.nbuv.gov.ua/bitstream/handle/123456789/162343/10-Shakhovska.pdf?sequence=1

2. Дослідження ефективності методів біометричної автентифікації / Г.Є. Ляшенко, А.А. Астраханцев – Режим доступа: World Wide Web. – URL https://www.researchgate.net/publication/323728995_Doslidzenna_efektivnosti_metodiv_biometricnoi_avtentifikacii

3. Живучість та резильєнтність критичної інфраструктури – 2023: збірник матеріалів міжнародної науково-практичної конференції, м. Київ, 19 жовтня 2023 р., ІПМЕ ім. Г.Є. Пухова НАН України. – 2023. – 149-152 с. – Режим доступа: World Wide Web. – URL https://ipme.kiev.ua/wp-content/uploads/2023/11/Матеріали_конференції_Survivability_and_Resilience-2023-4.pdf.

4. Single Sign-On [Електронний ресурс] // https://auth0.com – Режим доступа: World Wide Web. – URL:https://auth0.com/docs/authenticate/single-sign-on

5. Single sign-on (SSO) [Електронний ресурс] / Taina Teravainen // techtarget.com – Режим доступа: World Wide Web. – URL:https://www.techtarget.com/searchsecurity/definition/single-sign-on

6. What is SSO? | How single sign-on works [Електронний ресурс] // Cloudflare – Режим доступа: World Wide Web. – URL:https://www.cloudflare.com/learning/access-management/what-is-sso/

7. How SSO prevents security threats and aids compliance? [Електронний ресурс] // linkedin.com – Режим доступа: World Wide Web. – URL:https://www.linkedin.com/pulse/how-sso-prevents-security-threats-aids-compliance-rainbowsecure/

8. Identification vs. Authentication: What's the Difference? [Електронний ресурс] // hypr.com – Режим доступа: World Wide Web. – URL:https://blog.hypr.com/identification-vs-authentication

9. What Is Identification, Authentication and Authorization? [Електронний ресурс] // notarize.com – Режим доступа: World Wide Web. – URL:https://www.notarize.com/blog/what-is-identification-authentication-and-authorization

10. Как работает single sign-on (технология единого входа)? [Електронний ресурс] /Станислав Тибекин // habr.com – Режим доступа: World Wide Web. – URL:https://habr.com/ru/companies/nixys/articles/563244/

11. Типи ідентифікації користувачів [Електронний ресурс] // infobezpeka.com – Режим доступа: World Wide Web. – URL:http://www.infobezpeka.com/publications/?id=92

12. Identification and authentication [Електронний ресурс] // IBM MQ – Режим доступа: World Wide Web. – URL:https://www.ibm.com/docs/en/ibm-mq/7.5?topic=mechanisms-identification-authentication

13. Царьов Р.Ю. Біометричні технології: навч. посіб. [для вищих навчальних закладів] / Р.Ю. Царьов, Т. М. Лемеха. – Одеса: ОНАЗ ім. О.С. Попова, 2016. – 140 с.: іл.

14. Захаров В. П., Рудешко В. І. З-38 Біометричні технології в XXI столітті та їх використання правоохоронними органами: посібник. – 2-ге вид., доп. / В. П. Захаров, В. І. Рудешко. – Львів: ЛьвДУВС, 2015. – 492 с.

15. Khushbu Pandey, Reshma Lilani, Pooja Naik, Geeta Pol, 2014, Human Face Recognition Using Image Processing, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) ICONET – 2014 (Volume 2 – Issue 04), URL: https://www.ijert.org/human-face-recognition-using-image-processing

16. Біометрична ідентифікація і автентифікація особи за геометрією обличчя О.В. Нечипоренко, Я.В. Корпань. Режим доступа: World Wide Web. –

URL:https://www.researchgate.net/publication/321423112_Biometricna_identifikacia_i_avtentifikacia_osobi_za_geometrieu_oblicca

17. Методи обробки зображень [Електронний ресурс] // nmetau.edu. – Режим доступа: World Wide Web.– URL https://nmetau.edu.ua/file/07_7.5_lbr_gr_rbr_.pdf

18. Template Matching [Електронний ресурс] // opencv.org– Режим доступа: World Wide Web. – URL: https://docs.opencv.org/3.4/de/da9/tutorial_template_matching.html

19. Аналіз методу Віоли-Джонса для систем комп'ютерного зору [Електронний ресурс] // conferences.vntu.edu.ua – Режим доступа: World Wide Web. – URL: https://conferences.vntu.edu.ua/index.php/mn/mn2021/paper/viewFile/13162/11056

20. Histograms of Oriented Gradients for Human Detection [Електронний ресурс] // lear.inrialpes.fr – Режим доступа: World Wide Web. – URL: https://lear.inrialpes.fr/people/triggs/pubs/Dalal-cvpr05.pdf

21. 12 найкращих інструментів моніторингу персоналу [Електронний ресурс] // clevercontrol.com – Режим доступа: World Wide Web. – URL: https://clevercontrol.com/uk/12-best-employee-monitoring-tools/

22. The workflow of the face recognition process in image matching [Електронний ресурс] // opencv.org– Режим доступа: World Wide Web. – URL: https://it.telkomuniversity.ac.id/en/the-workflow-of-the-face-recognition-process/

23. Що таке Python? [Електронний ресурс] // opencv.org– Режим доступа: World Wide Web. – URL: http://www.plug.org.ua/documentation/about-python

24. PyCharm [Електронний ресурс] // – Режим доступа: World Wide Web. – URL: https://www.jetbrains.com/pycharm/

25. Електронний HTML і CSS довідник українською мовою [Електронний ресурс] // html-css.co.ua– Режим доступа: World Wide Web. – URL https://html-css.co.ua

26. Використання OpenCV і Face Recognition в системах розпізнавання облич на одноплатних комп'ютерах типу Raspberry Pi [Електронний ресурс] //

evergreens.com. – Режим доступа: World Wide Web. – URL: https://evergreens.com.ua/ua/articles/open-cv-face-recognition.html

27. Dlib [Електронний ресурс] // dlib.net – Режим доступа: World Wide Web. – URL: http://dlib.net

28. Tutorial Series: How To Build Web Applications with Flask [Електронний ресурс] //– Режим доступа: World Wide Web. – URL: https://www.digitalocean.com/community/tutorials/how-to-create-your-first-web-application-using-flask-and-python-3

29. SQLAlchemy, набір інструментів SQL з відкритим кодом і об'єктно-реляційний картограф [Електронний ресурс] // blog.desdelinux.net – Режим доступа: World Wide Web. – URL: https://blog.desdelinux.net/uk/sqlalchemy-un-kit-de-herramientas-sql-y-el-mapeador-relacional-de-objetos-open-source/

30. Біологічне забруднення [Електронний ресурс] // www.renovablesverdes.com – Режим доступа: World Wide Web. – URL: https://www.renovablesverdes.com/uk/contaminacion-biologica/

ДОДАТКИ

Додаток А

ФРАГМЕНТ КОДУ ПІДПРОГРАМИ РОЗПІЗНАВАННЯ ОБЛИЧЧЯ В РЕАЛЬНОМУ ЧАСІ

```
import argparse
import pickle
from collections import Counter
from pathlib import Path

import cv2

import face_recognition
from PIL import Image, ImageDraw

ENCODINGS_PATH = Path("output/encodings.pkl")
BOUNDING_BOX_COLOR = "blue"
TEXT_COLOR = "white"

# Create directories if they don't already exist
Path("training").mkdir(exist_ok=True)
Path("output").mkdir(exist_ok=True)
Path("validation").mkdir(exist_ok=True)

parser = argparse.ArgumentParser(description="Recognize faces in an image")
parser.add_argument("--train", action="store_true", help="Train on input data")
parser.add_argument(
    "--validate", action="store_true", help="Validate trained model"
)
parser.add_argument(
    "--test", action="store_true", help="Test the model with an unknown image"
```

```
)
parser.add_argument(
    "-m",
    action="store",
    default="hog",
    choices=["hog", "cnn"],
    help="Which model to use for training: hog (CPU), cnn (GPU)",
)
parser.add_argument(
    "-f", action="store", help="Path to an image with an unknown face"
)
args = parser.parse_args()


def encode_known_faces(
    model: str = "hog", encodings_location: Path = ENCODINGS_PATH
) -> None:
    """
    Loads images in the training directory and builds a dictionary of their
    names and encodings.
    """
    names = []
    encodings = []

    for filepath in Path("training").glob("*/*"):
        name = filepath.parent.name
        image = face_recognition.load_image_file(filepath)

        face_locations = face_recognition.face_locations(image, model=model)
        face_encodings = face_recognition.face_encodings(image, face_locations)
```

```
        for encoding in face_encodings:
            names.append(name)
            encodings.append(encoding)

    name_encodings = {"names": names, "encodings": encodings}
    with encodings_location.open(mode="wb") as f:
        pickle.dump(name_encodings, f)


def recognize_faces(
    image_location: str,
    model: str = "hog",
    encodings_location: Path = ENCODINGS_PATH,
) -> None:
    """
    Given an unknown image, get the locations and encodings of any faces and
    compares them against the known encodings to find potential matches.
    """
    with encodings_location.open(mode="rb") as f:
        loaded_encodings = pickle.load(f)

    input_image = face_recognition.load_image_file(image_location)

    input_face_locations = face_recognition.face_locations(
        input_image, model=model
    )
    input_face_encodings = face_recognition.face_encodings(
        input_image, input_face_locations
    )
```

```
    pillow_image = Image.fromarray(input_image)
    draw = ImageDraw.Draw(pillow_image)

    for bounding_box, unknown_encoding in zip(
        input_face_locations, input_face_encodings
    ):
        name = _recognize_face(unknown_encoding, loaded_encodings)
        if not name:
            name = "Unknown"
        _display_face(draw, bounding_box, name)

    del draw
    pillow_image.show()


def _recognize_face(unknown_encoding, loaded_encodings):
    """
    Given an unknown encoding and all known encodings, find the known
    encoding with the most matches.
    """
    boolean_matches = face_recognition.compare_faces(
        loaded_encodings["encodings"], unknown_encoding
    )
    votes = Counter(
        name
        for match, name in zip(boolean_matches, loaded_encodings["names"])
        if match
    )
    if votes:
```

```
        return votes.most_common(1)[0][0]


def _display_face(draw, bounding_box, name):
    """
    Draws bounding boxes around faces, a caption area, and text captions.
    """
    top, right, bottom, left = bounding_box
    draw.rectangle(((left, top), (right, bottom)), outline=BOUNDING_BOX_COLOR)
    text_left, text_top, text_right, text_bottom = draw.textbbox(
        (left, bottom), name
    )
    draw.rectangle(
        ((text_left, text_top), (text_right, text_bottom)),
        fill=BOUNDING_BOX_COLOR,
        outline=BOUNDING_BOX_COLOR,
    )
    draw.text(
        (text_left, text_top),
        name,
        fill=TEXT_COLOR,
    )


def validate(model: str = "hog"):
    """
    Runs recognize_faces on a set of images with known faces to validate
    known encodings.
    """
    for filepath in Path("validation").rglob("*"):
```

```
    if filepath.is_file():
        recognize_faces(
            image_location=str(filepath.absolute()), model=model
        )


def capture_from_camera(model: str = "hog", encodings_location: Path = ENCODINGS_PATH) ->
None:
    """
    Captures frames from the video camera and recognizes faces in real-time.
    """
    cap = cv2.VideoCapture(0)  # 0 corresponds to the default camera, change it if needed



    with encodings_location.open(mode="rb") as f:
        loaded_encodings = pickle.load(f)


        # Setting the frame size
    cap.set(cv2.CAP_PROP_FRAME_WIDTH, 640)  # Setting the frame width
    cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)  # Setting the frame height


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


        if not ret:
            print("Error reading frame from camera.")
            break


        input_face_locations = face_recognition.face_locations(frame, model=model)
        input_face_encodings = face_recognition.face_encodings(frame, input_face_locations)
```

```
        pillow_image = Image.fromarray(frame)
        draw = ImageDraw.Draw(pillow_image)

        for bounding_box, unknown_encoding in zip(input_face_locations, input_face_encodings):
            name = _recognize_face(unknown_encoding, loaded_encodings)
            if not name:
                name = "Unknown"
            """_display_face(draw, bounding_box, name)"""

            # add a signature with a name to the image
            top, right, bottom, left = bounding_box
            cv2.rectangle(frame, (left, top), (right, bottom), (255, 0, 0), 2)
            font = cv2.FONT_HERSHEY_DUPLEX
            cv2.putText(frame, name, (left + 6, bottom - 6), font, 0.5, (255, 255, 255), 1)

            if name == "Unknown":
                print("Unknown")
                exit()

        # show the video in real time
        cv2.imshow('Video', frame)

        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

    cap.release()
    cv2.destroyAllWindows()

if __name__ == "__main__":
```

Продовження додатку А

```
    if args.train:
        encode_known_faces(model=args.m)
    if args.validate:
        validate(model=args.m)
    if args.test:
        if args.f:  # If a file path is provided, use it
            recognize_faces(image_location=args.f, model=args.m)
        else:
            capture_from_camera(model=args.m)
```

ДОДАТОК Б

ФРАГМЕНТ КОДУ НАЛАШТУВАННЯ МАРШРУТІВ ТА ФУНКЦІЙ, ЯКІ ВІДПОВІДАЮТЬ НА HTTP-ЗАПИТИ

```
@app.route("/register", methods=['GET', 'POST'])
def register():
    form = RegistrationForm()
    if form.validate_on_submit():
        image_file = 0
        if form.picture.data:
            picture_file = save_picture(form.picture.data)
            image_file = picture_file
        hashed_password = bcrypt.generate_password_hash(form.password.data).decode('utf-8')
        if image_file == 0:
            flash('Please Upload Your Profile Picture.', 'danger')
            return redirect(url_for('register'))
        else:
            user = User(username=form.username.data, email=form.email.data,
password=hashed_password, image_file = image_file, face_reco_id = image_file)
        db.session.add(user)
        db.session.commit()
```

Продовження додатку Б

```
        flash('Your account has been created! You are now able to log in', 'success')
        return redirect(url_for('login'))
    return render_template('register.html', title='Register', form=form)


@app.route("/login", methods=['GET', 'POST'])
def login():
    if current_user.is_authenticated:
        return redirect(url_for('home'))
    form = LoginForm()
    if form.validate_on_submit():
        user = User.query.filter_by(email=form.email.data).first()
        if user and bcrypt.check_password_hash(user.password, form.password.data):
                login_user(user, remember=form.remember.data)
                next_page = request.args.get('next')
                flash('Login Successful, Your Face was Successfully Verified.', 'success')
                return redirect(next_page) if next_page else redirect(url_for('home'))
        else:
            flash('Login Unsuccessful. Please check email and password', 'danger')
    return render_template('login.html', title='Login', form=form)


@app.route("/logout")
def logout():
    logout_user()
    return redirect(url_for('home'))


@app.route("/account", methods=['GET', 'POST'])
@login_required
def account():
    form = UpdateAccountForm()
```

Продовження додатку Б

```
    if form.validate_on_submit():
        if form.picture.data:
            picture_path = os.path.join(app.root_path, 'static/profile_pics', current_user.image_file)
            os.remove(picture_path)
            picture_file = save_picture(form.picture.data)
            current_user.image_file = picture_file
            current_user.face_reco_id = picture_file
        current_user.username = form.username.data
        current_user.email = form.email.data
        db.session.commit()
        flash('Your account has been updated!', 'success')
        return redirect(url_for('account'))
    elif request.method == 'GET':
        form.username.data = current_user.username
        form.email.data = current_user.email
    image_file = url_for('static', filename='profile_pics/' + current_user.image_file)
    return render_template('account.html', title='Account',
                           image_file=image_file, form=form)


@app.route("/post/new", methods=['GET', 'POST'])
@login_required
def new_post():
    form = PostForm()
    if form.validate_on_submit():
        post = Post(title=form.title.data, content=form.content.data, author=current_user)
        db.session.add(post)
        db.session.commit()
        flash('Your post has been created!', 'success')
        return redirect(url_for('home'))
```

Продовження додатку Б

```
    return render_template('create_post.html', title='New Post',
                           form=form, legend='New Post')


@app.route("/post/<int:post_id>")
def post(post_id):
    post = Post.query.get_or_404(post_id)
    return render_template('post.html', title=post.title, post=post)


@app.route("/post/<int:post_id>/update", methods=['GET', 'POST'])
@login_required
def update_post(post_id):
    post = Post.query.get_or_404(post_id)
    if post.author != current_user:
        abort(403)
    form = PostForm()
    if form.validate_on_submit():
        post.title = form.title.data
        post.content = form.content.data
        db.session.commit()
        flash('Your post has been updated!', 'success')
        return redirect(url_for('post', post_id=post.id))
    elif request.method == 'GET':
        form.title.data = post.title
        form.content.data = post.content
    return render_template('create_post.html', title='Update Post',
                           form=form, legend='Update Post')


@app.route("/post/<int:post_id>/delete", methods=['POST'])
@login_required
```

Продовження додатку Б

```
def delete_post(post_id):
    post = Post.query.get_or_404(post_id)
    if post.author != current_user:
        abort(403)
    db.session.delete(post)
    db.session.commit()
    flash('Your post has been deleted!', 'success')
    return redirect(url_for('home'))


@app.route("/user/<string:username>")
def user_posts(username):
    page = request.args.get('page', 1, type=int)
    user = User.query.filter_by(username=username).first_or_404()
    posts = Post.query.filter_by(author=user)\
        .order_by(Post.date_posted.desc())\
        .paginate(page=page, per_page=5)
    return render_template('user_posts.html', posts=posts, user=user)
```