**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки**

**Кафедра обчислювальної техніки**

До захисту допущено:

Завідувач кафедри

Сергій, СТІРЕНКО

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2020 р.

**Дипломний проєкт**

**на здобуття ступеня бакалавра**

**за освітньо-професійною програмою «Комп'ютерні системи та мережі»**

**спеціальності 123 «Комп'ютерна інженерія»**

**на тему: «Реєстрація користувачів з використанням технології розпізнавання обличчя»**

Виконав:

студент IV курсу, групи ІО-64

Ганжа Денис Віталійович \_\_\_\_\_\_\_\_\_\_

Керівник:

Доцент, кандидат технічних наук

Роковий Олександр Петрович \_\_\_\_\_\_\_\_\_\_

Консультант з нормоконтролю:

Професор кафедри ОТ, д.т.н.,

Сімоненко Валерій Павлович \_\_\_\_\_\_\_\_\_\_

Рецензент:

\_\_\_\_\_\_\_\_\_\_

Засвідчую, що у цьому дипломному проєкті немає запозичень з праць інших авторів без відповідних посилань.

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_

Київ – 2020 року

**Національний технічний університет України**

**«Київський політехнічний інститут імені Ігоря Сікорського»**

**Факультет інформатики та обчислювальної техніки**

**Кафедра обчислювальної техніки**

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 123 «Комп'ютерна інженерія»

Освітньо-професійна програма «Комп'ютерні системи та мережі»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Сергій, СТІРЕНКО

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2020 р.

**ЗАВДАННЯ**

**на дипломний проєкт студенту**

**Ганжі Денису Віталійовичу**

1. Тема проєкту «Реєстрація користувачів з використанням системи розпізнавання обличчя», керівник проєкту Роковий Олександр Петрович, Доцент, кандидат технічних наук, затверджені наказом по університету від «07» травня 2020 р. № 1081-с

2. Термін подання студентом проєкту *26 травня 2020р.*

3. Вихідні дані до проєкту *див. технічне завдання*

4. Зміст пояснювальної записки *Аналіз і характеристика об’єкта проектування, обґрунтування оптимального варіанта реалізації мети цієї роботи, розробка додатку: вибір технологій та їх обґрунтування, основні рішення з реалізації додатку. Висновки.*

5. Перелік графічного матеріалу (із зазначенням обов’язкових креслеників, плакатів, презентацій тощо) *приниципова схема, функціональна схема, структурна схема*

6. Консультанти розділів проєкту

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Розділ | Прізвище, ініціали та посада  консультанта |  | Підпис, дата | |
|  | завдання  видав | завдання прийняв |
| нормоконтроль | Сімоненко В. П., професор, д.т.н. |  |  |  |

7. Дата видачі завдання 01.09.2019

Календарний план

|  |  |  |  |
| --- | --- | --- | --- |
| № з/п | Назва етапів виконання  дипломного проєкту | Термін виконання  етапів проєкту | Примітка |
| *1.* | *Затвердження теми роботи* | *01.09.2019* |  |
| *2.* | *Вивчення та аналіз завдання* | *15.12.2019-15.03.2020* |  |
| *3.* | *Розробка архітектури додатку* | *15.03.2020-25.03.2020* |  |
| *4.* | *Написання програмної частини* | *25.03.2020-05.04.2020* |  |
| *5.* | *Тестування та виправлення помилок* | *05.04.2020-15.04.2020* |  |
| *6.* | *Оформлення пояснювальної записки* | *15.04.2020-20.05.2020* |  |
| *7.* | *Захист програмного продукту* | *25.04.2020* |  |
| *8.* | *Передзахист* | *26.05.2020* |  |
| *9.* | *Захист* | *18.06.2020* |  |

Студент Денис ГАНЖА

Керівник Олександр РОКОВИЙ

**Анотація**

В бакалаврському дипломному проекті розроблено модифікований алгоритм розпізнавання обличчя. З розвитком новітніх технологій, які призводить до автоматизації більшості галузей суспільного життя, реєстрація або авторизації осіб через розпізнання обличчя – не є виключенням із наведеної тези.

Запропонований алгоритм може бути використаний для цільового використання: реєстрації нових та авторизації вже існуючих осіб через систему розпізнавання обличчя у режимі реального часу. Сфера застосування алгоритму велика, від приватного - до комерційного використання.

За основу модифікації було взято алгоритм Local Binary Patterns Histograms (LBPH) з використанням бібліотеки OpenCV. Мова програмування – Python, з урахуванням інтегрованого середовища розробки - Pycharm.

**Аннотация**

В бакалаврском дипломном проекте разработан модифицированный алгоритм распознавания лица. С развитием новейших технологий, которые приводят к автоматизации большинства отраслей общественной жизни, регистрация или авторизации лиц через распознавание лица - не исключение из приведенного тезиса. Предложенный алгоритм может быть использован для целевого использования: регистрации новых и авторизации уже существующих лиц через систему распознавания лиц в режиме реального времени. Сфера применения алгоритма большая, от частного - к коммерческому использованию.

За основу модификации был взят алгоритм Local Binary Patterns Histograms (LBPH) с использованием библиотеки OpenCV. Язык программирования - Python, с учетом интегрированной среды разработки - Pycharm.

**Abstract**

A modified face recognition algorithm has been developed in the bachelor's thesis project. With the development of new technologies, which leads to the automation of most areas of public life, registration or authorization of persons through facial recognition - is no exception to this thesis.

The proposed algorithm can be used for its intended use: registration of new and authorization of existing persons through the face recognition system in real time. The scope of the algorithm is large, from private to commercial use.

The modification was based on the Local Binary Patterns Histograms (LBPH) algorithm using the OpenCV library. Programming language - Python, taking into account the integrated development environment - Pycharm.

**ВІДОМІСТЬ ДИПЛОМНОГО ПРОЄКТУ**

*ДП.6406.01.000 ВП*

*НТУУ «КПІ ім. Ігора Сікорського», ФІОТ*

*Група ІО - 64*

*Літ.*

*Аркуш*

*Дата*

*Підп.*

*№ документа*

*Н.контр.*

*Затв.*

*Зм.*

T

1

1

*Реєстрація користувачів з використанням системи розпізнавання обличчя*

*Відомість дипломного проекту*

# Арк.

# рив

# шв

*Сімоненко В.П.*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № з/п | Формат | Позначення | Найменування | Кількість листів | Примітка |
| 1 | А4 |  | Завдання на дипломний проєкт | 2 |  |
| 2 | А4 | ДП 6406. 02.000 ТЗ | Технічне Завдання | 2 |  |
| 3 | А4 | ДП 6406. 03.000 ПЗ | Пояснювальна записка | 59 |  |
| 4 | А4 | ДП 6406. 04.000 Д1 | Принципова схема | 1 |  |
| 5 | А4 | ДП 6406. 05.000 Д2 | Функціональна схема | 1 |  |
| 6 | А4 | ДП 6406. 06.000 Д3 | Структурна схема | 1 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

*Роковий О.П*

*Ганжа Д.В.*

*Перевір.*

*Розробив*

**Технічне завдання**

**до дипломного проєкту**

на тему «Реєстрація користувачів з використанням системи розпізнавання обличчя»

**Зміст**

[1. Найменування та область застосування………………...... 9](#_Toc514367725)

[2. Підстави для розробки…………………………………………....... 9](#_Toc514367726)

[3. МЕТА та призначення розробки………………………………...... 9](#_Toc514367727)

[4. Джерела розробки…………………………………………………......9](#_Toc514367728)

[5. Технічні вимоги……………………………………………………..... 9](#_Toc514367729)

[5.1. Вимоги до програмного продукту, що розробляється……………….. 9](#_Toc514367730)

[5.2.Вимоги до пристрою, де відбувається тренування моделей…............. 9](#_Toc514367731)

[6. Етапи розробки……………………………………………………....... 10](#_Toc514367732)

**1.НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ**

Найменування: «Реєстрація користувачів з використанням системи розпізнавання обличчя».

Область застосування: програма може використовуватись в різних проектах, як надійний засіб для авторизації

**2.ПІДСТАВИ ДЛЯ РОЗРОБКИ**

Підставою для розробки є завдання на виконання бакалаврського дипломного проекту, затверджене кафедрою обчислювальної техніки Національного технічного університету України «Київський політехнічний інститут ім. Ігоря Сікорського».

**3.МЕТА та призначення розробки**

Метою розробки є створення системи реєстрації користувачів з використанням технології розпізнавання обличчя.

**4.Джерела розробки**

Джерелом розробки є науково-технічна література, публікації в спеціалізованих періодичних виданнях, довідники по платформах дистанційного навчання, публікації в мережі Інтернет по даній темі.

**5.Технічні вимоги**

**5.1. Вимоги до пристрою, де відбувається тренування моделей**

* Мінімальний обсяг оперативної пам'яті - 4 ГБ;
* Розмір сховища - 64 ГБ;
* Частота процесора - 2,4 ГГц;
* Архітектура - 64-бітна.
* Наявність мови програмування Python та бібліотеки OpenCv(з усіма додатками).
* Наявність середовища розробки PyCharm.

**6.** **Етапи розробки**

|  |  |
| --- | --- |
|  | Дата |
| Вивчення необхідної літератури | 21.02.2020 |
| Складання і узгодження технічного завдання | 08.03.2020 |
| Написання вступної частини та огляд рішень | 22.03.2020 |
| Розробка архітектури додатку | 04.04.2020 |
| Написання програмної частини | 11.04.2020 |
| Тестування та виправлення помилок | 04.05.2020 |
| Оформлення документації дипломного проекту | 17.05.2020 |
| Попередній захист та проходження нормативного контролю | 04.06.2020 |
| Захист дипломного проекту | 15.06.2020 |

**Пояснювальна записка**

**до дипломного проєкту**

на тему: «Реєстрація користувачів з використанням системи розпізнавання обличчя»

Київ – 2020 року

**Зміст**

СПИСОК СКОРОЧЕНЬ 13

ВСТУП 14

РОЗДІЛ 1.ОГЛЯД ІСНУЮЧИХ РІШЕНЬ ТА ЇХ АНАЛІЗ 15

1.1 Розвиток програм розпізнавання обличяя 15

1.2 Алгоритм EigenFaces (PCA) 18

1.3 Алгоритм FisherFaces (LDA) 20

1.4 Алгоритм LBPH 22

1.5 Використання бібліотеки OpenCV 24

1.6 Порівняння основних алгоритмів розпізнавання обличчя 26

ВИСНОВКИ ДО РОЗДІЛУ 1 27

РОЗДІЛ 2. РОЗШИРЕНИЙ АЛГОРИТМ LBPH З ВИКОРИСТАННЯМ ДОДАТКОВИХ ФУНКЦІЙ 28

2.1 Метод Сегментації зображення 30

2.2. Штатна обробка фотографій 33

ВИСНОВКИ ДО РОЗДІЛУ 2 39

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТЕХНОЛОГІЇ РОЗПІЗНАВАННЯ ОБЛИЧЧЯ З ЇЇ ПОДАЛЬШИМ УДОСКОНАЛЕННЯМ 40

3.1 Як працює розпізнавання обличчя OpenCV 40

3.2 Огляд проекту 42

3.3 Опис файлів проекту 45

3.4 Тестування реалізованих методів 53

ВИСНОВКИ ДО РОЗДІЛУ 3 57

ВИСНОВКИ 58

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ 59

ДОДАТКИ 61

# СПИСОК СКОРОЧЕНЬ

ImageWare – (система FaceID) - Візуальна технологія.

Vissage Technology – (система Vissage Gallery) - Візуальна технологія.

Visionics – (система FaceIt) - Візуальна технологія.

FRT - (англ. Facial recognition technology) - Технологія розпізнавання обличчя.

ASID – (англ. Automated Suspect Identification System) - Автоматизована Система Ідентифікації Підозрюваних.

NFC – (англ. Near field communication) - Комунікація найближчого поля.

AMD – (англ. Advanced Micro Devices) - Передові мікротехнології.

PCA – (англ. Principal Component Analysis) - Метод головних компонент.

LDA – (англ. linear discriminant Analysis ) - Лінійний Дискримінантний Аналіз.

OpenCV – (англ. Open Source Computer Vision Library) -  Біблиотека комп’ютерного хору з відкритим вихідним кодом.

HOG – (англ. High End Systems) – Операційна система, яка використовує градієнт.

TIFF (англ. Tagged Image File Format) — формат зберігання растрових графічних зображень

**ВСТУП**

Технічний прогрес не стоїть на місці. Кожного дня нині існуючі технології переходять у площину «супер-», або «нано-» технологій. Багаторівнева автоматизація, комп’ютеризація рутинної роботи, автоматичний аналіз великих масивів даних – перелік нескінченний, це все, що вже існує й використовується як для приватних осіб, так і для великих компанії.

За мотивами актуальності, дана тема, як показує практика, є дуже розповсюдженою. Системи розпізнання обличчя обладнуються як на вулицях міст, так і в офісах, навіть деякі відеокамери, в приміщеннях до яких доступ обмежений для фізичних осіб, також обладнуються «real-time» фіксацією та ідентифікацією особи, банківські операції, та багато інших.

Проблематика даної теми складається із визначення підходу для більш кращого, швидшого, а також точного розпізнання об’єкта, його ідентифікація, з подальшим пошуком інформації, як на приватних масивах інформації (баз даних фірми, компанії, контактні книги осіб), так і загальнодоступні бази даних (державні реєстри, доступ до яких не обмежується, соціальні сервіси – Google, Facebook, Instagram, тощо).

**РОЗДІЛ 1**

**ОГЛЯД ІСНУЮЧИХ РІШЕНЬ РЕЄСТРАЦІЇ КОРИСТУВАЧІВ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ РОЗПІЗНАВАННЯ ОБЛИЧЧЯ**

* 1. **Розвиток програм розпізнавання обличчя.**

Технологія розпізнавання обличчя виходить ще з 1960-х років. Першою людиною, яка презентувала такий спосіб був Вудро Вілсон Бледсоу. У фундаменті методу він виклав аналіз через так звані «лінії сітки». Даний метод був першим, а отже і достатньо примітивним та простим. Сутність методу полягала у відображені точок (особливостей) людиною-виконавцем розпізнавання на суб’єкті відображання. Далі, після знаходження та позначення таких точок, інформацію слід було занести до комп’ютера.Як вбачається, така технологія не була дуже швидкою у використанні. Загалом, в проміжку на одну годину, людина-виконавець могла розпізнати близько 40 осіб (обличчя), це була значна кількість, навіть для перших програм такого типу.

Наступною провідною ланкою, після 1960-х років, стала підтримка алгоритму Бледсоу у Стенфордському науково-дослідному інституті. Виявилася тенденція, з огляду на зручність та точності розпізнавання. Нова технологія справлялась краще людей, тобто був запропонований новий метод в якому ланку - людини-виконавця було або зменшено, або видалено взагалі. Вінцем 20-го сторіччя була новітня модель, яку віднайшли та покращили із попередніх версій у Рурському університеті в Бохумі. Нова технологія могла похизуватись своєю точністю, яка відразу знайшла своє місце у системах банківських установ, а також у повітряному ремеслі. Саме з того моменту увесь ринок технологій розпізнавання обличчя почав своє становлення та розвиток у більш глобальному масштабі. Також, при популяризації новітніх технологій розпізнаванню обличчя уряд США ініціював так зване «оцінювання» у проміжку між 1993 по 2010 року. Результати оцінювання вражали: загальний перебіг помилок з найперших програм, то останніх (на той момент часу) знизився у більше ніж у 270 разів.

Так як історична ідея була цілком зрозуміла, у цій роботі проведемо огляд запропонованих дефініцій терміну «розпізнання обличчя», для подальшого розуміння проблематики та аналізу алгоритмів. В підходах до розуміння поняття «розпізнання обличчя», виділяють такі:

- практичний додаток теорії розпізнання образів, до завдань якої входить автоматична локалізація обличчя на фотографії та, у разі необхідності, ідентифікація персон за обличчям.

Ще одне поняття містить наступний зміст: «розпізнання обличчя» - розпізнавання особи як біометричний програмний додаток, здатне однозначно ідентифікувати або верифікувати людину шляхом порівняння і аналізу шаблонів на основі контурів особи людини. [2] Окрім того, під розпізнаванням обличчя (Facial recognition technology (FRT)) розуміють таке, - це автоматична локалізація людського обличчя на зображенні або відео і, при необхідності, ідентифікація особистості людини на основі наявних баз даних.

Здебільшого, всі основні складові терміну «розпізнання обличчя» дуже схожі за своєю структурою, та відрізняються в залежності від використання самої методики розпізнання.

Найбільш поширенні групи методів розпізнання осіб (обличчя) є такими:

• нейропсихологічні моделі;

• нейрофізіологічні моделі;

• інформаційно - процесуальні моделі;

• комп'ютерні моделі розпізнавання.

До першої категорії (нейропсихологічні моделі) відносять процес обробки суб’єкта, або фото, в яких основною метою для пошуку «схожості» використовуються дотинки із фактичного «психологічного портрету», таким чином з кожною наступною обробкою фото, відбувається штучне навчання взятого коду, а саме: розуміння та розпізнавання психологічних особливостей та моделей обличчя людини.

До другої категорії (нейрофізіологічні моделі) відноситься формульний підхід, який направлений на більш загальний, ніж на точковий підхід, а саме, щодо фактичного знаходження та ідентифікації людини за манерою ходьби, антропологічною та фізіологічною особливостями окремих частин тіла, та обличчя.

Третя група складається із механічної обробки кадрів, від сприйняття до актуалізації.

До четвертої категорії – найбільш розповсюдженої, відноситься використання запасу машинного рівня, який є обмеженим тільки з точки зору апаратної можливості персонального комп’ютера. Навчена модель штучно розуміє, що їй необхідно для пошуку.

Слід звернути увагу на те, що всі моделі використовується в площині їх симбіозу. В глобальному процесі - одна модель сплітається із іншою. Один процес виходить на другий. Можливе як поетапне поєднання моделей розпізнання обличчя так і паралельне, яке за наслідком має дати найбільш точну відповідь на запит користувача.[1]

Найновіші моделі, які використовуються в широкому загалі мають наступні назви: Smith & Wesson (система ASID — Automated Suspect Identification System); ImageWare (система FaceID); Imagis, Epic Solutions, Spillman, Miros (система Trueface); Vissage Technology (система Vissage Gallery); Visionics (система FaceIt).

На думку аналітиків час звичайних паролів і пін-кодів вже спливає. У найближчому майбутньому безпечна аутентифікація користувача буде неможлива без використання додаткових апаратних середніх типів USB-токенів та сканерів відпечатків пальців. Користуючись у масштабах всього інтернету, класичні паролі вже перестають відповідати сучасним вимогам за захищеною інформацією.

Слід зазначити, що найвпливовіші фірми та корпорації активно пропонують технології, що знаходяться у голові з цільовим їх використанням для авторизації клієнтів. І одна з таких технологій ,що набула великої популярності – технологія авторизації через розпізнавання обличчя.

Пропоную розглянути технологію розпізнавання обличчя через призму вже існуючих алгоритмів , а саме: EigenFaces (PCA) , FisherFaces (LDA),LBPH, з котрих один буде використаний в дипломному проекті основою для створення покращеної системи авторизації користувачів.

**1.2 Алгоритм EigenFaces (PCA).**

Метод головних компонент (англ. Principal component analysis, PCA) - це фундаментальний та універсальний метод, який має безліч практичних застосувань. Мета методу - зменшення розмірності вихідних даних зі збереженням найбільш значимої інформації.

У найбільш простому, але не ефективнішому варіанті, алгоритм роботи наступний:

1. Наявну навчальну вибірку з N зображень розмірності xy представляємо у вигляді квадратної матриці xy на xy. Для цього "витягуємо" кожне зображення в вектор довжиною xy. Кожен вектор утворює стовпець у матриці. Відсутні xy - N стовпців доповнюємо нулями.

2. Обчислюємо власні вектори цієї матриці і упорядковуємо їх в порядку убування відповідних їм власних значень. Це і є наші "власні особи".

3. Беремо M за один вектор. Як знайти оптимальне число M власних векторів - це окрема тема для дослідження. В практиці використовується таке поняття: - правило «зламаною тростини».

Таким чином, щоб знайти вектор в "просторі осіб", який відповідає цьому зображенню, розкладаємо зображення по кожному з M власних векторів, обчисливши скалярний добуток (власні вектора мають ту ж розмірність xy, що і зображення). Набір з M значень утворює вектор в просторі осіб.

Для розпізнавання особи на зображенні потрібно знайти відповідний цьому зображенню вектор в просторі осіб і визначити до якого вектору з навчальної вибірки він найближче. Для оцінки відстані доцільно використовувати дистанцію Махаланобіса (рис. 1.2). [4]

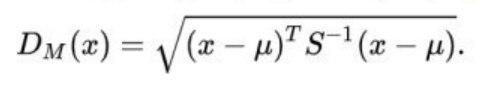


Рис. 1.2 Вигляд дистанція Махаланобіса

Алгоритм EigenFaces спочатку проектує всі навчальні приклади в підпростір аналізу головних компонент, потім проектує задане фото в цей підпростір, а потім відбувається пошук найближчих схожих ознак між тренувальними прикладами і заданим фото.

Використання всього набору численних ознак в процесі розпізнавання осіб (обличчя) може істотно уповільнити роботу алгоритму і знизити точність одержуваного рішення. Тому важливим для зниження розмірності в процесі вирішення задачі розпізнавання, є отримання найбільш інформативних ознак.

Метод головних компонент дозволяє зменшити число змінних за рахунок вибору найбільш мінливих з них. Він добре проявив себе в практичних додатках.

Ситуація дуже змінюється коли на об’єкт пошуку припадає значна доля світла (або висвітлення), тобто ефективність методу втрачає свою дієвість. Ця ситуація трапляється коли метод головних компонент (PCA) вибирає підпростір таким чином, щоб якомога більше апроксимувати вхідний набір даних всього діапазону з об’єкта, а не виконати дискримінацію між класами осіб (обличчя). Сам метод використовується ще на стадії фактичного предцифрової обробки як у кластерних так і інших можливих обробках.

Для більш подібного розбору: кожне фото презентується як вектор у просторі, де ця кількість векторів дорівнює окремим пікселям. [8]

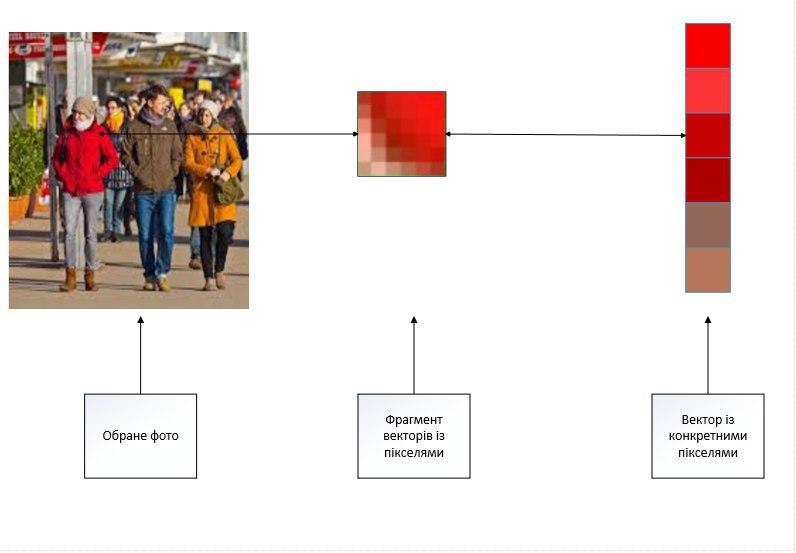


Рис. 1.3 Приклад роботи алгоритму PCA

Теоретичне підґрунтя методу являє собою в переведенні зображення в вимір ознак з найменшою розмірністю, в якому його можливо в подальшому найбільш точно охарактеризувати.

**1.3 Алгоритм FisherFaces (LDA).**

Наступним запропонованим алгоритмом, який має більш розгалужену систему є FisherFaces, в завдання якого покладено замінити свого попередника – EigenFaces.

В основі алгоритму FisherFaces лежить лінійний дискримінантний аналіз, який шукає проекцію даних, при якій класи максимально лінійно віддільні. Метод головних компонент (PCA), на відміну від лінійного дискримінантного аналізу (LDA), шукає проекцію даних, при якій буде максимізований розкид по всій базі даних осіб (без урахування класів). [6]

Іншими словами метод головних компонент зменшує розмірність, зберігаючи при цьому якомога більшу частину дисперсії в високому розмірному просторі. Лінійний діскрімінантний аналіз зменшує розмірність, зберігаючи при цьому якомога більше дискримінаційної інформації про клас.

Експериментально доведено, що в умовах сильного нижнього і бокового затемнення фото алгоритм FisherFace показав велику ефективність (більше 90%) в порівнянні з EigenFace (близько 50%).

FisherFaces працює досить добре при однаковому рівні освітленості зображення.

Слід пам’ятати, що в реальному житті не можна гарантувати завжди ідеальні параметри розміру, освітлення, повороту особи на зображеннях.

Згідно офіційній документації OpenCV, для отримання кращого показника розпізнавання потрібно приблизно 8 зображень на кожну людину, що створює труднощі при формуванні “dataset”. Рішення здавалося не оптимізовані належним чином, тому мета була замість представлення зображень у вигляді багатовимірного вектора - витягти локальні ознаки.

Нижче приведений «спіральний графік» співвідношень аналізу методів Eigenface та Fisherface.[7]

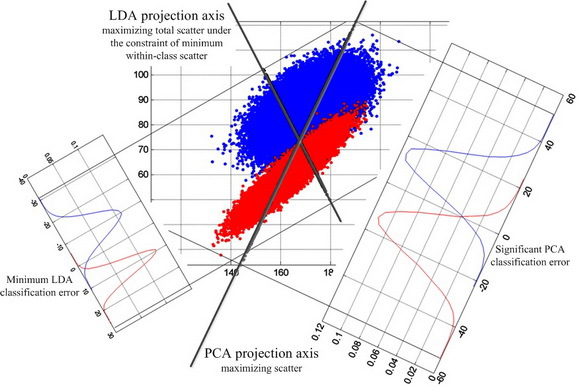


Рис. 1.4 Спіральний графік відмінності алгоритмів PCA та LDA

**1.4 Алгоритм LBPH.**

Вперше алгоритм був описаний у 1994 році (LBP) і з тих пір був визнаний потужнім засобом класифікації текстур. Крім того, було визначено, що коли LBPH поєднується з дескриптором гістограм орієнтованих градієнтів (HOG), це значно підвищує ефективність виявлення схожості в деяких наборах даних.

Алгоритм LBPH використовує 4 характеристики:

* + 1. Радіус: радіус використовується для побудови кругового місцевого двобічного шаблону і представляє радіус навколо центрального пікселя. Зазвичай він дорівнює 1. (рис. 1.5)

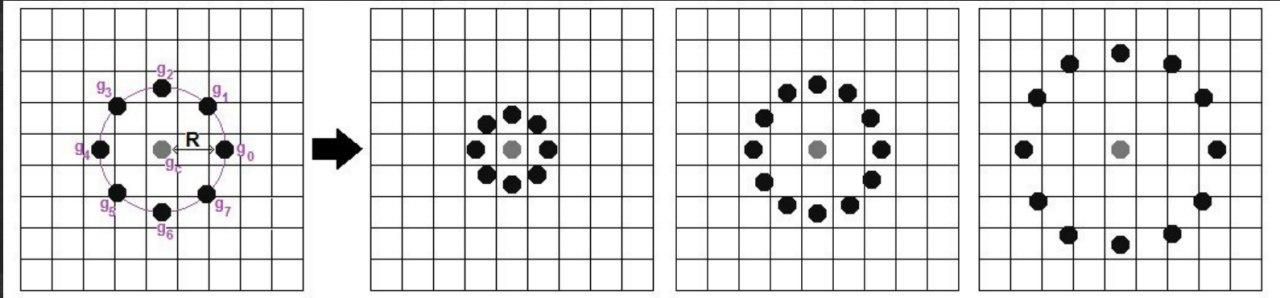


Рис. 1.5 Побудова двобічного шаблону

2. Дотичні точки: кількість точок вибірки для побудови кругового місцевого двостороннього шаблону. Чим більше точок вибірки включено, тим вище розрахункові затрати на пошук схожості.

3. Горизонтальна сітка: кількість кліток (пікселей) у горизонтальному напрямку. Чим більше пікселей, тим менша сітка, тим вище розмірність результатів.

4. Вертикальна сітка: кількість кліток (пікселей) у вертикальному напрямку. Чим більше пікселей, тим менша сітка, тим вище розмірність результатів.

Алгоритм Local Binary Patterns Histograms (LBPH) передбачає підсумовування локальних структур зображення, шляхом порівняння кожного пікселя із суміжними. Зображення розбивається на X та Y локальних областей, в кожній з яких виділяється центральний піксель і порівнюється із суміжними.

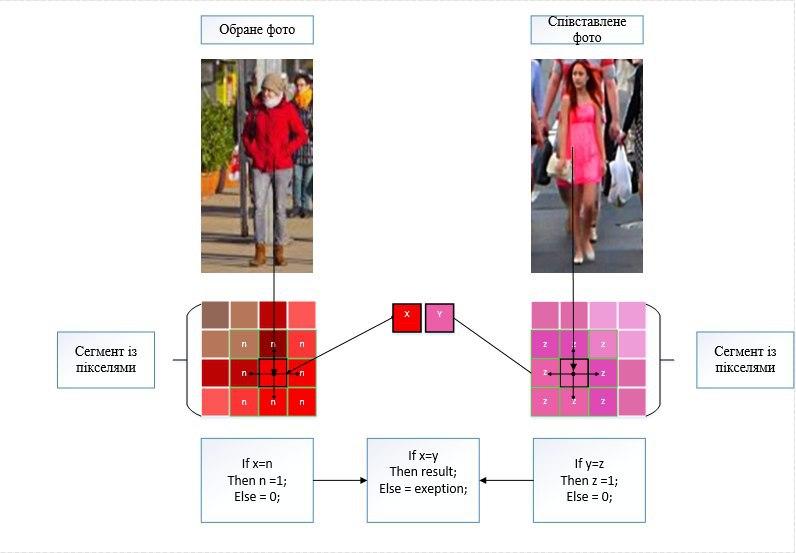


Рис. 1.6 Робота алгоритму LBPH

Якщо схожість наповнення пікселя, який стоїть рядом із порівнюваним,- відповідає або дорівнює першому, то такий піксель позначається 1, якщо ні - ставиться 0. [6]



Рис. 1.7 Кінцеве співставлення центральних пікселів із дотичними до нього клітинками

Підсумковий бінарний код (числовий вимір) є локальним бінарним шаблоном. Після вилучення з кожною локальною області гістограм локальних бінарних шаблонів відбувається їх з'єднання в підсумкову модель розпізнавання осіб. Фрагмент коду X порівнюється із кодом Y і у разі подібності видає результат, якщо ні, то або продовжує пошук, або видає помилку.

Алгоритм LBPH не чутливий до монотонних трансформацій відтінків сірого. Також, пошук інформації обмежується базою прив’язку та структури алгоритму.[5]

**1.5 Використання бібліотеки OpenCV.**

Для вирішення задачі розпізнавання осіб (обличчя) існує безліч інструментів, з яких найбільш поширеним є бібліотека з відкритими початковими кодами OpenCV, а саме - клас FaceRecognition.

OpenCV (Open Source Computer Vision) - це кроссплатформенна бібліотека комп'ютерного зору, метою якої є обробка зображень в режимі real-time. OpenCV містить 3 (три) варіанти алгоритму розпізнавання осіб (обличчя): EigenFaces, FisherFaces і LBPH (Local Binary Patterns Histograms), в кожному з яких в основі лежать свої методи: метод головних компонент, лінійний діскрімінантний аналіз і гістограми локальних бінарних шаблонів відповідно, які вже були представлені вище.

Використання всього набору численних ознак в процесі розпізнавання може істотно уповільнити роботу алгоритму і знизити точність одержуваного рішення. Тому важливим для зниження розмірності в процесі розв'язання задачі ідентифікації мовця, є отримання найбільш інформативних ознак, використовуваних алгоритмами розпізнавання.

Для того щоб вибрати найбільш підходящі ознаки, в кожному з алгоритмів використовуються свої методи: метод головних компонент, лінійний дискримінантний аналіз і гістограми локальних бінарних шаблонів.

Щоб отримати достовірні результати, для алгоритмів бібліотеки OpenCV необхідно виконати кілька вимог для зображень:

1. Усі зображення повинні бути представлені в градаціях сірого.
2. Зображення повинні бути центровані.
3. Зображення повинні мати однакові розміри.

Тестування 3 алгоритмів бібліотеки OpenCV вироблялося на основі ковзного іспиту.

Прикладом став наступний експеримент - 4ох емоцій, 20ти суб'єктів вибірки. [4]

На першому прогоні зображення 1-го спікера виключалися з навчальної множини, і становили лише тестову безліч.

На другому прогоні тестова безліч складалося тільки з зображень 2-го спікера, які виключалися з навчальної множини.

В результаті порівняльного аналізу ефективності за середнім значенням точності, кращий результат показав алгоритм FisherFaces. І при перевірці на статистичну значущість (T-test проводився з рівнем значущості 0.05) з'ясувалося, що він має статистично значущі відмінності в порівнянні з іншими алгоритмами. Однак, слід відзначити, що бібліотека OpenCV одним із своїх показників виділяє сприйняття сірого кольору, який у свою чергу є базовою негативністю алгоритму LBPH. З огляду на проведений тест, алгоритм LBPH з невеликим коефіцієнтом поступився алгоритму FisherFaces, проте інші показники виявились навіть вищими, ніж порівнюваний алгоритм.

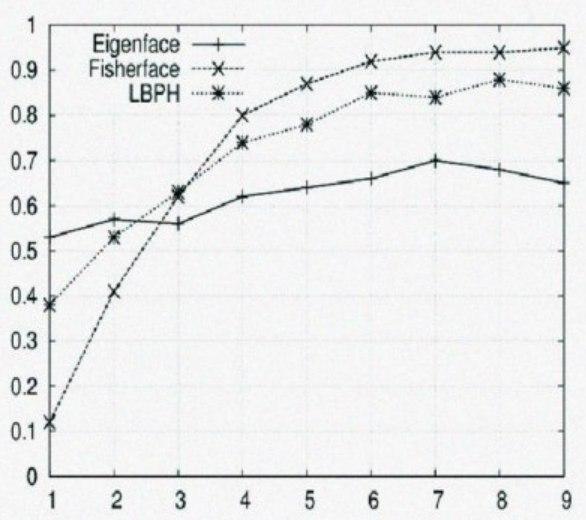


Рис. 1.8 Графік залежності алгоритмів у порівняльному кольоровому тесті.

**1.6 Порівняння основних алгоритмів розпізнавання обличчя.**

Таблиця 2.1

Основні переваги та недоліки запропонованих алгоритмів:

|  |  |  |
| --- | --- | --- |
| **Алгоритми:** | **Переваги:** | **Недоліки:** |
| **EigenFaces (PCA)** | 1. Мінімізація кількості вхідних даних з функцією зберігання найважливішої інформації. | 1. Виникають ускладнення з підвищеним вмістом світла. |
| 2. Максимізований розкид по всій базі даних осіб. | 2. Без урахування класів. |
| **FisherFaces (LDA)** | 1. Зменшення кількості вхідної інформації та підняття об’єктів за класами.  2. Шукає за лінією схожості об’єктів. | 1. Потрібна велика кількість зображень для розпізнавання об’єктів. 2. Немає збереження векторного формату, як у PCA. |
| **LBPH** | 1. Швидкодія та більший обсяг знаходження інформації серед баз вибірок. | 1. Алгоритм LBPH не чутливий до монотонних трансформацій відтінків сірого.  2. Звужений пошук порівняно з іншими алгоритмами. |

**ВИСНОВОК ДО ПЕРШОГО РОЗДІЛУ:**

На основі аналізу розглянутих алгоритмів можна зробити висновок, для підвищення швидкодії та їх ефективності необхідно розробити більш модифікований алгоритм для реєстрації нових користувачів з урахуванням особливостей технологій алгоритму LBPH.

В даному розділі були розглянуті основні алгоритми, з яких за основу було обрано алгоритм LBPH, який для розпізнавання та авторизації суб’єктів, використовує процес формування обробки інформації у бінарному коді.

На відміну від цього алгоритму пропонується розширення функціоналу метода LBPH, який враховує особливість освітлення, різних класів сприйняття, монотонності цвіту на основі використання попередньої обробки штатними та позаштатними інструментами, а також іншими інструментами, з метою підвищення ефективності дії алгоритму.

**РОЗДІЛ 2**

**РОЗШИРЕНИЙ АЛГОРИТМ LBPH З ВИКОРИСТАННЯМ ДОДАТКОВИХ ФУНКЦІЙ**

З огляду на дослідження запропонованих алгоритмів та їх подальшого використання, слід покращити або перетворити їх негативні характеристики.

Отже, з-першу, слід зупинитися на ідеї виключення залежності первинного суб’єкту, який бере участь у фактичній реєстрації/авторизації його у пропускних системах.

Так як з проведених тестів алгоритмів, з попереднього розділу, за основу подальшого використання був обраний алгоритм LBPH, отже, постає питання щодо вирішення таких проблем: використання штатних та позаштатних механізмів попередньої обробки фотографії, кількість вхідної інформації (фото), а також автоматизація даного процесу. Під автоматизацією вважаємо, що дія алгоритму від старту до отримання відповіді проходила в автоматичному режимі, аби викреслити фактор впливу на сприйняття кінцевим алгоритмом обробленої фотографії.

За своєю суттю попередня обробка фотографії являє собою покращення будь-яким чином вихідний файл-фото, який у подальшому процесу буде оброблювати алгоритм. Такі перетворення нададуть змогу алгоритму більш точно розпізнати вихідний файл, та як наслідок, прискорить пошук подібності (схожості).

Отже, все починається із отримання вихідного файлу-фото.

На відтворення файлу-фото, пропонуємо наступні методи:

Перший: завантаження відомого файлу-фото (який буде ідентифікуватись) з носія (приклад: флеш-карта, телефон із з’єднанням до апарату ідентифікації штатними програмами Wi-Fi / Bluetooth, та інші носії).

Другий: отримання фотографій безпосередньо у режимі реального часу. З особи, яка проходить ідентифікацію робляться знімки, які у подальшому оброблюються та направляються до алгоритму, який провадить пошук з подальшою ідентифікацією та реєстрацією особи-користувача.

При наступному детальному розборі виявилось, що перший спосіб має великий недолік. Людина, яка надає ідентифікатору інформацію для її подальшої обробки, може бути не тією за яку себе видає. Таким чином можливо обманути програму, так як ідентифікатор перевіряє сам фото-файл, а не людину, яка його подала.

Для реалізації другого способу, слід використати звичайну фотокамеру, або пристрій із подібним функціоналом. Людина, яка бажає отримати реєстрацію у відомій системи, становиться у визначеному місці (наприклад навпроти пристрою (камери)), де її одразу фотографують для подальшої обробки та пошуку відповідності у базі (базах).

Як вже зазначалось вище, за основу був обраний алгоритм LBPH, а отже графічні фотозйомки повинні бути у растровій графіці, тому що, в системі алгоритму провадиться пошук за схожими пікселями.

Растрове зображення (лат. Rastrum - граблі) - зображення, що представляє собою сітку (мозаїку) пікселів - кольорових крапок (зазвичай прямокутних).

Слід звернути увагу, на те, що один із негативних чинників растрового зображення – це великий розмір вихідного файлу (фото).

Для того щоб цей негативний фактор не впливав на програмний код, потрібно зосередити свою увагу на форматі самого файлу (фото).

Растрове зображення використовує наступні формати фото:

BMP або Windows Bitmap - зазвичай використовується без стиснення та на більшості сімейства Windows.

GIF (Graphics Interchange Format) - застарілий формат, який підтримує не більше 256 кольорів одночасно. Все ще популярний через підтримку анімації, яка відсутня в чистому PNG, хоча ПО починає підтримувати APNG.

PCX - застарілий формат, який дозволяв добре стискати прості мальовані зображення (при стисканні групи поспіль пікселів однакового кольору замінюються на запис про кількість таких пікселів і їх кольорі).

PNG (Portable Network Graphics) - растровий формат, в основі якого алгоритм стиснення Deflate, та інші формати растрової графіки.

Окремо слід виділити формат TIFF, який при збереженні ілюстрації не використовує до фото жоден з видів компресії (стиснення). У цьому форматі отримують максимально можливу ступінь якості та відповідності, збереженої у файлі копії зображення. Це єдиний формат, який використовується в професійному дизайні для зберігання зображень високої якості. Якісні TIFF-зображення можуть займати кілька сотень мегабайт. Базис формату дуже гнучкий та дозволяє зберегти зображення в різних варіаціях, що дозволяє отримати декілька копій одного зображення, а сам алгоритм вибере той, який буде найкраще підходити то розпізнавання.

Після того як вихідний файл-фото було зроблено у декількох варіаціях із невеликим проміжком часу, настає наступний етап – попередня обробка фото для подальшого сприйняття його алгоритмом.

Слід згадати, що обраний за основу алгоритм LBPH не чутливий до відтінків сірого кольору, а отже, у разі якщо фото містить велику кількість сірих пікселів - програма буде не в змозі віднайти подібні (схожі) фото у своїй базі.

Вирішенням цієї проблеми є використання метода сегментації зображень.

**2.1 Метод Сегментації зображення.**

Один з найпопулярніших методів позначення меж об'єктів – є сегментація. Цей метод є детермінований і статичний.

Під терміном сегментація, розуміють перетворення напівтонових або кольорових знімків в цілісні зображення, яке має меншу кількість тонів або ж кольорів, ніж початкові. У більш вузькому спектрів сегментацію називають перетворенням в бінарне (дворівневе) дерево, що має всього два рівня яскравості – мін. (0) і макс. (255). Об'єкт і фон розділені, визначення кількість об'єктів – першочергова задача. Місце розташування (координати, характеристика осі об'єкта відносно координатних осей), також геометричні характеристики (площа об'єкта, периметр, а також розміри) – все це допоможе розпізнати об’єкт в цілому.

Як вже було сказано раніше основна задачі сегментації – виділення однорідних областей. Однорідність – це приналежність області до конкретного класу. [9]

Перейдемо до видів сегментації зображень:

1.Обчислення порогу при порогової сегментації.

Порогова сегментація – це з'єднання схожих характеристик областей знімка в якесь число сегментів. У разі, коли яскравість вище допустимого порогу, то область відносять до одного сегмента, в іншому випадку до іншого. Найпростіший і найпопулярніший спосіб – розбиття на два сегменти (бінаризація). Порогова сегментація може виконуватися на базі апріорно встановлених порогів. Інший спосіб є тим, що пороги визначаються як межі гістограми зображення. Краще всього представити даний малюнок як приклад бінаризації шляхом поділу на два кольори, тобто методом двох мод. Це типовий спосіб для виділення площ вкритих снігом або льодом на тлі дерев і вже відталої землі за результатами досліджень.

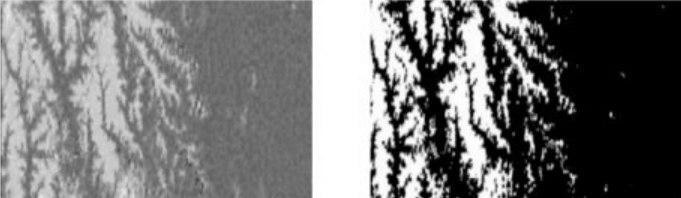


Рис. 2.1 Фото зліва – первинне зображення. Справа – проведення сегментації.

Більше труднощів виникає при проведенні сегментації, коли гістограми фону та об'єкта перекриваються «хвостами» при такій ситуації частина точок об'єкта може бути приписана до тла, а інша до самого об'єкта.

2.Сегментація шляхом нарощування областей.

У способі сегментації шляхом нарощування областей виділяються однорідні області. Розглянемо сегментацію шляхом нарощування галузей з використанням критерію однорідності за величиною векторів яскравості.

Схема алгоритму цього методу передбачає вибір стартового пікселя і розгляд суміжних з ним пікселів для перевірки близькості їх значень, наприклад, по евклідовій відстані. Якщо значення яскравості поточного і будь-якого суміжного пікселів виявляються близькими, то ці пікселі зараховуються

в одну область. Таким чином, формується область в результаті зрощування окремих пікселів. На наступному етапі область перевіряється на однорідність і, якщо результат перевірки виявляється негативним, то область розбивається на більш дрібні ділянки. Процес триває до тих пір, поки всі виділені області не пройдуть перевірку на однорідність.

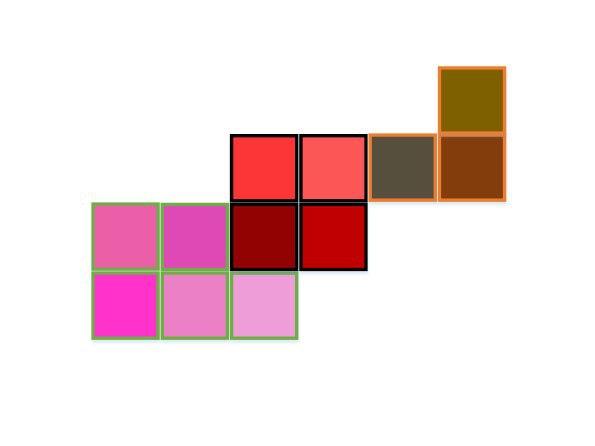


Рис. 2.2 Процес сегментації нарощування областей (за групами).

3.Сегментація шляхом виділення меж

Сегментація, як виділення кордонів передбачає задіяння оператора градієнта. Для розуміння того, що вже виявлено схожість між пікселями, задіється процедура поділу по порогу. Пікселі позначені, як граничні, об'єднуються в замкнуті криві, які оточені своїми ж областями.

У цьому методі, також, як і в інших важливих є критерій однорідності області, по характеристикам якого і обчислюється значення градієнта.

Питання конструювання кордонів сегментів на знімку градієнта позиціонується як самостійна завдання. Це - досить важка задача, яка може знайти рішення тільки при легких ситуаціях, наприклад, при виділенні локальних максимумів градієнта всіх рядків і стовпців знімка. Явний приклад такої ситуації - зображення частини краю Землі.

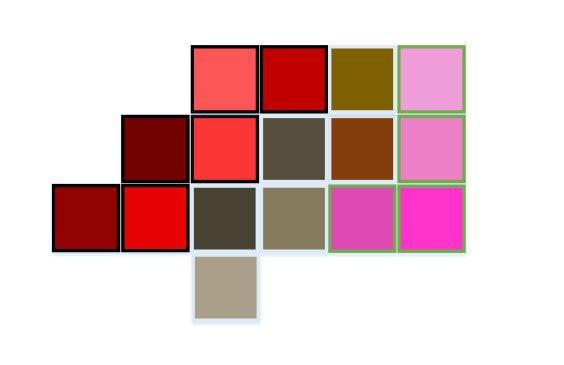


Рис. 2.3 З’єднання дотичних пікселів до однієї групи за їх схожості.

Поряд із роботою метода сегментації фото, вбачається використання штатної обробки первинних фотографій для подальшого пошуку за допомогою алгоритму. [11]

**2.2 Штатна обробка фотографій.**

До штатної обробки фото слід віднести:

1. Редагування балансу білого:

Чітке визначення балансу білого дуже часто може підвищити передачу кольору знімка.

Неправильний баланс білого може значно ускладнити обробку фотографії шляхом зниження таких факторів як: контраст і насиченість фото.



Рис. 2.4 Фото зліва - без редагування балансу білого, фото справа з використанням корективів балансу білого.

2. Експозиція:

Фото може містити деякі ділянки затемнення або вицвіту, з правильно підібраним ракурсом фото набуває чіткості.



Рис. 2.5 При використанні експозиції ділянки першого фото виказують затемнення, останнє фото – ділянки висвітлення, у свою чергу друге фото містить достатній рівень експозиції.

3. Шумозаглушення:

Є кілька видів візуального шуму, деякі його типи видаляються легко, деякі складно. За рахунок того, що фото було зроблено за високої чутливості ISO, відбувається виграш від шумозаглушення. Як раз такий шум, викликаний в результаті високого ISO, відноситься до того типу з якими відносно легко боротися.



Рис. 2.6 На фото зліва присутній шум, на фото справа після обробки відсутній.

4. Корекція об'єктива.

Є три основні і найбільші проблем у роботі з об'єктивом:

а) Віньєтування - помітно більше всього на низьких щаблях, в залежності від об'єктива його прояв варіюється (затемнення фотографії).

б) Спотворення - найсильніше проявляється, коли використовується ширококутний об'єктив або об’єктив камери телефону.

в) Хроматичні аберації – так само помітні на низьких ступенях, в крайніх частинах знімка і в частинах з дуже контрастними деталями.

Цей спосіб (корекція об'єктива) більш нейтральний, і буде використовуватись в залежності від здатності пристрою суб’єкта-користувача.

5. Різкість і чіткість фото:

Різкість і чіткість являє собою корекцію згладжування, яке є у камери або об'єктива. Підвищення різкості необхідно здійснювати обережно, оскільки в результаті неправильного його використання можуть з'явитися інші проблеми.

6. Контраст і криві:

Фото зроблені навпроти сонця або ж поблизу з яскравим світлом, часто мають низький контраст (оскільки знаходяться під впливом відблисків). Надмірне використання контрастності може надати знімку неприродний вигляд.

Нижче приведені приклади контрастного фото та без додавання контрасту.

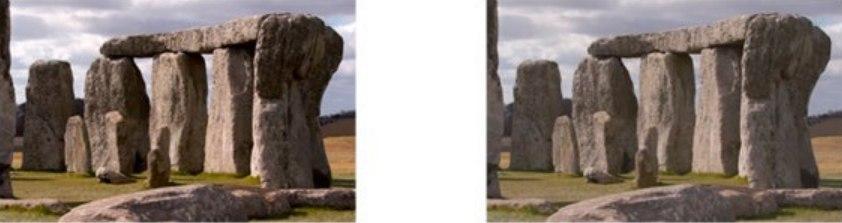


Рис. 2.7 Фото зліва – насичене контрастом, фото справа – без вирівнювання контрасту та кривих.

7. Композиція: кадрування.

Велику кількість знімків можна поліпшити простим кадруванням, яке посилює композицію.

8. Ретуш: передача кольору і часткова корекція.

Передача кольору – коригування насиченості, атмосфери та інших параметрів

у фото.



Рис. 2.8 Фото зліва не має дефектів які могли потрапити на об’єктів чи поле хору камери. Справа навпаки – фото із дефектами.

9. Розмір: збільшення і зменшення.

Щоразу при зміні розміру знімка необхідно змінювати розмір файлу (процес називається інтерполяцією цифрового зображення»).

10. Вихідна різкість:

Найчастіше це є останнім етапом обробки фото. Її налаштування залежить від пристрою відображення і може враховувати деякі речі, основані на розмірності та типі знімка.

Штатні методи являють собою обробку та застосування до первинних фото і інших методів, які мають на меті покращити сприйняття кінцевого фото обраним алгоритмом.

Застосування методів обробки фото повинно відбуватись у автоматичному режимі, таким чином воно економить час, а також позбавить використання суб’єкта-ідентифікатора, залишивши увесь масив дій на роботі самої програми, або додатку.

Природнє середовище застосування методів – відео-редактор. Здебільшого який відео-редактор використовувати, це питання дискусійне. Головне, щоб обробка вхідного зображення відповідала поставленій меті. Як приклад можливо обрати відео-редактор – Movavi.

Movavi – це програма, яка працює як звичайний редактор. Широке використання знайшла у системі Windows. Функціонал програми дозволяє використовувати в автоматичному режимі увесь масив для редагування фото: подавляти шуми, додавати або забирати кольори, тощо. [10]

Кінцевим етапом, після проведення попередніх етапів можливо стикнутися із проблемою, що роботі алгоритму щось завадить і отримати відповідь буде неможливо.

Для вирішення такої проблеми запропонований наступний алгоритм дій, а саме:

У користувача є можливість підтвердити свою особу за допомогою голосового вводу або звичайного паролю. Для надійності та швидкодії алгоритму може бути запропонований варіант використовувати подвійне підтвердження особи (Face-id та пароль) яке буде діяти за наступним алгоритмом:

Алгоритм авторизації користувача з додатковим рівнем безпеки:

1. If Face-id is accept;
2. then go to account ;
3. If Face-id is not accept;
4. then go to password;
5. If password is accept;
6. then go to account;
7. Else go to 4;
8. Finaly go to 1.

*Коментарі до псевдо-коду:*

1. Перевірка на відповідність особи з використанням Face-id

2. Доступ до профілю

3. Якщо Face-id не відповідає особі

4. Запит на введення паролю підтвердження особи

5. Якщо пароль вірній

6. Доступ до профілю

7. Якщо пароль невірний, вертаємося до пункту 4

8. У разі сторонніх збоїв, повторення алгоритму пункт 1.

**ВИСНОВОК ДО ДРУГОГО РОЗДІЛУ:**

За основу в роботі було обрано алгоритм LBPH, який був покращений шляхом врахування недоліків та використання додаткових функцій, зокрема, завдяки наявності попередньої обробки відбулось скорочення часу реєстрації, або авторизації користувачів

Запропоновано і обґрунтовано повний процес обробки фотографії, який дозволяє більш детальніше урахувати усі недоліки алгоритму LBPH.

З практичної точки зору використання запропонованого в роботі розширеного алгоритму підвищує ефективність роботи усієї системи.

Під час роботи були використані швидкодіючі компоненти за рахунок яких можливо досягнути покращених результатів.

**РОЗДІЛ 3**

**РЕАЛІЗАЦІЯ ТЕХНОЛОГІЇ РОЗПІЗНАВАННЯ ОБЛИЧЧЯ, З ЇЇ ПОДАЛЬШИМ УДОСКОНАЛЕННЯМ**

Для реалізації методу розпізнавання обличчя буде використана бібліотека OpenCV (детальний опис можна знайти у розділі 1, підпункт 1.5), у якій ми зможемо реалізувати метод LBPH(детальний опис можна знайти у розділі 1, підпункт 1.4).

Сам метод буде реалізований на програмній мові високого рівня Python з використанням інтегрованого середовища розробки Pycharm.

Python – одна з тих рідкісних мов програмування, які одночасно претендують на звання простих та потужних.

Офіційно Python представляють так:

Python – це проста в освоєнні та потужна мова програмування. Вона надає ефективні високорівневі структури даних, а також простий, але ефективний підхід до об’єктно-орієнтованого програмування. Його елегантний синтаксис і динамічна типізація поряд з тим, що він інтерпретується, роблять його ідеальною мовою для написання сценаріїв та швидкої розробки додатків в різних областях і на більшості платформ. [11]

Для реалізації даного методу потрібно спочатку встановити середовище розробки. Оскільки я буду використовувати мову програмування Python, то для мене найкращий вибір - це PyCharm який розроблений компанією JetBrains.

PyCharm – інтегроване середовище розробки для мови програмування

Python, котре надає засоби для аналізу коду, графічний зневаджувач, інструмент для запису юніт-тестів і підтримує веб-розробку на Django.

**3.1 Як працює розпізнавання обличчя OpenCV**

Щоб розпізнати обличчя потрібно пройти, як мінімум три етапи:

1. Збір даних
2. Навчання розпізнавача
3. Розпізнавання.

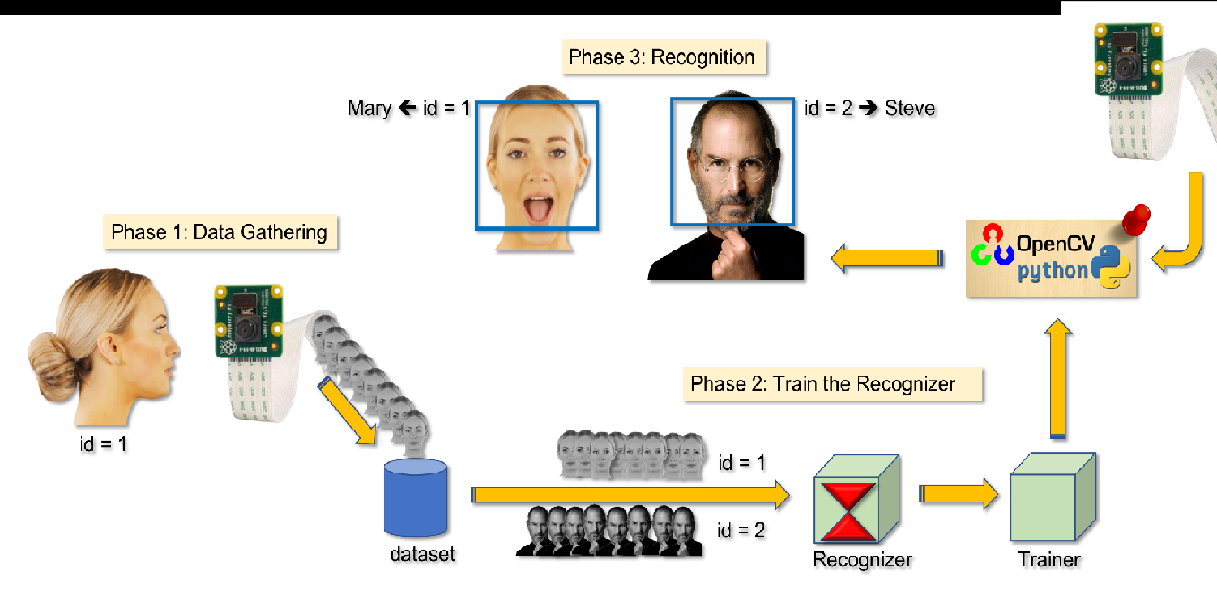


Рис 3.1 Етапи розпізнавання обличчя

Спочатку ми застосовуєм так зване - виявлення обличчя, котре визначає наявність і місцеположення обличчя на зображені, але не ідентифікує його. Далі ми вилучаємо так звані 128-мірні вектори котрі визначають кількість кожного обличчя на зображенні. Потім ми вводимо зображення або відеокадр в наш конвейер розпізнавання обличчь. Враховуючи вхідне зображення ми застосовуємо виявлення обличчя щоб визначити місце розташування обличчя на зображенні. Після цього наше вхідне обличчя проходить через нейронну мережу(модель навчання FaceNet).

Для навчання моделі розпізнавання обличчя кожна вхідна частина даних включає три зображення:

1)Зображення A (наше поточне обличчя)

2)Позитивне зображення(наше позитивне зображення, котре також містить поточне обличчя)

3)Негативне зображення(з іншої сторони не має такої ж ідентичності та може належати взагалі іншій людині.

Вся справа в тому, що зображення А та позитивне зображення – належать одній і тій самій особі, а негативне зображення не належить.

Нейронна мережа вираховує 128-мірні вектори для кожного обличчя, а потім налаштовує так звану ''вагу'' мережі так, що:

* 128-мірні вкладення зображення А та позитивного зображення знаходяться блище один до одного.
* в той же час відштовхує вкладення від негативного зображення.

Таким чином мережа навчається кількісно визначати обличчя і повертати, дуже схожі та зовсім не схожі вкладення, що підходять для роспізнавання обличчя.[10]

**3.2 ОГЛЯД ПРОЕКТУ**

Архітектура проекту в середовищі розробки Pycharm представлена на малюнку (Рис 3.1).

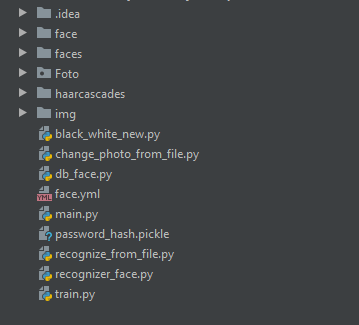


Рис 3.2 Архітектура проекту

Проект складається з дев’яти python файлів кожен з яких містить свої класи та методи , що взаємодіють між собою.

В стартовому вікні програма надає користувачу наступний список дій(рис 3.2):



Рис 3.3 Стартове вікно

* Зареєструватися
* Увійти в систему через розпізнавання обличчя
* Увійти в систему через фото зроблене при реєстрації(запасний варіант, якщо при вході в систему через розпізнавання обличчя виникла помилка чи несправність веб-камери)

При реєстрації буде запропоновано створити логін та пароль. Після проходження етапів реєстрації, користувач зможе увійти в систему. Увійшовши до системи потрібно додати своє обличчя в базу даних, для подальшої авторизації вже через систему розпізнавання обличчя.

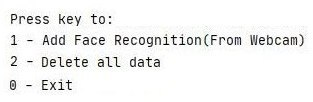


Рис 3.4 Набір команд після авторизації

Для цього потрібно обрати пункт 1 та зачекати поки ініціалізується веб- камера, яка зробить 30 фотографій обличчя(за прогресом можна слідкувати в полі виводу) і передасть зроблені фотографії класу TrainAI(детальний опис класу можна знайти у розділі 3.2), для того щоб нейронна мережа навчилась розпізнавати обличчя, після цього дані успішно будуть занесені в базу даних. Також можна видалити всі дані про користувача(наприклад, якщо цей користувач більше не може мати доступ до системи), обравши пункт 2.

При вході в систему через розпізнавання обличчя ініціалізується веб- камера, яка повинна ідентифікувати користувача котрий знаходиться перед нею. Якщо ідентифікація не вдалася буде запропоновано альтернативний варіант входу – ввід логіна та паролю.



Рис 3.5 Приклад авторизації з використанням паролю

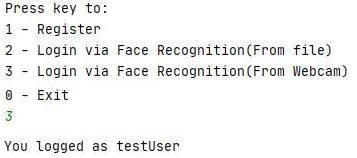
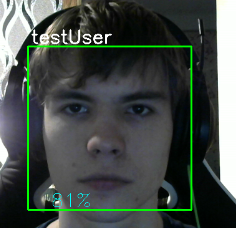
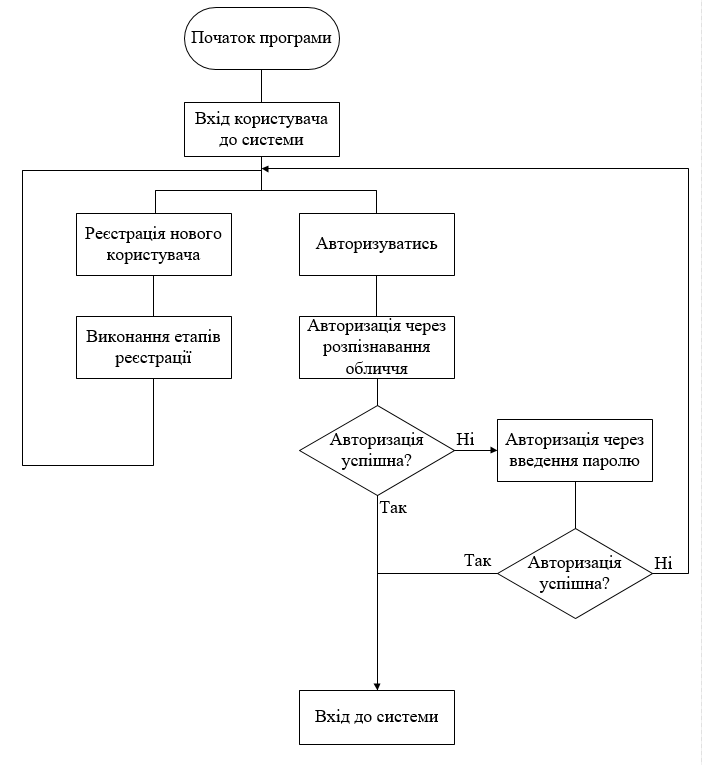


Рис 3.6 Приклад авторизації з використанням розпізнавання обличчя

Загальний алгоритм роботи системи представлений на малюнку (Рис 3.7).

Рис 3.7 Алгоритм за яким працює система



**3.3 ОПИС ФАЙЛІВ ПРОЕКТУ**

**db\_face.py** – цей файл фотографує користувача із веб-камери певну кількість разів (було вирішено, що 30 це оптимальна кількість). Коли програма помічає обличчя користувача, то виділяє його потім обрізає та зберігає в сірому фильтрі.

Файл містить один клас FaceByCam в якому є метод take\_photo\_ from\_webcam, а також метод \_\_init\_\_ , який ініціалізує змінні, і існує для того, щоб створювати екземпляри класів. Основні функції, що використовуються в цьому класі:

* cv2.VideoCapture(0,cv2.CAP\_DSHOW) – команда, запуску веб-камери.
* ret, img = cam.read() – команда зчитування зображення з веб-камери.
* Face\_detector = cv2.CascadeClassifier(self.haarcacades\_path) – вибір каскаду для розпізнавання обличчя(обирається каскад Хаара).

Каскади Хаара (Haar Cascade) - це метод виявлення об'єктів, що використовується для визначення місця розташування об'єктів на зображеннях. Алгоритм навчається на великій кількості позитивних і негативних зразків: позитивні зразки - це зображення, які містять об'єкт, що цікавить, а негативні зразки - це зображення, які містять що завгодно, крім шуканого об'єкта. Після навчання класифікатор може знайти об'єкт, що цікавить на нових зображеннях.

**Change\_photo\_from\_file.py**

Файл складається з класа PhotoCutter з методом change\_photo\_from\_file.

Клас PhotoCutter містить шлях до папки з зображеннями, який ми вказуємо при створенні екземпляра класу. Його суть полягає в тому, щоб знайти всі зображення, виділити на них обличчя, вирізати його. Потім функція OpenCV .cvtColor () перетворює об'єкт вхідного зображення в об'єкт в градаціях сірого та зберігає в іншій папці.

Звичайна практика обробки зображень - спочатку перетворити вхідне зображення у відтінки сірого. Це пов'язано з тим, що виявлення яскравості, на відміну від кольору, зазвичай дає кращі результати при виявленні об'єктів.

Клас містить наступні функції:

img = cv2.imread(f"{self.from\_folder}/{file}") – функція, що зчитує зображення з файлу.

OpenCV .cvtColor () – функція перетворення об’єкту в об’єкт градаціяї сірого.

Приклад перетворення:



Рис 3.8 Початкове зображення.



Рис 3.9 Результат виділення обличчя та перетворення його в обличчя в градації сірого

**Black\_white\_new.py**

Файл містить клас PhotoFilter з методами black\_white\_filter, remove\_noise.

Клас PhotoFilter містить шлях до папки, який ми вказуємо при створенні екземпляра класу.

black\_white\_filter – знаходить всі зображення в папці та застосовує до них метод ''виявлення країв зображення''.

В алгоритмі класифікації спочатку сканується зображення на предмет

«об’єктів », тобто при введенні зображення алгоритм знаходить всі об’єкти на цьому зображенні, а потім порівнює їх з характеристиками об’єкта, який ви намагаєтесь знайти. У разі такого класифікатора, як обличчя користувача, він буде порівнювати всі об’єкти, знайдені в зображенні, з характеристиками зображення обличчя користувача і якщо збіг знайдено, він говорить нам, що вхідне зображення містить обличчя користувача.

Приклад застосування методу ''виявлення країв'':



Рис 3.10 Початкове зображення



Рис 3.11 Результат виявлення краю.

Як ви можете бачити частина зображення, яка містить об’єкт, була розставлена точками / відділена за допомогою виділення краю.

remove\_noise – метод , що усуває / фільтрує зображення на предмет шуму.

У більшості випадків вихідні дані, які ми збираємо містять шум, тобто небажані функції, які ускладнюють сприйняття зображення. Хоча ці зображення можна використовувати безпосередньо для вилучення ознак, точність алгоритму може постраждати. Для отримання більшої точності, обробка зображень застосовується до зображення перед передачею його до алгоритму. Існує багато різних типів шуму, таких як гауссовський шум, шум солі і перцю і т.д. Ми можемо видалити цей шум з зображення, застосувавши фільтр, який видаляє цей шум або, хоча б, зводить до мінімуму його вплив.

**Train.py**

Файл містить клас TrainAI з методами getImagesAndLabels, train\_db, recognizer.train().

Клас TrainAI містить шлях до папки, який ми вказуємо при створенні екземпляра класу.

Опис методу train\_db:

1)Спочатку метод викликає getImagesAndLabels, щоб знайти всі фото з папки.

2)Після того, як всі фото були знайдені він отримує їх від методу getImagesAndLabels у вигляді списку.

3)Далі train\_db створює об’єкт recognizer класу LBPHFaceRecognizer\_create().

4) Після цього метод передає об'єкту recognizer список лиць, і за допомогою метода recognizer.train() навчає нейронну мережу розпізнавати дане обличчя.

5)В кінці викликається методrecognizer.write , який записує результат в файл

face.yml.

**Face.yml**

Файл в якому зберігається вся інформація про обличчя, завантажена методом train\_db. Інформація представлена в векторному вигляді. Кожне лице має свій персональний id.

**Recognize\_face.py**

Файл містить клас FaceRecognizerWebCam з методом recognize\_face\_from\_webcam. Суть методу полягає в тому, щоб розпізнати обличчя користувача перед веб-камерою, та вивести його ім’я і коефіцієнт схожості. Основні функції, що використовуються:

* recognizer = cv2.face.LBPHFaceRecognizer\_create() – створюємо об’єкт
* recognizer.read(self.config\_path) – зчитує файл face.yml в якому зберігається список фото завантажених попередньо методом traind\_db.
* face\_cascade = cv2.CascadeClassifier(self.cascade\_path) – каскад який ми використовуємо для розпізнавання обличчя.
* gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY) **–** застосування функції градації сірого.
* faces = face\_detector.detectMultiScale(gray, 1.3, 5) **–**  знаходимо обличчя на фотографії (фотографія робиться після кожного кадру з веб-камери)
* cam = cv2.VideoCapture(0) – запуск веб-камери
* cv2.rectangle(img, (x, y), (x + width, y + height), (255, 0, 0), 2) **–** команда фіксації обличчя.

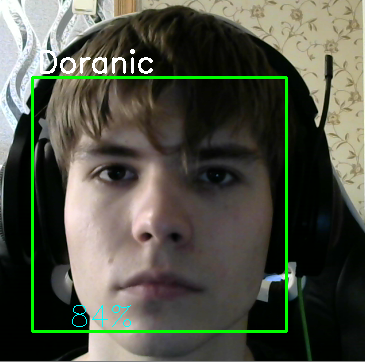


Рис 3.12 Приклад розпізнавання обличчя в реальному часі з коефіцієнтом схожості, по завантаженим попередньо фото.

**recognizer\_from\_file.py**

Файл містить клас FaceRecognizerWebCam з методом recognize\_face\_from\_file. Виконує таку ж роль, як і FaceRecognizerWebCam, але як об’єкт даних використовує не обличчя з веб-камери, а попередньо завантажені фото методом train\_db. Основна функція :

img = cv2.imread(self.file\_path) – зчитує фото з файлу.

**password\_hash.pickle**

В цьому файлі зберігаються паролі користувачів в зашифрованому вигляді.

Пароль шифрується за алгоритмом хешування SHA256.

SHA256 – це безпечний алгоритм хешування, який генерує вихідне хеш-значення в 256 біт.

В мові програмування Python, SHA-256 надається модулем hashlib.

Також в цьому файлі зберігаються імена користувачів та їх id.

**Main.py**

Основний файл в якому проходить реєстрація / вхід користувачів у систему.

Він містить клас ConsoleUserMenu з наступними методами:

1) get\_users() - метод повертає список всієї інформації про користувачів, або повертає пустий список,, якщо база даних пуста

2) register() - метод приймає 2 аргумента. Ім'я користувача та його пароль

Якщо таке ім'я вже присутнє в базі даних, то метод повертає False, якщо реєстрація пройшла успішно, то метод повертає True.

В іншому випадку Створює нового користувача, а пароль хешує за алгоритмом sha-256 з сіллю в 32 біта, який генерується випадковим чином та записується в файлі password\_hash.pickle в бінарному вигляді за допомогою бібліотеку pickle

3)login() - метод приймає 2 аргумента. Ім'я користувача та його пароль

Якщо таке ім'я вже присутнє в базі даних, то метод повертає True і просить ввести пароль, інакше повертає False.

4)home\_page() - метод, що складає основу консольного додатку. Виводить користувачу початкову інформацію, і дає початковий вибір дій, наприклад почати реєструватись, чи авторизуватись.

5)register\_page() – метод запитує у користувача ім’я та пароль і передеє керування методу register(). Після цього запитує у користувача наступні дії в залежності від результату реєстрації.

6)login\_page() - метод запитує в користувача ім'я та пароль і передає керування методу login(). Після цього запитує в користувача наступні дії

7)logged\_page() - метод викликається, коли користувач успішно зареєструвався та запитує наступні дії.

8)add\_face() - метод запитує в користувача папку, в який мають бути фотографії користувача. Та передає керування классам PhotoCutter, PhotoFilter, TrainAI.

9)add\_face\_by\_cam() - метод просто викликає методи класів FaceByWebCam, PhotoFilter, TrainAI.

10)login\_via\_face() - метод отримує список всіх користувачів через метод get\_users() та запитує в користувача шлях до його фотографії. Потім метод передає фотографію класу RecognizerFaceByFile для наступної ідентифікації. Повертає ім'я користувча при успішній авторизації

11)login\_via\_face\_by\_cam() - метод отримує список всіх користувачів через метод get\_users() та передає керування класу FaceRecognizerWebCam для наступної ідентифікації. Повертає ім'я користувча при успішній авторизації

12)delete\_all() - Видаляє всі дані про користувача (логін, пароль, фото).

**3.4 Тестування реалізованих методів**

Тестування програми – це обов’язкова перевірка для будь-якої завершеної програми. Вона допомагає, як перевірити працездатність системи, так і виявити можливі помилки та недоліки. Для оцінки програми перевіримо її працездатність використавши всі доступні нам функції.

Для початку створимо нового користувача. Щоб це зробити обираємо пункт 1.



Рис. 3.13 Вікно входу в систему

З’явилися поля вводу для імені користувача та паролю.

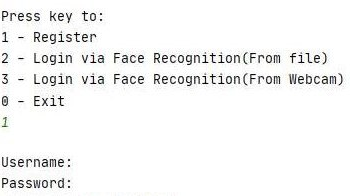


Рис. 3.14 Ввід даних для реєстрації

Вводимо дані , наприклад:

Username : User1.

Password : Pass1.

та натискаємо Enter. Система зареєструвала користувача User1 та виконала вхід.

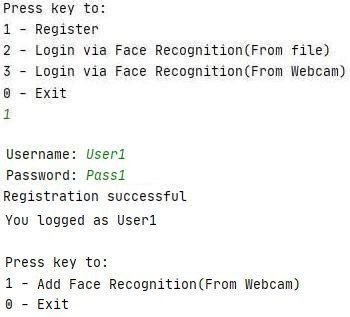


Рис. 3.15 Успішна реєстрація

Тепер потрібно додати своє обличчя в базу даних, для подальшої авторизації вже через систему розпізнавання обличчя. Щоб це зробити обираємо пункут 1 і чекаємо поки ініціалізується веб-камера та зробить 30 фотографій вашого обличчя(за прогресом можна слідкувати в полі виводу) і передасть зроблені фотографії класу TrainAI(детальний опис класу можна знайти у розділі 3.2). Це потрібно для того щоб нейронна мережа навчилась розпізнавати обличчя, після цього дані успішно будуть занесені в базу даних.

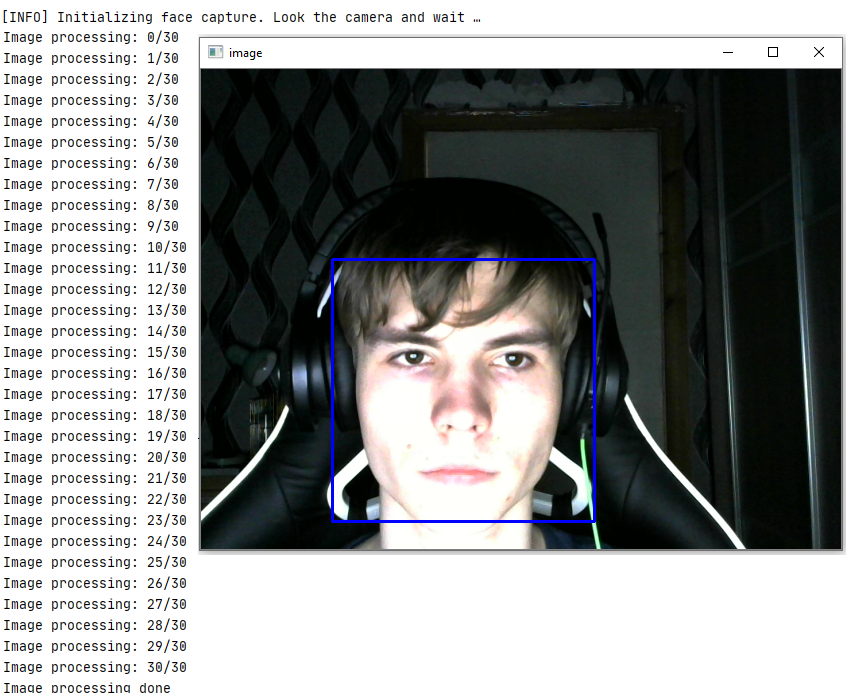


Рис. 3.16 Реєстрація обличчя

Наше обличчя занесено до бази даних і ми можемо авторизуватись. Обираємо пункт 3 (Login via Face Recognition(From Webcam)). Веб-камера має розпізнати ваше обличчя та ввійти до системи.

Рис. 3.17 Авторизація через розпізнавання обличчя



Розглянемо випадок, коли авторизація через веб-камеру не вдалася. Наприклад я закрив частину свого обличчя звичайним листом А4. Очевидно, що даних котрі отримує програма буде недостатньо для авторизації. Програма виведе наступну помилку та запропонує альтернативний спосіб авторизації – через ввід паролю.

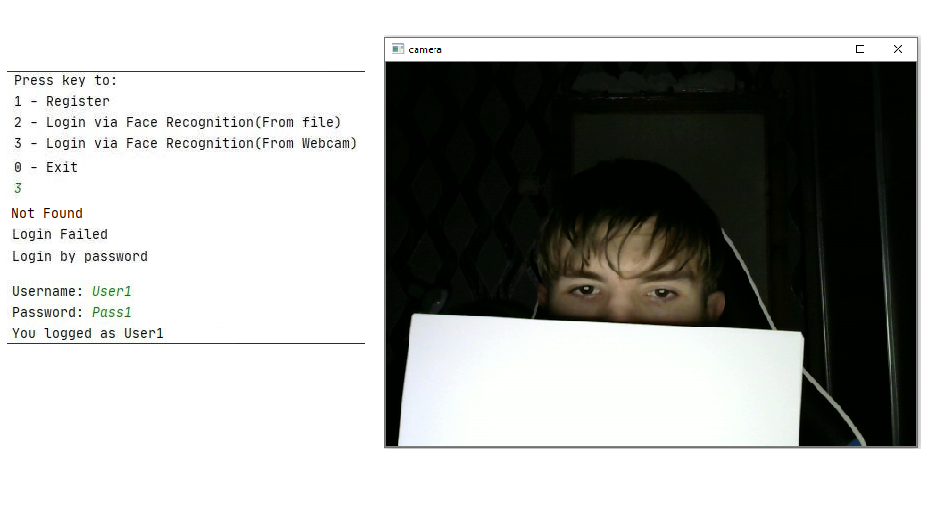


Рис 3.18 Авторизація через ввід паролю

Останній випадок, коли ми не можемо ініціалізувати веб-камеру для входу. Тоді ми обираємо пункт 2 (Login via Face Recognition(From file)). Нам буде запроновано ввести шлях до файлу(рекомендовано залишити поле пустим, тому, що шлях до фото вже прописаний в програмі), але можна обрати інше ваше фото (зроблене не веб-камерою при реєстрації), для цього потрібно буде вказати шлях.

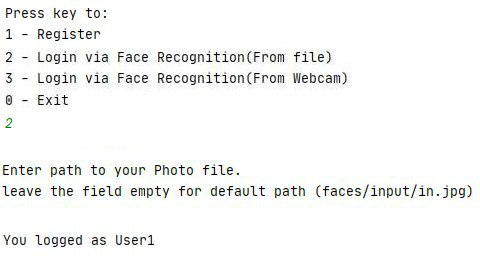


Рис 3.19 Вхід в систему з використанням фото

**ВИСНОВОК ДО ТРЕТЬОГО РОЗДІЛУ:**

В третьому розділі було розроблено програму, що може реєструвати та авторизувати користувачів використовуючи технологію розпізнавання обличчя. За основу було взято алгоритм Local Binary Patterns Histograms (LBPH) з використанням бібліотеки OpenCV. Він був покращений такими методами, як : шумозаглушення та виявлення краю. Також було додано можливість авторизації користувача з вводом паролю або фотографії. Останній варіант авторизації виглядає ненадійним, адже хтось може завантажити ваше фото і ввійти в систему(через це - в схемі роботи програми він не розглядається), тому для комерційного використання він не рекомендується, але для приватного – це досить зручний, запасний варіант. Також були проведені тестування, що довели працездатність програми в таких ситуаціях як: помилка при авторизації через веб-камеру, помилка ініціалізації веб-камери.

**ВИСНОВКИ**

В даному проекті запропонована удосконалена технологія розпізнавання обличчя, яка за допомогою врахування особливостей роботи алгоритму LBPH, бібліотеки OpenCV, мови Python та середовища Pycharm, показала швидкодію роботи усього алгоритму.

За рахунок використання методу Каскадів Хаара (англ. - Haar Cascade), програма з розпізнавання обличчя зможе навчатись розпізнаванню навіть із поганих зразків фото, що покращить наступний процес ідентифікації осіб. А застосування таких методів, як шумозаглушення та виявлення краю покращило здатність алгоритму - розпізнавати обличчя.

Одним із подальших шляхів вдосконалення технологій розпізнавання обличчя є застосування методів нейронного зв’язку, задля більшої швидкості та точності процесу, а також ширшого пошуку вихідних даних про особу із доступних мереж Інтернету.

# ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бойченко І. В. Identity and Access Management Authentication Management Systems.Управление идентификацией и доступом  
   Системы аутентификации // Бізнес журнал. [Режим доступу – електронне джерело] - [http://www.tadviser.ru/index.php/%D0%98%D0  
   %91\_\_%D0%90%D1%83%D1%82%D0%B5%D0%BD%D1%82%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F](http://www.tadviser.ru/index.php/%D0%98%D0%91__%D0%90%D1%83%D1%82%D0%B5%D0%BD%D1%82%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F)
2. Технологія розпізнавання обличчя. [Режим доступу - електронне джерело] - <https://securityrussia.com/blog/face-recognition.html>
3. EigenFaces. [Режим доступу - електронне джерело] - <https://delirium-00.livejournal.com/2764.html>
4. OpenCV. [Режим доступу - електронне джерело] - <http://masters.donntu.org/2018/fknt/makogon/library/article05.htm>
5. LBPH. [Режим доступу - електронне джерело] - <https://www.machinelearningmastery.ru/face-recognition-how-lbph-works-90ec258c3d6b/>
6. Антончик А. В, Дерюшев А. А. «Обзор методов распознавания лица на изображении». Белорусский государственный университет информатики и радиоэлектроники. / УДК 004.931 // [Режим доступу - електронне джерело] - [https://cyberleninka.ru/article/n/obzor-metod  
   ov-raspoznavaniya-litsa-na-izobrazhenii/viewer](https://cyberleninka.ru/article/n/obzor-metodov-raspoznavaniya-litsa-na-izobrazhenii/viewer)
7. Мищенкова Е. С. «Сравнительный анализ алгоритмов распознавания лиц» / Труды молодых ученых / УДК 004.021, ББК 32 // [Режим доступу - електронне джерело] - <https://cyberleninka.ru/article/n/sravnitelnyy-analiz-algoritmov-raspoznavaniya-lits/viewer>
8. 10 приемов обработки фотографий [Електронний ресурс] – Режим доступу до ресурсу: https://l-a-b-a.com/blog/show/322.
9. Swaroop C. H. A Byte of Python [Електронний ресурс] / Swaroop – Режим доступу до ресурсу: http://wombat.org.ua/AByteOfPython/AByteofPythonRussian-2.02.pdf.
10. Rosebrock A. OpenCV Face Recognition [Електронний ресурс] / Adrian Rosebrock. – 2018. – Режим доступу до ресурсу: <https://www.pyimagesearch.com/2018/09/24/opencv-face-recognition/.\>
11. Творошенко І. С. / Конспект лекцій, «Цифрова обробка зображень» / м. Харків – ХНУМГ ім. О. М. Бекетова,- 2017р. // [Режим доступу - електронне джерело: <https://eprints.kname.edu.ua/45649/1/2017%20%D0%BF%D0%B5%D1%87%2032%D0%9B%20_%D0%BA%D0%BE%D0%BD%D1%81%D0%BF%D0%B5%D0%BA%D1%82%20%D0%BB%D0%B5%D0%BA%D1%86%D1%96%D0%B9_%D0%A6%D0%9E%D0%97_4%20%D0%BA%D1%83%D1%80%D1%81_%D0%A2%D0%B2%D0%BE%D1%80%D0%BE%D1%88%D0%B5%D0%BD%D0%BA%D0%BE.pdf>

# ДОДАТКИ

**Схема Алгоритму**

*ДП.6406.04.000 Д1*

*НТУУ «КПІ імені Ігора Сікорського», ФІОТ*

*Група ІО - 64*

*Літ.*

*Аркуш*

*Дата*

*Підп.*

*№ документа*

*Н.контр.*

*Затв.*

*Зм.*

T

1. 66

1

*Реєстрація Користувачів з використанням технологій розпізнавання обличчя*

*Додаток А*

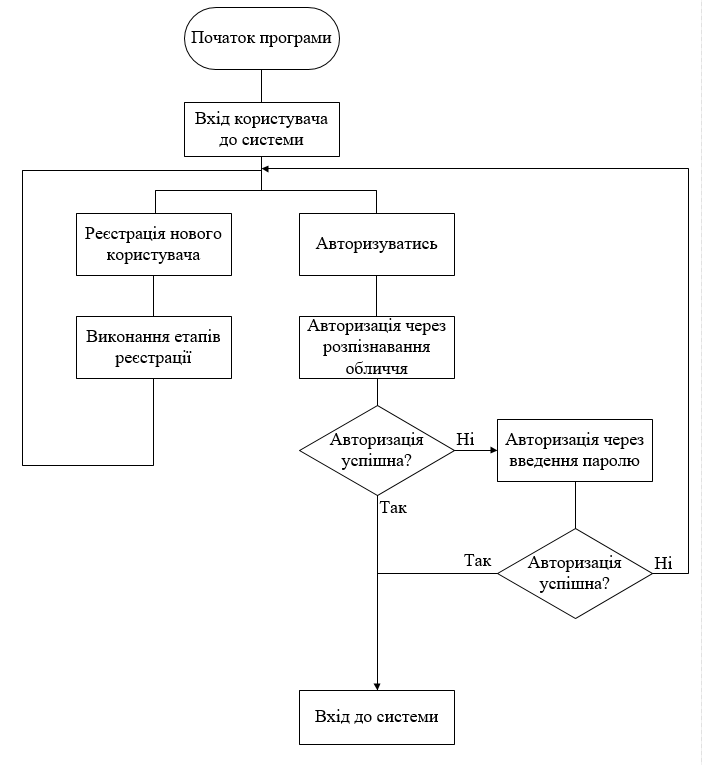
# в

# Арк.

# Перевірив

# Аркушів

*Симоненко В.П.*



**Функціональна Схема**

*ДП.6406.05.000 Д2*

*НТУУ «КПІ імені Ігора Сікорського», ФІОТ*

*Група ІО - 64*

*Літ.*

*Аркуш*

*Дата*

*Підп.*

*№ документа*

*Н.контр.*

*Затв.*

*Зм.*

T

1

1

*Реєстрація Користувачів з використанням технологій розпізнавання обличчя*

*Додаток Б*

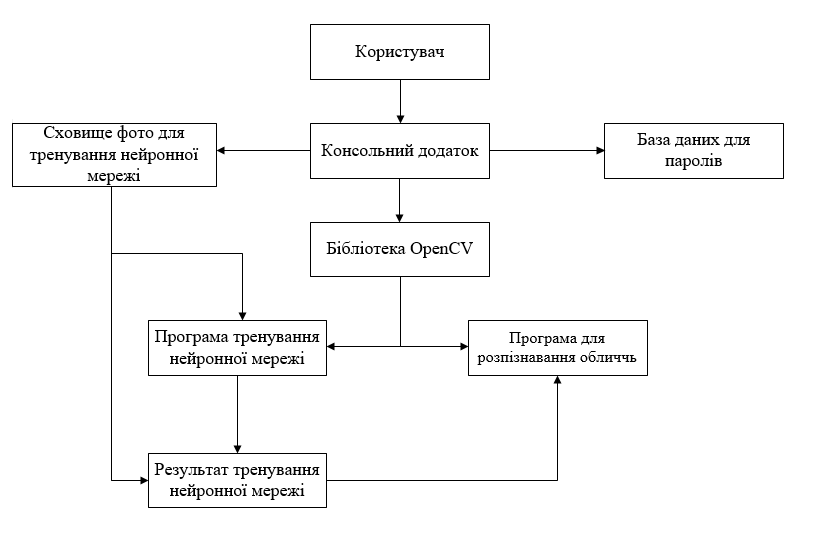
# в

# Арк.

# Перевірив

# Аркушів

*Симоненко В.П.*



**Діграма Класів**

*ДП.6406.06.000 Д3*

*НТУУ «КПІ імені Ігора Сікорського», ФІОТ*

*Група ІО - 64*

*Літ.*

*Аркуш*

*Дата*

*Підп.*

*№ документа*

*Н.контр.*

*Затв.*

*Зм.*

T

1

1

*Реєстрація Користувачів з використанням технологій розпізнавання обличчя*

*Додаток В*

# в

# Арк.

# Перевірив

# шв

*Симоненко В.П.*

