ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра інженерії програмного забезпечення

Пояснювальна записка

до бакалаврської роботи
на ступінь вищої освіти бакалавр

на тему: **«РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ ІДЕНТИФІКАЦІЇ ОБЛИЧЧЯ ДЛЯ СИСТЕМИ IOT МОВОЮ PHYTON»**

Виконав: студент 5 курсу, групи ПЗ-52 спеціальності
121 Інженерії програмного забезпечення
(шифр і назва спеціальності)

Тимошенко В.С.
(прізвище та ініціали)
Керівник Трінтіна Н.А.
(прізвище та ініціали)
Рецензент_____
(прізвище та ініціали)
Нормоконтроль_____
(прізвище та ініціали)

КИЇВ – 2021

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії Програмного Забезпечення

Ступінь вищої освіти - «Бакалавр»

Спеціальність -121 Інженерія Програмного Забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії Програмного Забезпечення

_____Негоденко О.В.

«____»_____2021 року

З А В Д А Н Н Я

НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

Тимошенко Володимир Семенович

(прізвище, ім‘я, по батькові)

1.Тема роботи: «Розробка програмного забезпечення на основі ідентифікації обличчя для системи IOT мовою Phyton»

Керівник роботи Тринтіна Наталія Альбертівна,

затверджені наказом вищого навчального закладу від — 12.03 2021 року №65.

2. Строк подання студентом роботи 01.06.2021.

3. Вхідні дані до роботи:

3.1 Розробка програмного забезпечення ідентифікації обличчя

3.2 Алгоритм авторизації та аутентифікації

3.3 Тестування интерфейсу користувача

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

4.1 Аналіз існуючих рішень для ідентифікації обличчя

4.2 Дослідження технічних документацій для створення програмного забезпечення ідентифікації обличчя

4.3 Розробка функціоналу програмного забезпечення для тестування

4.4 Аналіз та створення програмного забезпечення

5.Перелік графічного матеріалу :

5.1 Титульний стайд

5.2 Об’єкт, мета, предмет дослідження

5.3 Порівняльна характеристика аналогів.

5.4 Макети додатку

5.5 Висновки

6. Дата видачі завдання 19.04.2021

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково – технічної літератури	19.04.2021 – 22.04.2021	
2	Вимоги для розробки програмного забезпечення розпізнавання обличчя	24.04.2021 – 26.04.2021	
3	Аналіз перспективи програмного забезпечення	28.04.2021 – 29.04.2021	
4	Дослідження програмних засобів	30.04.2021 – 05.05.2021	
5	Моделювання об’єкту проектування	06.05.2021 – 22.05.2021	
6	Вступ, висновки, реферат	22.05.2021 – 26.05.2021	
7	Розробка презентації	27.05.2021	
8	Здача роботи	01.06.2021	

Студент Тимошенко Володимир Семенович

Керівник роботи Тринтіна Наталія Альбертівна

РЕФЕРАТ

Обсяг роботи 54 сторінок, 49 ілюстрацій, 4 таблиці, 1 формула, 15 джерел посилань.

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ ІДЕНТИФІКАЦІЇ ОБЛИЧЧЯ ДЛЯ СИСТЕМИ IOT МОВОЮ PHYTON

Об’єктом роботи є розпізнавання обличчя в реальному часі. Предметом роботи є програмний засіб для ідентифікації та автентифікації людей для подальшого фіксування в довідковій базі їх особистість, коли вони приходять і йдуть з дому, використовуючи розпізнавання обличчя, а також додаток для адміністрування системою, реєстрації в ній та візуалізації отриманих даних.

Метою роботи є створення захищеного від зловживання програмного засобу для ідентифікації та автентифікації людей для подальшого фіксування в довідковій базі їх особистість, коли вони приходять і йдуть з дому, використовуючи розпізнавання обличчя та додаток для адміністрування в системі, реєстрації в ній та візуалізації отриманих даних.

Методи розроблення: розробка програмного продукту для фіксації особистості людини, розробка веб додатка для адміністрування системи та візуалізації отриманих даних. Інструменти розроблення: безкоштовне, вільно поширюване інтегроване середовище розробки JetBrains PyCharm 2018.2.3, мова програмування Python та Javascript.

Результати роботи: виконано загальний огляд систем розпізнавання обличчя, а також способи захисту таких систем від зловживання. Розроблено програмний продукт для check-in/check-out людей з використанням розпізнавання обличчя. Розроблено програмний продукт для адміністрування даної системи, реєстрації в ній та візуалізації отриманих даних.

Для користування програмним продуктом наведеним вище потрібен комп’ютер чи ноутбук на базі ОС Windows, веб-камера та підключення до мережі Інтернет при першому запуску.

Цей програмний продукт може застосовуватися як для систем IoT, так і на підприємстві або закладі в якому має сенс контролювати присутність окремих людей та точний час який вони проводять авторизованими в систему.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

CV – Computer Vision;

DCNN – Deep Convolutional Neural Network;

EAR – Eye Aspect Ratio;

HOG – Histogram of Oriented Gradients;

ID – Identity;

IDE – Integrated Design Environment, інтегроване середовище розробки;

IP – Internet Protocol;

Jpg – Joint Photographic Group;

LBP – Local Binary Patterns;

MVC – Model View Controller;

NASA – The National Aeronautics and Space Administration;

SVM – Support Vector Machine;

БД – База даних;

ДНК (дезоксирибонуклеїнова кислота) – один із двох типів природних нуклеїнових кислот, що забезпечує зберігання, передачу з покоління в покоління і реалізацію генетичної програми розвитку й функціонування живих організмів;

ОС – Операційна система;

СКБД – Система керувань базами даних;

ТМ - територіальний менеджер

2D – 2-dimensional;

3D – 3-dimensional;

ВСТУП

Оцінка сучасного стану об’єкта дослідження або розробки. На сьогодні існує декілька способів для того, щоб ідентифікувати та автентифікувати людину. Ці способи можна розділити на не біологічні, такі як: паспорт, номер телефону, кредитна картка, ip-адреса, та біологічні, тобто ті, що використовують біологічні особливості, наприклад: відбитки пальців, голос, геометрична будова долоні, портрет обличчя, особливості малюнка сітківки ока, райдужна оболонка ока, судинні рисунки, ДНК або навіть “відбитки мозку”, тобто особливості нейронної активності кожної людини.

Ідентифікація та автентифікація людини за допомогою не біологічних та біологічних способів використовується в багатьох галузях та вже зараз є невіддільною частиною нашого життя. Використання біологічних особливостей людини для ідентифікації та автентифікації стає все більш популярним у всьому світі через великий спектр наукових викликів, велику кількість способів застосування у комерційних додатках, у контексті біометрії та питань безпеки.

Розпізнавання людини за допомогою обличчя має кілька переваг у порівнянні з іншими методами ідентифікації людини:

1. не потребує дорогого обладнання (достатньо звичайного ноутбука з веб-камерою);
2. не потрібен фізичний контакт із пристроями. Достатньо просто декілька секунд подивитись у камеру.

До недоліків розпізнавання людини за допомогою обличчя можна віднести:

1. не є надійною на 100%. Можливі помилки у розпізнаванні через декілька факторів: освітлення у приміщенні, кут огляду камери, також якщо вибірка довідкової БД з якою порівнюють обличчя людини, яке потрібно розпізнати, занадто велика.

Нині активно використовуються кілька десятків комп'ютерних методів розпізнавання обличчя: методи на основі нейронних мереж; геометричний метод

розпізнавання обличчя; метод головних компонент; метод гнучкого порівняння на графах; метод Віоли-Джонса; приховані моделі Маркова; локальні бінарні шаблони й т. д. У цій роботі розглядається метод розпізнавання обличчя, який використовує нейронні мережі та глибоке навчання.

Мета й завдання роботи. Метою кваліфікаційної роботи є створення програмного засобу для фіксації часу, коли працівник прийшов та пішов з робочого місця, використовуючи розпізнавання обличчя. Для досягнення цієї мети поставлено такі завдання.

- Дослідити сучасний метод автоматизованого розпізнавання обличчя.
- Дослідити існуючі методи захисту від зловживання систем розпізнавання обличчя.
- Розробити захищену від зловживання систему ідентифікації та автентифікації, використовуючи розпізнавання обличчя
- Розробити сервіс для реєстрування працівників у системі наведеній вище
- Розробити інструмент для адміністрування системи та візуалізації отриманих даних

Об’єкт, методи й засоби дослідження або розроблення. Об’єктом роботи є розпізнавання обличчя в реальному часі. Предметом роботи є захищений від зловживання програмний засіб для попередньої реєстрації, ідентифікації та автентифікації користувачів для подальшого фіксування в довідковій базі часу, коли вони приходять і йдуть з дому, використовуючи розпізнавання обличчя.

Методи розроблення: розробка програмного продукту для фіксації часу проведеного користувачами в домівці та додатку для адміністрування системи, реєстрація в ній та візуалізації отриманих даних. В якості інструменту створення програмного засобу було обрано безкоштовне, вільно поширюване інтегроване середовище розробки JetBrains PyCharm 2018.2.3 [1], мова програмування Python [2] та Javascript [3].

Можливі сфери застосування. Створений програмний продукт може застосовуватися в будь-якому підприємстві або закладі в якому має сенс фіксувати

точний час присутності окремих людей. Наприклад, в магазині для контролювання працівників.

РОЗДІЛ 1 АВТОМАТИЗОВАНЕ РОЗПІЗНАВАННЯ ОБЛИЧЧЯ

1.1 Поняття автоматизованого розпізнавання обличчя

На сьогодні, розпізнавання обличчя вже не є чимось фантастичним і вже впроваджується та використовується у багатьох галузях. Воно дуже швидко набирає популярності у всьому світі через великий спектр наукових викликів, велику кількість способів застосування у комерційних додатках, у контексті біометрії та у питаннях безпеки.

Відповідь на запитання «Чиє це обличчя?» – головна задача області розпізнавання обличчя. Люди мають природні здібності для цієї цілі, використовуючи власні перцептивні та когнітивні системи, тоді як машини потребують комплексні системи, що містять передові алгоритми та великі бази даних облич. Вивчення, розробка та проектування таких методів і технологій – це проблеми області автоматизованого розпізнавання облич.

Автоматизоване розпізнавання обличчя полягає у пошуку обличчя людини серед бази даних, що містить зображення облич інших людей.

Проте, для того щоб шукати обличчя у БД, спочатку треба локалізувати це обличчя на вхідному зображенні. Процес автоматизованого розпізнавання облич зображений на рис. 1.1.

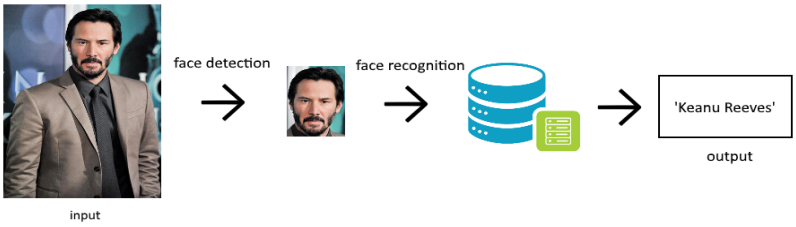


Рисунок 1.1 - Загальний процес автоматизованого розпізнавання обличчя

1.2 Основні складнощі розпізнавання обличчя

Більшість сучасних систем розпізнавання обличчя є досить чутливим до вхідних даних. В цьому розділі розглядаються основні складнощі з якими стикаються ці системи.

1.2.1 Варіації пози на зображенні

Не фронтальна щодо обличчя точка огляду камери значно ускладнює роботу систем розпізнавання обличчя, через те що з’являється велика кількість можливих варіацій, що проілюстровані на рис. 1.2, а також призводить до суттєвих змін у вигляді або формі обличчя.

Рисунок 1.2 – Деякі із можливих варіацій руху голови: а – вниз; б – вбік

1.2.2 Наявність елементів, які змінюють вигляд обличчя

Макіяж, зачіска, борода, бейсбольна кепка, окуляри – це лише декілька з тих елементів, які є причиною різноманітності вигляду одного й того ж обличчя. Інколи за наявності чи відсутності певних елементів навіть людина не здатна розпізнати одну й ту саму людину. Приклад проілюстрований на рис. 1.3.

Рисунок 1.3 – Деякі із можливих елементів, які змінюють вигляд обличчя:
а – наявність бороди; б – відсутність бороди, в – наявність окулярів

1.2.3 Зміни емоцій на обличчі

Людське обличчя здатне виражати широкий спектр різноманітних емоцій. Загалом, людські вирази обличчя можуть виражати злість, страх, сум, радість, здивування та ін. Емоції, які створюють видимий рух обличчя теж ускладнюють розпізнавання. Приклади руху обличчя наведені на рис. 1.4.

Рисунок 1.4 – Приклади руху обличчя через емоції: а – злість; б – радість, в – здивування

1.2.4 Старіння обличчя

Старіння людського обличчя – ще одна причина зміни вигляду обличчя, що може мати вплив на процес автоматичного розпізнавання облич, проте ця проблема виникає тільки якщо час між зображеннями обличчя однієї людини є суттєвим.

Рисунок 1.5 – Ілюстрація зміни вигляду обличчя через старіння

1.2.5 Варіації умов освітлення

Рівень освітленості може суттєво знизити якість роботи систем автоматизованого розпізнавання облич. Тіні на обличчі значно погіршують локалізацію та розпізнавання обличчя, так само як і занадто яскраве освітлення.

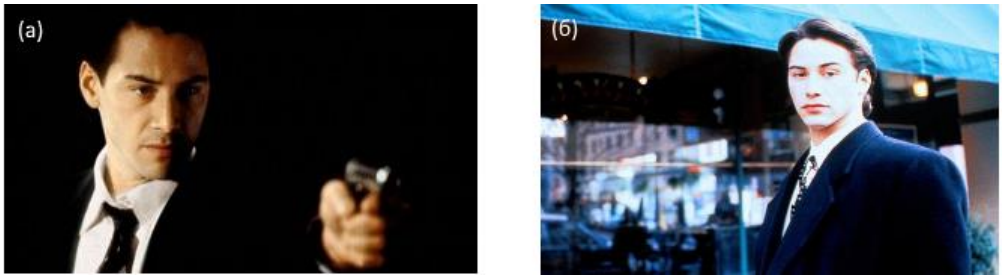


Рисунок 1.6 – Ілюстрація зображення обличчя з різним рівнем освітлення:

а – частково затемненим; б – освітленим

1.2.6 Роздільна здатність зображення

Ще одним з дуже важливих факторів, що впливають на якість автоматизованого розпізнавання облич є якість та роздільна здатність зображення обличчя. Налаштування камери теж можуть впливати на локалізацію та розпізнавання обличчя на зображені.

1.3 Огляд сучасного методу розпізнавання облич з глибоким навчанням

У цій кваліфікаційній роботі для розпізнавання обличчя використовується метод, описаний у статті Адама Гейтгейа [5]. Система розпізнавання обличчя з глибоким навчанням в реальному часі ставить перед собою кілька взаємопов'язаних проблем, які можна розділити на етапи:

1. Проаналізувати вхідне зображення і знайти всі обличчя, які знаходяться на ньому.
2. Далі потрібно сфокусуватись на кожному знайденому обличчю, розуміючи, що якщо обличчя особи може бути повернуто в незвичайному напрямку або бути при поганому освітленні, воно все ще належить одній і тій самій людині.
3. Система має вміти виділяти особливі риси обличчя, які допоможуть відрізнити дане обличчя від всіх інших.
4. Після виділення особливих рис обличчя, потрібно порівняти їх з усіма іншими особливими рисами відомих людей, щоб знайти відповідне ім’я.

Рисунок 1.7 - Загальний процес автоматизованого розпізнавання облич, використовуючи глибоке навчання

1.3.1 Локалізація обличчя

Автоматична технологія пошуку обличчя на зображенні вже давно використовується для камер, щоб переконатися, що всі особи знаходяться у фокусі, перш ніж зробити знімок. У розпізнаванні обличчя ця технологія використовується для того, щоб виділити область, на якій згодом шукаються особливі риси обличчя.

Для пошуку обличчя в цій кваліфікаційній роботі буде використовуватися метод HOG. Спочатку зображення, на якому шукаємо обличчя, потрібно зробити чорно-білим, даний метод не потребує кольору.

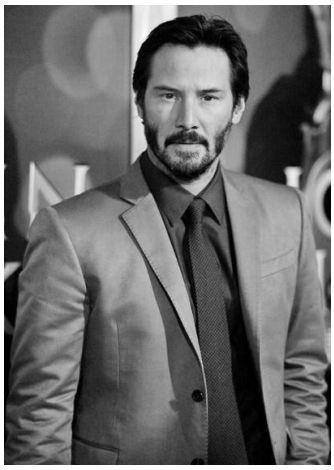


Рисунок 1.8 – Чорно-біле зображення на якому потрібно знайти обличчя

Потім по одному розглядається кожен піксель нашого зображення. Для кожного окремого пікселя дивляться на всі пікселі, які оточують його. Це робиться для того, щоб з'ясувати, наскільки темний піксель порівнюється з пікселями, які безпосередньо оточують його.

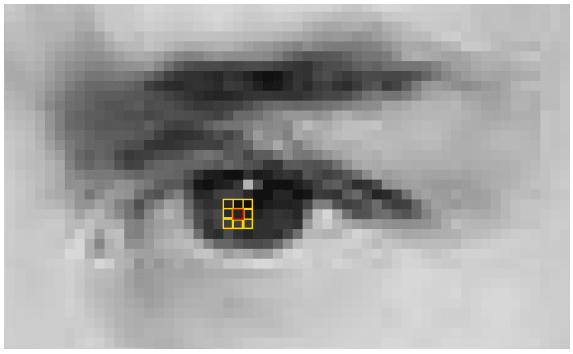


Рисунок 1.9 – Приклад порівняння пікселів із тими, що його оточують

Потім малюється стрілка, в якій стороні зображення стає темнішим, як це зображено на рис.1.10. Дивлячись на цей піксель і пікселі, що оточують його, зображення стає темнішим у верхньому правому куті.

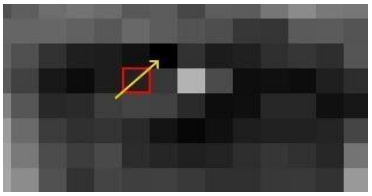


Рисунок 1.10 – Ілюстрація малювання стрілки переходу до темних пікселів

Якщо повторити цей процес для кожного окремого пікселя на зображенні, ви отримаєте заміну кожного пікселя стрілкою. Ці стрілки називаються градієнтами і показують потік від світлого до темного по всьому зображенню, незалежно від рівня яскравості зображення.

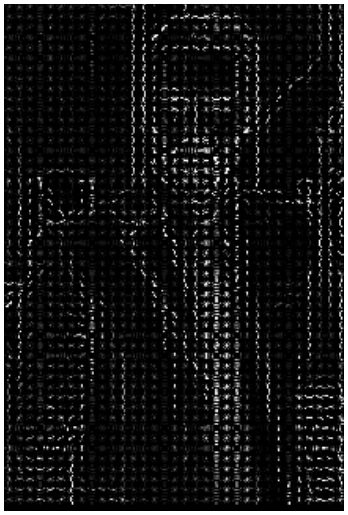


Рисунок 1.11 – Ілюстрація малювання стрілки переходу до темних пікселів для всіх пікселів на зображенні

Проте, збереження градієнта для кожного пікселя дає занадто багато деталей. Було би краще, якби ми могли бачити основний потік світла чи темряви на більш високому рівні, для того щоб бачити основну картину зображення. Для цього далі зображення розбивається на невеликі квадрати розміром 16x16 пікселів. У

кожному квадраті підраховується кількість точок градієнтів у кожному з основних напрямків (скільки точок вгору, точку вгору, праворуч тощо…). Після цього цей квадрат замінюється на зображення стрілками, які були найвиразнішими. Кінцевим результатом є перетворення оригінального зображення на дуже просте представлення, яке простим чином фіксує основну структуру обличчя. Щоб знайти обличчя в цьому зображенні, потрібно знайти частину нашого зображення, яка виглядає найбільш схожою на відомий шаблон HOG (рис. 1.12), витягнутий з набору облич інших осіб. Результат роботи цього методу наведений на рис. 1.13.

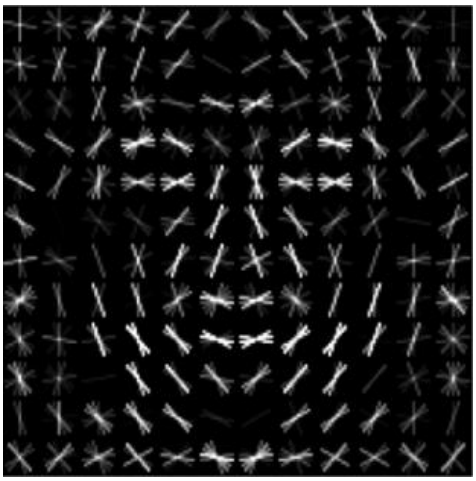


Рисунок 1.12 – Ілюстрація відомого шаблону HOG, витягнутого з набору облич інших осіб

Рисунок 1.13 – Ілюстрація результату роботи пошуку обличчя, використовуючи метод HOG

1.3.2 Вирівнювання та проектування облич

Один з можливих способів вирішення проблема варіації пози обличчя на зображенні, яка більш детально описана в розділі 1.2.1, це деформувати зображення обличчя таким чином, щоб очі і губи завжди знаходилися в місці вибірки на зображенні. Це значно полегшить порівняння облич на наступних етапах. Для цього використаємо алгоритм, що називається оцінкою орієнтирів обличчя. Є багато способів зробити це, але в цій кваліфікаційній роботі використовується підхід, винайдений у 2014 році Вахідом Каземі та Жозефіною Салліван [6].

Основна ідея полягає в тому, щоб зафіксувати 68 конкретних орієнтирів, проілюстрованих на рис. 1.14 та рис. 1.15, які існують на кожному обличчі: верхній частині підборіддя, зовнішній край кожного ока, внутрішній край кожної брови тощо.

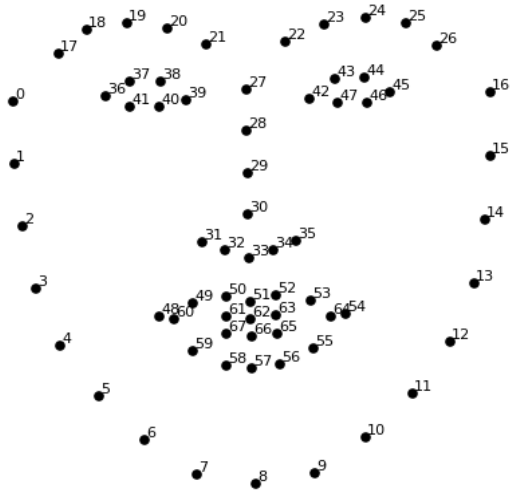


Рисунок 1.14 – Ілюстрація 68 орієнтирів, які існують на кожному обличчі

Рисунок 1.15 – Ілюстрація 68 орієнтирів, які існують на кожному обличчі

Далі, коли на зображені стає відомо де знаходяться ці 68 орієнтирів, використовуючи обертання та масштаб, зображення повертається, масштабується і зсувається так, щоб вирівняти позицію обличчя.

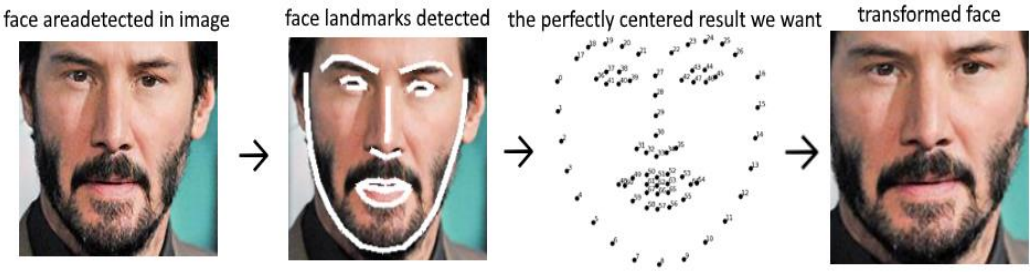


Рисунок 1.16 – Ілюстрація процесу вирівнювання позиції обличчя

1.3.3 Виділення особливих рис обличчя

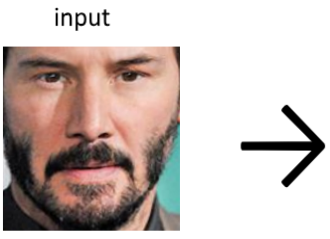
Підхід до розпізнавання облич полягає в тому, щоб порівнювати невідоме обличчя з обличчями усіх відомих для системи людей. Проте, якщо порівнювати 68 особливих рис невідомого обличчя з усіма відомими, це буде займати дуже багато часу, що не підходить до розпізнавання у реальному часі. Тому, застосовується спосіб, який виділяє кілька основних вимірювань з кожного обличчя. І після цього ці виміри порівнюються з особливими вимірами відомих для системи людей.

Дослідники виявили, що найточніший підхід полягає в тому, щоб, використовуючи глибоке навчання, дозволити комп'ютеру самому визначити які

вимірювання виділяти. Для цього потрібно навчити DCNN генерувати 128 вимірювань для кожного обличчя. Процес навчання розглядає 3 зображення обличчя одночасно: зображення відомої особи, інша фотографія тієї ж відомої особи та зображення невідомої особи.

Після цього алгоритм розглядає вимірювання, які він створює для кожного з цих трьох зображень. Потім вона трохи змінює нейронну мережу так, що вона гарантує, що вимірювання, які він генерує для першого фото відомої особи та другого фото тієї ж особи, співпадають більше, ніж вимірювання для другого фото відомої особи та фото невідомої особи.

Після повторення цього кроку мільйони разів для мільйонів зображень тисяч різних людей, нейронна мережа навчається генерувати надійні 128 вимірювань для кожної людини. Будь-які десять різних зображень однієї й тієї ж людини повинні дати приблизно такі ж вимірювання. Точний підхід для облич, який використовується в цій кваліфікаційній роботі, був розроблений в 2015 році дослідниками в Google , але існує багато схожих підходів.



128 Measurements Generated from Image

```
[-0.16934587  0.21119267  0.10881166 -0.00536802 -0.11413234 -0.00457576
 -0.05950853 -0.07121757  0.1464811  -0.05931316  0.31608611 -0.01693914
 -0.19227886 -0.0904272   0.00648136  0.13751175 -0.18110961 -0.05677556
 -0.04922068 -0.05608     0.10121522  0.02982391  0.044466    0.01970738
 -0.12244707 -0.36124352 -0.01270123 -0.12319976  0.13193756 -0.14588788
 -0.06098066 -0.05047747 -0.15855591 -0.03881312 -0.02228092  0.00088748
 -0.01171407 -0.13979995  0.16304959 -0.11054929 -0.18604958  0.00426564
  0.03990703  0.21193111  0.17865188 -0.07688195  0.03288184 -0.13892622
  0.1507431  -0.17455831  0.08597957  0.13423035  0.0594057   0.06197733
  0.12902085 -0.10602202  0.00479885  0.12684555 -0.19864519  0.01910157
  0.0816036   0.05344702 -0.0270997  -0.08730817  0.13898467  0.12609677
 -0.11780691 -0.18586333  0.04576039 -0.12693745 -0.04853077  0.07150833
 -0.09423302 -0.19472256 -0.28574505  0.04978726  0.32757476  0.16427727
 -0.25899389  0.01218985 -0.06745676 -0.09558664  0.11723085  0.12324563
 -0.02655697 -0.03606658 -0.11177334  0.03622038  0.14714807 -0.00228493
 -0.0132417   0.26546451  0.03645682 -0.04191261  0.00178798  0.13206264
 -0.19006455  0.00747772 -0.08279227 -0.03258264  0.05175696 -0.04398829
 -0.06583679  0.1257707  -0.19852726  0.1684498   0.01209629  0.04525413
  0.02384871 -0.06081514 -0.02585991 -0.01053698  0.06997328 -0.22386761
  0.19034445  0.10554815  0.05312395  0.16086845  0.05609111  0.03036838
  0.03242138 -0.03443946 -0.12684828 -0.06005421  0.00891294 -0.0598507
 -0.00361058  0.0104748 ]
```

Рисунок 1.17 – Приклад 128 вимірювань для обличчя

1.3.4 Порівняння облич

Все, що потрібно зробити для порівняння облич, це знайти людину в нашій базі даних відомих людей, обличчя якого має найближчі вимірювання до вимірювань обличчя вхідного зображення.

Це можна зробити за допомогою будь-якого базового алгоритму класифікації машинного навчання. В цій кваліфікаційній роботі буде використовуватись простий лінійний класифікатор SVM, але багато інших алгоритмів класифікації теж можуть бути застосовані для цього. Все, що потрібно зробити, це навчити класифікатор, який приймає вимірювання з нового зображення, і на вихід дає ім’я особи, яка є найбільш близькою.

Один із найпростіших способів сфальсифікувати дані для системи розпізнавання обличчя – використати фото або відео на якому присутнє обличчя потрібної людини. На сьогодні, в еру соціальних мереж, можна без складнощів знайти фото- відеоматеріал будь-якої людини, крім окремих випадків. Бо навіть, якщо людина не викладає в мережу матеріали зі своїм обличчям, це не означає, що хтось не може викласти якийсь матеріал в якому присутнє обличчя цієї людини. Більш того, обличчя людини може бути сфотографовано навіть без її відома, наприклад, під час прогулянки на вулиці.

Після того, як фото- відеоматеріал обличчя людини потрапляє до шахрая, все, що йому залишається зробити – показати перед камерою роздруковане обличчя потрібної людини чи просто фото цього обличчя зі свого смартфону.

РОЗДІЛ 2 МЕТОДИ ЗАХИСТУ ВІД ЗЛОВЖИВАННЯ СИСТЕМ РОЗПІЗНАВАННЯ ОБЛИЧЧЯ

2.1 Способи зловживання систем розпізнавання обличчя

Спуфінг – в контексті безпеки мережі, це випадок, коли особа або програма маскується під іншу за допомогою фальсифікації даних, і тим самим отримує незаконну перевагу. У випадку з системою розпізнавання обличчя існує не один спосіб спуфінгу.

2.1.1 Фото- та відеоматеріали з обличчям

Один із найпростіших способів сфальсифікувати дані для системи розпізнавання обличчя – використати фото або відео на якому присутнє обличчя потрібної людини. На сьогодні, в еру соціальних мереж, можна без складнощів знайти фото- відеоматеріал будь-якої людини, крім окремих випадків. Бо навіть, якщо людина не викладає в мережу матеріали зі своїм обличчям, це не означає, що хтось не може викласти якийсь матеріал в якому присутнє обличчя цієї людини. Більш того, обличчя людини може бути сфотографовано навіть без її відома, наприклад, під час прогулянки на вулиці.

Після того, як фото- відеоматеріал обличчя людини потрапляє до шахрая, все, що йому залишається зробити – показати перед камерою роздруковане обличчя потрібної людини чи просто фото цього обличчя зі свого смартфону.

2.1.2 3D-модель обличчя

Більш складний спосіб у реалізації – створення 3D-моделі обличчя. Ще більше полегшив у реалізації даного способу той факт, що фахівці з британських університетів Ноттінгема та Кінгстона розробили нейромережу, яка створює 3D-модель обличчя з однієї фотографії.

2.1.3 Інфрачервоні світлодіоди

Дослідники з Китаю створили бейсбольну кепку, на якій встановлені мініатюрні інфрачервоні світлодіоди, які розміщені таким чином, що інфрачервоні промені, які падають на обличчя власника головного убору, допомагають не тільки приховати його особистість, а й видати себе за іншу людину для проходження заснованої на розпізнаванні особи аутентифікації. Даний спосіб більш складний і вимагає використання глибокої нейронної мережі для розпізнавання статичного зображення особи і правильного проектування інфрачервоних променів на обличчя самозванця.

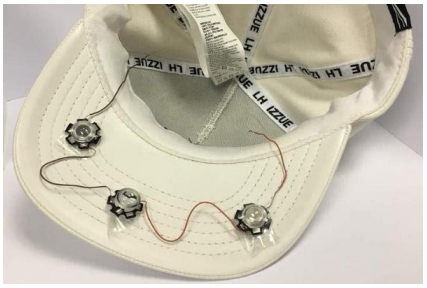


Рисунок 2.1 – Ілюстрація бейсбольної кепки, на якій встановлені мініатюрні інфрачервоні світлодіоди

2.2 Огляд деяких існуючих методів захисту від зловживання систем розпізнавання обличчя

2.2.1 Відслідковування моргання

Відслідковування моргання робить неефективним метод спуфінгу, який використовує фотоматеріали для фальсифікації обличчя. Оскільки саме цей спосіб впроваджений в програму з 3 розділу цієї кваліфікаційної роботи, розглянемо його більш детально.

Існує декілька способів відслідковування моргання, але в цій кваліфікаційній роботі розглядається метод із використанням співвідношенням сторін очей. Для цього слід локалізувати зображенні обличчя очей. Це не складно зробити, виділивши точки, які утворюють очі, з 68 зафіксованих точок, які є на кожному обличчі з розділу 1.3.2 цієї кваліфікаційної роботи.

Кожне око представлено 6-ю координатами вигляду (x, y), починаючи з лівого кута ока, а потім за годинниковою стрілкою навколо іншої частини області, як це проілюстровано на рис. 2.2.

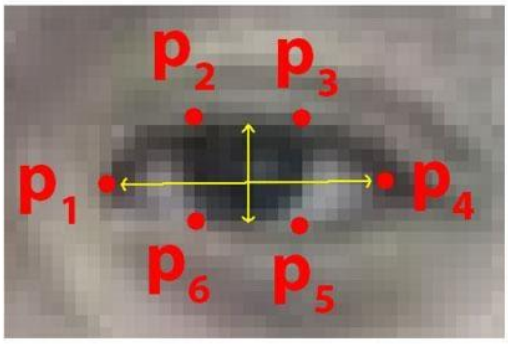


Рисунок 2.2 – Ілюстрація 6-ти орієнтирів обличчя пов’язаних з оком

Дивлячись на рис. 2.2, можна зробити висновок, що існує зв'язок між шириною і висотою цих координат. Ґрунтуючись на роботі Соукупової і Чеха в їхній статті 2016 року «Real-Time Eye Blink Detection using Facial Landmarks» [8], можна вивести рівняння (1), яке відображає це співвідношення, яке називається співвідношенням сторін очей (EAR).

$$EAR = \frac{||p_2 - p_6|| + ||p_3 - p_5||}{2||p_1 - p_4||} \quad (1)$$

Чисельник цього рівняння обчислює відстань між вертикальними орієнтирами очей, а знаменник обчислює відстань між горизонтальними орієнтирами очей. Знаменник подвоюється, оскільки є тільки один набір горизонтальних точок, але два набори вертикальнихточок. Під час моргання значення цього рівняння швидко падає до нуля, що дозволяє відслідковувати, коли людина блимає.

2.2.2 Виявлення рухливості обличчя

Спосіб виявлення рухливості обличчя звертає увагу на мікро-рухи окремих елементів обличчя, наприклад, губ та очей, що дозволяє відрізняти справжнє обличчя від обличчя на фото- відеоматеріалах. Зазвичай, для реалізації цього способу створюється класифікатор, для навчання якого потрібні два набори даних: зображення справжнього обличчя та фейкового людей які зареєстровані у системі розпізнавання облич. Цей метод не є дуже надійним та ефективний тільки проти фотографій та 2D-масок. Серед переваг варто виділити те, що цей метод здатний ефективно працювати при зміні умов реєстрації шаблону, наприклад: освітлення, шуми, якість зображення.

2.2.3 Відслідковування пульсу

Цей метод був розроблений в NASA для визначення частоти пульсу, використовуючи підключену камеру. В моменти скорочення серцевої мишці, сосуди людини наповнюються кров’ю. Завдяки цьому змінюється спектр відбитків від обличчя світла. Зміна цього спектру невидима для людського ока, проте, видима для відеокамери. Цю технологію можна використовувати для того, щоб відрізняти справжнє обличчя від фейку, через те, що на фото- відео матеріалах спектр відбитків світла від обличчя відсутній. Однак, цей метод дуже чутливий до світла та рухів обличчя.

2.2.4 Аналіз текстури

Основною ідеєю цього методу, задача якого запобігти спуфінг-атаці системи розпізнавання облич, є аналіз текстури зображень справжніх облич та фейків. Один із способів відобразити відмінності в яскравості і кольоровості, є обчислення LBP-характеристик із різних кольорових просторів та об'єднання всіх LBP в один вектор ознак. Серед переваг даного методу є те, що для його вхідних даних достатньо лише одного зображення та його невелика собівартість. Проте, цей метод вразливий при атаках з відеоматеріалами високої якості.

РОЗДІЛ 3 РОЗРОБКА СИСТЕМИ ІДЕНТИФІКАЦІЇ ДЛЯ ФІКСАЦІЇ ЧАСУ, ВИКОРИСТОВУЮЧИ РОЗПІЗНАВАННЯ ОБЛИЧЧЯ

3.1 Розробка веб частини додатку

Перед тим як розпізнавати обличчя певних людей потрібно, щоб ці люди були зареєстровані в системі, тобто потрібно знати їхні особливі риси обличчя, про які більш детально описані в розділі 1.3.3. Також повинен бути інструмент для аналізу та візуалізації отриманих даних. Тому, спочатку був розроблений MVC проєкт, використовуючи фреймворк Django [9] та СУБД PostgreSQL [10], для реєстрування користувачів в системі, їх адміністрування та візуалізації та аналізу отриманих даних.

3.1.1 Розробка додатку для реєстрації та адміністрування працівників

3.1.1.1 Опис організації інформаційної бази

На рис. 3.1 можна ознайомитись з логічною структурою БД системи.

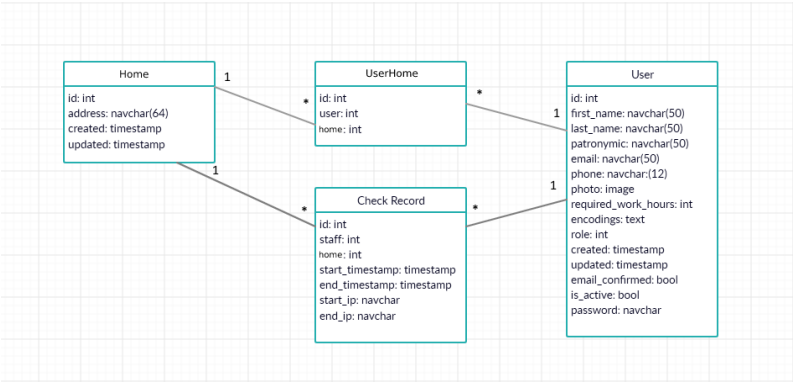


Рисунок 3.1 – ER діаграма структури БД системи

Home - список домівок та інформації щодо них.

User - список користувачів системи інформація щодо них

Check Record - список трек записів користувачів для контролювання часу відвідування

UserHome - таблиця для створення зв’язку “Many to many” між таблицями Home та User

3.1.1.2 Опис таблиць

Опис таблиці Shop наведено в табл. 1.

Таблиця 1. Таблиця Home

id	int	ідентифікатор таблиці
address	navchar(64)	адреса магазину
created	timestamp	дата та час створення екземпляру домівки
updated	timestamp	дата та час оновлення екземпляру домівки

Опис таблиці User наведено в табл. 2.

Таблиця 2. Таблиця User

id	int	ідентифікатор таблиці
first_name	navchar(50)	ім’я
last_name	navchar(50)	прізвище
patronymic	navchar(50)	по батькові користувача
email	navchar(50)	електронна пошта
phone	navchar(12)	номер мобільного телефону
photo	image	фото

required_work_hours	int	кількість годин, які користувач провів у мережі
encodings	text	вектор 128 вимірів для розпізнавання обличчя
role	int	роль користувача
created	timestamp	дата та час створення екземпляру користувача
updated	timestamp	дата та час оновлення екземпляру користувача
email_confirmed	bool	валідність електронної пошти
is_active	bool	статус активності
password	navchar	пароль

Опис таблиці Check Record наведено в табл. 3.

Таблиця 3. Таблиця Check Record

id	int	ідентифікатор таблиці
staff	int	ідентифікатор працівника
home	int	ідентифікатор домівки
start_timestamp	timestamp	дата та час початку роботи програми
end_timestamp	timestamp	дата та час кінця роботи програми
start_ip	navchar(16)	ip адреса з якої отримано start_timestamp
end_ip	navchar(16)	ip адреса з якої отримано end_timestamp

Опис таблиці UserHome наведено в табл. 4.

Таблиця 4. Таблиця UserHome

id	ідентифікатор таблиці
staff	ідентифікатор користувача
home	ідентифікатор домівки

3.1.1.3 Use case діаграми

В даній системі кожен користувач має одну з обраних ролей: працівник, територіальний менеджер або супервізор. Use case діаграми по ролям наведені на рис. 3.2, рис. 3.3, рис. 3.4 відповідно.

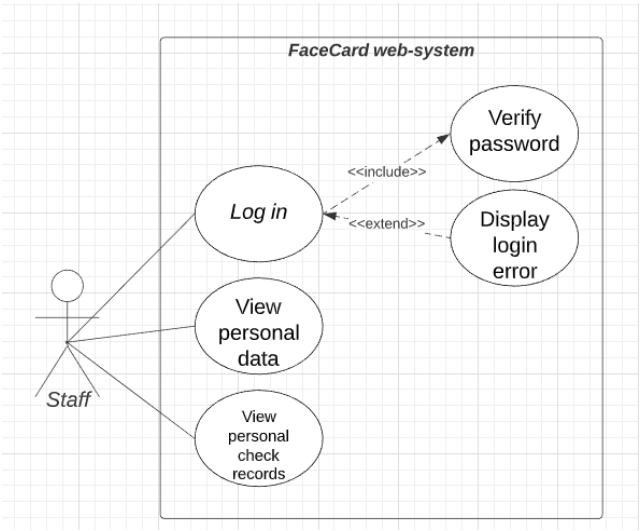


Рисунок 3.2 – Use case діаграма працівника

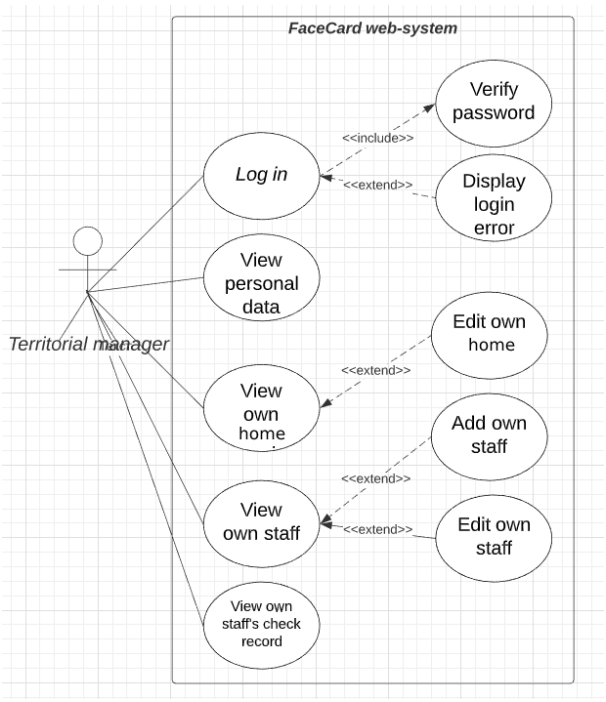


Рисунок 3.3 – Use case діаграма територіального менеджера

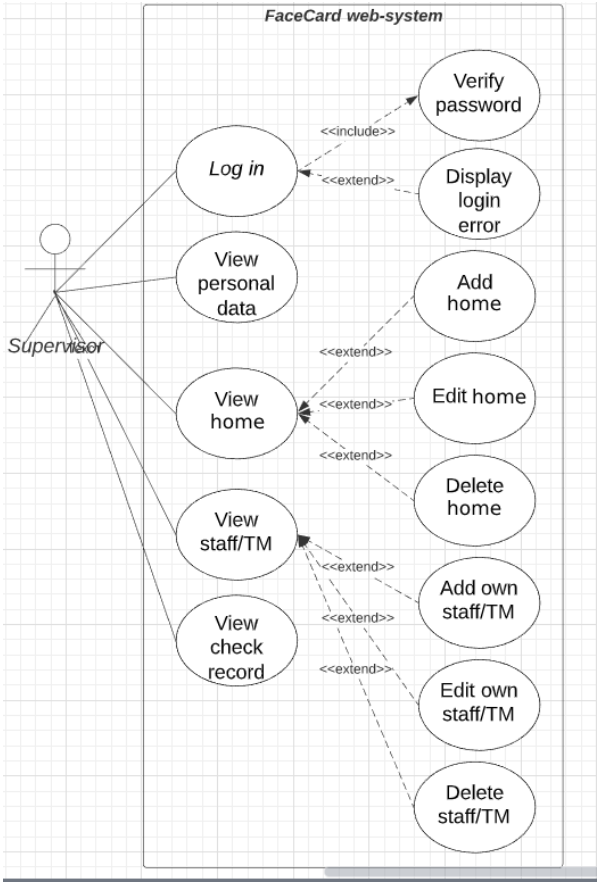


Рисунок 3.4 – Use case діаграма супервізора

3.1.1.4 Опис системи реєстрації та адміністрування

Для того, щоб зареєструватися в системі новий користувач заповнює форму попередньої реєстрації, а саме поле електронної пошти та пароль. Ознайомитись з цією формою можна на рис. 3.5. З формою для входу в систему можна ознайомитись на рис. 3.6

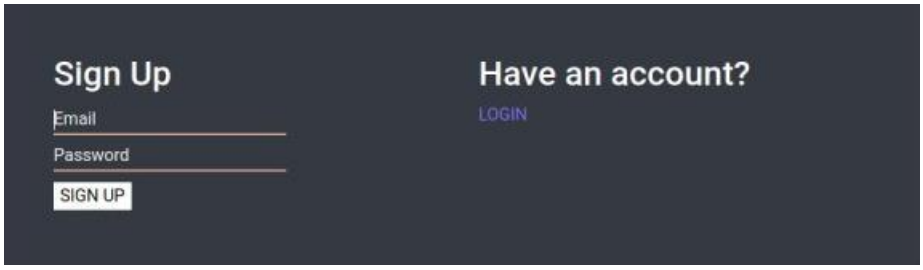


Рисунок 3.5 – Форма попередньої реєстрації

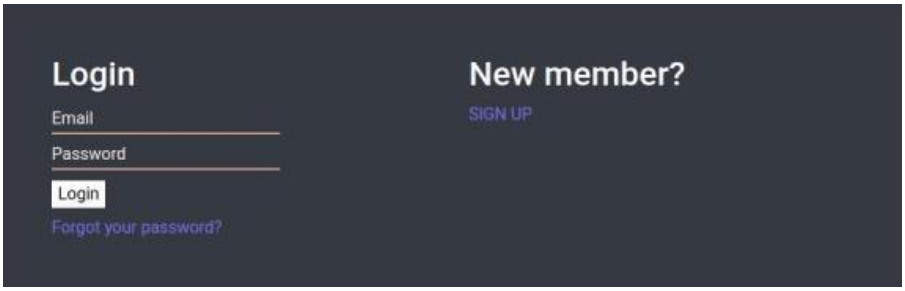


Рисунок 3.6 – Форма входу в систему

Після відправки форми попередньої реєстрації користувачу надсилається повідомлення про підтвердження електронної пошти. Приклад такого повідомлення наведено на рис. 3.7

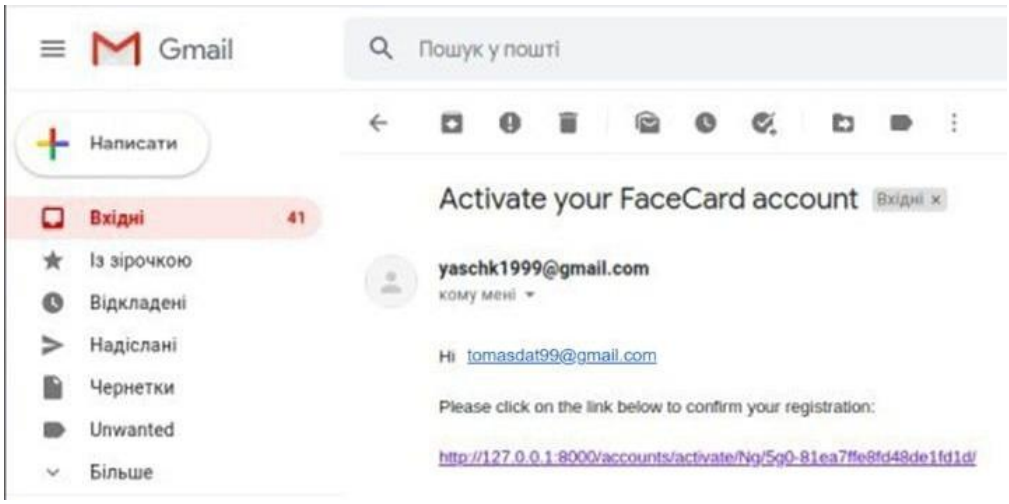


Рисунок 3.7 – Приклад тексту повідомлення про підтвердження електронної пошти

Далі користувач має обрати свою професію (роль). Від цього вибору буде залежати форма реєстрації. Приклад вибору ролі наведено на рис. 3.8. Порівняння форм в залежності від ролі наведено на рис. 3.9

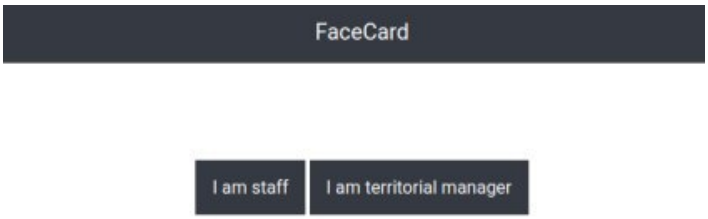


Рисунок 3.8 - Приклад форми вибору ролі

Рисунок 3.9 - Порівняння форм реєстрації в залежності від ролі: а - форма працівника; б - форма територіального менеджера

Користувач повинен обрати один або декілька домівок в яких він проживає. Список домівок формується з активних домівок заданих в панелі адміністратора. Приклад доданої адреси наведено на рис. 3.10. Для ролі користувача після вводу коректної інформації та натиснення кнопки «Зареєструватись», окрім перенесення всіх заповнених даних в БД, з фото будуть обчислені 128-ім вимірів особливих рис обличчя. Які теж зберігаються в довідковій таблиці БД, яка знаходиться на сервері.

Проте, після підтвердження будь-якої форми користувач не може залогінитися в систему, поки територіальний менеджер (в залежності від ролі користувача) не перевірить всі дані та не активує його на сторінці “Admin panel”. Також, в системі є форма відновлення паролю, якщо користувач його забув.

Рисунок 3.10 – Приклад доданої точки підприємства

Однак, якщо користувачів занадто багато додавати їх по одному може зайняти дуже багато часу. Тому також було розроблено програму для реєстрування великої кількості користувачів одразу.

Для того щоб скористатись нею потрібно створити папку під назвою «Staffs» в якій для кожної домівки потрібно створити папку, назва якої це id цієї точки. Далі у кожну створену папку потрібно скинути фото жителів, які проживають в цій домівці. Назва кожного фото повинна бути в такому форматі: «Staff_ID%Name%Surname%.jpg», де Staff_ID – id користувача,

Name – його ім’я та Surname – його прізвище. Варто виділити, що формат фото може бути не обов’язково jpg.

Після запуску програми для кожного фото будуть обчислені 128-ім вимірів особливих рис обличчя та потім повна інформація про працівника надсилається в БД на сервері. Якщо на якомусь фото буде знайдено більше 1 обличчя або не знайдено взагалі, програма не буде додавати до БД цього користувача та виведе на екран ім’я користувача, якому потрібно повторно сфотографуватись. Також якщо користувач вже зареєстрований в БД, то повторно він не буде додаватись. Фото

користувачів зареєстрованих таким чином спеціально не зберігаються в БД. Приклад результату роботи цієї програми наведено на рис. 3.11.

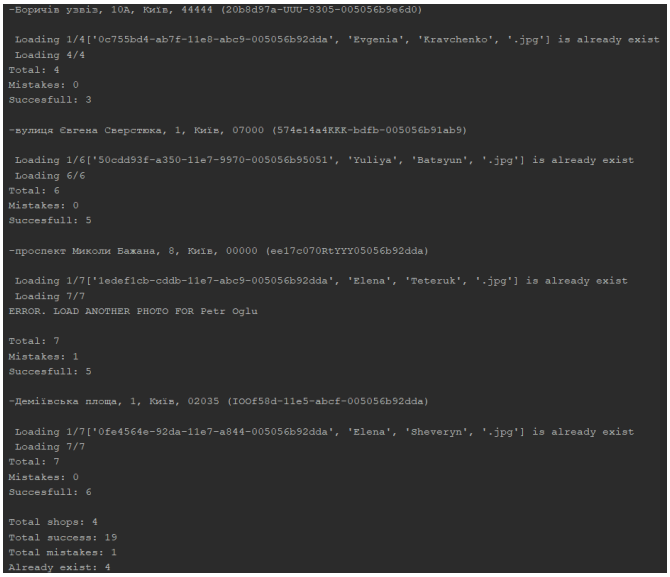


Рисунок 3.11 – Приклад результатів програми для реєстрування користувачів

Всі інструменти для адміністрування знаходяться на сторінці “Admin panel”, яка доступна лише для користувачів з роллю територіального менеджера. Менеджеру доступні абсолютно всі дані про активні домівки, а територіальному менеджеру лише ті, які відносяться до домівок за які він відповідає. Для більшої зручності в систему додано такі фільтри: за адресою домівки, за роллю користувача, за статусом активності користувача. Приклад сторінки “Admin panel” наведено на рис. 3.12.

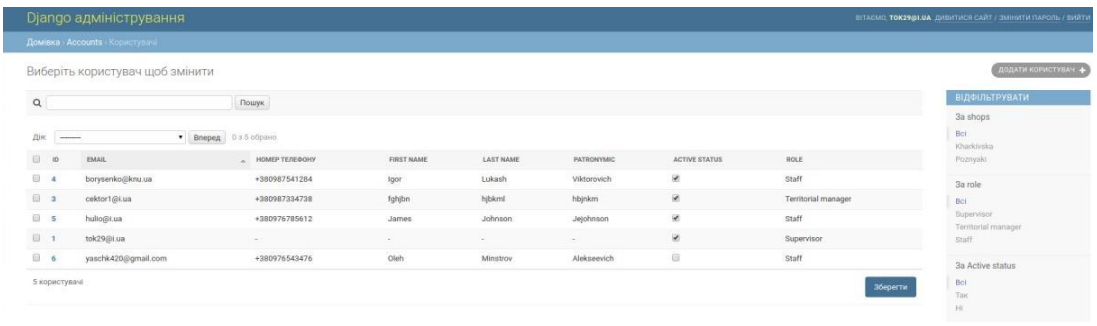


Рисунок 3.12 - Приклад сторінки “Admin panel”

З прикладом запису для відслідковування часу авторизації користувача можна ознайомитися на рис. 3.13. Програма трекінгу часу користувачів за особливим алгоритмом формує відповідний запис з усіма необхідними даними та відправляє його до БД.

ID	STAFF	SHOP	TIMESTAMP	START TIMESTAMP	END TIMESTAMP	START IP ADDRESS	END IP ADDRESS
28	Lukash Igor Viktorovich	Kharkivska	-	20 квітня 2021р. 18:04	20 квітня 2021р. 20:34	188.163.72.251	No connection

Рисунок 3.13 - Приклад запису для відслідковування часу роботи

3.1.2 Візуалізація та аналіз отриманих даних

Активний користувач після залогінення переадресовується на головну сторінку системи “Dashboard”. Відображення цієї сторінки залежить від ролі користувача, а список домівок для фільтрації формується з тих домівок з якими працює користувач. Приклад сторінки “Dashboard” для територіального менеджера та менеджера наведено на рис. 3.14. На цьому рисунку виводяться такі графіки та дані: кількість активних користувачів, найактивніший користувач минулого місяця, стан з’єднання з мережею інтернет, кількість авторизованих годин. Існує можливість фільтрації даних по зв’язаних з користувачем домівок.

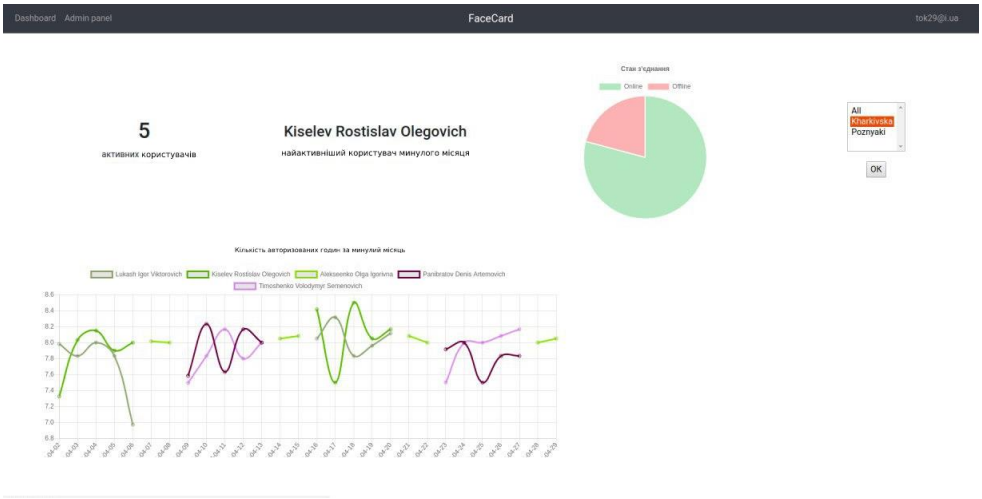


Рисунок 3.14 – Приклад сторінки “Dashboard” для територіального менеджера та менеджера

На рис. 3.15 можна ознайомитися зі сторінкою “Dashboard” для користувача. На цьому рисунку виводяться такі графіки та дані: середній час початку та закінчення авторизації, кількість авторизацій та кількість годин які користувач мав (має) відпрацювати в минулому (поточному) місяці.

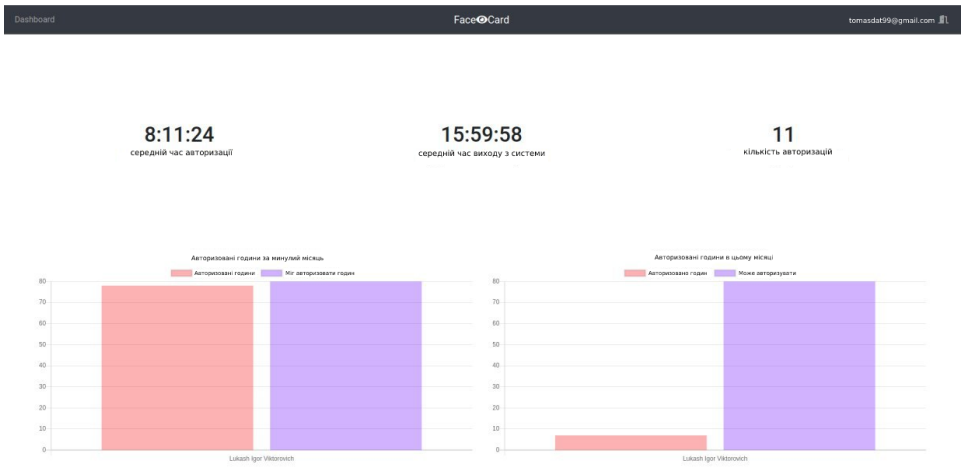


Рисунок 3.15 – Приклад сторінки “Dashboard” для працівника

Також, кожен користувач має сторінку свого профілю. Для переходу на цю сторінку потрібно натиснути на свою електронну пошту в верхній частині сайту. Ця сторінка відрізняється залежно від ролі користувача. На рис. 3.16 наведено приклад сторінки профілю для територіального менеджера (профіль менеджера виглядає так само, окрім ролі). На рис. 3.17 можна ознайомитись з тим, як виглядає сторінка користувача.

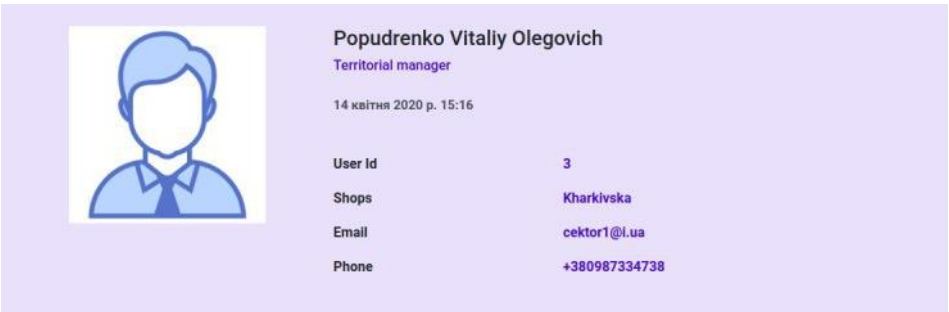


Рисунок 3.16 – Приклад профілю територіального менеджера

Рисунок 3.17 – Приклад профілю працівника

3.2 Розробка декстопної частини додатку

Після того, як БД з інформацією про користувачів готова, можна переходити до розробки додатку ідентифікації та автентифікації, використовуючи розпізнавання облич.

3.2.1 Загальний алгоритм десктопної частини додатку

Головна ідея цього додатку полягає в тому щоб створити програму, яка в режимі реального часу дозволяє фіксувати кількість авторизованих годин кожного користувача в його домівці. Загальний алгоритм програми проілюстрований на рис. 3.18.

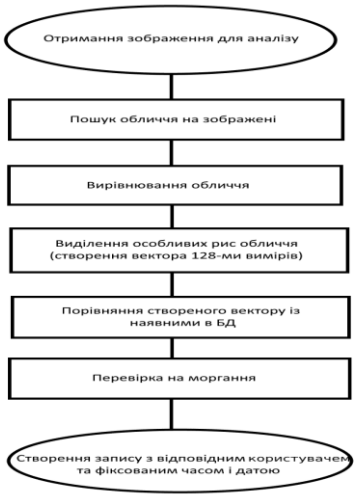


Рисунок 3.18 – Загальний алгоритм роботи програми

3.2.2 Система розпізнавання облич для фіксації часу

У частині програми, яка розпізнає обличчя, було вирішено використовувати бібліотеку «face_recognition» розроблену Адамом Гейтгейом [11]. Оскільки, для методів цієї бібліотеки для розпізнавання обличчя людини достатньо одного її зображення. Детальніше з методами розпізнавання обличчя, які використовуються в цій бібліотеці можна ознайомитись в розділі 1.3 цієї роботи.

Процес обробки зображення для розпізнавання обличчя був пришвидшений декількома способами:

1. Під час аналізу зображення з відеокамери воно зменшується в 4 рази. Це значно покращує швидкість роботи розпізнавання та майже не впливає на якість самого розпізнавання.

2. Викидання кожного n-го кадру з обробки. Число n задається в файлі налаштування додатку. Якщо не обробляти кожен кадр це також значно пришвидшує роботу програми.

3. Якщо обчислювати особливі характеристики обличчя зареєстрованих людей під час реєстрації та зберегти їх в БД, а не робити це під час запуску, то це помітно пришвидшує запуск самого додатку.

Для захисту від зловживання системою ідентифікації з розпізнаванням обличчя, до додатку була додана можливість увімкнути перевірку на моргання.

Також було передбачено можливі проблеми з підключенням до мережі Інтернет, через що було додано локальне сховище, а саме БД Sqlite3 [12]. Під час першого запуску програми, дані про співробітників цієї торгової точки зберігаються в локальному сховищі та оновлюються під час кожного запуску з підключеною мережею Інтернет, що дає можливість співробітникам бути розпізнаними системою, незважаючи на підключення до мережі Інтернету. Приклад збережених даних наведено на рис. 3.19.

	staff_id	full_name	encodings
1	4	Igor Lukash Viktorovich	[-0.05374091863632202, 0.003101664362475276, 0.060246679931879044, 0.0071317558176...
2	7	Volodymyr Timoshenko Semenovich	[-0.10834459215402603, 0.04603935778141022, 0.0011997763067483902, -0.040578778833...
3	9	Denis Panibratov Artemovich	[-0.12939496338367462, 0.18845589458942413, 0.15706050395965576, -0.01917357929050...
4	10	Olga Alekseenko Igorivna	[-0.006843926385045052, 0.1397002786397934, 0.05359087884426117, -0.02294782735407...

Рисунок 3.19 – Приклад збережених у локальному сховищі даних про співробітників

Запис із часом та даними про користувача збережеться в локальному сховищі та буде надісланий до БД на сервері при наступному запуску з доступом до мережі. Приклад збереженого у локальному сховищі фіксованого часу та інформації про користувача наведено на рис. 3.20. Після відправлення даних на сервер, дані про фіксований час користувача у локальному сховищі очищаються. З інструкцією користувача можна ознайомитись у додатку Б.

	home_id	staff_id	stamp	ip
1	1	4	2020-05-02 10:21:45.605709+00:00	No connection

Рисунок 3.20 – Приклад збереженого у локальному сховищі фіксованого часу та інформації про користувача

3.2.3 Структура файлу налаштування

Оскільки, ця версія додатку використовується для фіксації авторизованого часу користувачів, для того щоб мінімізувати похибку при розпізнаванні, було вирішено зробити так, щоб коли система порівнює обличчя користувача програми з обличчями з БД, вона порівнювала особливі характеристики обличчя тільки з характеристиками тих людей, які працюють саме в цій торговій точці. Адреса торгової точки задається id-кодом цієї точки під ключем “home id” словника “settings” в файлі з налаштуваннями “config.py”.

Ключ “blink status” в цьому словнику відповідає за увімкнення та вимкнення перевірки на моргання та має бути «true» або «false» відповідно. Якщо перевірка на моргання увімкнена, то значення порога моргання задається під ключем “eye ar thresh”, якщо співвідношення сторін очей падає нижче цього порогу, а потім піднімається вище цього порога, то зараховується моргання. А під ключем “eye ar consec frame blinks” записується кількість послідовних кадрів, які повинні відбутись із співвідношенням сторін очей меншим ніж значення порогу моргання для зарахування моргання. Значення ключа “frame counter” відповідає за значення n, при якому система розпізнавання облич обробляє кожен n-ий кадр. Ця функція додана для пришвидшення роботи програми.

Якщо система отримує кілька збігів для однієї і тієї ж людини, можливо, люди на фотографіях виглядають дуже схожими, для більш суворого порівняння осіб потрібно більш низьке значення допуску. В “recognition threshold” записується значення допуску, яке знаходиться в межах від 0 до 1, більш низьке значення робить порівняння особливих рис обличчя більш строгим.

Значення з ключем “console log” записується значення «true» чи «false», якщо значення параметру «true», то в консоль буде виводитись інформація про роботу додатку, ця функція дуже допомагає при налаштуванні порогу моргання.

Також для системи розпізнавання облич була додана можливість увімкнути пошук облич які знаходяться далі ніж на відстані витягнутої руки, при ввімкненні цієї функції швидкість роботи додатку уповільнюється. Для увімкнення цієї функції потрібно задати значення “true” під ключем «search small faces status».

Під ключем “num jitters” записується значення, яке відповідає значенню кількості перерахування особливих рис обличчя для більш точного результату, а під ключем “total blink(s)” записується значення кількості моргань, які повинен зробити користувач, для того, щоб зафіксувати час у БД.

Приклад текстового файлу налаштувань, який використовувався під час тестування у підприємстві наведено на рис. 3.21.

```
# settings
settings = {
            "shop id": 1,
            "blink status": False,
            "frame counter": 1,
            "console log": False,
            "recognition threshold": 0.6,
            "search small faces status": False,
            "num jitters": 2,
            "total blink(s)": 2,
            "eye ar thresh": 0.24,
            "eye ar consec frame blinks": 2,
            "time wait": 5,
        }
```

Рисунок 3.21 – Приклад вигляду файлу з налаштуваннями, які використовувались під час тестування

ВИСНОВКИ

Автоматичне розпізнавання обличчя на сьогодні є досить актуальною задачею через великий спектр наукових викликів, велику кількість способів застосування у комерційних додатках, у контексті біометрії та питаннях безпеки.

В цій кваліфікаційній роботі було виконано загальний огляд систем ідентифікації та автентифікації обличчя та основних складнощів, які можуть виникнути під час їх розпізнавання.

Було досліджено:

1. Процес автоматизованого розпізнавання обличчя та розглянуто деякі методи автоматизованого розпізнавання обличчя.
2. Методи зловживання системами розпізнавання обличчя.
3. Методи захисту від зловживання систем розпізнавання обличчя.

В результаті роботи, було розроблено:

1) Оптимізовано процес розпізнавання обличчя у реальному часі.
2) Програмний продукт для фіксації часу, коли користувач прийшов та пішов з дому, з використанням розпізнавання обличчя, який не реагує на фотографію обличчя.
3) Програмний продукт для реєстрації в даній системі.
4) Програмний продукт для адміністрування системи та візуалізації отриманих даних

Також було проведене тестування з подальшим впровадженням програмного продукту для фіксації часу, коли користувач прийшов та пішов з дому, з використанням розпізнавання обличчя.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. JetBrains PyCharm 2018.2.3 [Електронний ресурс] – Режим доступу до ресурсу: https://www.jetbrains.com/pycharm/.

2. Python [Електронний ресурс] – Режим доступу до ресурсу: https://www.python.org/.

3. Javascript [Електронний ресурс] – Режим доступу до ресурсу: https://www.javascript.com/

4. Windows [Електронний ресурс] – Режим доступу до ресурсу: https://www.microsoft.com/uk-ua/windows .

5. Geitgey A. Machine Learning is Fun! Part 4: Modern Face Recognition with Deep Learning [Електронний ресурс] / Adam Geitgey – Режим доступу до ресурсу: https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78.

6. Kazemi V. One Millisecond Face Alignment with an Ensemble of Regression Trees [Електронний ресурс] / V. Kazemi, J. Sullivan – Режим доступу до ресурсу: http://www.csc.kth.se/~vahidk/papers/KazemiCVPR14.pdf.

7. Schroff F. FaceNet: A Unified Embedding for Face Recognition and Clustering [Електронний ресурс] / F. Schroff, D. Kalenichenko, J. Philbin – Режим доступу до ресурсу:

https://www.cv-foundation.org/openaccess/content_cvpr_2015/app/1A_089.pdf.

8. Soukupova T. Real-Time Eye Blink Detection using Facial Landmarks [Електронний ресурс] / T. Soukupova, J. Cech – Режим доступу до ресурсу: http://vision.fe.uni-lj.si/cvww2016/proceedings/papers/05.pdf.

9. Django [Електронний ресурс] – Режим доступу до ресурсу: https://www.djangoproject.com/.

10. PostgreSQL [Електронний ресурс] – Режим доступу до ресурсу: https://www.postgresql.org/.

11. Geitgey A. Face Recognition [Електронний ресурс] / Adam Geitgey – Режим доступу до ресурсу: https://github.com/ageitgey/face_recognition.

12. Sqlite3 [Електронний ресурс] – Режим доступу до ресурсу: https://docs.python.org/3/library/sqlite3.html.

13. Савельєв Ю. Д. Використання конвуляційних нейронних мереж для генерації структурної моделі обличчя [Електронний ресурс] / Ю. Д. Савельєв – Режим доступу до ресурсу: http://cad.kpi.ua/attachments/093_2017dm_Savelyev.pdf.

14. Rosebrock A. Eye blink detection with OpenCV, Python, and dlib [Електронний ресурс] / Adrian Rosebrock. – 2017. – Режим доступу до ресурсу: https://www.pyimagesearch.com/2017/04/24/eye-blink-detection-opencv-python-dlib/.

15. Kelly A. Gates. Our Biometric Future: Facial Recognition Technology and the Culture of Surveillance (Critical Cultural Communication) / Kelly A. Gates., 2011.

ДОДАТОК А

Інструкція користувача

Для того, щоб запустити програму, слід двічі натиснути по ярлику програми. Після запуску програми відкриється вікно, в якому транслюється відео з веб-камери.

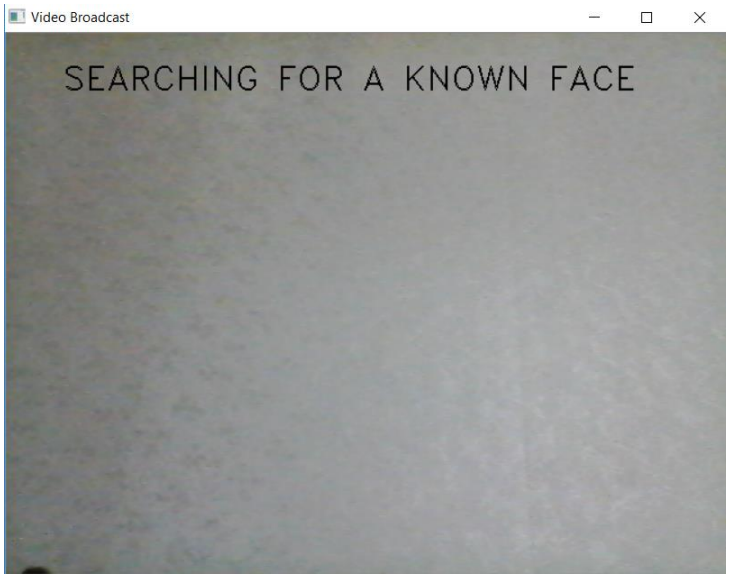


Рисунок А.1 – Ілюстрація вікна програми

Після цього потрібно навести веб-камеру на своє обличчя. Якщо користувач попередньо зареєструвався у системі, він має побачити жовту рамку навколо свого обличчя, та своє ім’я та прізвище, як це проілюстровано на рис. А.2

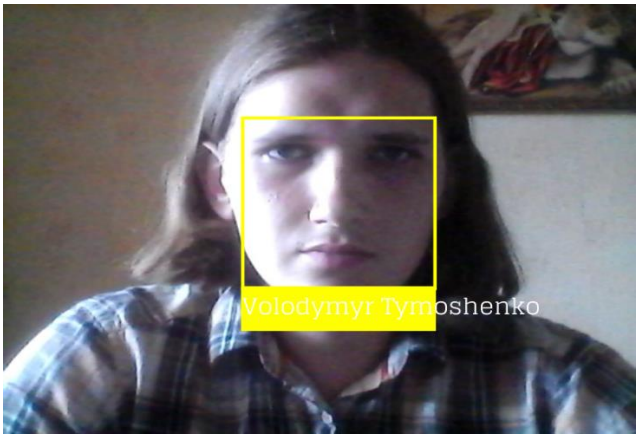


Рисунок А.2 – Ілюстрація розпізнаного обличчя

Далі потрібно трохи зачекати, поки програма перевірить чи справжнє це обличчя, чи фейк. Якщо обличчя справжнє навколо обличчя з’явиться зелена рамка, це означає, що людина зареєструвалась у системі. Приклад цього наведений на рис. А.3.

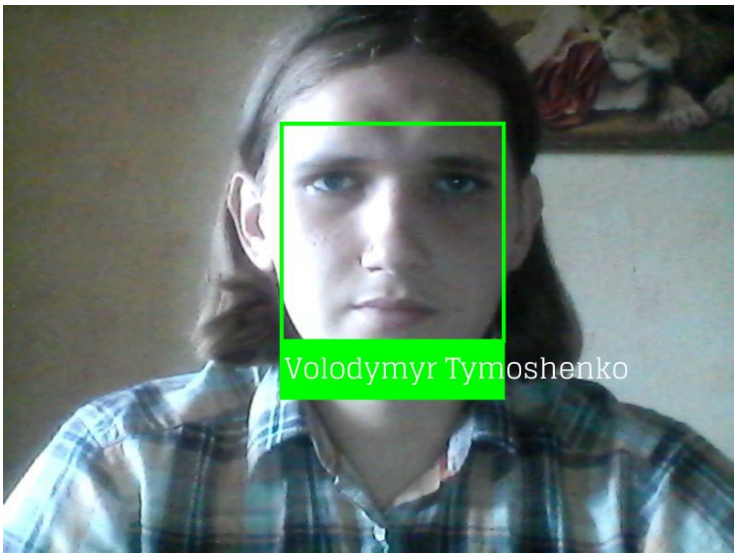


Рисунок А.3 – Ілюстрація про повідомлення про успішну авторизацію

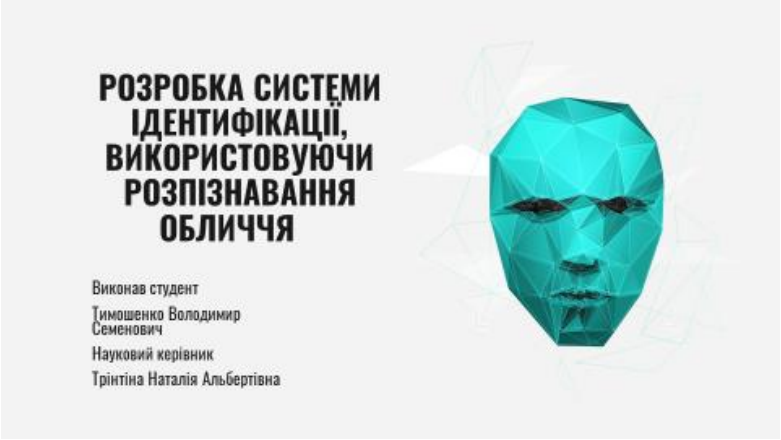


Рисунок А.4 – титульна сторінка презентації

Рисунок А.5 – мета роботи

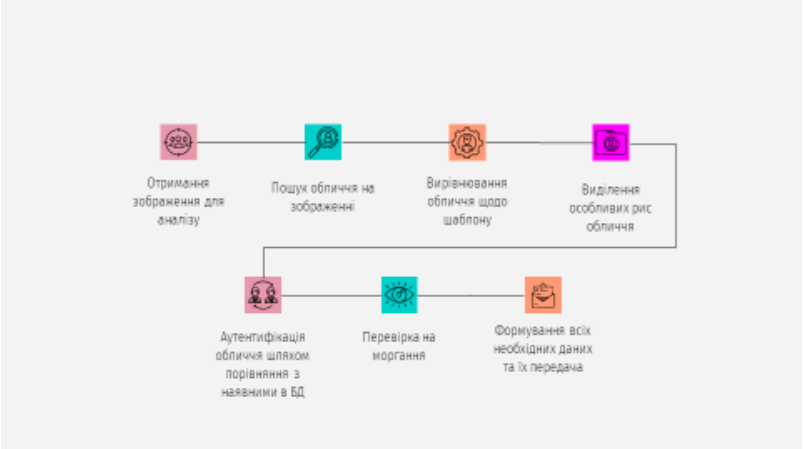


Рисунок А.6 – алгоритм роспізнавання обличчя

Рисунок А.7 – загальна архітектура

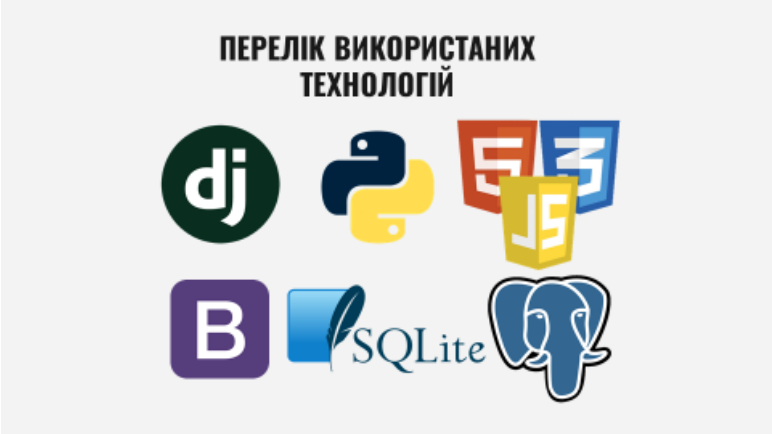


Рисунок А.8 – використані технології

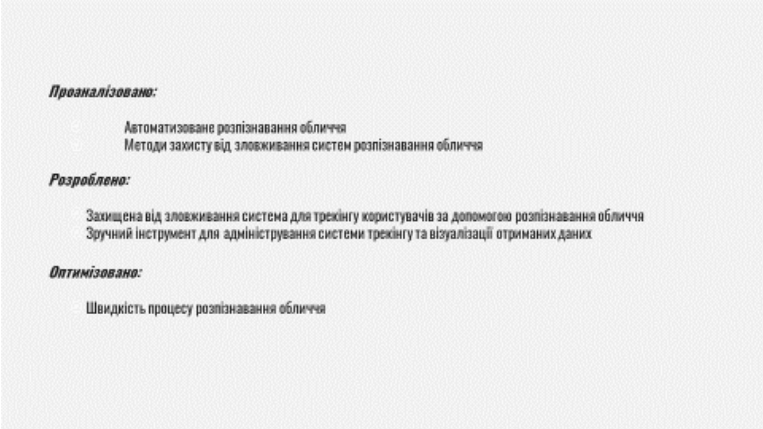


Рисунок А.9 – висновки

Рисунок А.10 – кінець презентації