

Спеціальність 122 «Комп'ютерні науки»

**«Система розпізнавання обличчя»**

## Зміст

<i>Анотація</i> .....	3
<i>Список аббревіатур та скорочень</i> .....	4
<i>Вступ</i> .....	5
<i>1 Аналіз літературних джерел</i> .....	8
1.1 Поняття комп'ютерного зору .....	8
1.2 Аналіз технології розпізнавання обличчя .....	8
1.3 Сфери використання CV .....	14
Висновки до розділу 1 .....	15
<i>2 Формальна постановка задачі побудови системи розпізнавання обличчя</i> ..	17
2.1 Основні завдання системи .....	17
2.2 Аналіз методів розпізнавання обличчя .....	18
2.3 Розроблений метод розпізнавання обличчя .....	19
2.4 Порівняння відомих рішень для розпізнавання .....	26
Висновки до розділу 2 .....	30
<i>3 Вибір програмних засобів</i> .....	32
Висновки до розділу 3 .....	33
<i>4 Практична реалізація та аналіз результатів</i> .....	34
4.1 Архітектура системи .....	34
4.2 Аналіз контрольного прикладу .....	36
Висновки до розділу 4 .....	38
<i>Висновки</i> .....	39
<i>Список літератури</i> .....	40
<i>ДОДАТОК</i> .....	44

## Анотація

**Система розпізнавання обличчя** – це проект, який зроблений за допомогою технології комп’ютерного зору (CV – computer vision). У роботі описано функціональні модулі системи для розпізнавання обличчя людини. Було досліджено уже існуючі програми такого типу та виділені їхні переваги та недоліки. Було проаналізовано актуальні методи та засоби створення системи розпізнавання обличчя людини, здійснено їх характеристику. Також здійснено вибір необхідної архітектури та вибір програмного вирішення поставленої задачі.

За темою роботи опубліковано 3 наукових праці:

1. Boyko, N., Basystiuk, O., & Shakhovska, N. (2018, August). Performance Evaluation and Comparison of Software for Face Recognition, Based on Dlib and Opencv Library. In 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP) (pp. 478-482). IEEE (*індексується в наукометричній базі даних Scopus*).
2. Rybchak Z. Analysis of computer vision and image analysis technics / Z. Rybchak, O. Basystiuk // ECONTECHMOD: an international quarterly journal on economics of technology and modelling processes. – Lublin: Polish Academy of Sciences, 2017. – Vol. 6, № 2. – P. 79-84.
3. Шаховська, Н. Б., & Басистюк, О. А. (2017). Розпізнавання обличчя за допомогою алгоритмів машинного навчання. *Штучний інтелект*, (3-4), 84-93.

Дослідження використано у створенні методичних вказівок до виконання лабораторних робіт для студентів першого (бакалаврського) рівня вищої освіти спеціальності 122 “Комп’ютерні науки”, Інтелектуальний аналіз даних. Ч. 2. Дослідження засобів прогнозування з використанням апарату штучних нейронних мереж.

## Список аббревіатур та скорочень

API	прикладний програмний інтерфейс (application programming interface)
CV	комп'ютерний зір (Computer Vision)
DCNN	глибинна згорткова нейронна мережа (deep convolutional neural network)
БД	база даних

## Вступ

У все більшій кількості сучасних програмних засобів усе частіше зустрічаються системи, у проектуванні і розробці яких був задіяні методи штучного інтелекту, а саме алгоритми машинного навчання. Такі системи знаходять своє місце навіть у повсякденному житті звичайних людей, а саме у пошукових системах, соціальних мережах, мобільних телефонах. Одні з найкращих результатів машинного навчання досягнуто у завдання розпізнавання, зокрема у розпізнаванні обличь. Більшість з таких реалізацій застосовуються у системах захисту на різних платформах. Проте також дуже поширене є застосування у сфері розваг.

Основними перевагами системи розпізнавання обличчя є:

- Автоматизація процесу ідентифікації особи. Забезпечує економію часу при ідентифікації особи. Система дозволяє зменшити затрати на ресурси та знизити ризики маніпуляції даними, оскільки всі потрібні дані зберігаються у базі даних.
- Збереження всіх транзакцій в базі даних (БД). Забезпечує можливість перегляду архіву роботи системи. Ця функція є надзвичайно корисною, тому що за її допомогою можна переглянути всі дані та у разі проблем знайти причину.
- Збереження даних про авторизованих у системі осіб. У базі даних міститься інформація про людей, які авторизувалися за допомогою систем. Це забезпечує можливість контролю за всіма транзакціями, проведеними певним користувачем, тобто дозволить відслідковувати усі процеси які відбуваються у системі, у випадку неполадок швидко їх виправляти.
- Розмежування прав доступу. Забезпечує різні права доступу до певних модулів системи. Корисне тим, що користувачі не мають прямого доступу до бази даних та не можуть та вносити туди зміни. У свою чергу, адміністрація має доступ до бази даних.

- Можливість винести базу даних відомих людей у відкритий доступ у мережі. Оскільки база даних може знаходитися на віддаленому сервері, то є змога розробити окрему базу даних для системи розпізнавання в розподіленому режимі, в якій будуть зберігатись дані по всіх відомих обличчях.

**Метою роботи** є дослідження перспективних напрямків і технологій комп'ютерного зору для пошуку, аналізу і розпізнавання обличчя та розроблення системи розпізнавання обличчя на основі проведених досліджень, порівняння ефективності різних варіацій локальних бінарних шаблонів щодо завдання розпізнавання обличчя в реальному часі.

Для досягнення поставленої мети необхідне рішення наступних завдань:

1. Огляд та аналіз існуючих рішень;
2. Дослідження вимог, методів і алгоритмів вирішення поставленого завдання;
3. Модифікація існуючих методів розпізнавання обличчя та порівняння з існуючими;
4. Розроблення структури програмного забезпечення;
5. Створення прикладної програми – система Розпізнавання обличчя.

**Об'єктом досліджень** є процес розпізнавання обличчя людини.

**Предметом досліджень** є методи та засоби опрацювання зображень за допомогою технологій комп'ютерного зору.

**Методи досліджень.** Для досягнення поставленої мети використано: методи об'єктно-орієнтованого аналізу та проектування – для створення моделі розпізнавання та проектування структури програмного засобу.

**Наукова новизна** роботи полягає у

Наукова новизна полягає в наступному:

1. Вперше запропоновано поєднання методу опорних векторів та нормалізації зображення в якості алгоритма розпізнавання обличчя.

2. Виділено 68 орієнтирів для розпізнавання особливих характеристик та позиціонування обличь, що стало основою для подальшого опрацювання зображень обличь.

3. Показано, що для центрально-симетричного LBP оператора оптимальним розбиттям зображення на блоки по відношенню витрат пам'яті та ефективності розпізнавання являється 4x4.

**Практична цінність** полягає у створенні системи, яка дозволить розпізнавати обличчя і вести їхній облік із збереженням результатів розпізнавання та можливістю їх перегляду і редагування, який відрізняється від аналогів вищою ефективністю. Дослідження використано у створенні методичних вказівок до виконання лабораторних робіт для студентів першого (бакалаврського) рівня вищої освіти спеціальності 122 “Комп’ютерні науки”, Інтелектуальний аналіз даних. Ч. 2. Дослідження засобів прогнозування з використанням апарату штучних нейронних мереж.

# **1 Аналіз літературних джерел**

## **1.1 Поняття комп'ютерного зору**

Комп'ютерний зір (англ. CV, computer vision) – технології, націлені на створення комп'ютерних засобів, які можуть проводити виявлення, стеження та класифікацію об'єктів.

Аналіз, прогнозування, розпізнавання вийшли на новий рівень із застосуванням технологій штучного інтелекту і машинного навчання. Останніми роками надзвичайно перспективною науковою областю є комп'ютерний зір.

Одна з найпопулярніших проблем комп'ютерного зору, яка зараз активно досліджується – розпізнавання обличчя. І останні розробки технологічних лідерів світу цьому доказ. У вересні 2017 року на Apple Special Event була представлена технологія Face ID [1]. Ця технологія розпізнавання активно впроваджена у соціальних мереж для позначення людей на фотографіях. Компанія Facebook є лідером цього напрямку. Алгоритм, розроблений компанією Facebook, показує ефективність розпізнавання у 93%.

Результатами широкого використання Face ID стало створення різного роду бібліотек та Application program interface (API) для розпізнавання обличчя. Розроблені рішення поділяються на спеціально для якоїсь області чи універсальні, написані під певну мову програмування чи з підтримкою всіх популярних мов. Часто серед усього різноманіття засобів важко зрозуміти, який з них підходить найкраще для вирішення конкретної проблеми.

Робота присвячена проблематиці комп'ютерного зору, а саме розпізнавання обличчя на основі бібліотек OpenCV [2] та dlib [3]. Під час виконання роботи оцінюватимуться переваги і недоліки кожної із них та доцільність використання цих бібліотек у проектах побудови систем розпізнавання.

## **1.2 Аналіз технології розпізнавання обличчя**

Проаналізуємо відомі рішення для задачі розпізнавання обличчя на основі машинного навчання, такі як:



- Facebook DeepFace,
- Apple Face ID,
- Microsoft Face API,
- Aware NexaFace,
- Samsung Face Recognition.

Сервіс **Facebook DeepFace** [4] призначений для допомоги користувачеві в соціальній мережі Facebook. Сервіс бере за основу опубліковані користувачами фотографії, на яких позначені люди, шукає на них обличчя і підв'язує до них імена, які були вказані користувачами. При наступних публікаціях фотографій системи DeepFace підкаже людину яка зображена на фотографії або нагадає вам, що можна позначити цю людину яку ви забули позначити.

Перевагою і особливістю цієї системи розпізнавання є її ефективність розпізнавання. Розробники досягнули результату 97.25% (результат вибірки людей 97.53%) [5] правильних відповідей у тесті, який полягає у розпізнаванні двох фотографіях та визначенні чи одне і те ж обличчя зображення на них. Робота у цій системі відбувається у вікні завантаження фотографій, відображеному на рис.1.1.

Особливості: використання попарних порівнянь для розпізнавання обличь.

Переваги:

- Висока точність розпізнавання (на рівні 97.25%),
- Велика кількість даних для навчання алгоритмів пошуку і розпізнавання обличь.

Недоліки:

- Значна потреба в апаратних ресурсах,
- Адаптованість системи розпізнавання тільки під соціальну мережу Facebook.

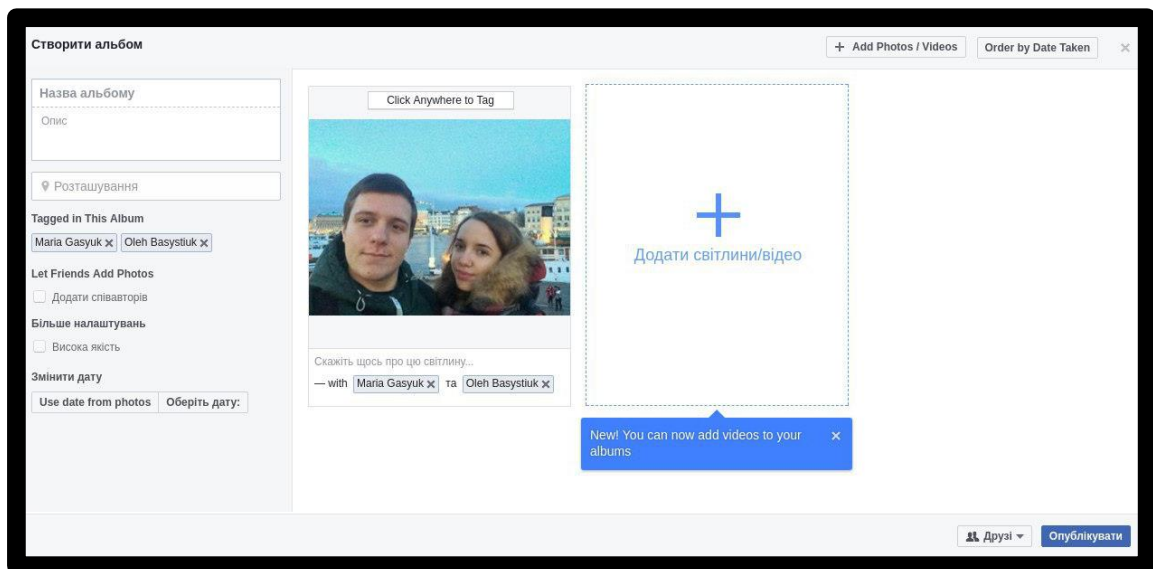


Рис.1.1. Головне вікно системи DeepFace

Система **Apple Face ID** [6] призначена для авторизації власника телефону. Результатом досліджень стало розроблення додаткової апаратної складової для системи розпізнавання, яка дозволяє відсіювати можливість обману системи розпізнавання за допомогою будь-яких 2D муляжів, а також дає змогу розпізнавати обличчя при поганому освітленні, оскільки апаратне забезпечення зчитує мітки на обличчю і порівнює з наявними [7].

Для початку роботи системи треба провести побудову макету обличчя. Обертаючи телефон навколо обличчя, за допомогою фронтальної камери і апаратного забезпечення виконується цей процес, відображений на рис.1.2. Після закінчення побудови макету система готова для роботи [7].



Рис.1.2. Основні етапи роботи системи Face ID

Особливості – унікальне апаратне забезпечення, розроблення для захисту і покращення ефективності розпізнавання обличчя.

## Переваги:

- Висока ефективність розпізнавання,
- Розроблене додаткове апаратне забезпечення,
- Захищеність системи від обману,
- Можливість розпізнавати при поганому освітлені.

## Недоліки:

- Потреба в особливому апаратному забезпеченні,
- Закрита технологія тільки для продукції Apple.

Сервіс **Microsoft Face API**. Призначена як інструмент для побудови систем розпізнавання. Містить у собі функції для пошуку, ідентифікації, групування, знаходження подібних обличчя, а також такі можливості як: визначення віку, статі, емоційного стану за допомогою обличчя [8]. Такий спосіб організації системи дозволяє надати доступ стороннім розробникам для розробки своїх додатків на основі цього API. Це звільняє розробника від потреб у апаратному забезпеченні, навчанні алгоритмів, оскільки всі ці процеси відбуваються на стороні сервера Microsoft. Проте, такий підхід робить розробку залежною від структури і відповідей цього API. Процес відправлення запиту на пошук обличчя відображено на рис.1.3.

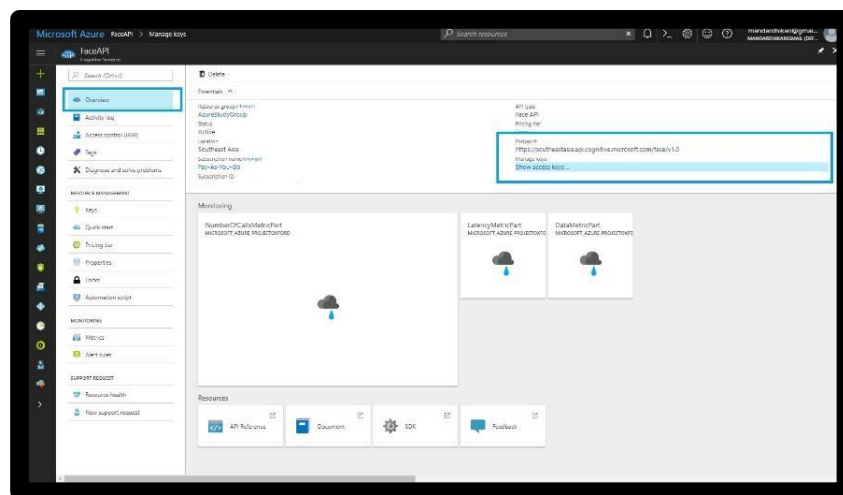


Рис.1.3. Головне вікно системи Face API

Особливості системи– задумувалась як інструмент, на основі якого розробники можуть будувати системи розпізнавання обличчя. У зв'язку з цим був створений у вигляді API.

Переваги:

- Хороша ефективність розпізнавання,
- Наявність величезної кількості засобів і методів для розпізнавання,
- Доступність,
- Усі операції відбуваються на стороні серверів Microsoft.

Недоліки:

- Не є закінченим рішенням, тобто це тільки інструмент, а не готовий продукт,
- Відсутня можливість додаткових налаштувань,
- Залежність розробки від стороннього API.

Систем Aware Nexa|Face. Розроблена у складі біометричної системи для ідентифікації і автентифікації особи. Використовується у цій системі як один із засобів підтвердження особи [9]. Окрім Nexa|Face, присутні такі системи: розпізнавання відбитків пальців Nexa|Fingerprint, сітківки ока Nexa|Iris, голосу Nexa|Voice, тексту Inquire, також можливим є підключення додаткових модулів, розроблених третіми особами.

У більшості випадків ці модулі використовують в комплексі, оскільки це гарантує додатковий захист і надійність системи. Для всього комплексу розроблена мобільна і веб версія, а для розгортання усієї системи потрібна клієнт-серверна архітектура.

На рис. 1.4. подано одне із вікон вказаної програми.

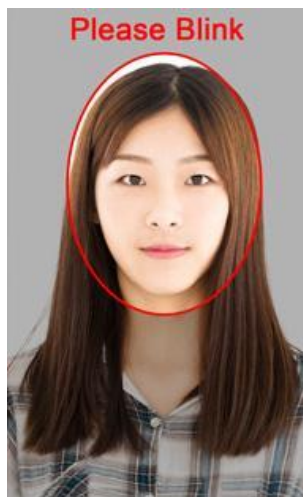


Рис.1.4. Головне вікно системи Nexo|Face

Особливості – комплексна біометрична системи з можливістю впровадження сторонніх модулів, а також кросплатформність цієї розробки.

Переваги:

- Висока ефективність розпізнавання,
- Наявність додаткових систем розпізнавання,
- Висока надійність і захищеність системи,
- Кросплатформність.

Недоліки:

- Комплексність системи,
- Висока вартість.

Система **Samsung Face Recognition** [10] розроблена інженерами Samsung для їхнього флагманської моделі S серії. Використовується як один із засобів підтвердження особи. Для розпізнавання використовує тільки фронтальна камера, тобто додаткового апаратного забезпечення немає, як у випадку з Apple Face ID. Проте, було створено спеціальну захищену папку (Secure Folder) і систему Кнох, які призначені для захисту важливих даних та їх інкапсуляції на рівні системи.

Проте, через відсутність додаткового апаратного забезпечення, яке могло б збільшити надійність і захищеність процесу розпізнавання, систему можна обдурити і отримати додатковий доступ. Розробники робили більше акцент на

доступності системи і простоті використання. Також, на основі системи розпізнавання обличчя, була створена система розпізнавання зіниці ока, яка себе досить добре зарекомендувала, але має деякі проблеми у разі поганого освітлення.

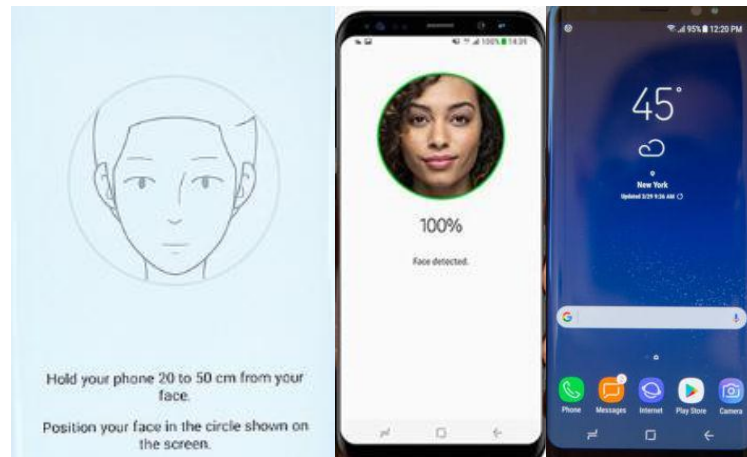


Рис.1.5. Головне вікно системи Samsung Face Recognition

Особливості – розроблена системи отримала свій подальший розвиток у системі розпізнавання зіниці.

Переваги:

- Висока ефективність розпізнавання,
- Наявність розпізнавання за допомогою зіниці,
- Додатковий шар захисту за допомогою Кнох системи і захищеної папки.

Недоліки:

- Можливість хибного розпізнавання,
- Проблема з розпізнаванням при поганому освітленні,
- Закрита технологія тільки для продукції Samsung.

### 1.3 Сфери використання CV

Можливості використання CV-технологій практично безмежні, комп'ютерний зір може застосовуватися майже в усіх аспектах нашого життя.

Прикладами систем застосування комп'ютерного зору є [11, 12]:

**Виробництво.** Комп'ютерний зір вже сьогодні знаходить застосування у великому і малому виробництві. Основною сферою застосування є

промислові роботизовані системи для автоматизації і контролю якості виробництва.

**Безпека.** Наведені вище системи FaceID та Nexa|Face – це прямі приклади систем безпеки, побудованих з використанням новітніх технологій комп'ютерного зору для розпізнавання обличчя. Основна сфера застосування – відслідковування і автоматизація доступу до будь-яких об'єктів.

**Транспорт.** Успіхи компанії Tesla у побудові системи автономного керування автомобілем. Основна сфера застосування – побудова системи безпечного і автономного керування автомобілем за будь-яких умов, без участі людини у процесі керування.

**Шопінг.** Компанія Amazon вже досить давно працює над розумними і автономними супермаркетами, без касирів, а також над автономними сервісами доставки товарів. Основна сфера застосування – підвищення економічної ефективності сфери торгівлі та підвищення якості й зменшенні ціни за рахунок повної автоматизації послуг з доставки товарів замовнику.

**Медицина.** Системи для аналізу медичних зображень і автоматизації оцінки результатів досліджень та надання базових приписів і рекомендацій по лікуванню на основі аналізу наявних зображень різного роду досліджень.

**Системи взаємодії.** Пристрої аналізу, розпізнавання і введення інформації для систем людино-машинної взаємодії. Основна сфера застосування – розуміння жестів, емоцій, будь-якої іншої візуальної інформації.

**Системи організації інформації.** Засоби роботи з зображеннями, наприклад системи для індексації баз даних зображень.

### ***Висновки до розділу 1***

Сфера використання технології комп'ютерного зору надзвичайно широка і дотична майже до усіх сфер життєдіяльності людини. Велика кількість систем

на ринку демонструє затребуваність і важливість розвитку цих технологій для підвищення рівня безпеки, зниження затрати виробництва, підвищення якості медичних послуг тощо.

Оцінюючи плюси і мінуси проаналізованих систем розпізнавання обличчя, констатуємо, що на ринку представлені переважно системи, які вимагають великих ресурсів для навчання, і тому можуть використовуватися великими компаніями. Рішення, які можуть використовуватися невеликими компаніями, показують не дуже високу точність розпізнавання. Також виявлено залежність якості розпізнавання від освітлення, позиціонування обличчя тощо. Тому вдосконалення методу розпізнавання обличчя є актуальною задачею.



## 2 Формальна постановка задачі побудови системи розпізнавання обличчя

### 2.1 Основні завдання системи

Процедура розпізнавання обличчя передбачає виконання таких кроків [13, 14]:

- **Знайти обличчя** – виникає незалежно від того, чи маємо завдання розпізнавати людей за фотографіями, чи розпізнати по відеоряду, чи у будь-який інший спосіб;
- **Позиціонування обличчя** – не часто зустрічаються фотографії, на яких людина стоїть прямо обличчям до об'єктива, найчастіше обличчя знаходиться повернуте. Тому виникає завдання позиціонувати обличчя так, ніби фотографія зроблена прямо;
- **Визначення унікальних особливостей обличчя** – цей крок можна назвати вже повноцінним етапом розпізнавання обличчя (попередні були підготовчі). Н цьому етапі аналізуємо зображення і отримуємо унікальні цифрові значення обличчя;
- **Ідентифікація особи** – порівнюємо отримані дані з уже наявними, якщо знаходимо подібні дані, виводимо ім'я особи, якщо ні, відповідно маємо незнайому ще нам особу.

У роботі реалізуємо усі кроки побудови системи розпізнавання обличчя з допомогою різних бібліотек, порівнюємо результати розпізнавання, отримані з використанням цих бібліотек, а також визначимо швидкодію роботи кожного етапу у різних бібліотеках комп'ютерного зору.

Результати досліджень можуть різнитися в залежності від конфігурації комп'ютера, на якому виконуватиметься розпізнавання. Також усі бібліотеки, які використанні для досліджень, налаштовані в режимі використання тільки CPU, без GPU(NVIDIA Cuda).

Перш за все, здійснимо аналіз існуючих методів розпізнавання обличчя та розроблення власного методу.

## 2.2 Аналіз методів розпізнавання обличь

Проаналізуємо методи, які застосовують для розпізнавання обличь.

Численні застосування показали хороші результати методів, що базуються на лінійній проєкції, такі як аналіз основних компонентів (PCA) [15], незалежний аналіз компонентів (ICA) [16], лінійний дискримінаційний аналіз (LDA) [17, 18], 2DPCA [19] та лінійний класифікатор регресії (LRC) [20]. Однак, через великі розбіжності в умовах освітлення, виразності обличчя та інші фактори ці методи можуть неадекватно представити обличчя. Основна причина полягає в тому, що лінії обличчя лежать на складному нелінійному і не опуклому колекторі у просторовому просторі.

Для вирішення таких випадків були запропоновані нелінійні розширення, такі як ядро PCA (KPCA), LDA ядра (KLDA) [21] або локальне лінійне вбудовування (LLE) [22]. Найбільш нелінійні методи, що використовують методи ядра, використовуються для відображення вхідних зображень обличчя у простір вищої розмірності, в якому колектор граней лінійний та спрощений. Тож можна застосовувати традиційні лінійні методи.

Хоча PCA, LDA та LRC розглядаються як алгоритми лінійного підпростору, помітно, що методи PCA та LDA зосереджені на глобальній структурі евклідового простору, тоді як підхід LRC орієнтований на локальну структуру колектора. Ці методи спрямовані на лінійний підпростір, що охоплюється зображеннями власних поверхонь. Відстань від простору обличчя – це ортогональна відстань до площини, тоді як відстань у просторі – це відстань уздовж площини від середнього зображення. Ці обидві відстані можна перетворити на відстані Махаланобіса і давати ймовірнісні інтерпретації [23].

Далі були розроблені: KPCA [24], ядро CA [25] та узагальнений лінійний дискримінантний аналіз [26]. Незважаючи на сильну теоретичну основу методів на основі ядра, практичне застосування цих методів у проблемах розпізнавання, однак, не дає значного поліпшення порівняно з лінійними методами.

Тому розроблено ще одне сімейство нелінійних методів проєкції. Вони успадкували простоту від лінійних методів та можливість обробляти складні дані

з нелінійних. Серед цих методів варто відзначити LLE [27] та збереження проєкції місцевості (LPP) [28]. Вони розробляють схему прогнозування лише для навчальних даних, але їх спроможність проєктувати нові елементи даних доволі низька.

У другій категорії методів знаходяться репрезентативні методи. Вони шукають особливості кожного зображення і в цьому мають певні переваги перед цілісними ознаками. Ці методи є стійкішими до локальних змін, таких як експресія, оклюзія та нерівність. Загальний репрезентативний метод – це локальні бінарні шаблони (LBP) [29,30]. Суміжні зміни навколо центрального пікселя простим, але ефективним способом описуються за допомогою LBP. Це інваріантне монотонне перетворення інтенсивності, також підтримуються невеликі варіації освітлення. Багато варіантів LBP пропонується для поліпшення вихідних LBP, таких як гістограма фазових моделей Габора [31] та локальна послідовність гістограм бінарної картини Габора [32,33]. Як правило, LBP використовується для спільного моделювання сусідніх елементів у просторовій, частотній та орієнтаційній областях [34].

### 2.3 Розроблений метод розпізнавання обличчя

Один з популярних алгоритмів пошуку обрисів обличчя на зображенні є алгоритм HOG, винайдений в 2005 році Навітаном Далалом (Navneet Dalal) і Білом Трігсом (Bill Triggs) [35]. Робота алгоритму складається з таких кроків:

- **Розрахунок значення градієнта** - Найпоширенішим способом розрахунку є використання одновимірної центрованої маски з подальшим фільтруванням кольору або насиченості зображення з використанням таких фільтрувальних ядер:

$$[-1, 0, 1] \text{ і } [-1, 0, 1]^T$$

- **Побудова гістограмних комірок** – комірки можуть бути прямокутної чи круглої форми, канали гістограми рівномірно розподіляються від 0 до 180 або від 0 до 360 градусів, в залежності від того, вираховується “без знаковий”(unsigned) чи “знаковий”(signed) градієнт [36]. Розробники

алгоритму визначили, що для розпізнавання обличчя якнайкраще підходить беззнаковий градієнт, використаний разом з 9-ма каналами гістограми.

- **Локальна нормалізація** – щоб визначити зміни освітленості та контрастності, сила градієнта повинна бути локально нормалізованою, що вимагає об'єднання комірок у більші блоки. Ці блоки, як правило, перекриваються, що означає, що кожна комірка неодноразово вноситься до фінального дескриптора. Існують дві основні геометрії блоків: прямокутні блоки R-HOG і круглі блоки C-HOG. Експериментальним шляхом було досліджено, що найоптимальніший для нашого завдання є спосіб об'єднання чотирьох комірок по 8x8 пікселів у один блок, тобто 16x16 пікселів на блок [37].
- **Нормалізація** – після об'єднання блоків, їхні значення треба нормалізувати, для уникнення впливу неякісних зображень. Існує чотири методи нормалізації, найкраще себе зарекомендував такий метод [36]:

- 

$$\text{L2-norm: } f = \frac{V}{\|V\|_2^2 + e^2},$$

де:

- $V$  – не нормалізований вектор блоків,
- $\|V\|_k$  – k-норма вектора,
- $k$  – достатньо константа(точне значення нам неважливе).

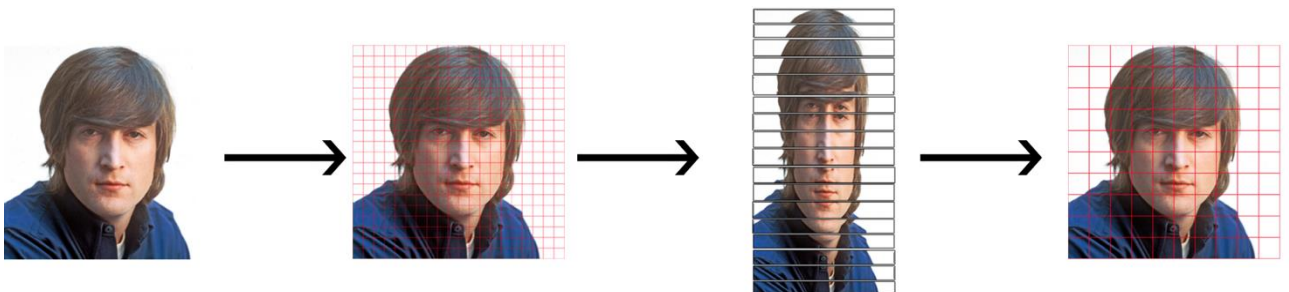


Рис.2.1. Спрощена візуалізація роботи алгоритму HOG

Після знаходження обличчя на зображенні, перед нами постає завдання його позиціонування, оскільки на переважній більшості зображень обличчя не є центровані, а знаходяться повернуті або від кутом, що погіршуватиме якість розпізнавання в подальшому. Для вирішення цієї проблеми, використовують алгоритм face landmark estimation (оцінки орієнтирів обличчя) [35]. Основна ідея полягає в пошуку 68 орієнтирів (landmarks), які присутні на кожному обличчі – верхня частина підборіддя, внутрішній край брів, зовнішній край очей, нижня точка носу, також верхня і нижня точка губи і тд. Після знаходження цих орієнтирів за допомогою простих маніпуляцій (повертання, збільшення або зменшення), отримуємо позиціоноване по центрі обличчя.

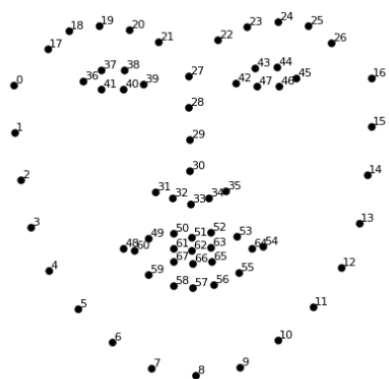


Рис.2.2. Усі 68 точок-орієнтирів, які розпізнаватиме алгоритм на обличчі

Для позиціонування ми використовуватимемо додаткову бібліотеку OpenFace.

Тепер ми дійшли до етапу, який безпосередньо пов'язаний із задачею розпізнавання обличчя. Щоб розрізняти обличчя, нам потрібно знайти у кожному з них особливості. І перше питання, яке перед нами постає, як організувати розпізнавання. Найпростіший спосіб розпізнати обличчя – це порівнювати невідоме обличчя, яке було отримано на другому кроці, зі усіма наявними в нашій базі даних обличчями. Відповідно, під час кожного розпізнавання ми звертаємося до бази даних. Ідея, на перший погляд, хороша, проте, якщо об'єми даних величезні, то це збільшуватиме час вибірки, додатково варто розглядати варіант винесення цих даних в хмару. Якщо ми говоримо про насправді велику

кількість даних (терабайти і більше), це породжує проблему затримок між запитом до серверу і відповідями з нього. А як зазначено у [2, 38], для більшості проектів із розпізнавання, швидкість розпізнавання є на першому місці. Тому такий спосіб розпізнавання не може бути використаний.

Через неефективність і високу обчислювальну складність попереднього способу було запропоновано виділяти декілька основних рис обличчя. У цьому випадку постає проблема вибору цих рис. На перший погляд, здається, що розмір або колір очей, довжина носу, величина губ, форма брів – це основні характеристики, за допомогою яких людина розпізнає обличчя. Проте, як показали дослідження [38], ці характеристики, не мають жодної цінності для комп'ютерного розпізнавання обличчя. Ця проблема пов'язана з тим, що комп'ютер не може оцінити обличчя загалом, а оцінює зображення по піксельно.

Для вирішення цієї нетривіальної задачі було запропоновано використовувати глибинну згорткову нейронну мережу (deep convolutional neural network, DCNN), яка буде навчена для визначення 128 унікальних числових характеристик обличчя.

Процес навчання такої нейронної мережі працює за таким принципом:

1. Завантажити зображення обличчя особи, яка нам вже відома (класифікована).
2. Завантажити інше зображення обличчя цієї ж людини.
3. Завантажити зображення обличчя іншої людини.
4. Нейронна мережа корегує результати отриманих значень так, щоб 128 характеристик зображень, завантажених на кроці 1 і 2, були якомога ближчими, а зображення, завантажене на кроці 3, відрізнялося від них якнайбільше.

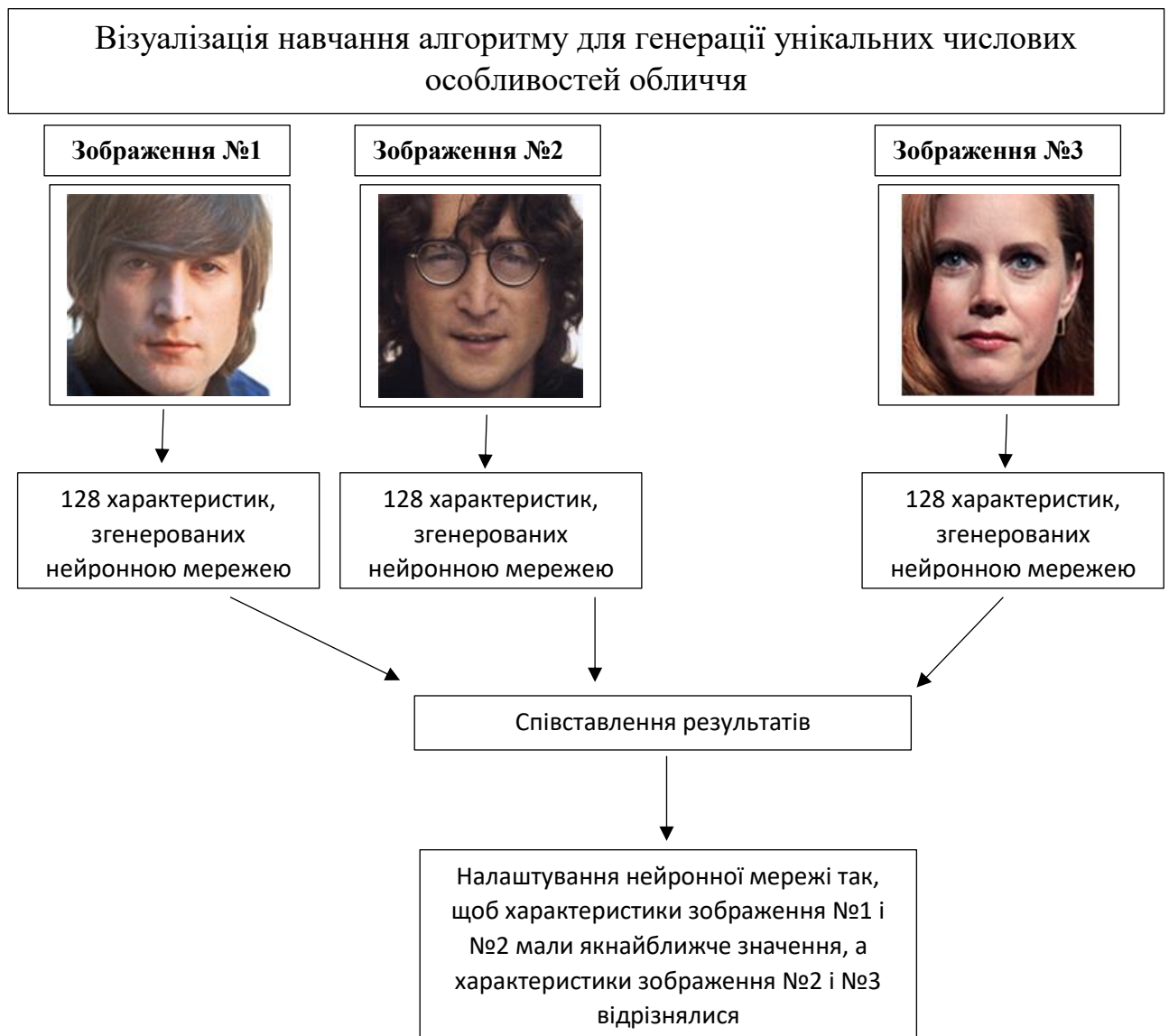


Рис.2.3. Візуалізація навчання алгоритму для визначення унікальних числових особливостей обличчя в майбутньому

Ідея скорочувати складні необроблені дані в список комп'ютерно генерованих номерів набула найбільшого розвитку саме в машинному навчанні і вперше використовувалася у сфері перекладу. Такий підхід опрацювання обличчя вперше був представлений інженерами компанії Google у 2015 році [14].

Процес тренування глибокої згорткової нейронної мережі для генерації унікальних числових характеристик обличчя є процесом складним, який вимагає величезної бази даних обличчя і значної обчислювальної здатності комп'ютера. Навіть з використанням графічної карти NVidia Telsa, оскільки для більшої

швидкості моделі тренують на відеокартах компанії NVidia з наявністю підтримки CUDA, процес навчання займав приблизно 24 години. Проте, як тільки нейронна мережа натренується, вона зможе генерувати унікальні характеристики для будь-якого обличчя, навіть для такого, яке вона ніколи не бачила до того. Отже, цей етап надзвичайно важливий. Нейронна мережа навчається тільки один раз, але від неї залежить ефективність розпізнаванні обличчя системою. У випадку, якщо немає можливості натренувати власну нейронну мережу для генерації характеристик, можна використати вже натреновану мережу [3].

Останній крок алгоритму полягає у порівнянні вже наявних даних, а саме 128 характеристик певного обличчя, які ми отримали на попередньому кроці, з наявними у нас даними про людей. Якщо ці дані співпадають, то ми можемо сказати, кого зображена на фотографії. Цей етап також може бути реалізований двома способами:

- **База даних** – як вище було описано, при великих об'ємах даних, це значно зменшуватиме продуктивність програми. Проте, як і раніше цей спосіб організації може бути використаний для маленьких об'ємів даних, коли маємо завдання ідентифікувати невелику групу людей і немає достатньої кількості фотографій для якісного тренування методу опорних векторів [16];
- **Метод опорних векторів(SVM)** – лінійний класифікатор методу опорних векторів (SVM) являє собою тренувальний набір даних з  $n$  точок вигляду [36]:

$$(\vec{x}_1, y_1), \dots, (\vec{x}_n, y_n)$$

де  $y_i \in \{1, -1\}$ , і кожен з них вказує клас, до якого належить точка  $\vec{x}_i$ . Кожен  $\vec{x}_i$  є  $r$ -вимірним дійсним вектором. Нам треба знайти «максимально розділову гіперплощину», яка відділяє групу точок  $\vec{x}_i$ , для яких  $y_i = 1$ , від групи точок, для яких  $y_i = -1$ , і визначається таким чином, що відстань між цією гіперплощиною та найближчою точкою  $\vec{x}_i$  з кожної з груп є максимальною (рис. 2.4).



Будь-яку гіперплощину може бути записано як множину точок  $\vec{x}$ , які задовольняють

$$\vec{w} * \vec{x} - b = 0$$

де  $\vec{w}$  є (не обов'язково нормалізованим) вектором нормалі до цієї гіперплощини. Параметр  $\frac{b}{\|\vec{w}\|}$  визначає зсув гіперплощини від початку координат вздовж вектора нормалі  $\vec{w}$ .

Усе, що потрібно зробити, це навчити SVM шукати найвдаліший збіг оброблюваного зображення із зображеннями з бази даних відомих обличь. Час роботи цього класифікатора займає мілісекунди, що ідеально підходить для швидкої ідентифікації обличь.

Зваживши всі за і проти, обираємо спосіб ідентифікації за допомогою методу опорних векторів. На цьому етапі також нема змісту порівнювати швидкодню, оскільки набір даних для навчання однаковий, відповідно і класифікатор створюємо один для двох бібліотек, отже на цьому етапі швидкість роботи буде однаковою для двох бібліотек.

Зазначимо особливості, які допоможуть підвищити продуктивність навчання [7]:

1. Між даними (у нашому випадку це 128 характеристик обличчя) має бути чіткі проміжки, щоб їх можна було розбити на групи, це залежить від того, наскільки добре натренований алгоритм визначення унікальних особливостей обличчя.
2. Більші обсяги навчальної вибірки не дадуть бажаного приросту швидкодії, а тільки збільшать час навчання алгоритму (так зване перенавчання, оверфітінг). Для підвищення продуктивності необхідно використовувати дані, відфільтровані від шумів. Також важливим є те, щоб дані з одного класу не перекривали дані іншого.

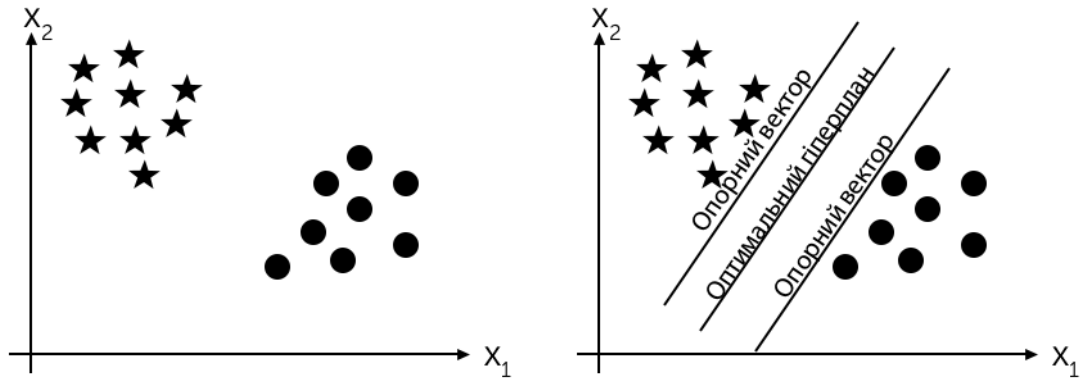


Рис.2.4. Приклад роботи SVM

Разом з SVM пропонуємо використовувати Local Binary Pattern. Математично це подано як:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p$$

Тут  $s$  – функція  $step(x)$  (крок), яка повертає 0, якщо  $x > 0$ , і 1 в іншому випадку,  $g_p$  – це значення яскравості пронумерованих сусідів пікселя,  $g_c$  – значення яскравості центрального пікселя.

Центрально-симетрична модифікація алгоритму розрахунку локальних бінарних шаблонів дає змогу скоротити витрати пам'яті і обчислювальну складність класифікації. Як граничне значення для кожного пікселя околиці приймається не значення яскравості центрального пікселя околиця, а значення яскравості протилежного відносно центра околиці пікселя. Це дає змогу скоротити кількість розрядів значень перетворень пікселів до чотирьох. Відповідно розмірність гістограми ознак зменшується до  $2^4 = 16$ . Це дає змогу використовувати таку модифікацію алгоритму для розпізнавання зображень у режимі реального часу.

## 2.4 Порівняння відомих рішень для розпізнавання

Порівняємо різні методи розпізнавання обличь. Перший нашим кроком буде пошук обличь на основі методу запропонованого вище, а саме HOG. Результатом виконання буде масив координат для кожного знайденого обличчя на фотографії.

Приклад організації цього алгоритму засобами бібліотеки dlib:

### **Face\_detector\_dlib.py:**

```
# Підключаємо системну і dlib бібліотеки
import sys
import dlib
from skimage import io

# Викликаємо базову функцію бібліотеки для пошуку обличь
face_detector = dlib.get_frontal_face_detector()

# Проходимося циклом по всіх введених зображеннях
for images in sys.argv[1:]:
    print("Processing image: {}".format(images))
    image = io.imread(images)
    faces_detected_image= face_detector (image, 1)
    io.imsave('face_recognition.jpg', faces_detected_image)
    print("Number of faces detected:{}".format ( len( faces_detected_image))

# Виводимо результат
win = dlib.image_window()
win.clear_overlay()
win.set_image(image)
win.add_overlay(dets)

#Очікує натискання кнопки "Enter" для продовження
dlib.hit_enter_to_continue()
```

### **Face\_detector\_opencv.py:**

```
# Підключаємо системну і OpenCV бібліотеки
import sys
import cv2
from skimage import io

# Викликаємо базову функцію бібліотеки для пошуку обличь
lbp_face_cascade = cv2.CascadeClassifier('data/lbpcascade_frontalface.xml')

# Проходимося циклом по всіх введених зображеннях
for images in sys.argv[1:]:
```

```

print("Processing image: {}".format(images))
image = io.imread(images)
faces_detected_img = detect_faces(lbp_face_cascade, image, scaleFactor=1.2)
io.imwrite('face_recognition.jpg', image)
print("Number of faces detected:{}".format ( len(faces))

# Виводимо результат
plt.imshow(convertToRGB(faces_detected_img))

#Очікує натискання кнопки "Enter" для продовження
cv2.waitKey(0)

```

Отримуємо такі результати виконання (таблиця 2.1)

Таблиця 2.1.

Порівняння результатів і часу, затраченого для пошуку обличь, за допомогою бібліотек OpenCV і dlib

	Експеримент №1	Експеримент №2	Експеримент №3	Експеримент №4
<b>OpenCV</b>				
<b>Час розпізнавання (сек.)</b>	0.5652	0.0934	0.6121	0.0523
<b>dlib</b>				
<b>Час розпізнавання (сек.)</b>	1.6584	0.2163	1.6889	0.1511

Переходимо до наступного кроку, на якому ми позиціонуватимемо знайдене обличчя. Для цього ми використаємо метод `face landmark estimation`, який побудуємо на основі додаткової бібліотеки `OpenFace`.

### **Face\_landmark\_estimation.py:**

```
# Підключаємо системну і OpenFace бібліотеки
import sys
import dlib
import openface
from skimage import io

# Викликаємо базову функцію бібліотеки для пошуку обличь
face_detector = dlib.get_frontal_face_detector()
face_pose_predictor = dlib.shape_predictor(predictor_model)
face_aligner = openface.AlignDlib(predictor_model)

# Проходимося циклом по всіх зображеннях і
for i, face_rectangle in enumerate(detected_faces):
    print("- Face #{} found at Left: {} Top: {} Right: {} Bottom: {}".format(i, face_rectangle.left(),
    face_rectangle.top(), face_rectangle.right(), face_rectangle.bottom()))
    pose_landmarks = face_pose_predictor(image, face_rect)
    alignedFace = face_aligner.align(534, image, face_rect, landmarkIndices =
    openface.AlignDlib.OUTER_EYES_AND_NOSE)

# Зберігаємо результат
io.imsave('aligned_face_{}.jpg'.format(i), image, alignedFace)
```

Результати виконання отримуємо такі:

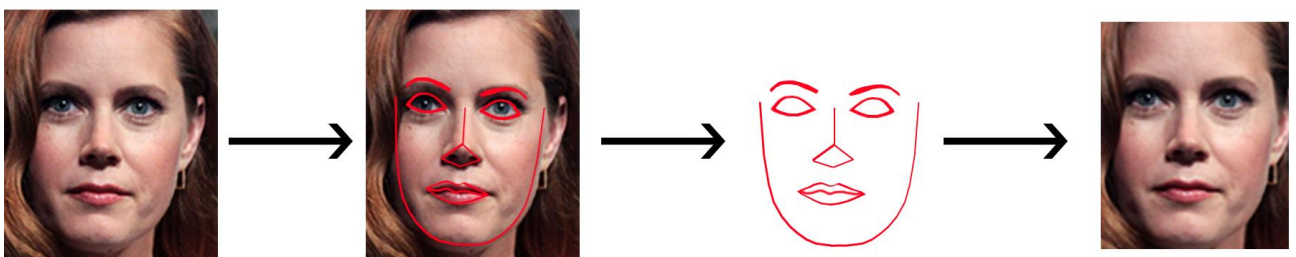


Рис.2.5. Візуалізація роботи алгоритму позиціонування обличчя

Після отримання характеристик необхідно здійснити пошук подібних характеристик з наявних даних. Як вже було описано вище, використовуватимемо для цього SVM та LBP. Ось як виглядає реалізація:

### **SVM\_example.py:**

```
import sys
import numpy as np
from sklearn import svm

x = np.vectors();
y = np.array();
x.append(np.vector([1, 2, 3, -1, -2, -3]))
y.append(+1)
x.append(np.vector([-1, -2, -3, 1, 2, 3]))
y.append(-1)

svm = dlib.svm_c_trainer_linear()
svm.be_verbose()
svm.set_c(10)

classifier = svm.train(x, y)

print("prediction for first sample: {}".format(classifier(x[0])))
print("prediction for second sample: {}".format(classifier(x[1])))

with open('saved_model.pickle', 'wb') as handle:
    pickle.dump(classifier, handle)
```

### ***Висновки до розділу 2***

У результаті аналізу методів розпізнавання обрано дві системи розпізнавання обличь, одна побудована на основі бібліотеки dlib, друга

побудована на основі OpenCV. Результати їхньої роботи подано на Рис. 2.6, де наводиться інтерфейс роботи аплікації, у якому ми бачимо розпізнане обличчя.

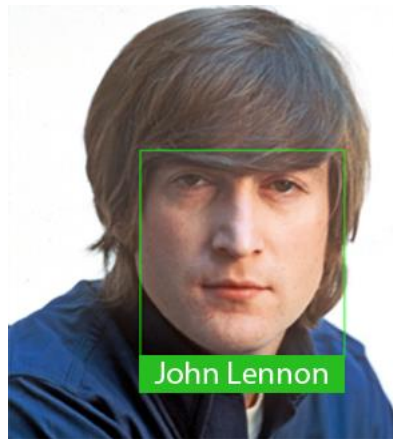


Рис.2.6. Приклад роботи системи розпізнавання обличь

На основі проведеного аналізу поданих бібліотек OpenCV та dlib встановлено, що для створення розподіленої інформаційної системи розпізнавання обличь не існує єдиних методів та технологій, які б поєднували всі етапи побудови системи. Існує величезна кількість методів для пошуку, позиціонування і загалом для організації процесу розпізнавання, які мають різну ефективність.

Вперше запропоновано поєднання методу опорних векторів та нормалізації зображення в якості алгоритма розпізнавання обличь.

### 3 Вибір програмних засобів

Для розроблення засобів з використанням методів штучного інтелекту створено велику кількість допоміжних бібліотек. Більшість з них розроблялися засобами мови Python, проте для подальшої популяризації були адаптовані під інші популярні мови програмування, такі як Java, C#, C++ і тд. Більшість бібліотек для розробки систем на основі алгоритмів машинного навчання розробляються під певну проблематику для кращої оптимізації і ефективнішого вирішення завдань цієї області У сфері комп'ютерного зору і розпізнавання обличч є дві основні бібліотеки, а саме:

- Dlib – набір інструментів C ++, який містить алгоритми машинного навчання та інструменти для створення складного програмного забезпечення на C ++ для вирішення реальних проблем. Наявність підтримки розробок для робототехніки, вбудованих пристроїв, мобільних телефонів та великих обчислювальних середовищ з високою продуктивністю. Dlib система з відкритим кодом дозволяє безкоштовно використовувати його в будь-яких розробках [39].
- OpenCV (Open Source Computer Vision Library) – бібліотека випускається за ліцензією BSD, отже, може безкоштовно використовуватися для академічних і комерційних розробок. Наявні інтерфейси для розробки з допомогою C++, Python і Java також підтримує усі популярні ОС, а саме Windows, Linux, Mac OS, iOS та Android. OpenCV була створена з високою обчислювальною ефективністю з фокусом для розробки real-time систем [40].

Оскільки маємо завдання розробити кросплатформенну систему, для цього якнайкраще підійде мова Java, оскільки код, написаний на мові Java, компілюється за допомогою JVM в проміжний бінарний формат, а потім запускається за допомогою JVM, тобто ця мова створена для кросплатформеної розробки. Найпопулярнішим IDE для розробки продуктів на мові Java є IntelliJ IDEA, також компанія розробник JetBrains надає академічну ліцензію для



студентів Національного університету «Львівська політехніка», отже у розробці використовуватиметься ця IDE.

### ***Висновки до розділу 3***

У розділі розглянуто і досліджено наявні програмні рішення для реалізації системи розпізнавання обличчя. Беручи до уваги наведені вище порівняння швидкодії відомих бібліотек, основою розробки було обрано бібліотеку OpenCV, вибір мови розробки базувався на потребах побудови кросплатформеної системи, щоб зробити можливим запуску системи на якнайбільшій кількості операційних систем.

## 4 Практична реалізація та аналіз результатів

### 4.1 Архітектура системи

Для функціонування даної системи необхідні наступні модулі:

1. Отримання зображення – відповідатиме за можливість системою отримати зображення для розпізнавання. У собі може містити як інтерфейс для введення шляху до зображення, так і працювати з камерами.
2. Опрацювання зображення – основний модуль проекрованої системи, у якому застосовуватимуться результати досліджень проведених вище. Основне завдання це отримане зображення опрацювати і виділити у ній усі наявні обличчя та відповідати зображенням з виділеними обличчями.
3. Розпізнавання обличчя – опісля системи виділяє унікальних точок-орієнтирів обличчя і проводить пошук подібних рис за допомогою SVM або у наявній базі даних.

Діаграма активності детально описує роботу системи з деталізацією по кожному модулю (Рис.4.1).

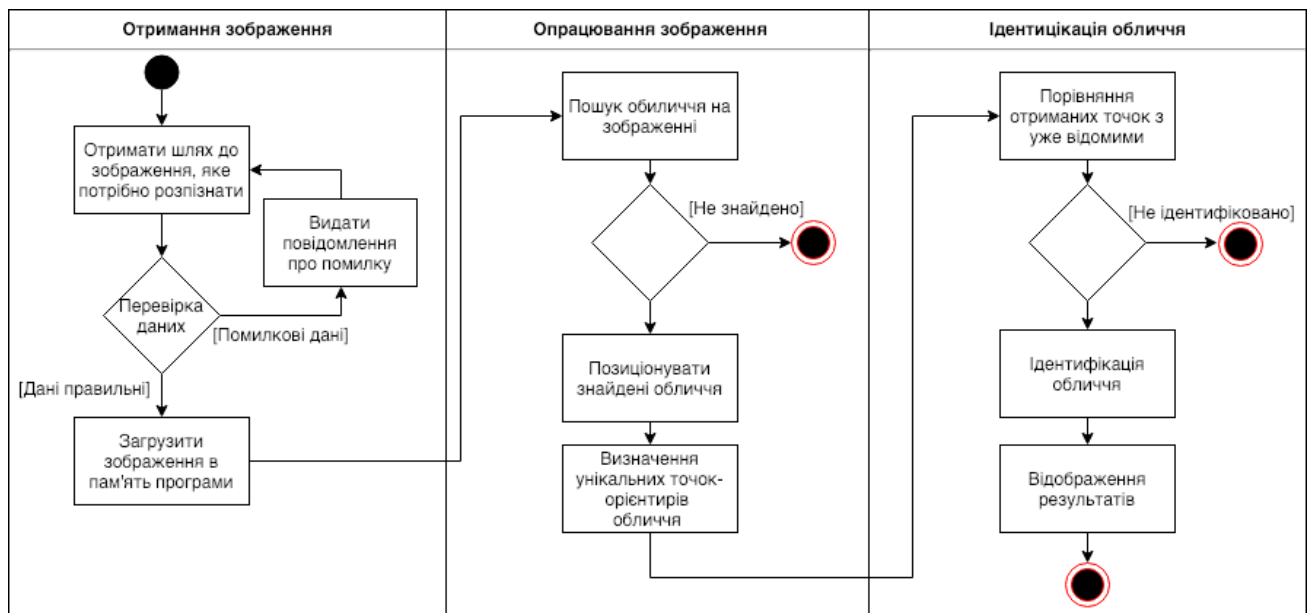


Рис.4.1. Діаграма активності

На діаграмі наявні три модуля, кожен з яких виконується лише після успішного завершення попереднього модуля. Вхідною точкою є введення даних у систему. Вихідною – виведення результатів розпізнавання обличчя.

Діаграма послідовності показує як передається фокус керування від одного модуля системи до іншого (Рис.4.2).



Рис.4.2. Діаграма послідовності.

Діаграма пакетів показує зв'язок між певними пакетами системи та визначає як вони взаємодіють між собою(Рис.4.3.). Напряму користувач працює з пакетом який містить в собі модулі графічного інтерфейсу. Цей пакет взаємодіє з усіма іншими передаючи їм дані. Конектор забезпечує з'єднання з сервером на якому міститься база даних.

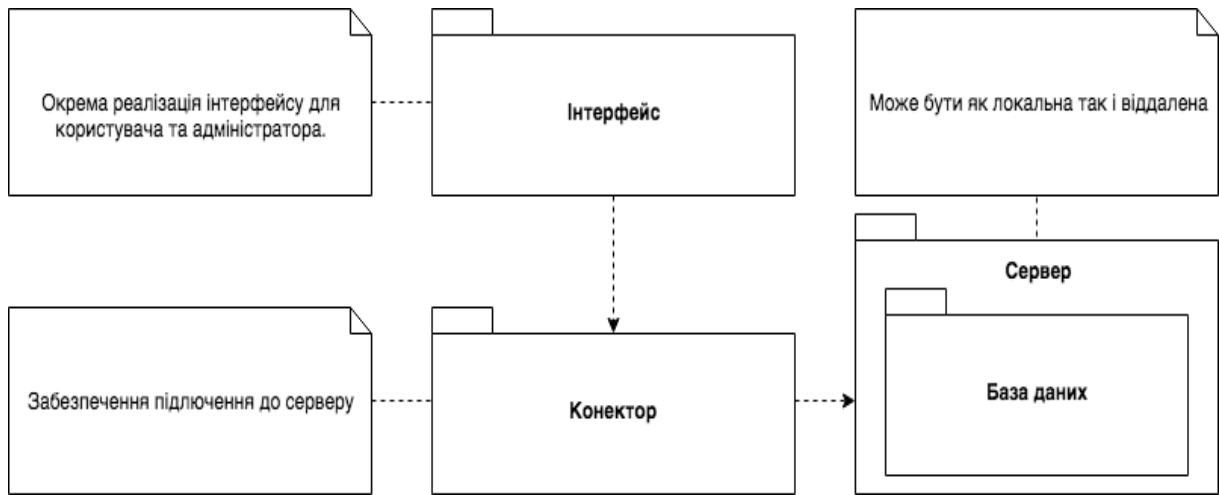


Рис.4.3. Діаграма пакетів

## 4.2 Аналіз контрольного прикладу

Наведено знімки екрану реалізованої системи.

Розроблений програмний продукт складається з базового інтерфейсу реалізованого в терміналі (Рис.4.4) і результат опрацювання введеного зображення (Рис.4.5):

```

Start recognition based on OpenCV 3.4.1
Input image path:
    
```

Рис.4.4. Запит на введення шляху до зображення

```

Start recognition based on OpenCV 3.4.1
Input image path: 3.jpg
Running DetectFaceDemo
Detected 1 faces
Writing faceDetection3.png
[ INFO:0] Initialize OpenCL runtime...
    
```

Рис.4.5. Результати опрацювання введеного зображення



Рис.4.6. Результати виконання розробленої програми



Рис.4.7. Результати виконання розробленої програми

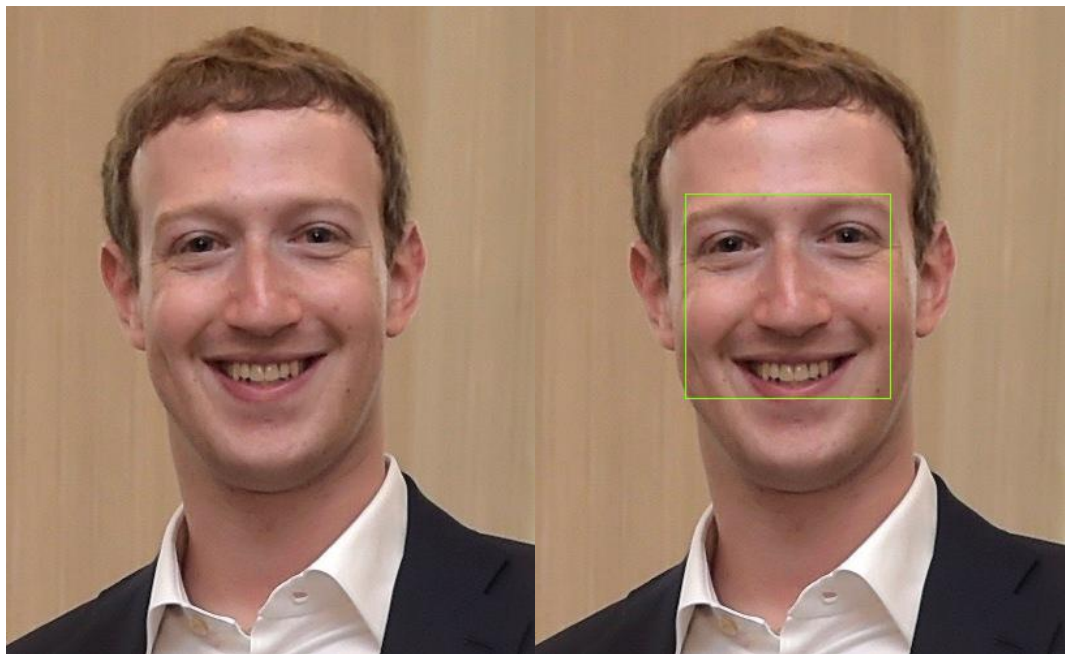


Рис.4.8. Результати виконання розробленої програми

На Рис. 4.6 – 4.8 подані результати опрацювання тестового набору даних розробленою системою. У зеленому квадраті обведено знайдене обличчя, яку передаватиметься на наступний етап вже безпосереднього співставлення з відомими обличчями.

#### ***Висновки до розділу 4***

У розділі описано архітектурну проектованої системи розпізнавання обличчя людини, також подано базовий інтерфейс програми та зображено рисунки до і після розпізнавання обличчя на зображенні, чим підтверджено працездатність створеної системи розпізнавання.

## Висновки

За результатами роботи визначено, бібліотека OpenCV продуктивніша, ніж dlib, має кращу швидкодію пошуку і розпізнавання обличь. Проведено дослідження найпопулярніших системи, які мають функціонал розпізнавання обличчя, досліджено особливості та виділено переваги та недоліки цих систем, також проведено порівняльну характеристику.

Вперше запропоновано поєднання методу опорних векторів та нормалізації зображення в якості алгоритма розпізнавання обличь. Виділено 68 орієнтирів для розпізнавання особливих характеристик та позиціонування обличь, що стало основою для подальшого опрацювання зображень обличь. Показано, що для центрально-симетричного LBP оператора оптимальним розбиттям зображення на блоки по відношенню витрат пам'яті та ефективності розпізнавання являється 4x4.

Вибрано архітектурне та програмне вирішення для реалізації майбутньої система. Реалізовані базовий набір методів системи та описано усі етапи проектування і ключові моменти побудови додатку розпізнавання обличь.

Перевагами розробленої системи є:

- **Оптимальність.** Система побудована на основі оптимальної бібліотеки комп'ютерного зору, дослідження ефективності найпопулярніших бібліотек було проведено у розділі 2.3. з детальним описом недоліків і переваг кожної з бібліотек.
- **Оптимізація і кросплатформеність.** Підтримка великої кількості операційних систем та хороша оптимізація, дозволяє запускати систему на багатьох комп'ютерах і засобах, що дає змогу набагато більшому колу користувачів користуватися розробкою.

## Список літератури

1. Face ID: [Electronic resource] – Access mode:  
[https://uk.wikipedia.org/wiki/Face\\_ID](https://uk.wikipedia.org/wiki/Face_ID)
2. OpenCV: OpenCV Tutorials [Electronic resource] – Access mode:  
<https://docs.opencv.org/2.4/doc/tutorials/tutorials.html>
3. Dlib Python API Tutorials [Electronic resource] – Access mode:  
<http://dlib.net/python/index.html>
4. Face Detection Algorithms and Techniques [Electronic resource] – Access mode:  
<https://facedetection.com/algorithms/>
5. Sominite T. Facebook Creates Software That Matches Faces Almost as Well as You Do [Electronic resource] / T. Sominite. – Access mode:  
<https://www.technologyreview.com/s/525586/facebook-creates-software-that-matches-faces-almost-as-well-as-you-do/>
6. About Face ID advanced technology [Electronic resource] – Access mode:  
<https://support.apple.com/en-us/HT208108>
7. Face ID for iPhone X [Electronic resource] – Access mode:  
<http://blog.maconline.com/face-id-iphone-x/>
8. What is the Azure Face API? [Electronic resource] – Access mode:  
<https://docs.microsoft.com/en-us/azure/cognitive-services/face/overview>
9. Aware. Biometrics Software Products [Electronic resource] – Access mode:  
<https://www.aware.com/biometrics/>
10. Samsung security. Face recognition. Iris recognition. [Electronic resource] – Access mode: : <http://www.samsung.com/uk/smartphones/galaxy-s8/security/>
11. Rybchak Z. Analysis of computer vision and image analysis technics / Z. Rybchak, O. Basystiuk // ECONTECHMOD: an international quarterly journal on economics of technology and modelling processes. – Lublin: Polish Academy of Sciences, 2017. – Vol. 6, № 2. – P. 79-84.
12. A toolkit for making real world machine learning and data analysis applications [Electronic resource] – Access mode:



[https://github.com/davisking/dlib/blob/master/python\\_examples/face\\_detector.py](https://github.com/davisking/dlib/blob/master/python_examples/face_detector.py)

13. Raja R. Face Detection Using OpenCV and Python [Electronic resource] / R. Raja. - Access mode: <https://www.superdatascience.com/opencv-face-detection/>
14. Raja R. Face Recognition Using OpenCV and Python [Electronic resource] / R. Raja. - Режим доступа: <https://www.superdatascience.com/opencv-face-recognition/>
15. Rosebrock A. Facial landmarks with dlib, OpenCV, and Python [Electronic resource] / A. Rosebrock. - Access mode: <https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/>
16. Turk, M., Pentland, A.: 'Eigenfaces for recognition', J. Cogn. Neurosci., 1991, 3, (1), pp. 71–86.
17. Bartlett, M.S., Movellan, J.R., Sejnowski, T.J.: 'Face recognition by independent component analysis', IEEE Trans. Neural Netw., 2002, 13, (6), pp. 1450–1464
18. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: 'Eigenfaces vs. \_sherfaces: recognition using class speci\_c linear projection', IEEE Trans. Pattern Anal. Mach. Intell., 1997, 19, (7), pp. 711–720
19. Hu, H., Zhang, P., De la Torre, F.: 'Face recognition using enhanced linear discriminant analysis', IET Comput. Vis., 2010, 4, (3), pp. 195–208
20. Yang, J., Zhang, D., Frangi, A.F., Yang, J.-Y.: 'Two-dimensional PCA: a new approach to appearance-based face representation and recognition', IEEE Trans. Pattern Anal. Mach. Intell., 2004, 26, (1), pp. 131–137.
21. Naseem, I., Togneri, R., Bennamoun, M.: 'Linear regression for face recognition', IEEE Trans. Pattern Anal. Mach. Intell., 2010, 32, (11), pp. 2106–2112.
22. Lu, J., Plataniotis, K.N., Venetsanopoulos, A.N.: 'Face recognition using kernel direct discriminant analysis algorithms', IEEE Trans. Neural Netw., 2003, 14, (1), pp. 117–126.
23. He, X., Yan, S., Hu, Y., Niyogi, P., Zhang, H.-J.: 'Face recognition using Laplacian faces', IEEE Trans. Pattern Anal. Mach. Intell., 2005, 27, (3), pp. 328–340.
24. Perlibakas, V.: 'Distance measures for PCA-based face recognition', Pattern

- Recognit. Lett., 2004, 25, (6), pp. 711–724
25. Kim, K.I., Jung, K., Kim, H.J.: 'Face recognition using kernel principal component analysis', *IEEE Signal Process. Lett.*, 2002, 9, (2), pp. 40–42.
  26. Bach, F., Jordan, M.: 'Kernel independent component analysis', *J. Mach. Learn. Res.*, 2003, 3, pp. 1–48
  27. Ji, S., Ye, J.: 'Generalized linear discriminant analysis: a unified framework and efficient model selection', *IEEE Trans. Neural Netw.*, 2008, 9, (10), pp. 1768–1782.
  28. Roweis, S., Saul, L.: 'Nonlinear dimensionality reduction by locally linear embedding', *Science*, 2000, 290, (5500), pp. 2323–2326.
  29. Xiaofei, H., Partha, N.: 'Locality preserving projections'. *Int. Conf. on Advances in Neural Information Processing Systems (NIPS'03)*, 2003, pp. 153–161.
  30. Ahonen, T., Hadid, A., Pietikäinen, M.: 'Face description with local binary patterns: application to face recognition', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2006, 28, (12), pp. 2037–2041.
  31. Suruliandi, A., Meena, K., Reena, R.R.: 'Local binary pattern and its derivatives for face recognition', *IET Comput. Vis.*, 2012, 6, (5), pp. 480–488.
  32. Zhang, B., Shan, S., Chen, X., Gao, W.: 'Histogram of Gabor phase patterns (HGPP): a novel object representation approach for face recognition', *IEEE Trans. Image Process.*, 2007, 16, (1), pp. 57–68.
  33. Yang, B., Chen, S.: 'A comparative study on local binary pattern (LBP) based face recognition: LBP histogram versus LBP image', *Neurocomputing*, 2013, 22, pp. 620–627.
  34. Zhang, B., Gao, Y., Zhao, S., Liu, J.: 'Local derivative pattern versus local binary pattern: face recognition with high-order local pattern descriptor', *IEEE Trans. Image Process.*, 2010, 19, (2), pp. 533–544.
  35. Cevikalp, H., Neamtu, M., Wilkes, M., Barkana, A.: 'Discriminative common vectors for face recognition', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2005, 27, (1), pp. 4–13.
  36. Support Vector Machines [Electronic resource] – Access mode: <http://scikit-learn.org/stable/modules/svm.html>

37. Schroff F. FaceNet: A Unified Embedding for Face Recognition and Clustering [Electronic resource] / F. Schroff, D. Kalenichenko, J. Philbin. – Access mode: [https://www.cv-foundation.org/openaccess/content\\_cvpr\\_2015/app/1A\\_089.pdf](https://www.cv-foundation.org/openaccess/content_cvpr_2015/app/1A_089.pdf)
38. Boyko, N., Basystiuk, O., & Shakhovska, N. (2018, August). Performance Evaluation and Comparison of Software for Face Recognition, Based on Dlib and OpenCV Library. In *2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP)* (pp. 478-482). IEEE.
39. Dlib C++ Library [Electronic resource] / Access mode: <http://dlib.net/>
40. OpenCV (Open Source Computer Vision Library) [Electronic resource] / Access mode: <https://opencv.org/>

## ДОДАТОК

### Код, що відповідає за розпізнавання

```
class CameraCapture {
static{ System.loadLibrary(Core.NATIVE_LIBRARY_NAME); }

    public static void main(String[] args) {

        System.out.println("Start recognition based on OpenCV " + Core.VERSION);

        new FaceRecognizer().run();

    }
}

public class FaceRecognizer {
public void run() {

    System.out.println("Running DetectFaceDemo");

    CascadeClassifier faceDetector = new
CascadeClassifier(getClass().getResource("lbpcascade_frontalface_improved.xml").getPath());

    Mat image = Imgcodecs.imread(getClass().getResource("5.jpg").getPath());

    MatOfRect faceDetections = new MatOfRect();

    faceDetector.detectMultiScale(image, faceDetections);

    System.out.println(String.format("Detected %s faces", faceDetections.toArray().length));

    for (Rect rectangle : faceDetections.toArray()) {

        Imgproc.rectangle(image, new Point(rectangle.x, rectangle.y), new Point(rectangle.x +
rectangle.width, rectangle.y + rectangle.height), new Scalar(5, 255, 125));

    }

    String filename = "faceDetection.png";

    System.out.println(String.format("Writing %s", filename));

    Imgcodecs.imwrite(filename, image);

}
}
```