

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ**

Кафедра прикладної математики та моделювання складних систем

Допущено до захисту

Завідувач кафедри ПМ та МСС

\_\_\_\_\_ Коплик І.В.  
(підпис)

«\_\_» \_\_\_\_\_ 20\_\_ р.

**КВАЛІФІКАЦІЙНА РОБОТА**

на здобуття освітнього ступеня «магістр»

спеціальність 113 «Прикладна математика»

освітньо–професійна програма «Наука про дані та моделювання складних систем»

тема роботи: **"АНАЛІЗ ТА ІДЕНТИФІКАЦІЯ ЗОБРАЖЕНЬ, ЯКІ  
МІСТЯТЬ СКЛАДНІ ТЕКСТУРНІ ПАТЕРНИ"**

**Виконавець:**

Студент факультету ЕЛІТ

Терьохін А.А. \_\_\_\_\_  
(прізвище, імя, по-батькові) (підпис)

**Науковий керівник:**

доцент, к.ф-м.н.  
(науковий ступінь, вчене звання)

Князь І.О. \_\_\_\_\_  
(прізвище, імя, по-батькові) (підпис)

Факультет  
Кафедра

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
електроніки та інформаційних технологій  
прикладної математики та моделювання складних систем

Рівень вищої освіти

другий  
(перший (бакалавр) або другий (магістр))

Галузь знань **11 Математика та статистика**

Спеціальність **113 Прикладна математика**

Освітня програма **освітньо–професійна «Наука про дані та моделювання складних систем»**

ЗАТВЕРДЖУЮ  
Завідувач кафедри ПМтаМСС  
Коплик І.В.  
«\_\_» \_\_\_\_\_ 20\_\_ р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Терьохіна Андрія Андрійовича.  
(прізвище, ім'я, по батькові)

1. **Тема роботи** Ааналіз та ідентифікація зображень, які містять складні текстурні патерни

**Керівник роботи** доцент, к.ф-м.н. Князь І.О  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання, посада)

затверджено наказом по факультету ЕлІТ від «\_\_» \_\_ 20\_\_ р. № \_\_\_\_\_

2. **Термін подання роботи студентом** «\_\_» \_\_\_\_\_ 20\_\_ р.

3. **Вихідні дані до роботи**

Навчальна та тестова вибірка бази даних зображень, що містять складні текстурні паттерни

4. **Зміст розрахунково–пояснювальної записки (перелік питань, що їй належить розробити)**

1. Літературний огляд
2. Побудова моделі
3. Програмна реалізація
4. Тестування та аналіз результатів

5. **Перелік графічного матеріалу**

Рис. 1.1 — Загальний алгоритм розпізнавання осіб

- Рис. 1.2 — Класифікація методів виявлення облич
- Рис. 1.3 — Характеристики біометричних систем
- Рис. 2.1 — Узагальнений алгоритм обробки кадрів відеопотоку
- Рис. 2.2 — Схема розпізнавання в алгоритмі Віюли-Джонса
- Рис. 2.3 — Ознаки Хаара
- Рис. 2.4 — Маски світлих та темних областей
- Рис. 2.5 — Додаткові ознаки Хаара
- Рис. 2.6 — Приклад навчальних зображень облич
- Рис. 2.7 — Схема алгоритму навчання
- Рис. 2.8 — Прямокутник ABCD
- Рис. 2.9 — Розділення зображення на прямокутні області та формування гістограми
- Рис. 2.10 — Застосування маски значущих областей
- Рис. 2.11 — Метод найближчого сусіда
- Рис. 2.12 — Блок-схема алгоритму обробки виявлених облич
- Рис. 2.13 — Блок-схема алгоритму виводу інформації щодо облич на форму
- Рисунок 2.14 – Консоль керування додатком
- Рисунок 2.15 – Область виводу відео-поток для збору навчальних даних
- Рисунок 2.16 – Область виводу результату класифікації

**6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх**

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

7. Дата видачі завдання « 2 » листопада 2020р.

### КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання роботи	Примітка
1	Літературний огляд	2.11.2020- 10.11.2020	
2	Побудова моделі	11.11.2020 – 20.11.2020	
3	Програмна реалізація	21.11.2020- 5.12.2020	
4	Тестування та аналіз результатів	6.12.2020- 15.12.2020	
5	Оформлення звіту	16.12.2020- 20.12.2020	
6			
7			

Здобувач вищої освіти

\_\_\_\_\_ Терьохін А.А.  
(підпис) (прізвище та ініціали)

Керівник роботи

\_\_\_\_\_ доцент, к.ф-м.н. Князь І.О  
(підпис) (прізвище та ініціали)

## РЕФЕРАТ

**Кваліфікаційна робота:** с.58, 19 рисунків, 20 формул, 19 джерел.

**Актуальність теми.** Останнім часом технологія відеоаналітики набула все більшого поширення, використання комп'ютерного зору для автоматичного збору інформації та ідентифікації людей за обличчям у послідовності зображень, отриманих із відеокамер реального часу та збережених відеозаписів. Ця технологія широко використовується у транспорті, системах відеоспостереження та охорони. Основним рішенням цієї проблеми є одночасне використання цієї технології в системах контролю доступу та ідентифікації особи.

**Мета переддипломної практики:** дослідити розпізнавання осіб у відео потоках, дослідити ефективність різних варіацій локальних двійкових шаблонів при розпізнаванні облич у режимі реального часу.

**Об'єкт дослідження:** процес розпізнавання облич.

**Предмет дослідження:** методи і алгоритми виявлення осіб в кадрі та безпосереднє розпізнавання знайдених облич.

Ключові слова: СИСТЕМА ВИЯВЛЕННЯ ОСІБ, СИСТЕМА РОЗПІЗНАВАННЯ ОБЛИЧ, РОЗПІЗНАВАННЯ ОБРАЗІВ, МЕТОД ВІОЛІ-ДЖОНСА, ЛОКАЛЬНІ БІНАРНІ ШАБЛОНИ, ЦЕНТРАЛЬНО-СИМЕТРИЧНІ ЛОКАЛЬНІ БІНАРНІ ШАБЛОНИ.

## ЗМІСТ

	С.
ВСТУП .....	7
РОЗДІЛ 1 СИСТЕМИ РОЗПІЗНАВАННЯ ОСІБ.....	8
1.1 Огляд методів виявлення облич на цифрових зображеннях.....	9
1.2 Попередня обробка зображень .....	15
1.3 Методи розпізнавання осіб.....	16
1.4 Оцінка ефективності систем розпізнавання.....	19
1.5 Сучасні системи розпізнавання осіб.....	20
1.5.1 Система «FaceVACS» компанії «Cognitec Systems» .....	21
1.5.2 Система «NEC's Face Recognition» компанії «NEC».....	21
1.5.3 Система «LUNA SDK» компанії «VisionLabs».....	23
1.5.4 Система «VeriLook SDK» компанії «Neurotechnology» .....	24
РОЗДІЛ 2 РОЗРОБКА СИСТЕМИ РОЗПІЗНАВАННЯ ОСІБ У ВІДЕОПОТОКАХ .....	26
2.1 Етапи обробки кадрів відеопотоку .....	26
2.1.1 Виявлення осіб методом Віоли-Джонса.....	28
2.1.2 Фільтр Гаусса.....	36
2.1.3 LBP перетворення .....	36
2.1.4 Маска значущих областей зображення.....	37
2.1.5 Метод найближчого сусіда .....	38
2.2 Інструментарій розробки .....	40
2.3 Опис класів .....	41
2.4 Опис інтерфейсу керування додатком.....	47
ВИСНОВОК.....	50
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	51
ДОДАТОК А.....	51

## ВСТУП

На сьогодні переважна більшість підприємств встановлює системи для контролювання доступом персоналу на свої об'єкти. Дякуючи цим біометричним ідентифікаторам значною мірою підвищується безпека безпосередньо самих підприємств та співробітників в тому числі.

У минулому, для ідентифікування персоналу, на входах до підприємств були встановлені турнікети, що зчитували відбитки пальців чи картки, на сьогоднішній день, враховуючи бурхливий розвиток біометричних способів ідентифікування, фірми та корпорації змінюють методи розпізнавання, більш зручнішими та точнішими. Приміром, розпізнають свій персонал із відеопотоку камер спостереження в режимі реального часу.

По мірі того, як корпоративний попит на системи біометричного розпізнавання швидко зростає, аналітики Trend Force очікують, що інтерес і попит на технології розпізнавання обличчя будуть зростати в найближчі роки. Основною сферою застосування цих технологій, по суті, буде система моніторингу та безпеки, а сфера їх використання буде щороку розширюватися [1].

У зв'язку з цим була визначена мета даної випускної кваліфікаційної роботи - розробка програмної системи контролю доступу в відеопотоках на основі алгоритмів розпізнавання осіб.

Для досягнення поставленої мети потрібно:

- Виявити класифікацію алгоритмів розпізнавання;
- Провести аналіз існуючих на ринку систем розпізнавання;
- Визначити для розробки основні інструментальні засоби;
- Спроекувати систему розпізнавання.

## РОЗДІЛ 1 СИСТЕМИ РОЗПІЗНАВАННЯ ОСІБ

Процес розпізнавання обличчя складається з ряду різних завдань, за допомогою яких можна по-різному ідентифікувати людину за допомогою відео та фото. Загалом, у цьому процесі розглядаються наступні етапи: Після того, як система отримала зображення з камер спостереження, алгоритми виділяють особистості характеристики (встановлення). Далі слідує етап розпізнавання, в якому зображення трансформується (воно вирівнюється, збільшується, змінюється його яскравість тощо) і надається певна форма. Після розпізнавання знаки обчислюються, а потім порівнюються зі стандартами. Цей останній етап порівняння називається залежно від системи верифікація чи ідентифікація.

**Верифікація:** Зразки порівнюють схемою «1:1». Для визначення характеристик особи системою використовується порівняння біометричного зразка з біометричним шаблоном, що зберігається в базі даних, і дає відповідь на запитання «Чи є ця людина тією людиною, з шаблоном якої було порівняння?»

**Ідентифікація:** відібрані зразки порівнюються за схемою один зразок до загальної кількості шаблонів у базі.

Рисунок 1.1 зображає загальний алгоритм розпізнавання.

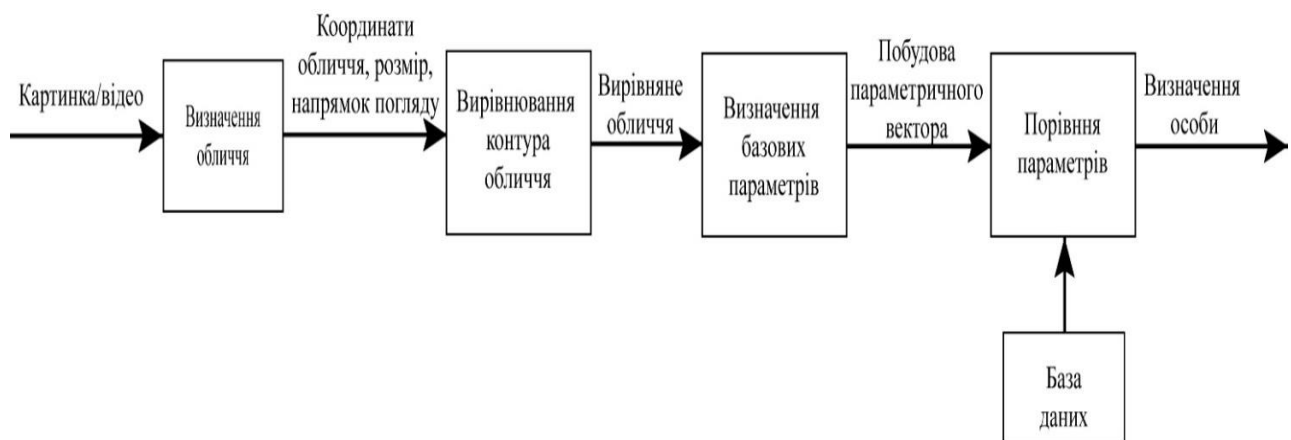


Рис. 1.1 — Загальний алгоритм розпізнавання



## 1.1 Огляд методів виявлення облич на цифрових зображеннях

Зображення у вигляді цифрових даних з камер відеоспостереження після їх передачі на комп'ютер, обробляються за допомогою спеціальних алгоритмів, які визначають положення лицьової області на основі їх головні особливості (очі, рот, брови, ніс і т.д.). Методів виявлення облич існує досить велика кількість і більшість з них – комбінацію інших методів. Всі методи можна розділити на дві категорії: методи засновані на основі знань, які побудовані на досвіді людини і методи, що виявляють за зовнішніми ознаками (методи які потребують проведення етапу навчання системи, обробкою тестових зображень). Класифікацію методів виявлення облич наведено на рисунку 1.2.

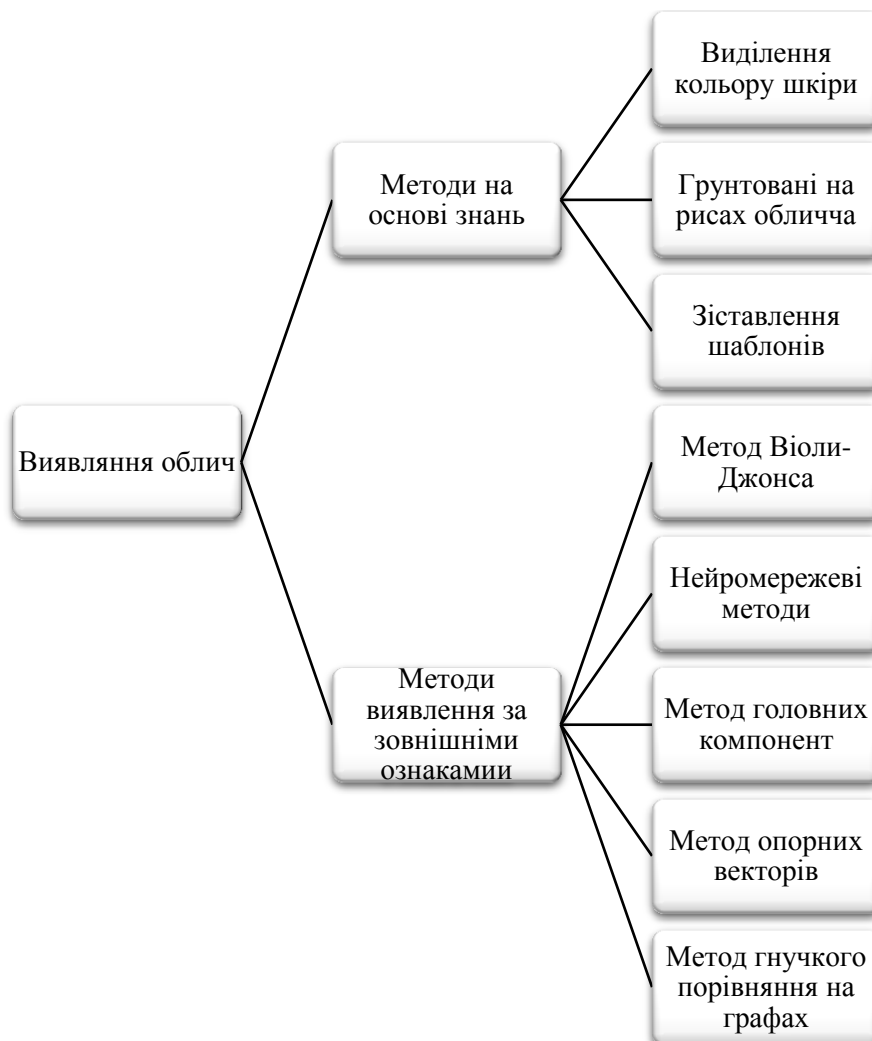


Рис. 1.2 — Класифікація методів виявлення облич

Методи, що засновані на знаннях, використовують дані про особу, її форму, риси, текстуру чи колір шкіри. Ці методи виділяють ряд правил (властивостей і властивостей особи), яким повинен відповідати фрагмент з кадру, щоб вважатися людським обличчям. Ряд таких правил визначається досить просто. Всі правила-це формалізоване знання, яким людина користується, коли визначає, стоїть перед ним людина чи ні.

Основні правила: області рота, носа і очей розрізняються по яскравості щодо іншої частини обличчя; очі на обличчі завжди розташовані симетрично один одному. Опираючись на ці та інші властивості, що схожі між собою, будуються алгоритми, які при виконанні перевіряють на зображенні наявність правил.

До цієї групи методів також відноситься більш загальний – метод порівняння з шаблоном. Цей метод описує властивості окремих областей особи та їх задане взаємне розташування, так визначається стандарт особи (шаблон), до якого потім прирівнюють вихідне зображення.

Методи на основі знань поширені досить широко та мають непогані показники, але показують гарні результати тільки на зображеннях які мають хороше розширенням, з нескладним фоном і без шумів. На кадрах відеопотоку чи камер спостереження, що встановлені в громадських місцях, де є різні ракурси і повороти осіб, а також мінливий стан освітлення та присутня велика кількість об'єктів, що заважають на задньому плані, велика ймовірність виникнення помилок.

Методи, що виявляють осіб за зовнішніми ознаками мають інший підхід до цієї проблеми, вони не намагаються формалізувати процеси, що відбуваються в мозку людини в явному вигляді, а навпаки намагаються неявно виявляти властивості і закономірності зображення особи, і при цьому застосовують методи і машинного навчання та математичної статистики. Методи даної групи позбавлені згаданих раніше недоліків і тому застосовуються в системах відеоспостереження частіше. Виявлення осіб в цих методах

здійснюється перегляду усіх прямокутних фрагментів зображень кінцева ціль якого – до якого класу належить зображення: до класу зображень без фігури обличчя чи до класу, що містять обличчя .

Враховуючи такий великий об'єм роботи методи мають надмірність та обчислювальну складність. Для того, щоб пришвидшити пошук людей і зменшити кількість обчислень, автори алгоритмів і методів для більш швидкого розпізнавання фрагментів перегляду мають кілька додаткових методів.

Найбільш доречні та помітні з них розглянуті нижче:

Метод Віоли-Джонса (Viola-Jones object detection). Даний метод запропонували Пауло Віола та Майкл Джонс у 2001 році. Цей метод був першим, що показував високі результати в обробці зображень у реальному часі. Цей метод має багато реалізацій і знаходиться в бібліотеці для комп'ютерного зору OpenCV (Функція `cvHaarDetectObjects ()`) [2].

Детальніше метод розглянемо в розділі 2.

Перевагами даного методу є:

- Висока точність виявлення повернених на кут до 30 градусів осіб (якщо кут повороту більше, ефективність знижується);
- Висока швидкість роботи (досягається класифікатором каскаду);

Недоліки даного методу:

- Тривале навчання. Алгоритму потрібно проаналізувати значну кількість тестових зображень;
- Коли йде виявлення, положення людини може бути обмежене..

Метод гнучкого порівняння на графах (Elastic graph matching). Цей метод відноситься до 2D моделювання. Суть методу слід використовувати при виборі діаграм, що описують людину (людина представлена у вигляді сітки з індивідуальним розташуванням країв та кутових точок).

Етап розпізнавання відбувається так – еталонний граф, що є основним параметром розпізнавання, залишається без зміни, а інші графи деформуються під впливом структури особи з прив'язкою до основних точок обличчя: ширина губ,

лінія носа, відстань між очима, вухами і т.п. Чим більше точок буде використано, тим точніше буде працювати система розпізнавання, але на обробку об'єкта збільшується час.

Недоліки методу:

- Складний процес введення в базу даних нових шаблонів;
- Складність алгоритму розпізнавання потребує значних обчислювальних потужностей;
- Розміри баз даних обернено пропорційні швидкості дії аналітичної системи.

Приховані моделі Маркова (СММ). Метод побудований на статистичному порівнянні об'єкта з базою даних шаблону. Приховані моделі Маркова використовують статистичні властивості сигналів та враховують їх просторові властивості. Елементи моделі Маркова: початкова ймовірність станів, безліч прихованих станів, безліч спостережуваних станів, матриця перехідних ймовірностей. Кожний елемент має модель і то того ж свою власну. При виявленні перевіряються всі згенеровані моделі Маркова і визначається найбільша ймовірність виявлення послідовності спостережень для моделі, згенерованої об'єктом.

Недоліки:

- Система оптимізує тільки час обробки і реакцію на власну модель, але не може звести до мінімуму час прориву інших моделей.
- Досить повільна швидкодія;
- Не оптимальний алгоритм навчання та низька роздільна здатність;

Метод головних компонент (МГК). Головна мета МГК – зменшення кількості ознак без втрати значущої інформації, щоб як можна краще описувати найбільш стандартні образи, що належать особам. У задачі розпізнавання осіб цей метод головним чином використовують для того, щоб обличчя уявити як вектор малої розмірності, а потім порівняти з базою даних еталонних векторів.

Набір з власних векторів, що був отриманий на початковій вибірці один раз, кодує інші зображення осіб, які можна уявити як зважену комбінацію векторів власних. Якщо використовується обмежена кількість векторів власних, то можна отримати апроксимацію вхідному зображенню особи, яку можна зберігати в базі даних, як вектор коефіцієнтів, що служитиме одночасно і як ключ пошуку в базі даних.

МГК чудово працює в прикладних програмах. Але, коли в образі людини відбуваються суттєві зміни чи в освітленості ефективність знижується.

Метод опорних векторів (Support Vector Machines, SVM) – набір алгоритмів схожих на навчання з вчителям, використовуються для регресійного аналізу і задач класифікації.[4] Сутність даного методу – знаходження гіперплощини в просторі ознак, що відрізняє клас зображень осіб від зображень «не осіб». При цьому з усіх можливих гіперплощин, які ділять два класи, необхідно вибрати гіперплощину, відстань до якої від кожного класу максимальна.

Переваги даного методу:

- Стійкий до перепідготовки;
- Робоча швидкість вище якщо порівнювати з нейронними мережами;
- За рахунок зниження точності можна контролювати чутливість до шуму.

Недоліки методу:

- Точність роботи поступається іншим методам.

Методи Нейромереж. Методи, що мають у своєму складі майже десяток різних алгоритмів. Головною характеристикою цих мереж є їх здатність заздалегідь вчитися на ряді готових прикладів. Під час навчання мережа автоматично відстежує ключові функції і налагоджує взаємозв'язок між ними. Потім, щоб розпізнати раніше невідомі об'єкти, нейронна мережа використовує набутий досвід. Методи нейронних мереж вважають надскладними для реалізації, але мають найкращі розпізнавальні здібності.

Переваги даного методу:

- Висока точність роздільної здатності відображається при правильному пошуку параметрів мережі.

Недоліки методу:

- Висока обчислювальна складність;
- Важко формалізувати архітектуру мереж (кількість нейронів, шари, тип зв'язку).
- Складний процес змін (будь-яка зміна вимагає перекваліфікації мережі).

## 1.2 Попередня обробка зображень

Щоб підвищити ефективність розпізнавання і якість виділення облич, в системах розпізнавання облич проводять для вхідних зображень етапи попередньої обробки.

Зображення після визначення обличчя на кадрі і визначення положення голови, розміру, пози і ключових особливостей воно відображається шляхом нормалізації: масштабоване, закодоване, перетворене в горизонтальну лінію, що з'єднує центри очей. Для зменшення рівня шуму під час попередньої обробки використовуються різні фільтри (гауссовський, медіанний та ін.).

Методи попередньої обробки досить різноманітні і залежать від завдань досліджень. Найчастіше зустрічаються:

- нелінійні фільтри – фільтри, що використовують для видалення імпульсного шуму (окремих точок з максимальною (білої) чи мінімальною (чорної) яскравістю) на зображенні;
- фільтр Гаусса – фільтр розмиття зображення, найчастіше використовують, якщо зображення має дрібні деталі, які не вимагають відділення від фону і їх можна розмити;
- медіанні фільтри – фільтри, що використовують для збереження перепадів яскравості контурів і придушення імпульсних шумів.

### 1.3 Методи розпізнавання осіб

Ще однією важливою частиною автоматичних систем розпізнавання осіб є самі алгоритми розпізнавання людини. Сьогодні таких алгоритмів багато, і кожен з них має свої особливості, швидкість і надійність.

Алгоритми розпізнавання діляться на дві категорії, в залежності від технології розпізнавання - двовимірні, розпізнавання здійснюється по геометрії особи (2D-технологія) і Тривимірні, в яких розпізнавання здійснюється за структурою черепа (3D-технологія).

2D-виявлення працює з плоскими, двовимірними зображеннями і розпізнаванням обличчя, аналізує його текстуру і області з високою контрастністю, виявлення сильно утруднюється при порушенні освітлення або положення особи.

3D-виявлення системи стійке до таких змін. При створенні моделі особи враховуйте особливості будови черепа. Хоча 3D-системи мають великий плюс, через низьку швидкість немає можливості обслуговувати велику кількість користувачів в режимі ідентифікації, 3D-технологія ще не широко використовується. Технології 3D-виявлення вартість обладнання для таких систем набагато вище, ніж у 2D, і вимагає великих обчислювальних ресурсів [5].

У сучасній системі відеоспостереження в основному існують ефективні алгоритми виявлення, усі вони закріплені за характерними точками, що визначають значення пікселів. Нижче ви знайдете короткий опис окремих підгруп. Методи, зафіксовані в характерних точках, не оцінюють особливостей пікселів, а координують характерні точки зображення. Такими характерними точками можуть бути центри очей, положення носа, лінія брів, рота тощо. До цього класу методів відносяться активні моделі зовнішнього вигляду та активні моделі форм.



Активні моделі зовнішнього вигляду (Active Appearance Models, ААМ)-це статистичні моделі зображень, які можуть бути адаптовані до реального зображення за допомогою різних типів деформацій. Цей тип моделей у двовимірній версії був запропонований Тімом Кутса і Крісом Тейлором в 1998 році. [6] Модель активного зовнішнього вигляду містить два типові параметри: параметри, пов'язані з формою (параметри форми), і параметри, пов'язані зі статистичною піксельною моделлю зображення або текстури (параметри зовнішнього виду). Перед використанням моделі слід завершити набір видалених зображень. Розмітка зображень проводиться вручну.

Активні моделі форми (Active Shape Models, ASM) враховують статистичні зв'язки у взаємному розташуванні антропометричних точок. На кожному зображенні вибірки експерт розмічає розташування антропометричних точок. Для приведення координат всіх зображень в єдину систему зазвичай проводиться узагальнений аналіз, в якому всі точки перетворюються в масштаб і центруються. Потім обчислюється середня форма і коваріаційна матриця для всього набору зображень. На основі матриці коваріації обчислюються власні вектори, які потім сортуються в порядку убуття відповідних власних значень. Локалізації ASM моделі на новому, не включеному в вибірку зображенні, відбувається в процесі вирішення проблеми оптимізації..

Методи, засновані на значеннях пікселів, використовують колір або яскравість для розпізнавання виявлених осіб. Найпростіші подібні методи:

Eigenfaces - Запропонований в 1991 році алгоритм став першим успішним розпізнавачем, який набув популярності.

Основна ідея алгоритму Eigenfaces полягає у відображенні характерних особливостей зображення обличчя за допомогою двовимірних зображень у відтінках сірого. Щоб ідентифікувати особу, алгоритм порівнює отриману людину з уже зареєстрованим шаблоном в базі даних і визначає коефіцієнт різниці. Ця відмінність між візерунками показує ступінь подібності.

Алгоритм показує високі результати при використанні в добре освітлених кімнатах коли відбуваються зміни умов використання даного алгоритму, показники його ефективності дуже знижуються.

Fisherfaces - нащадок Eigenface, забезпечує більш високу точність розпізнавання при змінах світла або обличчя. На відміну від методу eigenfaces, в основі Fisherfaces лежить лінійний дискримінантний аналіз LDA, а саме лінійний дискримінант Фішера

Дія алгоритму Fisherfaces заснована на пошуку проекції даних, в якій класи зображень обличчя максимально відокремлені один від одного. Ця відмінність від Eigenface вирішує проблему високої чутливості до змін світла.

Локальні двійкові шаблони (LBP) - це ефективний спосіб полегшити розпізнавання облич. При розпізнаванні методом LBP кожному пікселю зображення присвоюється значення яскравості за допомогою функцій, що описують його оточення. Отримане зображення розділено на області, для кожної з яких розраховується гістограма. Потім гістограми об'єднують і порівнюють за допомогою методів машинного навчання. Класична версія алгоритму використовує метод найближчого сусіда.

На відміну від EigenFaces, алгоритм стійкий до монотонних змін освітлення, що робить його придатним для розпізнавання осіб в системах обробки в реальному часі.

## 1.4 Оцінка ефективності систем розпізнавання

Існує велика різниця між аматорськими та спеціалізованими системами виявлення. Аматорські системи-виглядають красивіше, коштують дорожче, і до них пред'являються більш низькі вимоги. У зв'язку з тим, що в разі помилок, викликаних цими системами, наслідки, як правило, невеликі, їх вартість набагато нижче, ніж вартість професійних систем, до яких пред'являються серйозні вимоги.

Якщо ми використовуємо, наприклад, технологію розпізнавання осіб на фотографіях в соціальних мережах, і професійну систему, розроблену для розшуку людей. Перша обробляє фотографії, на яких люди зазвичай дивляться в об'єктив, а друга - з людьми, знятими в громадському місці, які не дивляться в камери відеоспостереження, а іноді навіть спеціально ховаються від них. Якщо перша система виявлення допустить помилку, нічого страшного не станеться, але якщо система пошуку шуканих людей допустить багато помилок, то це серйозна проблема.

Тому при виборі системи необхідно правильно визначити не тільки цілі, а й критерії оцінки ефективності цієї системи. Це означає, що адаптація системи до роботи за певних умов і порогові значення допустимих помилок мають бути враховані.

Оцінка ефективності системи сьогодні визначає два основні параметри. Вони відомі як Коефіцієнтом помилкового доступу (FAR - False Acceptance Rate) і Коефіцієнтом помилкової відмови (False Rejection Rate - FRR).

Коефіцієнт помилкової відмови показує ймовірність того, як часто особи помилково відкидаються, або іншими словами-система не знайшла в базі даних «своїх» людини. Чим нижчий цей коефіцієнт, тим вища точність розпізнавання.

Коефіцієнт помилкового допуску - це навпаки, зворотня ймовірність того, як часто системою справжність розпізнано помилково.

Є ще один фактор, що дозволяє порівнювати біометричні системи - коефіцієнт EER (рівний рівень помилок). Це коефіцієнт, при якому обидві похибки (похибка прийому та похибка відхилення) еквівалентні. Чим нижчий EER, тим точніша біометрична система

На рисунку 1.3 показані взаємозв'язки характеристик FAR, FRR і EER.

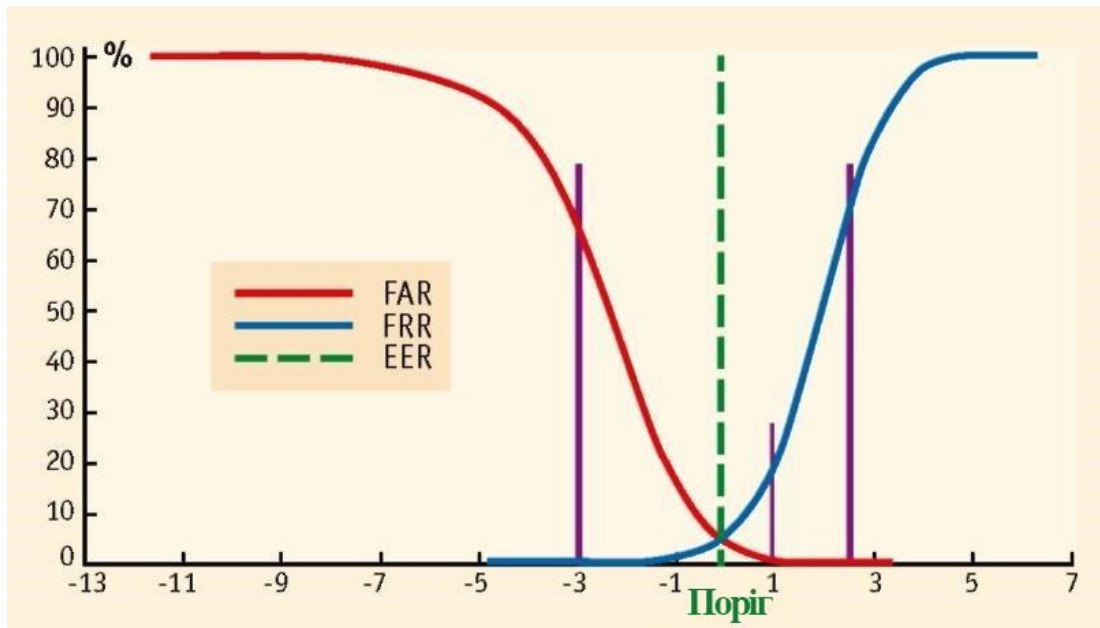


Рис. 1.3 — Характеристики біометричних систем

Діаграма використовується для порівняння систем та візуалізації зв'язку між помилками ROC (Receiver Operating Characteristic).

Крива ROC визначає, наскільки зразок повинен бути близький до шаблону, щоб система могла розглядати його як випадковість. При регулюванні порогових значень помилок, варто не забувати, що величини FAR і FRR взаємодоповнюючі і якщо їх зменшувати, то буде зменшуватися кількість помилкових розбіжностей, але збільшуватиметься кількість помилкових прийомів.

### 1.5 Сучасні системи розпізнавання осіб

З кожним роком інтерес до технологій розпізнавання осіб значно зростає, на сьогодні, в різних сферах життя використовуються кілька десятків.

Серед компаній, що розробляють такі системи, лідерами є: корпорація NEC (Японія), «VisionLabs» (Росія), «FACE ++» Neurotechnology (Литва), і Cognitec Systems GMBH (Німеччина).

### **1.5.1 Система «FaceVACS» компанії «Cognitec Systems»**

«FaceVACS-VideoScan» - програмне забезпечення просте у використанні, налаштовується для розпізнавання осіб через живий відеопотік, який пропонує компанія «Cognitec Systems».

«FaceVACS-VideoScan» складається з: Videomage Manager відеопотоків; Video scan Server, який координує всі компоненти системи і виконує основні біометричні операції; комп'ютерний вузол, який використовується для розподілу обчислювального навантаження; користувальницький інтерфейс; диспетчер сигналів, що приймає повідомлення про події і мобільних пристроїв; база даних операційної системи; і ряд інтеграторів.

Зараз технологія FaceVACS використовує алгоритм розпізнавання осіб V10T9. Цей алгоритм, стійкий до змін виразу та поворотів обличчя (на  $\pm 15^\circ$ ) часткового його закриття, використання окулярів та зміни освітлення [7].

Системи FaceVACS:

- порівняння осіб відбувається в реальному часі;
- дозволяє переглядати та звітувати статистичні дані про потоки;
- підтримка інтерактивної реєстрації з нерухомого;
- Використання API C ++ та API веб-служб.
- можливість відстеження кількох людей одночасно;

### **1.5.2 Система «NEC's Face Recognition» компанії «NEC»**

«NEC's Face Recognition» система розпізнавання осіб, розроблена японською компанією "NEC", дозволяє ідентифікувати людей за багаторічними знімками, а також по тому, коли знаходиться людина в окулярах. Всі виявлені

особи зберігаються в базі даних, тому при необхідності ви можете захопити всю історію відеозапису і переглянути дату і час кожного збереженого зображення.

Технологія NEC перевершує багато інших систем виявлення по своїй швидкості і точності. Має хорошу продуктивність в різних ситуаціях, включаючи роботу з відео низької якості і сильно стиснутими зображеннями. NEC аналізує індивідуальні риси обличчя (розмір, ніс і рот, форма зіниць), їх взаємне розташування, а потім на основі цієї інформації знаходить відповідну людину в базі даних. Система включає в себе кілька модулів, що реалізують такі алгоритми:

- Алгоритм PSM (Perturbation Space Method), дозволяє ефективно справлятися зі складнощами пов'язаними з розташуванням особи в кадрі (особа нахилена або під певним кутом).
- Використовується узагальнений метод відповідності (GMFD), який забезпечує високу швидкість виявлення і високу точність розпізнавання осіб. Метод GMFD заснований на нейронних мережах і виконує попередній пошук пар очей.
- Метод ARBM (Adaptive Regional Blend Matching), який зменшує вплив невеликих змін на обличчі (наприклад, виразу обличчя, окуляри, головні убори) на точність розпізнавання [8].

Система розпізнавання обличчя NeoFace має такі особливості:

- Моніторинг в режимі реального часу;
- Множинне розпізнавання;
- Ідентифікація на основі індивідуальних рис обличчя;
- Можливість пошуку подій по базі даних;
- Ведення журналу зображень осіб;
- Незалежне розпізнавання напрямку погляду і характеристик особи (Окуляри, борода і вираз обличчя);
- Масштабований і необмежений розмір Баз Даних;

- "Drag and Drop" управління;
- Стійкість до повороту особи на  $\pm 15^\circ$  і нахилу голови до  $45^\circ$  в будь-якому напрямку від фронтального положення.

### 1.5.3 Система «LUNA SDK» компанії «VisionLabs»

«LUNA SDK» - спеціалізоване програмне забезпечення, супроводу і розпізнавання облич людей на цифрових фотокартках чи в відеопотоці від компанії «VisionLabs». Двигун має одні з найкращих у світі реальних показників комплектності та точності.

Процес розпізнавання осіб в LUNA SDK працює на основі глибинних нейронних мереж і складається з декількох ключових кроків: визначення положення та розміру всіх осіб (ідентифікація), визначення рис обличчя та перетворення його в стандартизовану форму (вирівнювання), вилучення дескрипторів (Числові вектори з рисами обличчя), порівняння обличчя з базою даних зображень та запобігання заміщення людей.

Характеристики і особливості LUNA SDK:

- Повністю розроблена на C ++;
- Спостереження і контролю в реальному часі;
- Робота в режимі багатопотокової обробки даних;
- Використання алгоритму розпізнавання осіб на основі глибинних нейронних мереж;
- Оптимізоване керування пам'яттю;
- Компактні дескриптори, що забезпечують обчислювальну ефективність;
- Підтримка різних платформ, включаючи Linux і Windows.

Дескриптори LUNA навчені на мільйонах осіб з різних джерел і забезпечують високу точність при різних умовах, таких як банки, системи відеоспостереження та соціальні медіа. [9].

#### 1.5.4 Система «VeriLook SDK» компанії «Neurotechnology»

«VeriLook SDK» - технологія ідентифікації осіб, яку розробили фахівці компанії «Neurotechnology». Розпізнавання осіб системою з можливістю одночасного багаторазового розпізнавання осіб в кадрі і швидка ідентифікація особи (знаходить до 100 000 осіб в секунду) робить дану систему просто не замінною. VeriLook SDK доступний в якості комплекту розробки програмного забезпечення і підтримує безліч пристроїв на Mac OS X, iOS, Android, Windows і Linux. [10] Алгоритм VeriLook реалізує локалізацію особи з використанням алгоритмів обробки цифрових зображень заснованих на глибоких нейронних мережах.

Основними перевагами системи VeriLook є відсутність контакту з інструментами сканування і швидке впровадження функцій біометричної ідентифікації в прикладні системи клієнта, і є ряд інших переваг:

- Одночасна обробка декількох осіб;
- Гендерна класифікація. При необхідності стать може бути визначена для кожної людини на зображенні;
- Живе розпізнавання обличчя. VeriLook визначає, чи відображається особа в живому відеопотоці або фотографії;
- Розпізнавання емоцій. VeriLook аналізує шість основних емоцій:
- Атрибути особи. VeriLook може бути налаштоване для виявлення певних атрибутів під час вилучення особи - посмішки, відкритого рота, закритих очей, окулярів, бороди або вусів;
- Визначення якості зображення особи. Поріг якості може використовуватися під час реєстрації особи, щоб гарантувати, що в базі даних будуть зберігатися тільки шаблони кращої якості.
- Кілька зразків однієї і тієї ж особи. Ці зразки можуть бути зараховані з різних джерел і в різний час, що дозволяє поліпшити якість зіставлення;



- Невеликі візерунки особи. Малюнок особи може мати розмір 4 кілобайт;
- Функції режиму Узагальнення. Цей режим створює набір узагальнених лицьових функцій з декількох зображень одного і того ж об'єкта;
- Алгоритм VeriLook здатний порівнювати межі, які були захоплені в інфрачервоному спектрі.

## РОЗДІЛ 2 РОЗРОБКА СИСТЕМИ РОЗПІЗНАВАННЯ ОСІБ У ВІДЕОПОТОКАХ

### 2.1 Етапи обробки кадрів відеопотоку

Є два основних етапи обробки кадру відеопотоку. На першій фазі розпізнавання визнання людей у кадрі, а на другій - безпосередньо саме розпізнавання виявлених осіб.

Для розробки системи, в даній роботі використовується метод розпізнавання осіб Віюлі-Джонса, а для розпізнавання - методу найближчого суду з використанням гістограми центрально-симетричних локальних двійкових зразків.

Розпізнавання облич в системі засноване на пошуку мінімальної відстані між гістограмою вхідного зображення особи і гістограмами, що зберігаються в базі даних.

Слід зазначити, що ефективність вибраних алгоритмів суттєво залежить від різних факторів, таких як: положення обличчя, фон, освітлення тощо. Тому для забезпечення правильної роботи системи повинні бути створені сприятливі умови для її використання:

- Нейтральний вираз обличчя;
- Особа, яка не перекривається іншими об'єктами;
- Тестова вибірка повинна зніматися в однакових умовах освітлення;
- Фронтальне, чи близьке до нього положення особи.

Крім базових фаз виявлення і розпізнавання, в системі, що виявляє знайдені особи, існують проміжні фази: Гаусовий фільтр, що зменшує ефект шуму після розпізнавання осіб, а також маска зі значними областями, що знімає ефект кутових областей зображення, що містять фон.

Узагальнений алгоритм редагування кадрів, що розробляється включає в себе наступні етапи: ідентифікація осіб, обробка осіб за допомогою Гаус-

фільтра, трансформація знайдених осіб з подальшим застосуванням маски значущих ділянок, розрахунок гістограм знайдених осіб, Класифікація осіб найближчим сусідом гістограми. Загальний алгоритм представлений на рисунку 2.1.



Рис. 2.1 — Узагальнений алгоритм обробки кадрів відеопотоку

Також за складеним узагальненим алгоритмом обробки відеопотоку була зібрана та розроблена необхідна функціональність:

- Введення відеопотоку з підключеною до комп'ютера камери в реальному часі чи можливість завантаження відео з файлу;
- Виведення інформації про приладдя розпізнавання особи до певного класу;
- графічне відображення гістограми і LBP уявлення відслідковується особи;
- Можливість навчання і додавання класів осіб з використанням камери через інтерфейс програми;
- Можливість налаштування параметрів роботи алгоритмів.

### 2.1.1 Виявлення осіб методом Віоли-Джонса

У 2001 році Віола і Джонс запропонували алгоритм для розпізнавання осіб, який став проривом в області розпізнавання осіб. Метод використовує технологію ковзного вікна. Тобто рамка, розміром, меншим, ніж вихідне зображення, рухається з деяким кроком по зображенню, і за допомогою каскаду слабких класифікаторів визначає, чи є в даному вікні особа. Метод змінного вікна ефективно використовується в різних завданнях комп'ютерного зору і розпізнавання об'єктів.

Даний метод складається з 2-х під алгоритмів: алгоритм навчання і алгоритм розпізнавання. На практиці швидкість роботи алгоритму навчання не важлива. Вкрай важлива швидкість роботи алгоритму розпізнавання. За введеної раніше класифікації можна віднести до структурних, статистичним і нейронних методам.

Метод має такі переваги:

- використання простих класифікаторів показує хорошу швидкість і дозволяє використовувати цей метод в потоці;
- можливе виявлення більше однієї особи на зображенні.

Однак метод складно навчається, так як для навчання потрібна велика кількість тестових даних і передбачає великий час навчання, який вимірюється днями. Спочатку алгоритм був запропонований для розпізнавання тільки осіб, але його можна використовувати для розпізнавання інших об'єктів. Одним з вкладів Віоли і Джонса було застосування таблиці сум, яку вони назвали інтегральним зображенням, детальний опис якого буде дано нижче.

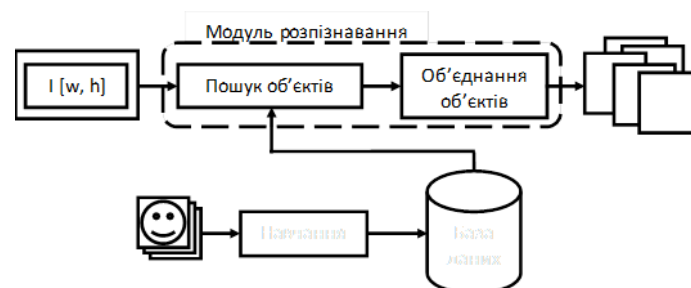


Рис. 2.2 — Схема розпізнавання в алгоритмі Віоли-Джонса

Узагальнена схема алгоритму вигляди наступним чином: перед початком розпізнавання алгоритм навчання на основі тестових зображень навчає базу даних, що складається з ознак, їх паритету і кордони. Детальніше про паритет, ознаці і кордоні буде розказано в наступних пунктах. Далі алгоритм розпізнавання шукає об'єкти на різних масштабах зображення, використовуючи створену базу даних. Алгоритм Віоли-Джонса на виході дає весь люд знайдених необ'єднаних об'єктів на різних масштабах. Наступне завдання - прийняти рішення про те, які з знайдених об'єктів дійсно присутні в кадрі, а які - дублі.

### Ознаки класу

В якості ознак для алгоритму розпізнавання авторами були запропоновані ознаки Хаара, на основі вейвлетів Хаара. Вони були запропонований угорським математиком Альфредом Хара в 1909 році.

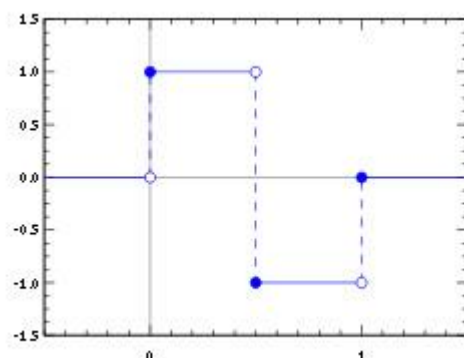


Рис. 2.3 — Ознаки Хаара

У задачі розпізнавання осіб, загальний нагляд, що серед всіх осіб області очей темніше області щік. Розглянемо маски, що складаються зі світлих і темних областей.



Рис. 2.4 — Маски світлих та темних областей

Кожна маска характеризується розміром світлою і темною областей, пропорціями, а також мінімальним розміром. Спільно з іншими спостереженнями були запропоновані наступні ознаки Хаара, як простір ознак в задачі розпізнавання для класу осіб.

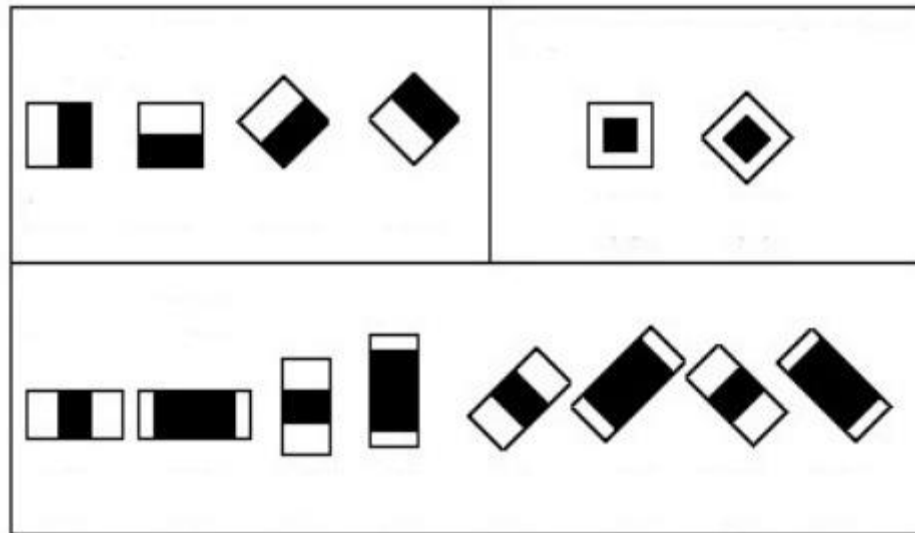


Рис. 2.5 —Додаткові ознаки Хаара

Ознаки Хаара вказують значення точки перепаду яскравості по осі X і Y відповідно. Тому загальний ознака Хаара для розпізнавання осіб представляє набір двох суміжних прямокутників, які лежать вище очей і на щоках. Значення ознаки обчислюється за формулою (2.1):

$$F = XY \quad (2.1)$$

де X - сума значень яскравості точок закриваються світлою частиною ознаки, а Y - сума значень яскравості точок закриваються темною частиною ознаки.

Видно, що якщо вважати суми значень інтенсивностей для кожної ознаки це зажадає значних обчислювальних ресурсів. Віолою і Джонсом було запропоновано використовувати інтегральне представлення зображення, докладніше про нього буде далі. Таке уявлення стало досить зручним способом обчислення ознак і застосовується також і в інших алгоритмах комп'ютерного зору, наприклад SURF.

### Схема навчання

Узагальнена схема алгоритму навчання виглядає наступним чином. Є тестова вибірка зображень. Розмір тестової вибірки близько 10 000 зображень. На рисунку показаний приклад навчальних зображень облич. Алгоритм навчання працює з зображеннями в відтінках сірого.



Рис. 2.6 — Приклад навчальних зображень облич

При розмірі тестового зображення 24 на 24 пікселя кількість конфігурацій однієї ознаки близько 40 000 (залежить від мінімального розміру маски). Сучасна реалізація алгоритму використовує близько 20 масок. Для кожної маски, кожної конфігурації тренується такий слабкий класифікатор, який дає найменшу помилку на всій тренувальній базі. Він додається в базу даних. Таким чином алгоритм навчається. І на виході алгоритму виходить база даних з  $T$  слабких класифікаторів. Узагальнена схема алгоритму навчання показана на рисунку 2.7.

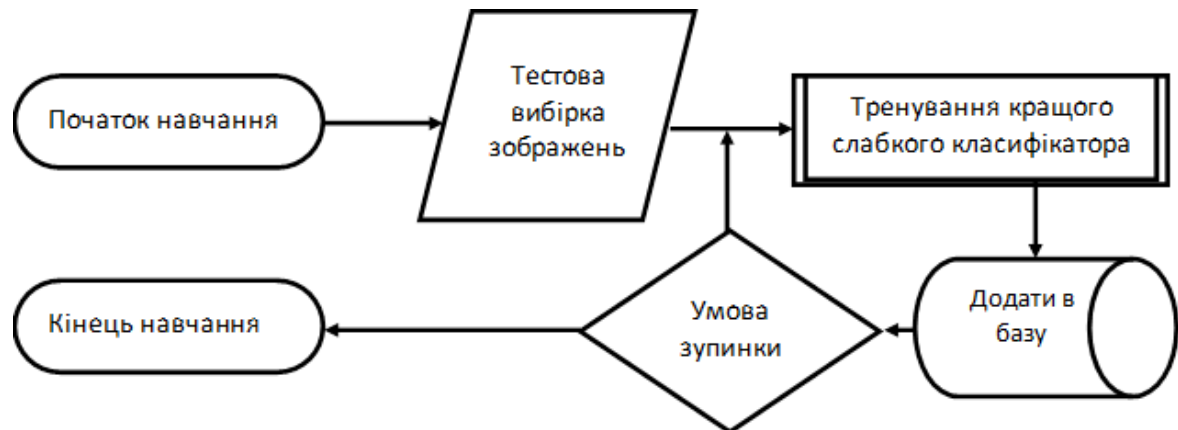


Рис. 2.7 — Схема алгоритму навчання

Навчання алгоритму Віола-Джонса - це навчання алгоритму з учителем. Для нього можливо така проблема як перенавчання. Показано, що AdaBoost може використовуватися для різних проблем, в тому числі до теорії ігор, прогнозуванні. У даній роботі умова зупинки є досягнення заздалегідь заданого кількості слабких класифікаторів в базі.

Для алгоритму необхідно заздалегідь підготувати тестову вибірку з  $l$  зображень, що містять шуканий об'єкт і  $m$ , що не містять. Тоді кількість всіх тестових зображень буде:

$$n = l + m \quad (2.2)$$

$$X = \{x_1, x_2, \dots, x_n\} \quad (2.3)$$

де  $X$  - безліч всіх тестових зображень, де для кожного заздалегідь відомо чи присутній шуканий об'єкт чи ні і відображено в безлічі  $Y$ .

$$Y = \{y_1, y_2, \dots, y_n\} \quad (2.4)$$

де

$$y_i = \begin{cases} 1, \text{ об'єкт присутній на зображенні } x_i \\ 0, \text{ інше} \end{cases}$$

Під ознакою  $j$  будемо розуміти структуру виду

$$j = \{\text{маска, положення, розмір}\}$$



Тоді відгуком ознаки буде  $f_j(x)$ , який обчислюється як різниця інтенсивностей пікселів у світлій і темній областях. Слабкий класифікатор має вигляд:

$$h_j(x) = \begin{cases} 1, & p_j f_j(x) < p_j \theta_j \\ 0, & \text{інше} \end{cases} \quad (2.5)$$

Завдання слабкого класифікатора - вгадувати присутність об'єкта в більше ніж 50% випадків. Використовуючи процедуру навчання AdaBoost створюється дуже сильний класифікатор складається з  $T$  слабких класифікаторів і має вигляд:

$$H(x) = \begin{cases} 1, & \sum_{t=1}^T a_t h_{j(t)}(x) \geq \frac{1}{2} \sum_{t=1}^T a_t \\ 0, & \text{інше} \end{cases} \quad (2.6)$$

Цільова функція навчання має наступний вигляд:

$$T, h_{j(1)}, h_{j(2)}, \dots, a_1, \dots, a_T = \operatorname{argmin} \sum_{i=1}^n |H(x_i, T, h_{j(1)}, \dots, h_{j(T)}, a_1, \dots, a_T)| \quad (2.7)$$

Інтегральне представлення зображень

Інтегральне уявлення можна представити у вигляді матриці, розміри якої збігаються з розмірами вихідного зображення  $I$ , де кожен елемент розраховується так:

$$H(x, y) = \sum_{i=0, j=0}^{i \leq x, j \leq y} I(r, c) \quad (2.8)$$

де  $I(r, c)$  - яскравість пікселя вихідного зображення.

Кожен елемент матриці  $H[x, y]$  являє собою суму пікселів в прямокутнику від  $(0,0)$  до  $(x, y)$ . Розрахунок такої матриці займає лінійний час. Для того, щоб обчислити суму прямокутної області в інтегральному уявленні зображення потрібно всього 4 операції звернення до масиву і 3 арифметичні операції. Це дозволяє швидко розраховувати ознаки Хаара для зображення в навчанні і розпізнаванні.

Наприклад, розглянемо прямокутник ABCD рисунок 2.8.

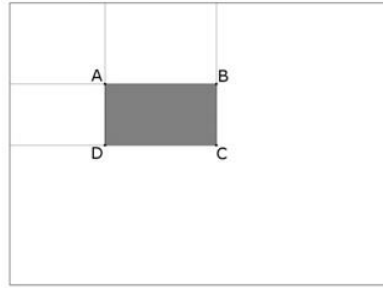


Рис. 2.8 — Прямокутник ABCD

Суму всередині прямокутника ABCD можна виразити через суми і різниці суміжних прямокутників за формулою:

$$\Sigma ABCD = II(A) + II(C) - II(B) - II(D) \quad (2.9)$$

#### Навчання

Перед початком навчання ініціалізуються ваги  $w(Q, i)$ , де  $q$  - номер ітерації,  $i$ -номер зображення.

$$w_{1,i} = \begin{cases} \frac{1}{2^l}, y_i=1 \\ \frac{1}{2^m}, y_i=0 \end{cases} \quad (2.10)$$

Після процедури навчання вийде  $T$  слабких класифікаторів і  $T$  значень.

$$a. p_j = \{1, -1\}, \theta_j = [\theta_{min}, \theta_{max}]. \quad (2.11)$$

$$t = 1..T \quad (2.12)$$

$$w_{t,i} = \frac{w_{t,i}}{\sum_{k=1}^n w_{t,k}} \quad (2.13)$$

$$foreach j \quad (2.14)$$

$$e_j = \min_{p,\theta} \sum_{i=1}^n w_{t,i} |h_j(x_i) - y_i| \quad (2.15)$$

$$J^{(t)} = arg \min_j (e_j) \quad (2.16)$$

$$\beta_t = \frac{e_{j^{(t)}}}{1 - e_{j^{(t)}}} \quad (2.17)$$

$$w_{t+1,i} = \begin{cases} w_{t,i} \beta_t, & h_{j^{(t)}}(x_i) = y_i \\ w_{t,i} & \end{cases} \quad (2.18)$$

$$a_t = \log \frac{1}{\beta_t} \quad (2.19)$$

На кожній ітерації циклу відбувається оновлення ваг так, що їх сума буде дорівнює 1. Далі для всіх можливих ознак відбувається підбір таких значень

$p, \theta, j$  що значення помилки  $e_j$  буде мінімально на цієї ітерації. отриманий ознака  $J(t)$  (на кроці  $t$ ) зберігається в базу слабких класифікаторів, оновлюються ваги і обчислюється коефіцієнт  $\alpha_t$ .

У запропонованому в 2001 році оригінальному алгоритмі була описана процедура отримання оптимального ознаки на кожній ітерації. Передбачається використання алгоритму AdaBoost і повний перебір можливих параметрів кордону і паритету.

#### Розпізнавання

Після навчання на тестовій вибірці є навчена база знань з  $T$  слабких класифікаторів. Для кожного класифікатора відомі: ознака Хаара, що використовується в цьому класифікаторі, його положення всередині вікна розміром  $24 \times 24$  пікселя і значення порога  $E$ .

На вхід алгоритму надходить зображення  $I(x, y)$  розміром  $W \times H$ , де  $I(x, y)$  - яркостная складова зображення. Результатом роботи алгоритму є безліч прямокутників  $R(x, y, w, h)$ , що визначають положення осіб в оригінальному документі  $I$ .

Алгоритм сканує зображення  $I$  на декількох масштабах, починаючи з базової шкали: розмір вікна  $24 \times 24$  пікселя і 11 масштабів, при цьому кожен наступний рівень в 1.25 рази більше попереднього, за рекомендацією авторів. Алгоритм розпізнавання виглядає наступним чином:

For scale=1 to 11

For t=1 to T

Для поточного значення scale масштабується розмір ковзаючого вікна і крок.

Обчислюється  $T$  значень слабких класифікаторів з бази, визначеної на етапі навчання. Для того, щоб класифікатор виніс ухвалу про те, що в даному вікні є присутнім шуканий шаблон - необхідно, щоб виконалася умова сильного класифікатора :

$$H(x) = \begin{cases} 1, \text{ якщо } \sum_{t=1}^T a^{(t)} h_j^{(t)} \geq \frac{1}{2} \sum_{t=1}^T a^{(t)} \\ 0, \text{ інше} \end{cases} \quad (2.20)$$

### Недоліки алгоритму Віюлі-Джонса

Базовий алгоритм Віюлі-Джонса (далі базовий алгоритм) має ряд недоліків:

- тривалий час роботи алгоритму навчання. Під час навчання алгоритм повинен аналізувати велику кількість тестових зображень;
- велика кількість близько розташованих один до одного результатів через застосування різних масштабів і ковзного вікна.

### 2.1.2 Фільтр Гаусса

Для видалення шуму на зображеннях обличчя використовували гаусівський фільтр. Гауссовий фільтр - це фільтр розмиття зображення, який використовує звичайний розподіл для обчислення перетворення, застосованого до кожного пікселя зображення.

Розмиття за Гаусом усуває небажаний шум зображення та мінімізує його вплив на класифікацію.

### 2.1.3 LBP перетворення

LBP оператор вперше запропонували в 1996 році для класифікації текстур [13]. Але пізніше знайшов застосування і для розпізнавання осіб. Але пізніше його використовували для розпізнавання облич. Сутність оператора може бути застосована до пікселів зображення порогового перетворення, де значення яскравості оброблюваних пікселів порівнюється зі значеннями яскравості пікселів, що його оточують. Результат порівняння кожного

оточуючого пікселя з обробленим пікселем конкатенуються в двійкове число.

Після використання оператора LBP зображення розміщується на прямокутних областях, для кожної з яких враховуються гістограми, оскільки в цій області часто є пікселі з різними суттєвими ознаками яскравості.

Отримані гістограми нормалізуються, конкатенуються і використовуються в подальшому в якості ознак класифікації. Приклад поділу зображення на прямокутні області і формування гістограм на рисунку 2.9.

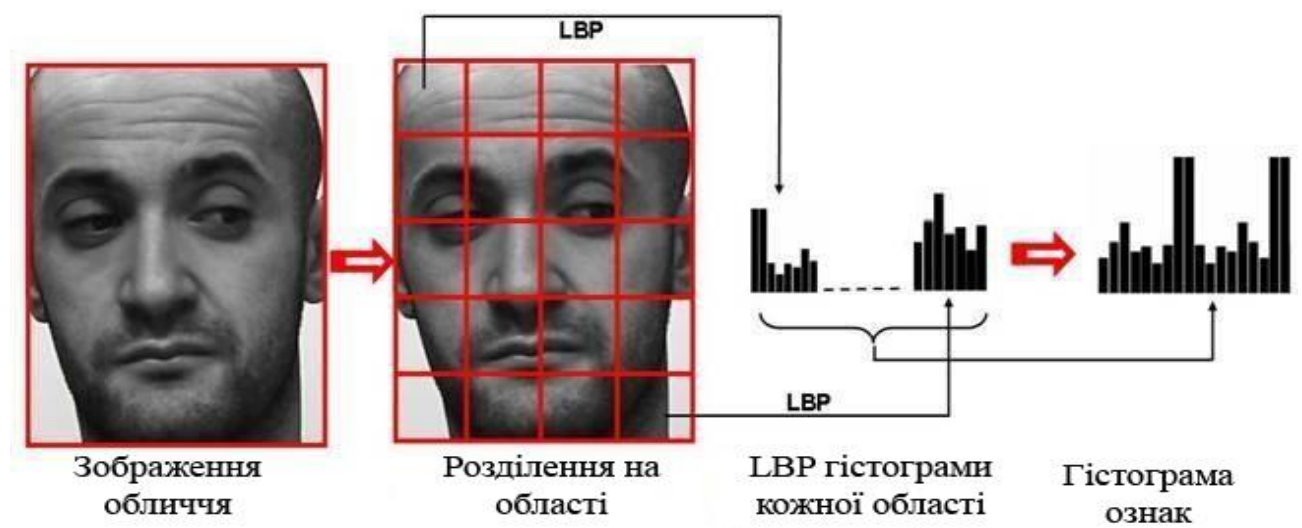


Рис. 2.9 — Розділення зображення на прямокутні області та формування гістограм

В результаті виходить опис зображення особи на трьох рівнях локалізації. При цьому такий опис не залежить від монотонних змін освітлення.

#### 2.1.4 Маска значущих областей зображення

Зображення облич, одержані після виявлення, квадратні. Але особа не займає весь простір такої картини. Тому логічно було б виключити вплив на рішення класифікатора областей зображення, де немає особи..

Найпростіший спосіб вирішити проблему - нанести маски на ключові ділянки зображення. Ця маска має той самий розмір, що і оброблене

зображення. Ненульові пікселі в масці відповідають значимим областям. У нашому випадку важлива область.

При вирішенні класифікації задач за допомогою локальних двійкових шаблонів доцільно застосовувати маску важливих областей після перетворення LBP та перед обчисленням гістограми. Таким чином, усі вторинні пікселі зображення в ракурсі підсумовуються в одне значення. Застосування маски значущих областей до зображення, створеного оператором LBP, наведено в рисунку 2.10.

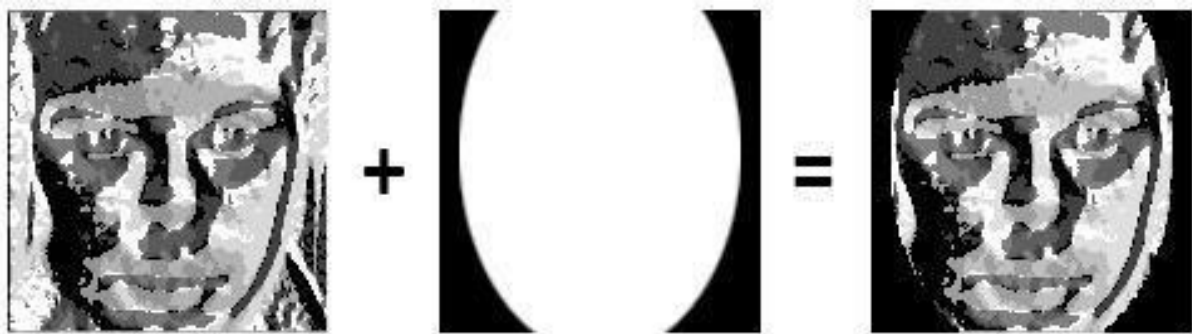


Рис. 2.10 — Застосування маски значущих областей

Як видно з рисунка 2.10, в результаті застосування даної операції пікселі зображення, які не повинні впливати на результат класифікації, мають нульове значення яскравості.

### 2.1.5 Метод найближчого сусіда

Цей метод є простим алгоритмом класифікації, суть якого полягає в тому, що об'єкт належить до класу, до якого він найближчий. Наприклад, на рисунку 2.11 зелене коло відповідно до цього алгоритмом повинен бути класифікований як червоний трикутник.

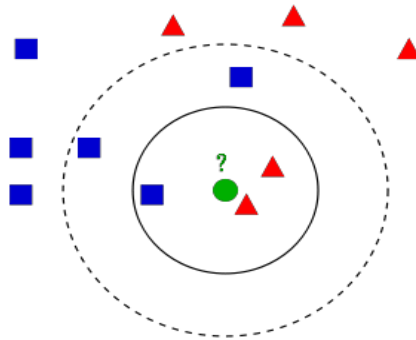


Рис. 2.11 — Метод найближчого сусіда

Щоб поліпшити результати, ви також використовуєте метод, в якому об'єкт належить до класу, до якого належить більшість його сусідів по сусідству певного розміру. Однак при вирішенні проблеми класифікації людей такий підхід негативно позначається на роботі класифікатора.

Цей метод застосовується, коли ціна помилки неправильної класифікації велика, а помилки даних малі. Основним недоліком цього методу є його чутливість до окремих (можливо, помилкових) значень даних. Проте, метод показує високу ефективність застосування в широкому спектрі завдань класифікації [14].

## 2.2 Інструментарій розробки

Розробка системи відбувалася на об'єктно-орієнтованій мові програмування Python в середовищі розробки Microsoft Visual Studio Code.

Microsoft Visual Studio Code - лінійка продуктів компанії Майкрософт, що включають інтегроване середовище розробки програмного забезпечення і ряд інших інструментальних засобів.

В якості мови програмування була вибрана мова Python - одна з найпопулярніших мов на сьогодні. Вибір був обґрунтован одночасно потужним та простим та зрозумілим синтаксисом, а також можливостями мультиплатформенності, що стане нагодою у подальших проектах. вбудовану підтримку узагальнень, делегатів і подій, що полегшить реалізацію. Структури даних високого рівня разом із динамічною семантикою та динамічним зв'язуванням роблять її привабливою для швидкої розробки програм, а також як засіб поєднування наявних компонентів.

Python підтримує модулі та пакети модулів, що сприяє модульності та повторному використанню коду. Інтерпретатор Python та стандартні модулі наявні як у скомпільованій, так і у вихідній формі на всіх основних платформах. В мові програмування Python підтримується кілька парадигм програмування, зокрема: об'єктно-орієнтована, процедурна, функціональна та аспектно-орієнтована.

Також вагомим критерієм на користь вибору даної мови програмування був досвід розробки, отриманий за час навчання.

Для розробки системи, було вирішено використовувати модулі `opencv-python` (в якості основної), `numpy`, `PIL`.

Основний модуль `OpenCV` містить алгоритми для обробки, реконструкції та очищення зображень, розпізнання образів, захоплення відео, стеження за об'єктами, калібрування камер та ін. Модуль розповсюджується за ліцензією `BSD`, отже, може вільно використовуватися в академічних і комерційних цілях.



## 2.3 Опис класів

В даному підрозділі наведено опис основних класів, функцій і методів розробленої програми.

Головний клас додатку `Face_classifier` містить методи для збору даних з веб-камери `add_face`, тренування на отриманих даних, а також сам метод класифікації облич. Так само він використовує в своїй роботі клас бібліотеки `opencv` осіб `FaceDetector` і LBP перетворювач зображень `LBPTransformer`.

В процесі створення тренувального датасету, кожне розпізнане обличчя зберігається в обрану теку. `RecognizedFace` при цьому посилається на відповідний йому ідентифікатор особи. Додаток не містить графічного інтерфейсу і керується консольними командами, що дозволяє, в подальшому, використовувати даний додаток на малопотужних пристроях. Детальний опис елементів інтерфейсу і механізмів для користувача взаємодії з ними будуть описані в розділі, присвяченому розробці інтерфейсу системи.

### RecognizedFace

Головний клас `RecognizedFace` містить методи для розпізнання та збору облич як даних для навчання, тренування, метод розпізнання облич.

Атрибути класу `RecognizedFace`:

*face\_detector* – об'єкт, що містить каскади Хаара для подальшого виявлення облич

*recognizer* – об'єкт, що посилається на клас модуля `opencv cv2.face.LBPHFaceRecognizer_create()`, описує CS-LBP перетворювач. Даний клас в подальшому виконуватиме центрально-симетричне LBP перетворення зображення.

*cam* – об'єкт, що виконує зчитування фреймів з відео-потoku

*ret* – атрибут класу, що повертає значення `True/False`, якщо кадр був коректно зчитаний

*frame* – об’єкт, що містить в собі поточний кадр з відео-потoku

*cv2.rectangle*- прямокутна область, яка відповідає стану розпізнання в кадрі відеопотоку

*count* – об’єкт-лічильник кадрів

*key* – об’єкт для розпізнання кнопки закриття вікна потоку

*imagePath* – об’єкт адреси до виявлених облич

*PIL\_img* – клас для обробки кадру

*Img\_numpy* – клас для зберігання кадру у вигляді numpy-масиву (для збільшення швидкодії)

*id* – ідентифікатор обличчя

*faces* - функція-класифікатор, що приймає такі параметри як масштабний коефіцієнт, кількість сусідів і мінімальний розмір виявленої особи.

*faceSamples* – масив для тимчасового зберігання облич, які були зафіксовані під час збору даних

*path* – адреса теки, що має містити в собі обличчя для тренування класифікатора

*font* – об’єкт з налаштуваннями шрифту виводу тексту поверх відеопотоку

*minW* – параметр ширини розміру кадру

*minH* – параметр висоти розміру кадру

*gray* – клас для перетворення фрейму в градації сірого

*ScaleFactor* - параметр, який визначає, наскільки зменшений розмір зображення при кожному масштабі зображення. Він використовується для створення піраміди масштабу.

*minNeighbors* - параметр, який вказує, скільки сусідів повинен мати кожен прямокутник, щоб зберегти його. Більше число дає менші помилкові спрацьовування.

*minSize* - мінімальний розмір прямокутника, який слід вважати обличчям.

*confidence* – параметр точності розпізнання обличчя

*recognizer* – виконує класифікацію облич на базі LBP-гістограм

*cv2.imshow* – клас для відображення відео-потoku

*cv2.waitKey* - клас для відображення відео-потoku

*cv2.destroyAllWindows* - клас для припинення роботи всіх вікон,

пов'язаних з відео-потокami

Метод *add\_face*:

Даний метод призначений для первинного розпізнання та збору облич в базу. Для коректної роботи алгоритму класифікації облич, необхідно багато True Positive зображень (зображень осіб) та True Negative зображень (зображень без осіб) для навчання класифікатора *faceCascade*.

Клас для розпізнання облич *faceCascade* був натренований вбудованими засобами OpenCV. Для його завантаження подаємо аргументом класу *faceCascade* шлях до файлу “haarcascade\_frontalface\_default.xml”, що був взятий з теки модуля OpenCV.

Після цього відбувається налаштування камери за допомогою класу *cam*. Всередині циклу завантажуються вхідні фрейми, перетворюючи їх у відтінки сірого, використовуючи клас *gray*. Зчитуємо та зберігаємо фрейми до теки з адресою *path*, доки не відлічимо 30 кадрів за допомогою параметру *count*, в яких буде виявлене обличчя з використанням класу *face\_detector*. Дostroкове переривання процесу збору даних можна виконати, нажавши клавішу “ESC”. Блок-схема алгоритму розпізнання облич наведена на рис. 2.12.

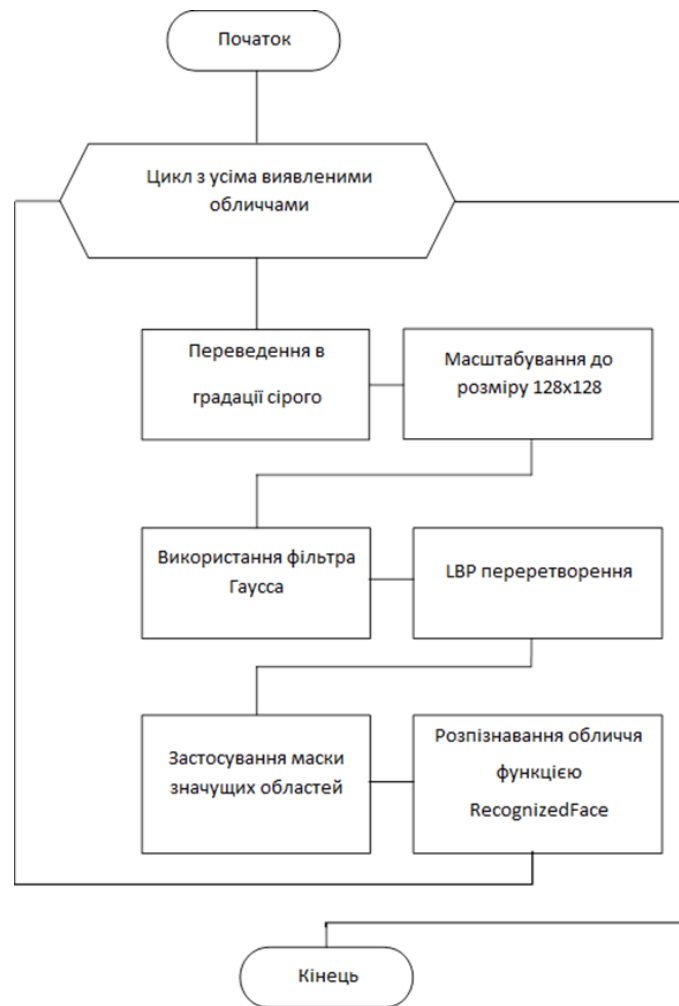


Рис. 2.12 — Блок-схема алгоритму обробки виявлених облич

Метод *train\_detector*:

Даний метод, використовує функцію головного класу *FaceClassifier* - *getImageAndLabels* для збору образів облич з теки з адресою *path*. Далі модель *recognizer* тренується на базі образів, та зберігає параметри моделі у файл. На цьому етапі ми повинні взяти всі дані з нашого набору і передати тренеру, після чого відбудеться навчання. Як результат отримаємо файл 'trainer.yml', який збережемо для подальшого використання.

Метод *recognize*:

Параметри натренованої моделі в форматі ".yml" подаються до об'єкту *recognizer*. Також застосовуємо клас *faceCascade* для подальшого розпізнання облич, де в якості аргументу подаємо адресу до файлу "haarcascade\_frontalface\_default.xml", що був взятий з теки модуля *Opencv*.

Далі ініціюємо роботу клас камери *cam*, встановлюємо розміри відео-потoku. Та визначаємо мінімальну область вікна для розпізнання обличчя.

Кадри відеопотоку фіксуються в циклі с можливістю перивання циклу за допомогою клавіші 'ESC'. В ході виконання циклу, отримані кадри передаються класу *cv2.cvtColor()*, за допомогою якого виконується перетворення зображення в градації сірого кольору.

Безпосередньо, перед класифікацією облич, необхідно провести процедуру їх виявлення за допомогою класу *faceCascade.detectMultiScale()*, що отримує на вхід: зображення (*frame*), на якому проводиться пошук осіб, а так само дані для настройки параметрів виявлення, а саме фактор збільшення скануючого вікна (*scaleFactor*), мінімальна кількість найближчих сусідів (*minNeighbors*) і мінімальний розмір області виявленого обличчя (*minSize*).

В ході наступного етапу алгоритму класифікації, проводимо ітерації за масивом виявлених облич *faces* та помічаємо області, де були виявлені обличчя зеленою рамкою за допомогою вбудованого методу *cv2.rectangle*, передаючи координати вершин області. Далі, переведені в градації сірого обличчя (*об'єкт gray*) подаються на вхід навченому класифікатору *recognize*. Класифікатор повертає ідентифікатор обличчя (атрибут *id*) та параметр *confidence*, що характеризує відсоток співпадання, а інакше – точність розпізнання.

У разі перевищення параметром *confidence* порогового значення, яке свідчить, що модель успішно класифікувала особу, яка надала дані для тренування моделі – перезаписуємо параметр *id* на її ім'я, інакше – перезаписуємо на текст "Unknown".

В ході останнього етапу алгоритму розпізнання, виконуємо рендер написів імені особи за індексом, що повернула модель та точність розпізнання (параметр *confidence*) на кадр відео-потoku та відображаємо кадр у вікні потoku. У разі натиснення на клавішу "ESC", відбудеться вихід з циклу *while*, здійсниться вивід повідомлення щодо закриття вікна, призупинення потoku *cam.release()*, та закриття вікна потoku *cv2.destroyAllWindows()*.

Блок-схема алгоритму розпізнання облич наведена на рис. 2.13.

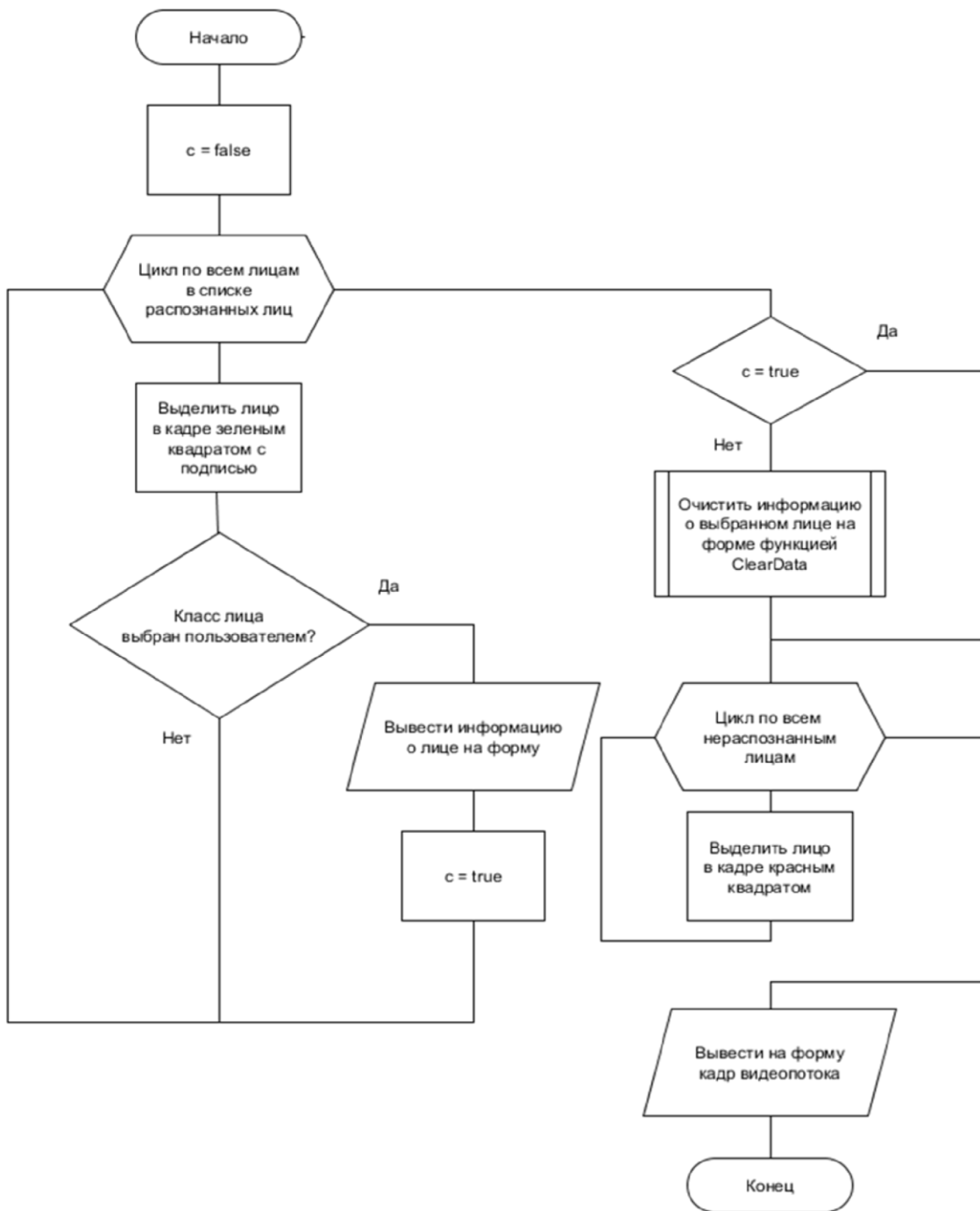


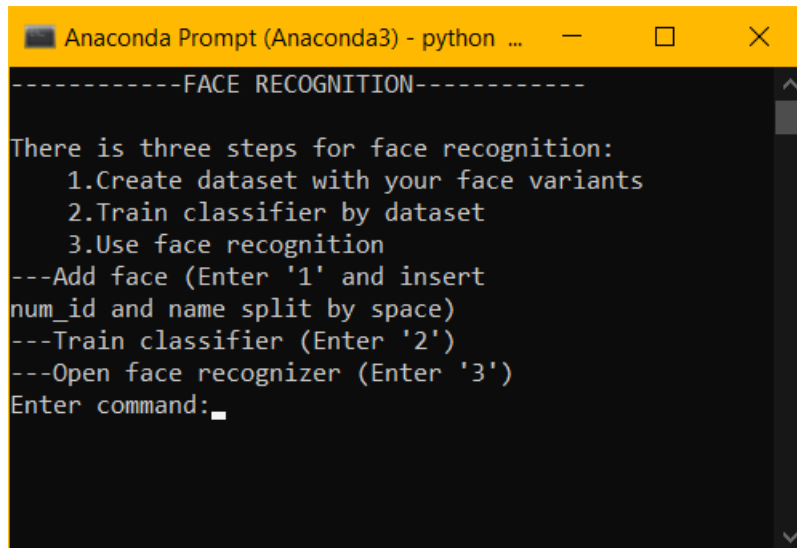
Рис. 2.13 — Блок-схема алгоритму виводу інформації щодо облич на форму

Повний код програми представлений в додатку А.

## 2.4 Опис інтерфейсу керування додатком

Керування додатком було вирішено реалізувати за допомогою консольного інтерфейсу та відповідних команд керування. Консоль надає можливість активувати методи додання нових облич до бази, тренування на усіх доданих обличчях, а також сам метод розпізнання. Обраний метод керування надає здатність використовувати додаток на малопотужних пристроях: мобільні пристрої, одноплатні ЕОМ.

Після введення консольних команд керування, додаток, в залежності від введеної команди, виводить нове вікно для розпізнання та додання нових облич, повідомлення щодо статусу класифікатора, або ж вікно класифікації облич, в якому безпосередньо і відображається результат роботи класифікатора. Загальний вигляд консолі керування додатком представлений на рисунку 2.14.



```
Anaconda Prompt (Anaconda3) - python ...
-----FACE RECOGNITION-----
There is three steps for face recognition:
  1.Create dataset with your face variants
  2.Train classifier by dataset
  3.Use face recognition
---Add face (Enter '1' and insert
num_id and name split by space)
---Train classifier (Enter '2')
---Open face recognizer (Enter '3')
Enter command: _
```

Рисунок 2.14 – Консоль керування додатком

Розглянемо детальніше процес керування додатком. Після введення команди «1», необхідно ввести ідентифікатор особи, наприклад, «1» та її ім'я через розділовий знак. Далі відкриється область виводу відео-потоків для збору даних.

Область виводу відеопотоку в процесі збору даних призначена для зворотнього зв'язку з користувачем під час збору даних для тренування моделі. У вікні відео-потoku відображається кожен поточний фрейм, де *faceCascade* розпізнає обличчя. Виявлені об'єктом обличчя зберігаються у теку для подальшого тренування моделі. Як джерело відеопотоку може використовуватися підключена до комп'ютера веб-камера.

Зупинка обробки і виведення кадрів з обраного джерела здійснюється натисканням кнопки “ESC”. Область виводу відео-потoku для збору даних представлена на рис 2.15.

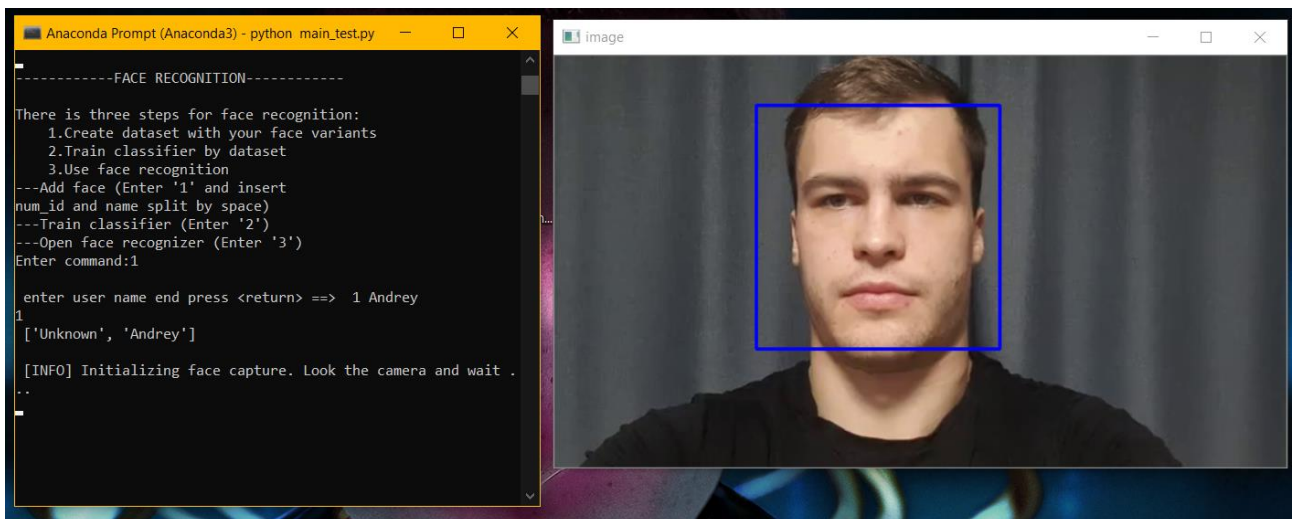


Рисунок 2.15 – Область виводу відео-потoku для збору навчальних даних

Після введення команди «2», відбудеться тренування на всіх даних, що містяться в теці *path*. Після тренування в консоль буде виведено повідомлення щодо завершення процесу.

Відкриття вікна відеопотоку з функціоналом класифікації на базі тренуваної моделі можливо за введенням команди «3». Дане вікно відображує всі поточні фрейми та виділяє усі виявлені обличчя.

Якщо класифікатор розпізнає обличчя, то, відповідно, на кадрі відеопотоку, над виявленим обличчям буде виведено ім'я особи, а також



відсоток точності співпадання LBP-гістограм. Область виводу результату відеопотоку представлена на рис. 2.16.

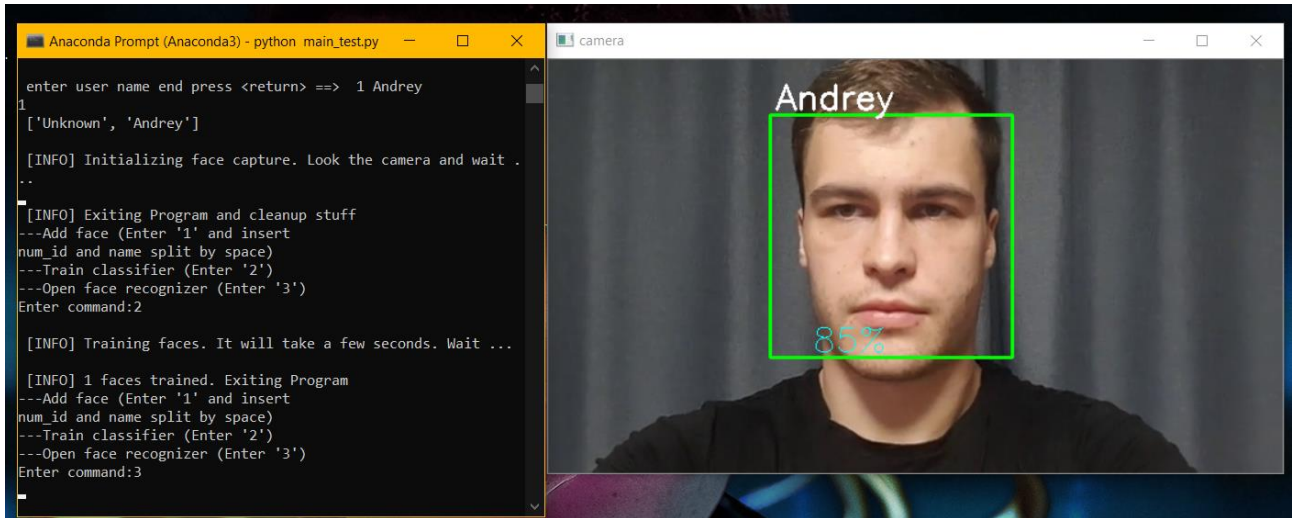


Рисунок 2.16 – Область виводу результату класифікації

## ВИСНОВОК

У цій дипломній роботі була розроблена система розпізнавання осіб у відеопотоках на основі методу Віоли-Джонса і методу локального двійкового шаблону.

Розроблена система може бути використана при вирішенні різних завдань відеоаналізу, має, перш за все, безпосереднє застосування в системах контролю доступу та ідентифікації особистості.

Основними напрямками подальшого розвитку розвинутої системи є вдосконалення роботи класифікатора осіб, для яких доцільно замінити алгоритми класифікації більш досконалішими.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Analysis market recognition technology [Электронный ресурс]. // TrendForce. Режим доступа: <https://www.trendforce.com>
2. Метод Виолы — Джонса [Электронный ресурс]. // Wikipedia. – Режим доступа: [https://ru.wikipedia.org/wiki/Метод\\_Виолы\\_—\\_Джонса](https://ru.wikipedia.org/wiki/Метод_Виолы_—_Джонса)
3. Видеоаналитика и распознавание лиц [Электронный ресурс]. // Wikipedia. – Режим доступа: [https://video-praktik.ru/st\\_videoanalitika.html](https://video-praktik.ru/st_videoanalitika.html)
4. Метод опорных векторов [Электронный ресурс]. // Wikipedia. – Режим доступа: [https://ru.wikipedia.org/wiki/Метод\\_опорных\\_векторов](https://ru.wikipedia.org/wiki/Метод_опорных_векторов)
5. Актуальность 2D алгоритмов в определенных задачах автоматического распознавания [Электронный ресурс]. // Реноме. – Режим доступа: <https://moluch.ru/conf/tech/archive/2/138>
6. Хабрахабр - крупнейший ресурс для IT-специалистов. [Электронный ресурс]. // Режим доступа – <https://habr.com/post/133826/>
7. Системы технической безопасности и охраны «МТИ». FaceVACS – VideoScan [Электронный ресурс]. // Режим доступа: <http://www.security.mti.ua/products/sistemy-videonabludeniya/Soft-raspoznavanie-liz/Cognitec/152-facevacsvideoscan>
8. NEC. Facial Recognition [Электронный ресурс]. // Режим доступа: [https://ru.nec.com/solutions/security/technologies/face\\_recognition.html](https://ru.nec.com/solutions/security/technologies/face_recognition.html)
9. VisionLabs. LUNA SDK [Электронный ресурс]. // Режим доступа: <https://visionlabs.ai/ru/luna-platform-info.html>
10. Neurotechnology. VeriLook SDK [Электронный ресурс]. // Режим доступа: <http://www.neurotechnology.com/>
11. Локальные бинарные шаблоны [Электронный ресурс]. // Режим доступа: [https://ru.wikipedia.org/wiki/Локальные\\_бинарные\\_шаблоны](https://ru.wikipedia.org/wiki/Локальные_бинарные_шаблоны)

12. Татаренков Д. А. Анализ методов обнаружения лиц на изображении [Электронный ресурс]. // Молодой ученый. — 2015. — №4. — С. 270-276. — Режим доступа: <https://moluch.ru/archive/84/15524/>

13. Броневи́ч А. Н. Лекции по методам машинного обучения [Электронный ресурс]. // Режим доступа: [http://window.edu.ru/resource/files/lect\\_Lepskiy\\_Bronevich.pdf](http://window.edu.ru/resource/files/lect_Lepskiy_Bronevich.pdf)

14. Методы ближайшего соседа и k-ближайших соседей [Электронный ресурс]. // Режим доступа: [http://studbooks.net/2429081/informatika/metody\\_blizhayshego\\_sosedya\\_blizhayshih\\_sosedey](http://studbooks.net/2429081/informatika/metody_blizhayshego_sosedya_blizhayshih_sosedey)

15. Язык программирования С#. Классика Computers Science. 4-е изд. / А. Хейлсбер, М. Торгерсен, С. Вилтамут, П. Голд. СПб.: Питер, 2011, 784 с.

16. About OpenCV [Электронный ресурс]. // Режим доступа: <http://opencv.org/about.html>

17. EmguCV [Электронный ресурс]. // Режим доступа: [http://www.emgu.com/wiki/index.php/Main\\_Page](http://www.emgu.com/wiki/index.php/Main_Page)

18. Разработка мультимедийных приложений с использованием библиотек OpenCV и IPP [Электронный ресурс]/ А.В. Бовырин [и др.].— Электрон. текстовые данные.— М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 515 с.— Режим доступа: <http://www.iprbookshop.ru/39564>

СТО 4.2-07-2014 Система менеджмента качества. Общие требования к построению, изложению и оформлению документов учебной деятельности.— Введ. 9.01.2014. — Красноярск: ИПК СФУ, 2014. — 60 с.

## ДОДАТОК А

### Лістинг коду

```
# -*- coding: utf-8 -*-

import cv2 #власне модуль OpenCV
import os #знадобиться для роботи з файловою системою
import numpy as np #більш швидкі масиви
from PIL import Image #для перетворень в градації сірого

#Основной класс
class Face_classifier():
    def __init__(self):
        self.face_detector = cv2.CascadeClassifier('haarcascade_frontalface_default.xml') #визначник облич
        self.names = ['Unknown'] #Лістинг з іменами людей в базі

#метод збору даних для навчання
def add_face(self):
    # Для кожної особи вводимо ідентифікатор та її ім'я
    self.face_id, face_name = input('\n enter user name end press <return> ==> ').split()
    self.names.append(face_name)
    print(self.face_id, '\n',self.names)
    print("\n [INFO] Initializing face capture. Look the camera and wait ...")
    self.count = 0 #змінна для підрахунку кількості кадрів для навчання
    self.cam = cv2.VideoCapture(0)

#Встановлюємо роздільну здатність
self.cam.set(3, 640)
self.cam.set(4, 480)

#збираємо кадри
while(True):
    ret, img = self.cam.read() #зчитуємо фрейми в img
```

```

gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) #перетворюємо в градації сірого
faces = self.face_detector.detectMultiScale(gray, 1.3, 5) #детектимо обличчя з урахуванням
масштабу

#будуємо область розпізнання (прямокутник)
for (x,y,w,h) in faces:

    cv2.rectangle(img, (x,y), (x+w,y+h), (255,0,0), 2)
    self.count += 1

    # зберігаємо обличчя
    cv2.imwrite("dataset/User." + str(self.face_id) + '.' + str(self.count) + ".jpg", gray[y:y+h,x:x+w])
    cv2.imshow('image', img)

k = cv2.waitKey(100) & 0xff # Для закриття вікна тиснемо 'ESC'
if k == 27:
    break

elif self.count >= 30: # Рообимо 30 кадрів з обличчям та завершуємо запис
    break

print("\n [INFO] Exiting Program and cleanup stuff")

# Закриваємо вікно
self.cam.release()
cv2.destroyAllWindows()

#метод повернення тренувальних даних, повертає обличчя та id особи
def getImagesAndLabels(self, path):
    #парсимо адреси до тренувальних файлів
    imagePath = [os.path.join(path,f) for f in os.listdir(path)]
    faceSamples=[]
    ids = []

    for imagePath in imagePath:

```

```

PIL_img = Image.open(imagePath).convert('L') # конвертуємо в градації сірого
img_numpy = np.array(PIL_img,'uint8') #переводимо зображення в numpy-масив

#Парсимо id з назви файлу
id = int(os.path.split(imagePath)[-1].split(".")[1])
faces = self.face_detector.detectMultiScale(img_numpy)

#отримуємо дані, що будемо повертати
for (x,y,w,h) in faces:
    faceSamples.append(img_numpy[y:y+h,x:x+w])
    ids.append(id)

return faceSamples,ids

#метод тренування класифікатора
def train_detector(self):
    path = 'dataset' #Назва папки з даними для навчання

    #Ініціалізуємо розпізнавач та класифікатор
    recognizer = cv2.face.LBPHFaceRecognizer_create()
    detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
    print ("\n [INFO] Training faces. It will take a few seconds. Wait ...")

    faces,ids = self.getImagesAndLabels(path)
    # Навчаємо класифікатор
    recognizer.train(faces, np.array(ids))

    #Зберігаємо параметри класифікатора в файл
    recognizer.write('trainer/trainer.yml')

    # Виводимо число осіб, які тепер можемо розпізнавати
    print("\n [INFO] {0} faces trained. Exiting Program".format(len(np.unique(ids))))

```

```
#метод для класифікації
def recognize(self):

    recognizer = cv2.face.LBPHFaceRecognizer_create()
    recognizer.read('trainer/trainer.yml')
    cascadePath = "haarcascade_frontalface_default.xml"
    faceCascade = cv2.CascadeClassifier(cascadePath);

    font = cv2.FONT_HERSHEY_SIMPLEX

    #ініціалізуємо лічильник особи
    id = 0

    # ініціалізуємо веб-камеру та запускаємо вікно
    cam = cv2.VideoCapture(0)
    # задаємо параметри відео
    cam.set(3, 640)
    cam.set(4, 480)

    # Визначаємо мінімальний розмір області розпізнання обличчя
    minW = 0.1*cam.get(3)
    minH = 0.1*cam.get(4)

    while True:
        #зчитуємо кадри відео
        ret, img =cam.read()

        #перетворюємо в градації сірого
        gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)

        #задаємо параметри виявлення облич
        faces = faceCascade.detectMultiScale(
```



```

gray,
scaleFactor = 1.2,
minNeighbors = 5,
minSize = (int(minW), int(minH)),
)
#мітимо розпізнані обличчя
for(x,y,w,h) in faces:
    #будуємо навколо обличчя прямокутник
    cv2.rectangle(img, (x,y), (x+w,y+h), (0,255,0), 2)
    #отримаємо id особи та точність класифікації
    id, confidence = recognizer.predict(gray[y:y+h,x:x+w])

    # дивимося процент співпадання та визначаємо яким буде ім'я особи та точність
    if (confidence < 100):
        id = self.names[id]
        # виводимо точність
        confidence = " {0}%".format(round(100 - confidence))
    else:
        id = "unknown"
        confidence = " {0}%".format(round(100 - confidence))
    #виводимо ім'я особи та точність
    cv2.putText(img, str(id), (x+5,y-5), font, 1, (255,255,255), 2)
    cv2.putText(img, str(confidence), (x+5,y+h-5), font, 1, (255,255,0), 1)

# виводимо фрейм
cv2.imshow('camera',img)

# для виходу тиснемо "ESC"
k = cv2.waitKey(10) & 0xff
if k == 27:
    break

# Закриваємо вікно виводу потоку

```

```

print("\n [INFO] Exiting Program and cleanup stuff")
cam.release()
cv2.destroyAllWindows()

```

```

print("""
-----FACE RECOGNITION-----

```

There is three steps for face recognition:

- 1.Create dataset with your face variants
- 2.Train classifier by dataset
- 3.Use face recognition"""

```
)
```

```
recognizer = Face_classifier()
```

```
while True:
```

```
    a = int(input(
```

```
    ""---Add face (Enter '1' and insert
    num_id and name split by space)
```

```
    ---Train classifier (Enter '2')
```

```
    ---Open face recognizer (Enter '3')
```

```
    Enter command:"))
```

```
    if a == 1:
```

```
        recognizer.add_face()
```

```
    elif a == 2:
```

```
        recognizer.train_detector()
```

```
    elif a == 3:
```

```
        recognizer.recognize()
```