

Міністерство освіти і науки України

Луцький національний технічний університет

(повне найменування закладу вищої освіти)

Факультет комп'ютерних та інформаційних технологій

(повне найменування факультету)

Кафедра комп'ютерної інженерії та кібербезпеки

(повне найменування кафедри)

КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»

НЕЙРОМЕРЕЖЕВІ ЗАСОБИ РОЗПІЗНАВАННЯ ОБЛИЧ У
ВІДЕОПОТОКАХ НА ОСНОВІ МЕТОДА ВІОЛИ-ДЖОНСА

NEURAL NETWORK TOOLS FOR FACE RECOGNITION IN
VIDEO STREAMS BASED ON THE VIOLA-JONES METHOD

спеціальність 123 Комп'ютерна інженерія

(шифр і назва спеціальності)

освітня програма Комп'ютерна інженерія

(назва освітньої програми)

Виконав: здобувач вищої освіти
групи КІс-21

Іванісік Владислав Васильович

(підпис)

Керівник:

д.е.н., професор

Ляшенко Оксана Миколаївна

(підпис)

Кваліфікаційну роботу

допущено до захисту

« 2 » червня 2023 р.

Гарант освітньої програми:

к.т.н., доцент

Лавренчук Світлана Василівна

(підпис)

Луцьк – 2023 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій
Кафедра комп'ютерної інженерії та кібербезпеки
Ступінь вищої освіти: бакалавр
Галузь знань: 12 Інформаційні технології
Спеціальність: 123 Комп'ютерна інженерія
Освітня програма: «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ
Завідувач кафедри

« _____ » _____ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Іванісіку Владиславу Васильовичу
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи *Нейромережеві засоби розпізнавання облич у відеопотоках на основі метода Віоли-Джонса*

Керівник роботи *д.е.н., професор Ляшенко Оксана Миколаївна*

затверджені наказом закладу вищої освіти від «28» грудня 2022 року № 982/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи 01.06.2023р.

3. Вихідні дані до роботи *Джерелом розробки є науково-технічна література та публікації в періодичних виданнях з даного питання, опубліковані зарубіжні та вітчизняні роботи в даній області, різні інтернет-ресурси технічного спрямування*

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

Вступ

Теоретична частина

Аналітична частина

Рекомендаційна частина

Висновки

5. Перелік графічного (ілюстративного) матеріалу:

Практична цінність розроблена система може використовуватись при вирішеннях різних завдань відеоаналітики, і, в першу чергу, має безпосереднє застосування в системах контролю доступу і ідентифікації особистості

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
<i>Аналіз проблеми за темою роботи та постановка завдань дослідження</i>	<i>Ляшенко О.М.</i>		
<i>Теоретичне дослідження та практична реалізація</i>	<i>Ляшенко О.М.</i>		
<i>Практична реалізація об'єкта проектування</i>	<i>Ляшенко О.М.</i>		
<i>Висновки</i>	<i>Ляшенко О.М.</i>		

7. Дата видачі завдання 01.11.2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	<i>Обґрунтування теми</i>	До 15.11.2022 р.	
2.	<i>Огляд літератури із досліджуваної проблеми</i>	До 15.12.2022 р.	
3.	<i>Розділ 1</i>	До 02.02.2023 р.	
4.	<i>Розділ 2</i>	До 02.03.2023 р.	
5.	<i>Висновки та пропозиції</i>	До 02.04.2023 р.	
6.	<i>Формування списку використаних джерел</i>	До 15.04.2023 р.	
7.	<i>Формування додатків</i>	До 02.05.2023 р.	
8.	<i>Оформлення ілюстративного матеріалу</i>	До 15.05.2023 р.	
9.	<i>Нормоконтроль</i>	До 25.05.2023 р.	
10.	<i>Інструментальна перевірка на академічний плагіат</i>	До 01.06.2023 р.	
11..	<i>Представлення кваліфікаційної роботи бакалавра до захисту</i>	До 07.06.2023 р.	

Здобувач вищої освіти

(підпис)

Іванісік В.В.

(прізвище, ініціали)

Керівник кваліфікаційної роботи

(підпис)

Ляшенко О.М.

(прізвище, ініціали)

АНОТАЦІЯ

Іванісік В.В. Нейромережеві засоби розпізнавання облич у відеопотоках на основі методу Віоли-Джонса. Рукопис.

Кваліфікаційна робота бакалавра ОП "Комп'ютерна інженерія" спеціальності 123 Комп'ютерна інженерія. Луцький національний технічний університет. Луцьк, 2023.

Кваліфікаційна робота складається зі вступу, трьох розділів, висновків, списку використаних джерел.

Перший розділ присвячено огляду відеоаналітики, технології, яка використовує комп'ютерний зір для автоматичного збору інформації про обличчя людей у послідовності кадрів, отриманих з відеокамер у реальному часі або з відеозаписів.

У другому розділі було розглянуто програмне забезпечення, цю технологію можна використовувати в системах відеоспостереження, системах безпеки, транспорті.

Розділ третій. Створено розпізнавання обличчя в реальному часі. Розпізнавання облич у відеопотоках, дослідження ефективності різних варіацій локальних бінарних шаблонів щодо задачі розпізнавання облич у реальному часі, розробка системи розпізнавання облич.

Об'єкт – процес розпізнавання обличчя.

Предмет – алгоритми та методи виявлення облич у кадри та безпосереднього розпізнавання знайдених облич.

Мета роботи – покращення розпізнавання облич в реальному часі.

розпізнавання облич у відеопотоках, дослідження ефективності різних варіацій локальних бінарних шаблонів стосовно завдання розпізнавання облич в реальному часі, розробка системи розпізнавання облич.

Ключові слова: метод Віоли-Джонса, фільтр Гаусса, перетворення LBP.

ANNOTATION

Ivanisik V.V. Neural Network Tools for Face Recognition in Video Streams Based on the Viola-Jones Method. Manuscript.

Bachelor's Qualification Work of the Educational Program "Computer Engineering" in the specialty 123 Computer Engineering. Lutsk National Technical University. Lutsk, 2023. The qualification work consists of an introduction, three chapters, conclusions, and a list of references.

The first chapter is devoted to the review of video analytics, a technology that uses computer vision to automatically collect information about people's faces in sequences of frames captured from video cameras in real time or from video recordings.

The second chapter discusses software; this technology can be used in video surveillance systems, security systems, and transportation.

Chapter three. Real-time face recognition was created. Face recognition in video streams, investigation of the effectiveness of different variations of local binary patterns for real-time face recognition tasks, development of a recognition system.

Object – the process of facial recognition. Subject – algorithms and methods for detecting faces in a frame and directly recognizing the detected faces. The purpose of the work is to improve real-time facial recognition, recognize faces in video streams, study the effectiveness of different variations of local binary patterns for real-time facial recognition tasks, and develop a facial recognition system.

Keywords: Viola-Jones method, Gaussian filter, LBP transformation.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 ОГЛЯД МЕТОДІВ РОЗПІЗНАВАННЯ ОБЛИЧ.....	10
1.1 Огляд існуючих методів розпізнавання облич	10
1.2 Метод гнучкого порівняння на графах.....	12
1.3 Згорткові нейронні мережі.....	14
1.4 Метод прихованих Марковських моделей.....	15
РОЗДІЛ 2 РОЗПІЗНАВАННЯ ОБЛИЧЧЯ МЕТОДОМ ВІОЛИ-ДЖОНСА.....	18
2.1 Фактори, що впливають на якість розпізнавання облич	18
2.2 Аналіз системи розпізнавання обличчя у відеопотоках	19
2.3 Метод Віоли-Джонса.....	22
РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ	29
3.1 Інструментарій розробки.....	29
3.2 Загальний алгоритм обробки кадрів	30
3.3 Реалізація основних класів.....	32
3.4 Тестування розробленої системи	37
ВИСНОВОК.....	39
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	40
ДОДАТКИ.....	42

ВСТУП

Розпізнавання облич – це ключова біометрична технологія для посилення безпеки та розвитку інтелектуальних систем. Хоча вже тривалий час ведуться роботи над створенням ефективних алгоритмів, штучні нейронні мережі (НМ) визнані найбільш перспективним рішенням.

Сьогодні НМ широко використовуються у відеоаналітиці – технології, що застосовує комп'ютерний зір для автоматичного аналізу відеоданих (з камер або записів) в реальному часі.

Проте, незважаючи на велику кількість існуючих нейромережевих моделей, вони стикаються зі значними труднощами при розпізнаванні деформованих зображень (наприклад, зміщених, повернутих чи масштабованих). Успішне вирішення цієї проблеми в реальних умовах залежить від оптимального вибору архітектури та стратегії навчання нейромережі.

Підвищений інтерес до біометричних технологій зумовлений їхньою високою ефективністю у соціальних та комерційних сферах. Зокрема, розпізнавання облич стало незамінним інструментом у правоохоронній діяльності, наприклад, автоматичний моніторинг підозрюваних, пошук зниклих осіб та активне відеоспостереження під час надзвичайних ситуацій. У фінансовому секторі біометрія застосовується для ідентифікації в банкоматах та системах віддаленого керування рахунками.

Біометрична технологія використовується для верифікації особи (паспорти, водійські права, імміграційні картки) та в інформаційній безпеці для розпізнавання облич, що забезпечує контрольований доступ до комп'ютерів, програм, баз даних, медичних записів та електронної комерції.

Традиційні системи ідентифікації покладаються на паролі, ключі чи картки, які є вразливими до забування чи втрати. На противагу цьому, біометричні системи використовують унікальні фізичні характеристики людини (як-от відбитки пальців, форма долоні, райдужна оболонка, сітківка чи особливості обличчя). Ці характеристики значно важче підробити, і вони забезпечують однозначну ідентифікацію особи.

Незважаючи на численні дослідження в цій галузі, проведені в усьому світі протягом останніх десятиліть, не розроблено методів надійного виявлення та розпізнавання людських облич за будь-яких умов.

Наразі використовується та модифікується велика кількість методів розпізнавання облич, серед яких є класичні методи (метод Віоли-Джонса, пружні графи, аналіз головних компонент (РСА) та геометричні методи розпізнавання) та нейромережеві методи (згорткові нейронні мережі (CNN), метод прихованої моделі Маркова (НММ) та метод локальних бінарних патернів (LBP) та інші).

Навіть при спільному наборі чинників, що впливають на якість, кожен із методів розпізнавання облич має свою унікальну стійкість до певних змін, що і визначає його переваги та недоліки. Ці фактори охоплюють як умови зйомки – рівень і кут освітлення, відстань до камери та кут нахилу обличчя відносно неї – так і зміни в зовнішньому вигляді самої людини, зокрема стан міміки, настрій, вікові трансформації, наявність бороди чи вусів.

Наявні методи розпізнавання облич часто розв'язують проблему із неприйнятними часовими затримками. Однак існують підходи, які можуть суттєво прискорити цей процес, наприклад, шляхом комбінування методу Віоли-Джонса з локальними бінарними патернами. Дана робота присвячена розробці такої системи, яка вирішує цю проблему за допомогою методу Віоли-Джонса та локальних бінарних шаблонів.

Метою роботи є розробка системи виявлення облич у кадрі відеопотоку за методом Віоли-Джонса та розпізнавання знайдених облич за методом найближчого сусіда з використанням локальних бінарних шаблонів.

Об'єктом дослідження: методи розпізнавання облич, включаючи виявлення облич у кадрі відеопотоку за методом Віоли-Джонса.

Предмет дослідження: система виявлення облич у кадрі відеопотоку.

На основі поставленої мети необхідно виконати наступні завдання:

– здійснити аналітичний огляд розпізнавання облич, а також порівняння з іншими відомими методами;

- розробити систему, яка включає виявлення облич у кадрі відеопотоку за методом Віоли-Джонса та розпізнавання знайдених облич за методом найближчого сусіда з використанням локальних бінарних шаблонів;
- здійснити тестуванням працездатності розробленої системи з різними налаштуваннями методів розпізнавання та наборів ознак.

РОЗДІЛ 1

ОГЛЯД МЕТОДІВ РОЗПІЗНАВАННЯ ОБЛИЧ

1.1 Огляд існуючих методів розпізнавання облич

Завдання розпізнавання обличчя в кадрах відеопотоку можна розділити на два етапи. Перший етап це визначення облич у кадрі, а другий – пряме розпізнавання знайдених облич.

В загальному, весь процес розпізнавання облич на зображеннях можна зобразити наступною схемою (рис.1.1) [18]:

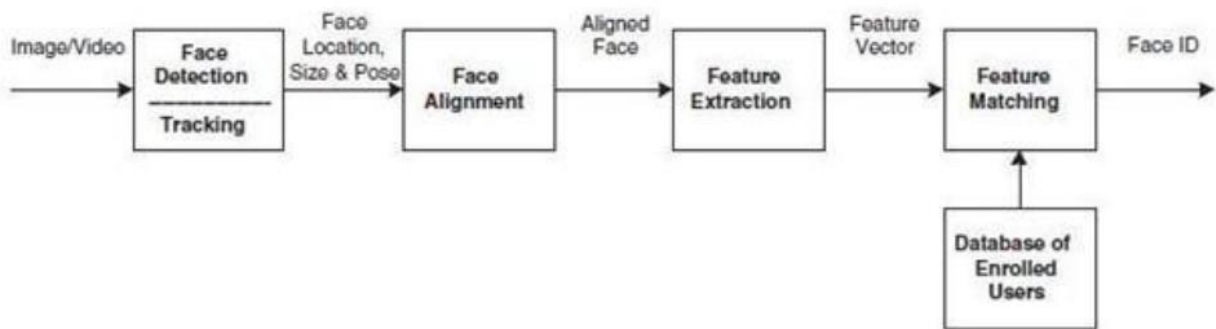


Рисунок 1.1 – Процес розпізнавання обличчя

Процес розпізнавання облич починається з виявлення та точної локалізації обличчя на зображенні. Після цього, зображення обличчя нормалізується, тобто вирівнюється як геометрично, так і за рівнем яскравості, для забезпечення стандартних умов обробки. Наступним кроком є обчислення унікального набору ознак, за яким слідує безпосередньо розпізнавання – зіставлення цих визначених ознак зі зразками, що зберігаються в базі даних. Принципова відмінність між існуючими алгоритмами полягає виключно у методах, які вони використовують для обчислення ознак та подальшого порівняння їхніх сукупностей.

Кожен метод має свої переваги та недоліки і їх можна умовно поділити на дві основні групи: класичні та нейромережеві методи. До класичних методів, що базуються на математичних моделях та статистичному аналізі ознак, можна віднести:

- метод Віоли-Джонса, який використовується переважно для швидкого виявлення обличчя на зображенні;
- метод гнучкого порівняння на графах, що складає графічну модель обличчя для розпізнавання;
- аналіз головних компонент, який зменшує розмірність даних, використовуючи «власні обличчя» (eigenfaces) для ідентифікації;
- геометричні методи, що базуються на вимірюванні відстаней між ключовими точками обличчя (очі, ніс, рот).

До нейромережевих методів, які використовують машинне навчання для вивчення складних патернів відносять:

- згорткові нейронні мережі, що автоматично видобувають високоабстрактні ознаки з зображень, забезпечуючи високу точність;
- метод прихованої моделі Маркова, що використовується для моделювання послідовних даних, які може бути застосовано для розпізнавання обличчя як послідовності ознак;
- метод локальних бінарних патернів для опису текстури зображення.

Кожен з перелічених методів має свої переваги та недоліки, що впливають на швидкість та якість розпізнавання. Наприклад, якщо розглядати задачу автоматичного визначення статі людини з використанням BIF ознак (biologically inspired features) [9] та методу опорних векторів [10], то алгоритм може включати наступні кроки, як наведено у роботі А.С. Спижевой та Д.В. Баландіна:

- 1) геометрична нормалізація та нормалізація яскравості особи;
- 2) обчислення біологічно обумовлених ознак;
- 3) зниження розмірності вектора-ознак за допомогою методу головних компонентів або методу, заснованого на використанні дерев рішень;
- 4) класифікація статі за допомогою методу опорних векторів з rbf ядром, реалізований у бібліотеці OpenCV [11].

Розглянемо особливості, переваги та недоліки деяких запропонованих методів.

1.2 Метод гнучкого порівняння на графах

Даний метод полягає в еластичному порівнянні графів, які описують зображення обличчя. В даному випадку обличчя виглядають як графи з вершинами і ребрами. При цьому, при ідентифікації, еталонний граф не змінюється, в той час як інший змінює форму. Це відбувається задля того, щоб найкращим чином підігнати граф, що змінюється, до зразкового. Графи в таких системах розпізнавання можуть виглядати і як прямокутна решітка, і як утворена антропометричними точками людини структура.

Аналіз обличчя за допомогою EBGM (Elastic Bunch Graph Matching). Графік групи облич (посередині) зіставляється з новим зображенням обличчя (ліворуч). Для кожного вузла (маленькі чорні крапки) у графі групи є локальний експерт (на який вказують стрілки), з якого можна синтезувати так зване фантомне обличчя та вивести абстрактні ознаки, такі як стать, окуляри або борода. У цьому випадку фантомне обличчя виглядає європеоїдним, хоча оригінальне обличчя наполовину азіатське, оскільки граф групи включає лише європеоїдні обличчя (рис. 1.2).

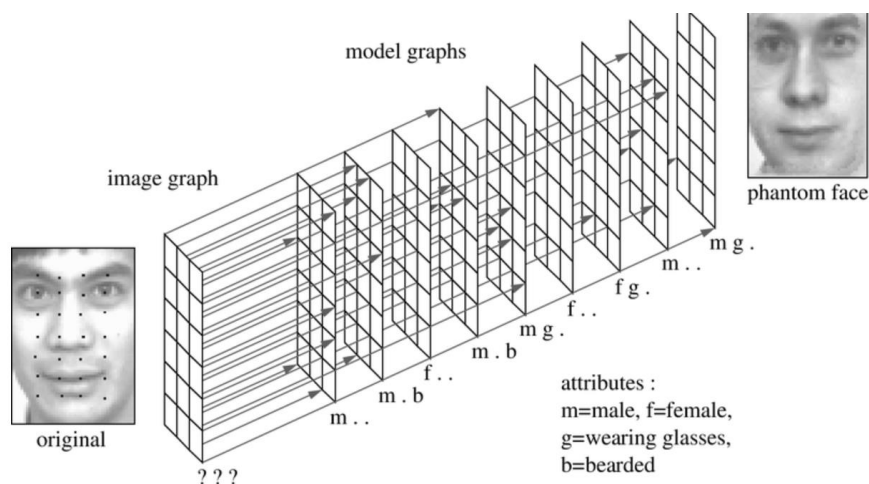


Рисунок 1.2 – Аналіз обличчя за допомогою EBGM.

На вершинах графа обчислюються значення ознак. Частіше за все використовують комплексні значення фільтрів Габора або їх впорядкованих наборів – Габорівських вейвлет (строї Габора). Ці значення обчислюються в

певній локальній області вершини графа локально шляхом згортки значень яскравості пікселів з фільтрами Габора.

Відстань між суміжними вершинами необхідна для зважування ребер графа. За допомогою деякої цінової функції можна власне обчислити відстань між двома графами. Дана функція для виконання обчислення враховує рівень деформації графа і різницю між значеннями ознак.

Зміна форми графа відбувається таким чином: кожна з його вершин зсувається на певну відстань і в певних напрямках, відносно її початкового розташування. Далі відбувається вибір такої позиції вершини, при якій буде мінімальною різниця між значеннями ознак (відгуків фільтрів Габора) в вершині деформованого графа і відповідної їй вершині еталонного графа. Ця дія буде виконуватися для всіх вершин графа по чергово до тих пір, поки не буде знайдено найменшу загальну відмінність між ознаками еталонного і зміненого графів. В цьому випадку мірою відмінності між початковим зображенням обличчя і еталонним графом буде значення цінової функції деформації при такому положенні деформуючого графа. Така процедура деформації виконується для всіх еталонних осіб, що додані до бази даних системи. Результатом розпізнавання системи є еталон, який має найкраще значення цінової функції деформації.

Даний метод дуже стійкий до змін міміки обличчя та незначних поворотів у площині, оскільки пружне порівняння ефективно компенсує ці деформації. На відміну від глобальних методів (як PCA), EBGM моделює структурні відносини між елементами обличчя. Теоретично, часткове закриття обличчя впливає лише на закриті вузли, але не на загальну топологію графа.

До недоліків цього методу можна віднести складність обчислень, т.б. обчислення локальних ознак (векторів Габора) та подальше пружне зіставлення є обчислювально дорогим процесом, що може призводити до повільної роботи у реальному часі.

1.3 Згорткові нейронні мережі

Сучасні дослідження налічують близько десяти різновидів нейронних мереж (НМ), серед яких найпопулярнішими є ті, що ґрунтуються на багат шаровому перцептроні і застосовуються для класифікації вхідних зображень чи сигналів згідно з попереднім навчанням. Процес навчання НМ є ключовим: він відбувається на основі навчальних прикладів, під час чого відбувається налаштування ваг міжнейронних зв'язків за допомогою методу градієнтного спуску як оптимізаційного рішення.

Основні етапи цього процесу включають автоматичне вилучення ключових властивостей, визначення їхньої важливості та побудову взаємозв'язків між ними. Очікується, що навчена нейронна мережа зможе застосовувати набутий досвід для розпізнавання невідомих образів завдяки своїм узагальнювальним здібностям. Наразі, згідно з аналізом літератури, згорткова нейронна мережа (CNN) демонструє найкращі результати у завданні ідентифікації людини (рис 1.3).

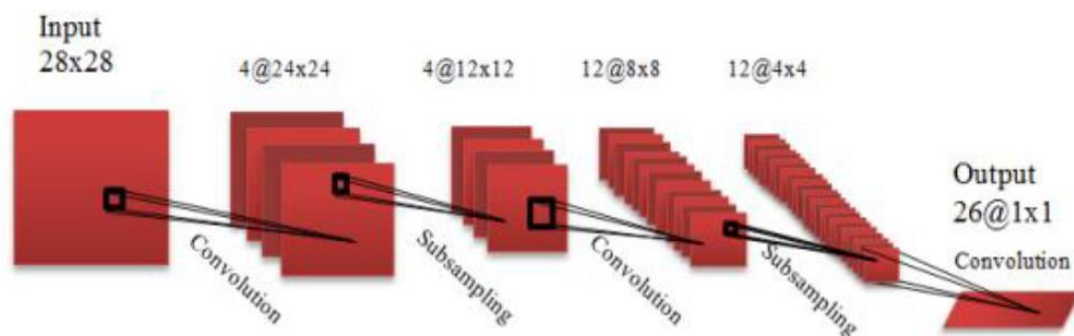


Рисунок 1.3 – Архітектура згорткової нейронної мережі

Згорткову нейронну мережу протестували на базі даних ORL. Ця база даних містить в собі зображення людей з різними змінами, такими як зміни масштабу, освітлення, повороти тощо. За результатами тестування, точність розпізнавання цією мережею сягає 96%.

На відміну від класичних методів, CNN автоматично навчаються виділяти найбільш релевантні ознаки (контури, текстури, кути) без ручного програмування. Це дозволяє мережі самостійно ідентифікувати важливі патерни, ігноруючи другорядні шуми, спричинені змінами. Завдяки використанню шарів згортки та підвибірки (пулінгу), CNN досягають трансляційної інваріантності. Це означає, що мережа здатна розпізнати обличчя, навіть якщо воно трохи зміщено в кадрі або змінений його розмір (масштаб). Глибокі архітектури, треновані на великих наборах даних, можуть успішно узагальнювати та компенсувати вплив різного освітлення, тіней, міміки та кута повороту обличчя (ракурсу). У завданнях верифікації та ідентифікації людей CNN регулярно демонструють найвищу точність порівняно з усіма іншими методами, що критично важливо при роботі з деформованими зображеннями.

Незважаючи на високу ефективність, використання CNN для розпізнавання зі змінами має певні обмеження. Для навчання глибоких CNN, стійких до всіх можливих змін (різні зачіски, вік, окуляри, борода, погане освітлення), необхідні величезні набори даних. Якщо навчальна вибірка не містить різноманітних варіацій, ефективність на реальних, деформованих зображеннях буде низькою. Глибокі CNN вимагають значних обчислювальних ресурсів (потужних GPU) як для навчання, так і для виведення (інференсу). Це ускладнює їхнє розгортання на пристроях з обмеженими ресурсами. CNN можуть бути обмануті мінімальними, ледь помітними для людського ока змінами у зображенні (шумом), що може призвести до неправильного розпізнавання навіть у ідеальних умовах. Складно точно визначити, які саме ознаки використовує мережа для розпізнавання обличчя (наприклад, чи фокусується вона на очах, чи на загальному контурі). Це ускладнює діагностику та покращення її роботи у специфічних, складних випадках.

1.4 Метод прихованих Марковських моделей

Приховані марковські моделі (ПММ) – це найпопулярніший алгоритм розпізнавання образів. Приховані марковські моделі – це математичні

представлення стохастичного процесу, який створює серію спостережень на основі попередньо збережених даних. Статистичний підхід у ПММ має багато переваг, включаючи надійну математичну основу, потужні методи навчання та декодування, ефективні можливості обробки послідовностей та гнучку топологію для синтаксису та статистичної фонології. Недоліки пов'язані з поганою розрізняльністю моделей та ірраціональними припущеннями, необхідними для побудови теорії ПММ, зокрема, незалежністю наступних фреймів ознак (тобто вхідних векторів) та марковським процесом першого порядку. Розроблені алгоритми у статистичній структурі на основі ПММ є надійними та ефективними в реальних сценаріях. Крім того, приховані марковські моделі часто використовуються в реальних додатках для реалізації систем розпізнавання та розуміння жестів.

ПММ є розширенням звичайних Марковських ланцюгів і базуються на концепції прихованих станів та спостережуваних подій. Приховані стани – це внутрішні стани системи, які не можна спостерігати безпосередньо (наприклад, фонема, яку вимовляє людина, або справжній стан людини – хвора чи здорова). Спостережувані події – це дані, які ми фіксуємо (наприклад, аудіосигнал слова, або результат медичного тесту) (рис.1.4).

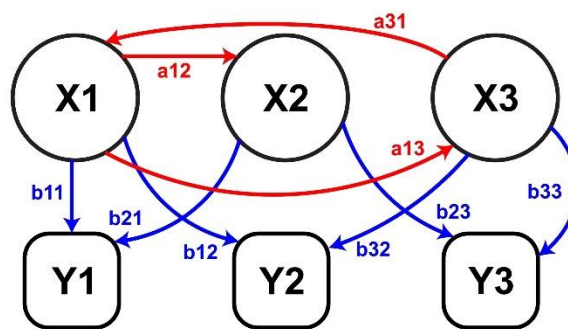


Рисунок 1.4 – Приховані Марковські моделі

Модель описується трьома основними наборами ймовірностей:

- ймовірність переходу з одного прихованого стану в інший;
- ймовірності спостереження: ймовірність того, що в даному прихованому стані буде спостерігатися певна подія;

– початкові ймовірності: ймовірність того, що процес почнеться з певного прихованого стану.

ПММ не є ідеальним вибором для сучасного розпізнавання статичних зображень (як-от складні сцени або 2D-обличчя), оскільки вони погано враховують просторові кореляції між пікселями. У більшості сучасних завдань, пов'язаних із зображеннями, згорткові нейронні мережі (CNN) значно перевершують ПММ за точністю та стійкістю до деформацій.

РОЗДІЛ 2

РОЗПІЗНАВАННЯ ОБЛИЧЧЯ МЕТОДОМ ВІОЛИ-ДЖОНСА

2.1 Фактори, що впливають на якість розпізнавання облич

На якість та точність розпізнавання облич впливає ціла низка взаємопов'язаних зовнішніх та внутрішніх факторів, які стосуються як самого зображення, так і особливостей алгоритмічної системи.

Насамперед, вирішальну роль відіграють умови зйомки, т.б. освітлення та ракурс. Недостатнє, надмірне або нерівномірне освітлення, а також сильні тіні чи контрове світло значно ускладнюють виділення та аналіз рис обличчя, тоді як обличчя, повернуте вбік, схилене або зняте під нетиповим кутом (у «профіль»), часто знижує точність розпізнавання.

Важливими є якість і стан зображення, включаючи його роздільну здатність та наявність шумів; зображення з низькою деталізацією або великою кількістю цифрових артефактів є складними для аналізу.

Значний вплив мають і властивості самого обличчя на момент зйомки. Міміка та вираз обличчя (наприклад, широка посмішка чи зведені брови) можуть тимчасово спотворити ключові риси, ускладнюючи порівняння з еталонною базою. Довготривалі зміни, такі як старіння, зміни зачіски, наявність окулярів, макіяжу або рослинності на обличчі (борода, вуса), також можуть суттєво впливати на результат розпізнавання.

Нарешті, якість самого алгоритму розпізнавання та якість навчальних даних є фундаментальними. Слабкі або застарілі алгоритми, а також бази даних, що не враховують різноманітність демографічних груп, освітлення чи ракурсів, можуть призводити до систематичних помилок або упереджень системи, особливо щодо певних груп людей.

Програма FERET (Face Recognition Technology), ініційована DARPA та дослідницькою лабораторією армії США, слугує ключовим інструментом для об'єктивної оцінки ефективності алгоритмів розпізнавання облич. Щорічно FERET публікує результати порівняльних тестів, які наразі зосереджені виключно на комерційних системах, без розкриття інформації про їхні внутрішні

принципи роботи. На цей час комерційним лідером у цих звітах є система NeoFace, розроблена компанією NEC.

До створення бази даних FERET велика кількість робіт повідомляла про видатні результати розпізнавання (зазвичай >95% правильного розпізнавання) на базах даних обмеженого розміру (зазвичай <50 осіб). Лише деякі з цих алгоритмів повідомляли про результати на зображеннях, що використовували спільну базу даних, не кажучи вже про досягнення бажаної мети – оцінки за стандартним протоколом тестування, який включав окремі навчальні та тестові набори. Як наслідок, не існувало методу для проведення обґрунтованих порівнянь між різними алгоритмами.

База даних FERET дозволила дослідникам розробляти алгоритми на спільній базі даних та повідомляти про результати в літературі, використовуючи цю базу даних. Результати, представлені в літературі, не забезпечували прямого порівняння між алгоритмами, оскільки кожен дослідник повідомляв про результати, використовуючи різні припущення, методи оцінювання та зображення. Незалежно проведені оцінки FERET дозволили провести пряму кількісну оцінку відносних сильних та слабких сторін різних підходів.

Крім того, розробники оцінок застосували два алгоритми розпізнавання облич (РСА та кореляцію) для забезпечення базової продуктивності.

2.2 Аналіз системи розпізнавання обличчя у відеопотоках

Як зазначалося раніше, обробка кадрів відеопотоку розробленою системою повинна включати два основних етапи. Перший етап – розпізнавання обличчя за методом Віюлі-Джонса. Другий етап – розпізнавання знайдених облич за допомогою гістограм локальних бінарних шаблонів і методу найближчого сусіда. Однак продуктивність цих алгоритмів суттєво залежить від таких факторів, як освітлення, положення людини тощо. Тому доцільно відразу описати умови використання розробленої системи, за яких може бути забезпечена її коректна робота:

– Допускається лише монотонна зміна освітлення. Дослідницький і тестовий набір потрібно знімати в однакових умовах освітлення.

– Використовується фронтальне положення або положення біля обличчя. Нейтральні вирази обличчя на зображеннях.

– Обличчя не накладаються на інші об'єкти.

Також бажано відразу описати необхідний розроблений функціонал:

– Обробка відеопотоку з камери, підключеної до комп'ютера, в реальному часі.

– Можливість налаштування параметрів роботи, які використовуються для ідентифікації та розпізнавання алгоритмів.

– Відображення інформації про розпізнане обличчя, включаючи міру приналежності до певного класу, графічне відображення гістограми та представлення LBP відстежуваного обличчя.

– Можливість навчання та додавання класів за допомогою камери через інтерфейс програми.

Для підвищення точності розпізнавання облич, крім базових етапів виявлення та ідентифікації, доцільно вводити проміжні кроки обробки.

Спочатку, застосування фільтра Гауса до знайдених облич допомагає ефективно знизити вплив шуму, що підвищує якість подальшого розпізнавання.

Наступним важливим кроком є робота з ознаками: до облич, трансформованих за допомогою оператора LBP (Local Binary Pattern), необхідно застосувати маску значущих областей. Цей прийом дозволяє усунути вплив фону, який часто потрапляє у кутові частини локалізованого зображення обличчя, фокусуючи систему лише на релевантних рисах.

Отже, загальний алгоритм розпізнавання включає послідовне виконання таких кроків:

– Виявлення обличчя у кадрі.

– Обробка знайдених облич фільтром Гауса.

– LBP-перетворення облич із подальшим накладанням маски значущих ділянок.

- Розрахунок гістограм LBP-ознак.
- Класифікація облич за гістограмами із застосуванням методу найближчого сусіда.

Кінцевим результатом цього процесу є формування списку відстежуваних облич, що містить їхні ідентифікаційні характеристики та точні координати їхніх прямокутних областей у кадрі.

Узагальнену блок-схему алгоритму обробки кадру розробленою системою наведено на (рис. 2.1).



Рисунок 2.1 – Узагальнена блок-схема алгоритму обробки кадрів відеопотоку

Подібним чином проводиться процедура навчання. Виявлене в кадрі обличчя послідовно обробляється згідно з описаним алгоритмом, розраховані гістограми облич навчальної вибірки кожного класу зберігаються. Розпізнавання ґрунтується на знаходженні мінімальної відстані між гістограмою вхідного зображення обличчя та гістограмами, які зберігаються в базі даних.

2.3 Метод Віоли-Джонса

Метод Віоли-Джонса, розроблений у 2001 році, є надзвичайно ефективним і широко застосовується на практиці, зокрема у системах відеоспостереження, завдяки своїй здатності швидко знаходити об'єкти в реальному часі. В основі цього алгоритму лежать три ключові компоненти: використання інтегрального подання зображення для обчислення ознак Хаара, створення класифікатора за допомогою алгоритму адаптивного бустингу (це процедура послідовної побудови композиції алгоритмів машинного навчання, коли кожен наступний алгоритм намагається компенсувати недоліки композиції всіх попередніх алгоритмів) та об'єднання класифікаторів у єдину каскадну структуру (базується на побудові послідовності класифікаторів, яка називається каскадом: кожен наступний прагне врахувати помилки попереднього). Цей метод вирізняється дуже низькою ймовірністю помилкового визначення обличчя та високою точністю, яка може сягати 90%, навіть коли обличчя спостерігається під кутом до 30 градусів. Цей алгоритм доступний у бібліотеці OpenCV.

Виділимо основні принципи, на яких базується робота цього методу:

- Цілісне представлення зображень.
- Пошук обличчя за допомогою знаків Хаара.
- Каскадна класифікація на основі використання бустингу.

Для розрахунку яскравості прямокутної ділянки зображення використовується інтегральне представлення [18]. Це представлення часто використовується в багатьох розроблених алгоритмах комп'ютерного зору.

Інтегральне представлення дозволяє швидко розрахувати загальну яскравість довільного прямокутника на заданому зображенні, причому час обчислення залежить від площі прямокутника.

Інтегральним представленням зображення є матриця, розміри якої збігаються з розмірами вхідного зображення. У кожному елементі такої матриці зберігається сума інтенсивностей таких пікселів, які розташовані ліворуч і вище цього елемента. Елементи матриці розраховуються за такою формулою:

$$I(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y')$$

де $I(x, y)$ – значення точки (x, y) інтегрального зображення; $i(x, y)$ – значення інтенсивності вхідного зображення.

Використання інтегрального представлення зображення дає можливість обчислювати однотипні ознаки, але з різними геометричними параметрами, за один і той же час, оскільки розрахунок матриці інтегрального представлення займає лінійний час, пропорційний кількості пікселів у зображення.

Просту прямокутну хаароподібну ознаку можливо визначити як різницю сум пікселів областей усередині прямокутника, який може бути в будь-якому місці та масштабі на первинному зображенні. Такий варійований набір ознак називають 2-прямокутковою ознакою (англ. 2-rectangle feature). Віола та Джонс також визначили 3-прямокутні та 4-прямокутні ознаки. Ці значення вказують на певні характеристики певної області зображення. Кожен тип ознак може вказувати на наявність (або відсутність) певних характеристик на зображенні, таких як контури або зміни текстури. Наприклад, 2-прямокуткова ознака може вказувати, де проходить межа між темною та світлою областями (рис. 2.2, а).

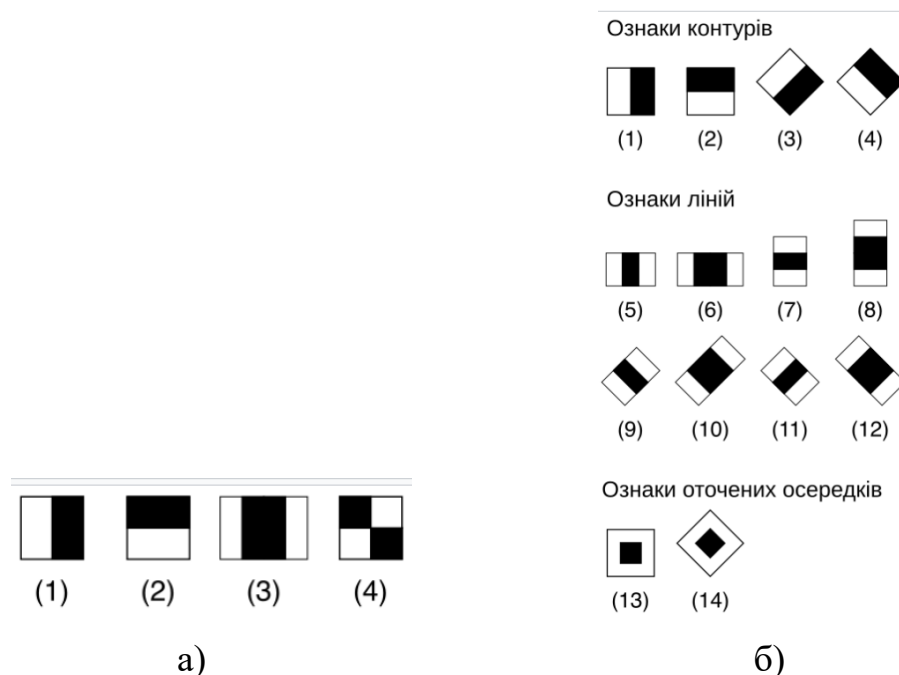


Рисунок 2.2 – Ознаки та додаткові ознаки Хаара

Лієнгарт і Майдт[4] запровадили поняття нахиленої (на 45°) хаароподібної ознаки. Її було використано для збільшення розмірності набору ознак у спробі покращити виявлення об'єктів на зображеннях. Це вдалося, оскільки деякі з цих ознак здатні описувати об'єкт краще. Наприклад, 2-прямокутнікова нахилена хаарова ознака може вказувати на присутність контуру під кутом 45° (рис.2.2, б).

Результатом обчислення такої ознаки на інтегральному представленні буде:

$$F=U-V$$

де U — сума значень яскравості точок, охоплених світлою частиною об'єкта, а V — сума значень яскравості точок, охоплених темною частиною об'єкта. Такі особливості описують різницю в яскравості вздовж двох осей зображення.

Пошук облич здійснюється за допомогою так званого вікна сканування, розмір якого в оригінальному алгоритмі становить $24*24$ пікселя.

Вікно переміщується по зображенню з кроком 1 піксель, і для кожної його позиції розраховуються знаки Хаара з різним масштабом і положенням у вікні, що відкривається. При цьому саме сканування виконується однаково для різних масштабів вікна сканування. Знайдені ознаки передаються в класифікатор, який за їх значенням визначає, чи є область зображення, пов'язана з вікном, обличчям чи ні.

Каскадна класифікація – це структурний підхід, що значно прискорює процес розпізнавання облич. Суть методу полягає в організації послідовності (каскаду) слабких класифікаторів, навчених за допомогою процедури посилення (наприклад, AdaBoost). Така ієрархічна структура дозволяє на ранніх етапах і з мінімальними обчислювальними витратами швидко відфільтрувати ті ділянки зображення, які з високою ймовірністю не містять цільового об'єкта (обличчя), фокусуючи складнішу обробку лише на найбільш перспективних областях. Приклад каскадної структури класифікаторів наведено на (рис. 2.3).

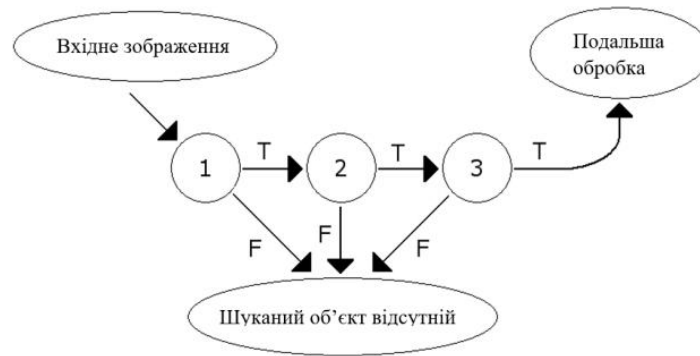


Рисунок 2.3 – Каскадний класифікатор

Кожен послідовний рівень каскадної структури навчається за допомогою алгоритму AdaBoost. Процес навчання передбачає поступове збільшення кількості використовуваних ознак, доки не буде досягнуто необхідного рівня точності виявлення цільового об'єкта при одночасному дотриманні заданого порогу помилки першого роду (хибнопозитивних спрацьовувань).

Рівні каскаду додаються послідовно: якщо сукупна помилка першого роду для всієї структури ще не досягнута, до каскаду вводиться додатковий шар. Навчальний негативний набір для цього нового шару формується з усіх помилкових спрацьовувань, зібраних під час роботи поточної версії каскаду.

У результаті цієї каскадної класифікації система видає набір областей зображення, що містять об'єкт. На наступному етапі відбувається видалення вкладених повторів, які виникають через масштабування вікна сканування, забезпечуючи, щоб один об'єкт був виявлений лише один раз. Для подальшої високоточної обробки всі знайдені обличчя уніфікуються: вони перетворюються в градації сірого та масштабуються до стандартного розміру 128*128 пікселів.

Фільтр Гауса. Фільтр Гауса використовується для усунення шуму на зображеннях обличчя. Фільтр Гауса — це фільтр розмиття зображення, який використовує нормальний розподіл для обчислення перетворення, яке застосовується до кожного пікселя зображення. Нормальний розподіл для двох вимірів описується формулою [21]:

$$G(u, v) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/(2\sigma^2)}$$

де r - радіус розмиття; $r = x^2 + y^2$, σ — стандартне відхилення розподілу Гаусса.

Ця формула визначає поверхню, яка виглядає як зосереджені кола з нормальним розподілом від центральної точки. Пікселі, де розподіл відмінний від нуля, використовуються для побудови матриці згортки, застосованої до вхідного зображення. Значення пікселя має найбільшу вагу (має найвище значення Гауса), а сусідні пікселі мають меншу вагу залежно від їх відстані.

Гаусова фільтрація є важливим інструментом в області обробки зображень, особливо для зменшення шуму. Застосовуючи гаусове ядро, фільтр надає центральним пікселям більшої ваги, ніж навколишнім областям, ефективно зменшуючи шум і зберігаючи структуру зображення. Важливий параметр σ контролює область застосування фільтра і подальше згладжування. Гаусові фільтри чудово видаляють випадкові, малопомітні шуми на зображенні, що робить їх життєво важливими у багатьох програмах обробки зображень.

Можна побачити приклад того, як працює розмиття за Гаусом для одновимірного масиву (рис. 2.4).



а)

б)

в)

Рисунок 2.4 – Застосування фільтра Гауса на зображенні. а) початкове зображення, б) ядро розміром 10x10, в) ядро розміром 20x20

Трансформація LBP (Local Binary Patterns, локальні бінарні шаблони) – це метод у комп'ютерному зорі для опису та класифікації текстур, який перетворює зображення на бінарну карту, де кожен піксель кодується як двійковий код, що відображає співвідношення яскравості його сусідів до центру. Цей процес дозволяє виділити унікальні характеристики текстури для подальшого розпізнавання об'єктів, облич, або аналізу зображень, що стало базою для

багатьох покращених алгоритмів. Оператор LBP вперше був запропонований у 1996 році для класифікації текстур [13].

Вектор ознак LBP у найпростішому вигляді знаходиться так:

- Розділити вікно пошуку на комірки (наприклад, 16x16 пікселів на кожне комірка)
- Для кожного пікселя в комірці порівняти з кожним з 8-8 сусідів (зліва зверху, зверху по центру, зліва знизу, праворуч зверху і т. д.). Обходити пікселі по колу, тобто за годинниковою або проти годинникової.
- Якщо значення центрального пікселя більше за сусіднє, записати «0», інакше записати «1». В результаті виходить восьмирозрядне двійкове число (яке зазвичай конвертується в ціле десяткове для зручності).
- Обчислити гістограму в комірці за частотою появи кожного «числа» (тобто всі комбінації пікселів, які більші і менші від центрального). Цю гістограму можна подати як 256-розмірний вектор ознак .
- Опціонально нормалізувати гістограму.
- Конкатенувати (нормалізовані) гістограми всіх осередків. Це дасть вектор ознак всього вікна.

Отриманий вектор ознак тепер може бути оброблений за допомогою Support vector machine, або іншими алгоритмами машинного навчання класифікації зображень. Подібні класифікатори можуть бути використані для системи розпізнавання обличчя або текстурного аналізу.

Метод найближчого сусіда. Отримані гістограми LBP класифікуються за методом найближчого сусіда, як і в оригінальному дослідженні LBP стосовно проблеми розпізнавання обличчя [14]. Метод найближчого сусіда - це простий алгоритм класифікації, суть якого полягає в тому, що об'єкт належить до класу, до елемента якого він найближче (рис. 2.5).

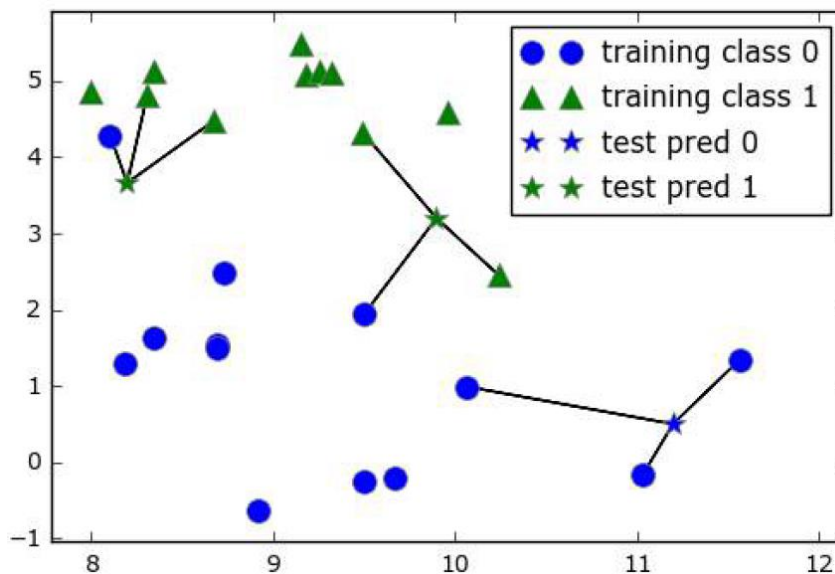


Рисунок 2.5 – Прогнози, отримані для набору даних *forge* за допомогою моделі трьох найближчих сусідів

Цей метод класифікації застосовується переважно у випадках, коли ціна неправильної класифікації є надзвичайно високою, а якість і чистота вихідних даних (відсутність суттєвих помилок) підтримується на належному рівні. Незважаючи на свою високу ефективність у широкому спектрі класифікаційних завдань, основним його недоліком є значна чутливість до аномальних або помилкових значень окремих точок даних, що може суттєво спотворити результат[24].

Окрема увага приділяється вибору метрики, яка безпосередньо визначає спосіб розрахунку відстані між гістограмами ознак. Для досягнення максимальної точності класифікації критично важливо підібрати метрику, що найбільш адекватно відобразить відмінності між гістограмами, які належать до різних класів. Наприклад, в оригінальних дослідженнях для цього часто застосовується відстань χ^2 (*хі-квадрат*), яка розраховується за такою формулою [14]:

$$D(H_1, H_2) = \sum_i \frac{(H_1(i) - H_2(i))^2}{H_1(i) + H_2(i)}$$

РОЗДІЛ 3

ПРАКТИЧНА РЕАЛІЗАЦІЯ

3.1 Інструментарій розробки

Дослідження показали, що центрально-симетричний оператор LBP (CS-LBP) може дещо поступатися іншим локальним бінарним шаблонам, якщо зображення розбивається на малу кількість блоків. Однак, при використанні великої кількості субдоменів, його показник точності класифікації відстає від конкурентів у середньому не більше ніж на 3%. Примітно, що на деяких тестових наборах даних CS-LBP навіть перевершує інші шаблони.

З огляду на ці результати, центрально-симетричні локальні бінарні шаблони є цілком доцільними для використання у системі розпізнавання облич у відеопотоках. Їхня ключова перевага полягає у високій швидкості роботи при показниках точності, що майже не поступаються іншим модифікаціям LBP. Детальне порівняння швидкості системи при використанні різних варіацій LBP буде представлено у розділі тестування.

Для досягнення оптимального балансу між точністю розпізнавання та витратами пам'яті, було визначено, що найкращим розбиттям зображення для CS-LBP є сітка 4×4 . Саме це співвідношення буде використано в розробленій системі.

Система розроблена з використанням об'єктно-орієнтованої мови програмування C# в середовищі розробки Microsoft Visual Studio 2017 [20].

Наявність механізму «збірка сміття» в C# дозволить ефективно організувати роботу зі структурами списків, на яких базується система.

Щоб полегшити процес розробки, було вирішено використовувати бібліотеку OpenCV.

OpenCV є основною бібліотекою для роботи з візуальними даними, пропонуючи набір функцій для трьох ключових напрямків: обробки зображень, реконструкції/очищення та розпізнавання образів.

Бібліотека містить величезну кількість інструментів для маніпуляцій із зображеннями. Це включає базові операції, такі як зміна колірному простору

(наприклад, перетворення в градації сірого або HSV), геометричні перетворення (масштабування, обертання, афінні перетворення) та порогова обробка. OpenCV також забезпечує засоби для фільтрації шумів (наприклад, фільтри Гауса, медіанний фільтр) та виконання морфологічних операцій (ерозія, дилатація), які є критично важливими для попередньої підготовки даних.

Для очищення зображень від шуму та дефектів OpenCV пропонує спеціалізовані алгоритми, зокрема нелокальне засобування (Non-Local Means Denoising), що ефективно видаляє шум, зберігаючи деталі. У контексті реконструкції, бібліотека підтримує функції для: виправлення оптичних спотворень, спричинених об'єктивом; відновлення тривимірної інформації про об'єкти за допомогою пари камер (реконструкція глибини), об'єднання кількох зображень в одну панораму.

Це один з напрямків. OpenCV містить реалізації класичних і сучасних алгоритмів для:

- виявлення об'єктів (включаючи відомий метод Віоли-Джонса для швидкого виявлення облич, а також модулі для інтеграції з сучасними глибокими нейронними мережами);
- розпізнавання осіб (використання методів, таких як Eigenfaces, Fisherfaces та LBP);
- відстеження об'єктів (алгоритми для моніторингу руху об'єктів у відеопотоці, наприклад, Mean-Shift, трекери, що базуються на глибокому навчанні);
- аналіз ознак (виявлення та опис ключових точок на зображенні (SIFT, SURF, ORB, FAST), що є основою для порівняння та зіставлення зображень.

3.2 Загальний алгоритм обробки кадрів

Відповідно до загального алгоритму обробки кадрів відеопотоку можна скласти діаграму потоків даних розробленої системи. Ця діаграма зображена на (рис. 2.6).

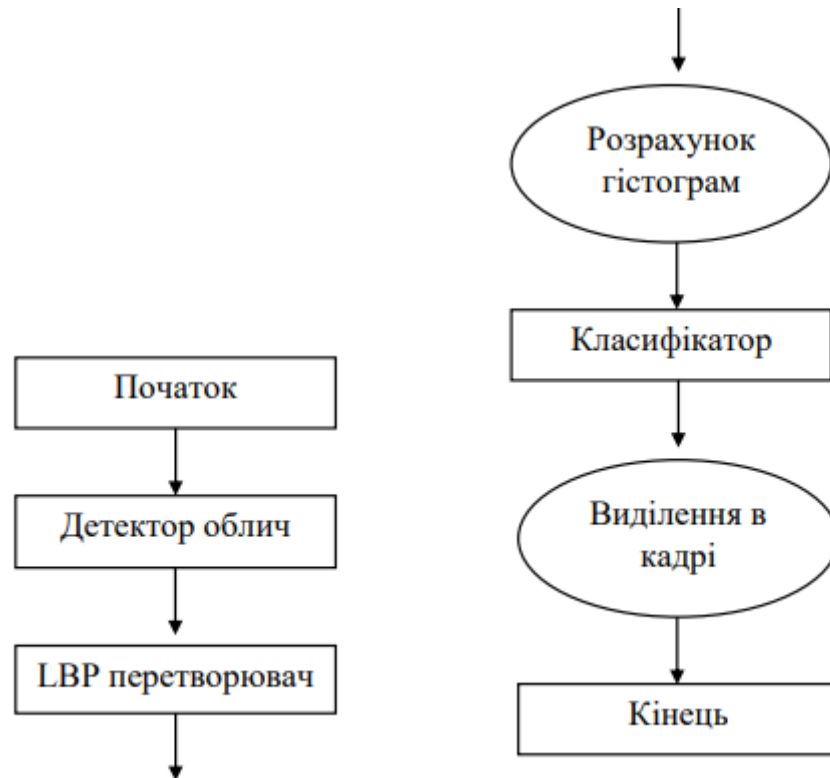


Рисунок 2.6 – Діаграма системних потоків даних

На основі діаграми потоку даних системи, доцільно структурувати програмне забезпечення за допомогою класів, які відповідають ключовим етапам перетворення даних. До таких сутностей належать:

- Детектор обличчя (Face Detector).
- Конвертер LBP (LBP Converter).

Класифікатор має бути реалізований як метод основного класу програми. Таке рішення забезпечує тісну взаємодію зі списками розпізнаних та нерозпізнаних облич, які зберігаються як атрибути цього головного класу. Крім того, необхідно створити окремі класи, що описують категорію обличчя та розпізнане обличчя, а також реалізувати функцію для обчислення гістограм.

Детектор обличчя повинен втілювати алгоритми, описані у відповідному розділі документа. Його функція полягає в отриманні кадрів відеопотоку та пошуку облич. У результаті роботи детектора формується список прямокутних областей, що відповідають положенню обличчя у кадрі.

Саму реалізацію каскадного класифікатора доцільно виконати за допомогою бібліотеки Emgu CV, оскільки вона вже містить готову реалізацію

цього алгоритму з набором добре навчених каскадів. Усі структури даних та облікові записи будуть описані з використанням бібліотеки колекцій мови C#.

Конвертер LBP повинен реалізовувати алгоритм перетворення зображення центральньо-симетричним оператором LBP. На вхід конвертера будуть подаватися зображення обличчя. Вихідними даними є перетворені зображення LBP.

Клас головної форми додатку міститиме елементи інтерфейсу програми, а також список розпізнаних і нерозпізнаних облич, якими система буде оперувати при обробці кадрів. Крім того, в цьому класі реалізовані функції обчислення гістограм і графічного відображення результатів роботи системи. Детальний опис елементів інтерфейсу та користувацьких механізмів взаємодії з ними буде описано в розділі розробки інтерфейсу системи.

Клас головної форми додатку слугує центральним ядром системи, поєднуючи елементи користувацького інтерфейсу з ключовою логікою обробки.

Цей клас не тільки містить усі елементи інтерфейсу, але й відповідає за зберігання та управління списками розпізнаних та нерозпізнаних облич, які використовуються системою під час послідовної обробки відеокадрів.

Крім того, у цьому ж класі реалізовано важливі допоміжні функції, зокрема алгоритми для обчислення гістограм та механізми для графічного відображення результатів роботи системи на екрані.

3.3 Реалізація основних класів

Emgu CV для каскадної класифікації (детектор облич) та реалізує окремий клас для CS-LBP перетворення. Структуру класів:

- Основний клас (з класифікатором).
- Детектор Обличчя.
- Конвертер LBP.

Ця структура забезпечує чітке розділення відповідальності між класами, використовуючи Emgu CV для ресурсомісткої роботи з каскадами та власну реалізацію для специфічного CS-LBP перетворення.

Клас: LBPConverter (Конвертер LBP). Цей клас відповідає за реалізацію алгоритму перетворення зображення, використовуючи центрально-симетричний локальний бінарний шаблон (CS-LBP) (лістинг 3.1).

Лістинг 3.1 – Клас: LBPConverter

```

using Emgu.CV;
using Emgu.CV.Structure;

/// <summary>
/// Клас для перетворення зображення обличчя в LBP-простір за допомогою центрально-симетричного оператора.
/// </summary>
public class LBPConverter
{
    // Припустимо, що оператор CS-LBP має радіус 1 і 8 сусідів.
    private const int Radius = 1;
    private const int Neighbors = 8;
    /// <summary>
    /// Конвертує вхідне зображення обличчя в зображення LBP.
    /// </summary>
    /// <param name="faceImage">Вхідне зображення обличчя (градації сірого, 128x128).</param>
    /// <returns>Перетворене LBP зображення (насправді, це матриця LBP-значень).</returns>
    public Image<Gray, byte> ConvertToCSLBP(Image<Gray, byte> faceImage)
    {
        return faceImage.Copy();
    }
    /// <summary>
    /// Обчислює гістограму LBP з перетвореного зображення з розбиттям 4x4.
    /// </summary>
    /// <param name="lbpImage">LBP зображення.</param>
    /// <returns>Масив, що представляє гістограму LBP ознак.</returns>
    public float[] ComputeLBPHistogram(Image<Gray, byte> lbpImage)
    {
        return faceImage.Copy();
    }
    /// <summary>
    /// Обчислює гістограму LBP з перетвореного зображення з розбиттям 4x4.
    /// </summary>
    /// <param name="lbpImage">LBP зображення.</param>
    /// <returns>Масив, що представляє гістограму LBP ознак.</returns>
    public float[] ComputeLBPHistogram(Image<Gray, byte> lbpImage)
    {
        // Реалізація розбиття 4x4 та обчислення гістограм
        // (логіка розрахунку гістограм та їх нормалізації)
        return new float[256 * 16]; // Приклад розміру гістограми: 256 бінів * 16 блоків (4x4)
    }
}

```

Кінець лістингу 3.1

Клас: FaceDetector (детектор обличчя). Цей клас використовує вбудовану в Emgu CV реалізацію каскадного класифікатора (Віюли-Джонса) для виявлення облич (лістинг 3.2).

Лістинг 3.2 – Клас: FaceDetector

```

using Emgu.CV;
using System.Drawing; // Для Rectangle

/// <summary>
/// Клас для виявлення облич у кадрі за допомогою каскадного класифікатора (Emgu CV).
/// </summary>
public class FaceDetector
{
    private CascadeClassifier _cascade;
    private const string HAAR_CASCADE_PATH = "haarcascade_frontalface_alt.xml"; // Шлях до файлу каскаду

    public FaceDetector()
    {
        // Ініціалізація каскадного класифікатора (Emgu CV)
        _cascade = new CascadeClassifier(HAAR_CASCADE_PATH);
    }

    /// <summary>
    /// Знаходить обличчя в кадрі відеопотоку.
    /// </summary>
    /// <param name="frame">Вхідний кадр.</param>
    /// <returns>Масив прямокутних областей, де виявлено обличчя.</returns>
    public Rectangle[] DetectFaces(IInputArray frame)
    {
        // Використання вбудованого методу Emgu CV для детекції
        return _cascade.DetectMultiScale(
            frame,
            1.1, // Масштабний коефіцієнт
            10, // Мінімальна кількість сусідів
            new Size(20, 20) // Мінімальний розмір об'єкта
        );
    }
}

```

Кінець лістингу 3.2

Клас: MainFormLogic. Головний клас, що містить логіку класифікації та управляє списками облич. Класифікатор реалізовано як метод цього класу. Частина коду наведено в додатку А. Головний метод обробки кадру та класифікації (реалізація класифікатора) з переліком основних операцій наведено у лістингу 3.3. Представлена логічна схема обробки, без деталізації методів.

Необхідно забезпечити такий перелік функціоналу:

1. Виявлення обличчя в кадрі.
2. LBP-перетворення.
3. Обчислення гістограм.
4. Класифікація (Метод найближчого сусіда).
5. Оновлення списків (графічне відображення результатів).
6. Реалізація функції графічного відображення результатів.

Лістинг 3.3 – Клас: MainFormLogic

```

public void ProcessFrameAndClassify(Mat currentFrame)
{
    // 1. Виявлення обличчя в кадрі
    Rectangle[] detectedRegions = _detector.DetectFaces(currentFrame);

    foreach (var region in detectedRegions)
    {
        // Вирізання, конвертація у сірий та масштабування обличчя (наприклад, до 128x128)
        using (var faceMat = new Mat(currentFrame, region))
        using (var faceGray = faceMat.ToImage<Gray, byte>().Resize(128, 128, Emgu.CV.CvEnum.Inter.Cubic))
        {
            // 2. LBP-перетворення
            var lbpImage = _lbpConverter.ConvertToCSLBP(faceGray);

            // 3. Обчислення гістограми
            float[] histogram = _lbpConverter.ComputeLBPHistogram(lbpImage);

            // 4. Класифікація (Метод найближчого сусіда)
            string identifiedLabel = ClassifyFace(histogram);

            // 5. Оновлення списків (графічне відображення результатів)
            if (identifiedLabel != "Unknown")
            {
                // Додати до _recognizedFaces
            }
            else
            {
                // Додати до _unrecognizedFaces
            }

            // 6. Реалізація функції графічного відображення результатів
            // ... (тут логіка малювання рамок та міток на currentFrame)
        }
    }
}

```

Кінець лістингу 3.3

При розробці інтерфейсу програми доцільно обмежитися одним головним вікном, яке функціонально розділене на кілька ключових областей.

Це єдине вікно одночасно виконуватиме роль центру управління та відображення даних. Воно міститиме область для виведення обробленого відеопотоку з камери, надаючи користувачеві візуальний контроль над процесом розпізнавання. Поруч розташовуватиметься область для налаштування параметрів роботи програми, де можна буде змінювати робочі установки системи. Окремо будуть виділені область для роботи з класами (можливо, для управління базою даних осіб) та область для відображення детальної інформації про конкретну особу, яку обрав користувач зі списку розпізнаних облич. Загальний вигляд цієї форми (рис. 2.7).

Ключовим елементом інтерфейсу є область виведення відеопотоку, яка слугує для відображення оброблених кадрів у реальному часі.

Ця область також інтегрує всі необхідні елементи керування джерелом відео. Зокрема, тут розміщені кнопки для встановлення джерела відеопотоку чи то підключена до комп'ютера веб-камера, чи відеофайл. Для вибору файлу

користувач натискає кнопку «Відкрити файл», після чого з'являється діалогове вікно вибору; після підтвердження, шлях до обраного відеофайлу відображається на формі.

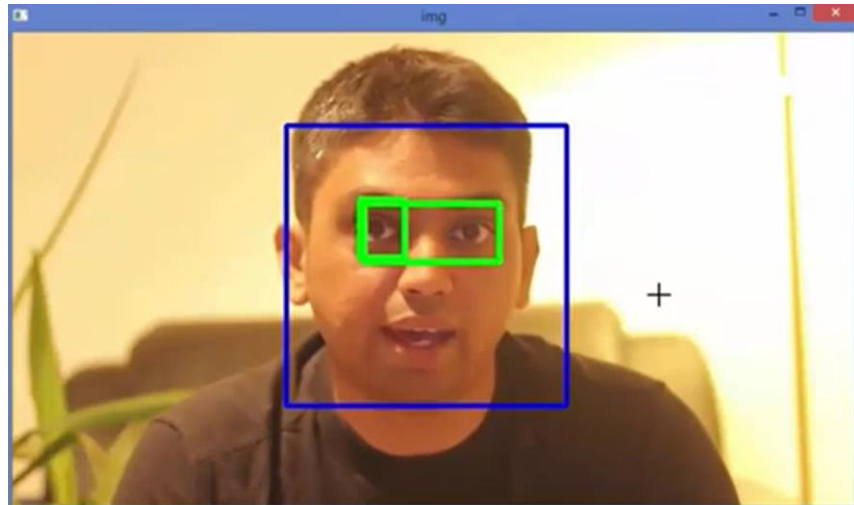


Рисунок 2.7 – Інтерфейс додатку

Крім того, ця зона містить універсальну кнопку, яка керує каналом захоплення та дозволяє запуснути або зупинити обробку і відображення кадрів з вибраного джерела. Запуск та зупинка обробки кадрів здійснюється єдиною кнопкою Пуск/Стоп. Область виведення відеопотоку зображена на (рис. 2.8).

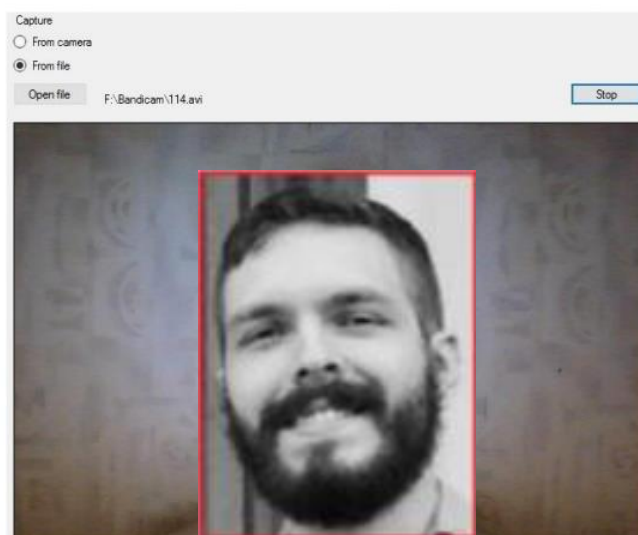


Рисунок 2.8 – Область виводу відеопотоку

Область відображення інформації про вибране користувачем обличчя призначена для відображення даних розпізнаних програмою облич. Джерелом

даних для відображення є клас обличчя, вибраний користувачем Face Classes, і розпізнане обличчя, що відповідає цьому класу.

У цій області форми відображаються зображення вибраного класу, прямокутна область поточного кадру, що відповідає обличчю цього класу, а також LBP-перетворення зображення обличчя. Крім того, поточний стан відображається між гістограмою розпізнаного обличчя та гістограмою його класу та графічним представленням гістограми LBP зображення обличчя (рис. 2.9).

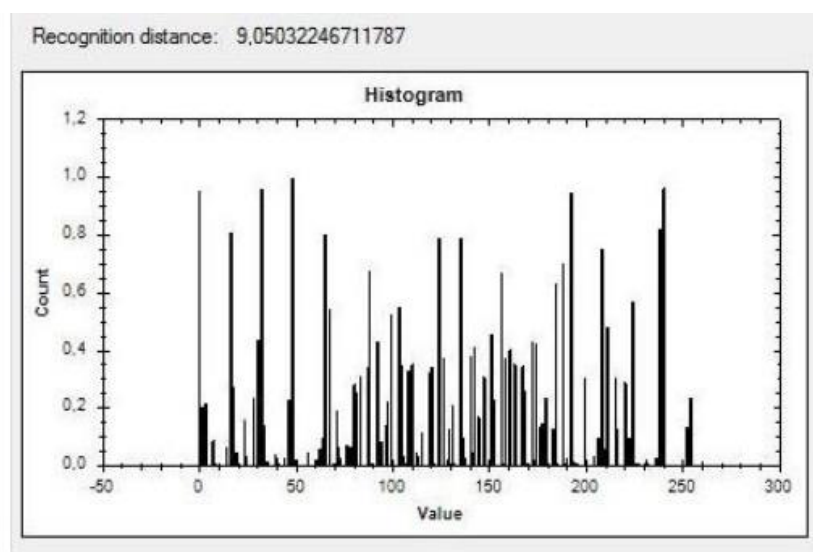


Рисунок 2.9 – Область виводу інформації про обране обличчя

3.4 Тестування розробленої системи

У цьому розділі проведено тестування розробленого класифікатора обличчя, а також проведемо порівняльне дослідження швидкодії системи при використанні різних варіацій оператора LBP.

Тестування класифікатора обличчя проводилося в умовах, які точно відповідають вимогам, описаним у розділі аналізу проблеми розпізнавання. Це означає, що відеопотік записувався за тих самих умов освітлення, що й еталонні зображення навчальної вибірки. Щоб забезпечити високий відсоток правильних класифікацій при русі, під час зйомки тестових даних були використані різні положення та ракурси обличчя в кадрі.

Всього під час тестування в програму було завантажено 20 класів облич. Для формування кожного класу було використано 10 зображень. З цих 20 класів 15 облич були створені за допомогою зображень людей з Інтернету. Ці класи граней були необхідні для того, щоб оцінити якість класифікатора на великому масиві даних, а також для тестування програми при читанні відеопотоку з файлу.

Результати тестування показали, що точність класифікатора при обробці відеопотоку становить більше 90% правильно розпізнаних кадрів.

Однак слід зазначити, що розроблена система чутлива до сильних немонотонних змін освітленості, а також до змін положення та нахилу розпізнаних облич, які не були враховані при захопленні зображень для формування класів облич. Якщо необхідні умови для коректної роботи програми не виконуються, точність класифікації знижується.

Вибір гістограм центрально-симетричних локальних бінарних шаблонів (CS-LBP) як ознак для класифікації був обґрунтований їхньою високою швидкістю роботи та економією пам'яті, досягнутою при незначних втратах у точності розпізнавання порівняно з іншими типами LBP-операторів. Порівняння їхньої якості в класифікації зображень було детально розглянуто в розділі 3 цієї роботи.

Однак, проведене дослідження досі не розкрило переваг CS-LBP з точки зору швидкості. Щоб точно визначити переваги продуктивності та оцінити їхній вплив на швидкість обробки відеопотоку, було проведено додаткове порівняння швидкості роботи додатків при використанні різних операторів LBP.

ВИСНОВОК

Була розроблена система розпізнавання облич для відеопотоків, яка базується на комбінації двох ключових методів: методу Віоли-Джонса для виявлення облич у кадрах та методу найближчого сусіда для їхньої класифікації. Для представлення ознак обличчя використовувалися центрально-симетричні локальні бінарні шаблони (LBP).

Тестування системи, проведене під час обробки відеопотоку з веб-камери в реальному часі, продемонструвало її високу ефективність, досягнувши приблизно 93% коректного розпізнавання.

В ході роботи також було проведено дослідження різних модифікацій локальних бінарних шаблонів. Результати підтвердили, що гістограми центрально-симетричних LBP є високошвидкісною та ефективною функцією ознак для розпізнавання облич у реальному часі, показуючи майже таку ж продуктивність, як й інші, більш складні варіанти LBP.

Розроблена система має пряме застосування у сфері відеоаналітики, зокрема в системах контролю доступу та ідентифікації особистості.

Важливою перевагою архітектури додатку є її модульність, яка дозволяє легко замінювати окремі компоненти. Це відкриває можливості для подальшого вдосконалення системи на різних апаратних платформах. Основний напрямок майбутнього розвитку полягає в удосконаленні класифікатора облич для підвищення загальної точності та надійності.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Який принцип роботи систем розпізнавання обличчя? Перспективи технології в Україні. URL: http://kristall-systems.net.ua/ua/novosti/kak_rabotaet_raspoznavanje_lits_perspektivyi_tehnologii_v_ukraine/ (дата звернення: 30.05.2023).
2. Розпізнавання обличчя: від теорії до практики. Wiki ТНТУ. URL: <http://surl.li/hmeoo> (дата звернення: 30.05.2023).
3. Бембель, О. С. (2020). Застосування методу Віолі-Джонса для розпізнавання тривимірних об'єктів (Bachelor's thesis, КПІ ім. Ігоря Сікорського).
4. Повідайко, П. М., & Скороход, М. П. (2013). Дослідження методів розпізнавання емоцій людини в охоронних системах.
5. Sosnovskiy, V., & Khlevniy, A. (2019). Аналіз та дослідження основних методів розпізнавання обличчя. *Computer-integrated technologies: education, science, production*, (35), 192-197.
6. Тимошин, Ю. А., & Орленко, С. П. (2018). Алгоритм розпізнавання обличчя людей на базі згорткової нейронної мережі. Адаптивні системи автоматичного управління, 1(32), 166-173.
7. Ahonen, A. Hadid, M. Pietikainen. Face Description with Local Binary Patterns: Application to Face Recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 1996, №28(12), С. 2037-2041.
8. Golubyak, I. V., & Kosarevych, R. Ya. (2017). Face recognition methods. *Problems of information technologies*, (2), 158-164.
9. Guo G, Fu Y, Dyer CR, Huang TS. Image-based human age estimation by manifold learning and locally adjusted robust regression. *IEEE Trans Image Process*. 2008 Jul;17(7):1178-88. doi: 10.1109/TIP.2008.924280. PMID: 18586625.
10. Corinna Cortes, Vapnik Vladimir. Support-vector networks. *Machine learning* 20.3. 1995. P. 273-297.
11. OpenCV Face Recognition. OpenCV. URL: <https://opencv.org/opencv-face-recognition/> (date of access: 11.06.2023).

12. Abdi H., Williams L.J. Principal component analysis. Wiley Interdisciplinary Reviews: Computational Statistics, 2(4). 2010. P. 433-459.
13. Geurts Pierre, Damien Ernst, Louis Wehenkel. Extremely randomized trees. Machine learning, 63.1. 2006. P. 3-42.
14. Li S.Z., Jain A. Handbook of Face Recognition. Springer New York, 2011. P. 398.
15. Шаховська, Н. Б., & Басистюк, О. А. (2017). Розпізнавання обличчя за допомогою алгоритмів машинного навчання. Штучний інтелект.
16. Histograms of Oriented Gradients for Human Detection. URL: <https://lear.inrialpes.fr/people/triggs/pubs/Dalal-cvpr05.pdf> (date of access: 13.05.2023).
17. FaceNet: A Unified Embedding for Face Recognition and Clustering URL: https://www.cv-foundation.org/openaccess/content_cvpr_2015/app/1A_089.pdf (date of access: 13.05.2023)
18. OpenFace. URL: <https://cmusatyalab.github.io/openface/> (date of access: 13.05.2023).
19. Abeled Faces in the Wild. URL: <http://vis-www.cs.umass.edu/lfw/> (date of access: 28.05.2023).
20. Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. University of Massachusetts, Amherst, Technical Report 07-49, October, 2007.
21. Elastic Bunch Graph Matching - Scholarpedia. URL: http://www.scholarpedia.org/article/Elastic_Bunch_Graph_Matching (date of access: 09.05.2023).
22. Engineering Journal | DOI – Peer Reviewed Journal | Low cost Journal | High Impact Factor Journal | DOI Journal | UGC Care Journal | Referred Journal. URL: <https://ijtre.com/images/scripts/2015021129.pdf> (дата звернення: 09.05.2023).

ДОДАТКИ

Додаток А

C#

```
using System.Collections.Generic;

/// <summary>
/// Основний клас програми, що містить інтерфейс, списки облич та логіку
класифікації.
/// </summary>
public class MainFormLogic
{
    private List<RecognizedFace> _recognizedFaces;
    private List<UnrecognizedFace> _unrecognizedFaces;

    private FaceDetector _detector;
    private LBPCConverter _lbpConverter;

    public MainFormLogic()
    {
        _recognizedFaces = new List<RecognizedFace>();
        _unrecognizedFaces = new List<UnrecognizedFace>();
        _detector = new FaceDetector();
        _lbpConverter = new LBPCConverter();
    }

    /// <summary>
    /// Головний метод обробки кадру та класифікації (реалізація
класифікатора).
    /// </summary>
    /// <param name="currentFrame">Кадр відеопотоку.</param>
    public void ProcessFrameAndClassify(Mat currentFrame)
    {
        // 1. Виявлення обличчя в кадрі
        Rectangle[] detectedRegions = _detector.DetectFaces(currentFrame);

        foreach (var region in detectedRegions)
        {
            // Вирізання, конвертація у сірий та масштабування обличчя
            (наприклад, до 128x128)
            using (var faceMat = new Mat(currentFrame, region))
            using (var faceGray = faceMat.ToImage<Gray, byte>().Resize(128,
128, Emgu.CV.CvEnum.Inter.Cubic))
            {
                // 2. LBP-перетворення
                var lbpImage = _lbpConverter.ConvertToCSLBP(faceGray);

                // 3. Обчислення гістограми
                float[] histogram =
                _lbpConverter.ComputeLBPHistogram(lbpImage);

                // 4. Класифікація (Метод найближчого сусіда)
                string identifiedLabel = ClassifyFace(histogram);

                // 5. Оновлення списків (графічне відображення результатів)
                if (identifiedLabel != "Unknown")
                {
                    // Додати до _recognizedFaces
                }
                else
                {

```

```

        // Додати до _unrecognizedFaces
    }

    // 6. Реалізація функції графічного відображення результатів
    // ... (тут логіка малювання рамок та міток на currentFrame)
}
}

/// <summary>
/// Класифікатор, реалізований як метод основного класу (метод
найближчого сусіда).
/// </summary>
/// <param name="histogram">Гістограма LBP ознак.</param>
/// <returns>Мітка розпізнаного обличчя.</returns>
private string ClassifyFace(float[] histogram)
{
    // Тут розміщується логіка порівняння гістограми з базою даних
    // за допомогою метрики (наприклад, Хі-квадрат) та пошуку
найближчого сусіда.
    // ...
    return "Unknown"; // або ім'я особи
}
}

// Допоміжні класи (для обліку облич)
public class RecognizedFace { /* ... */ }
public class UnrecognizedFace { /* ... */ }

```