

**БАКАЛАВРСЬКА РОБОТА**

**БР. ІІІ - 13.00.00.000 ІІЗ**

**Група ІІІ-21-4**

**Сивак Роман**

**2025**

**Івано-Франківський національний технічний університет нафти і газу**

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

**Сивак Роман Олегович**

(прізвище, ім'я, по батькові)

УДК 004  
(індекс)

## **БАКАЛАВРСЬКА РОБОТА**

**Побудова рішення для оптимізації та налаштування методів**

**розпізнавання образів**

(назва роботи)

**Інженерія програмного забезпечення**

(назва освітньої програми)

**121 - Інженерія програмного забезпечення**

(шифр і назва спеціальності)

**Робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело**

Здобувач освітнього рівня Сивак Р.О.  
(підпис, ініціали та прізвище здобувача)

Науковий керівник Процюк Василь Романович, к.т.н., доцент  
(підпис, прізвище, ім'я, по батькові, науковий ступінь, вчене звання керівника)

Допущено до захисту  
Завідувач кафедри

доц. Бандура В.В.  
(посада) (підпис) (дата) (ініціали та прізвище)

**Івано-Франківськ – 2025**



## 6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 28 квітня 2025 р.

Керівник \_\_\_\_\_

(підпис)

Завдання прийняв до виконання \_\_\_\_\_

(підпис)

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту	Примітка
1	Аналіз предметної області машинного навчання та методів розпізнавання образів	06.05.2025	виконано
2	Огляд методів та алгоритмів розпізнавання обличчя	17.05.2025	виконано
3	Представлення методологій побудови рішення для налаштування методів розпізнавання	27.05.2025	виконано
4	Розробка алгоритмів та етапів обробки алгоритму LBRH для розпізнавання обличчя	01.06.2025	виконано
5	Імплементация рішення для оптимізації та налаштування методів розпізнавання образів	06.06.2025	виконано
6	Оформлення пояснювальної записки дипломної роботи завідувачем кафедри	10.06.2025	виконано

Студент – дипломник \_\_\_\_\_

(підпис)

Керівник роботи \_\_\_\_\_

(підпис)

## АНОТАЦІЯ

Бакалаврська робота містить 79 сторінок, 25 рисунків, список використаних джерел із 45 найменуваннями.

**Мета роботи** полягає в аналізі існуючих алгоритмів розпізнавання обличчя, розробці та імплементації оптимізованого варіанту методу LBPН, а також оцінці його ефективності на практичних прикладах із використанням бібліотеки OpenCV.

**Об'єкт дослідження** - процеси та технології автоматичного розпізнавання облич на основі методів машинного навчання.

**Предмет дослідження** - алгоритми та моделі машинного навчання, що застосовуються для розпізнавання облич, зокрема метод LBPН та його вдосконалення.

**В першому розділі** проведено ґрунтовний аналіз предметної області розпізнавання облич та методів машинного навчання, що дозволило визначити потенціал та обмеження існуючих підходів.

**В другому розділі** представлено детальний опис методів розпізнавання облич та їх характеристик, що стало підґрунтям для вибору методу LBPН як базового для вдосконалення

**В третьому розділі** реалізовано практичну систему розпізнавання облич з використанням OpenCV, а також проведено експериментальну перевірку та оптимізацію алгоритму LBPН, що підтвердило доцільність обраного підходу

**Висновок:** реалізовано прототип системи розпізнавання обличчя з використанням бібліотеки OpenCV у середовищі Visual Studio. Проведено серію експериментів для оцінки точності та продуктивності рішення.

**КЛЮЧОВІ СЛОВА:** РОЗПІЗНАВАННЯ ОБЛИЧЧЯ; МАШИННЕ НАВЧАННЯ; ЛОКАЛЬНІ БІНАРНІ ШАБЛони; КОМП'ЮТЕРНИЙ ЗІР; OPENCV; EIGENFACES; ГЛИБОКЕ НАВЧАННЯ

## ANNOTATION

The bachelor's thesis contains 79 pages, 25 figures, a list of used sources with 45 names.

**The purpose of the work** is to analyze existing face recognition algorithms, develop and implement an optimized version of the LBPH method, as well as evaluate its effectiveness on practical examples using the OpenCV library.

**The object of the study** is the processes and technologies of automatic face recognition based on machine learning methods.

**The subject of the study** is algorithms and machine learning models used for face recognition, in particular the LBPH method and its improvement.

**The first section** provides a thorough analysis of the subject area of face recognition and machine learning methods, which allowed us to determine the potential and limitations of existing approaches.

**The second section** presents a detailed description of face recognition methods and their characteristics, which became the basis for choosing the LBPH method as the basic one for improvement.

**The third section** implements a practical face recognition system using OpenCV, and experimental verification and optimization of the LBPH algorithm are carried out, which confirmed the feasibility of the chosen approach.

**Conclusion:** a prototype of a face recognition system using the OpenCV library in the Visual Studio environment has been implemented. A series of experiments have been conducted to evaluate the accuracy and performance of the solution.

**KEYWORDS:** FACE RECOGNITION; MACHINE LEARNING; LOCAL BINARY PATTERNS; COMPUTER VISION; OPENCV; EIGENFACES; DEEP LEARNING

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ .....	8
ВСТУП .....	10
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ МАШИННОГО НАВЧАННЯ ТА МЕТОДІВ РОЗПІЗНАВАННЯ ОБРАЗІВ .....	14
1.1. Порівняльний аналіз алгоритмів та покращення методу локальних бінарних шаблонів для автоматичного розпізнавання обличчя .....	14
1.2. Постановка задачі та актуальність розпізнавання обличчя .....	17
1.3. Огляд машинного навчання, розпізнавання об'єктів та його застосування до розпізнавання обличчя.....	18
1.3.1. Вступ до машинного навчання та розпізнавання об'єктів .....	18
1.3.2. Огляд розпізнавання обличчя .....	19
1.3.3. Виклики у розпізнаванні обличчя .....	22
1.3.4. Огляд поточних робіт .....	23
1.4. Огляд методів та алгоритмів розпізнавання обличчя .....	28
1.4.1. Метод Eigenfaces (Власні обличчя) .....	28
1.4.2. Метод Fisherfaces (Обличчя Фішера) .....	29
1.4.3. Метод Local Binary Patterns Histograms (LBPН) .....	30
1.4.4. Сучасні методи на основі глибокого навчання (CNN-based) .....	31
РОЗДІЛ 2. ПРЕДСТАВЛЕННЯ МЕТОДОЛОГІЙ ПОБУДОВИ РІШЕННЯ ДЛЯ ОПТИМІЗАЦІЇ ТА НАЛАШТУВАННЯ МЕТОДІВ РОЗПІЗНАВАННЯ ОБРАЗІВ.....	33
2.1. Опис методу Eigenfaces .....	33
2.2. Опис методу Fisherfaces .....	35

					<b>БР.ІІ – 13.00.00.000 ПЗ</b>				
Змн.	Арк.	№ докум.	Підпис	Дата	Побудова рішення для оптимізації та налаштування методів розпізнавання образів  <b>Пояснювальна записка</b>	Літ.	Арк.	Акрушіє	
Розроб.		Сивак Р.О.						6	
Перевір.		Процюк В.Р.							
Реценз.									
Н. Контр.		Піх М.М.							
Затверд.		Бандура В.В.						<b>ІФНТУНГ ІІ-21-4</b>	

2.3. Дослідження методу Local Binary Patterns Histograms (LBPН) .....	36
2.4. Оригінальні та розширені локальні бінарні шаблонні гістограми .....	38
2.5. Представлення переваг і недоліків шаблонів .....	40
2.6. Розробка алгоритмів та представлення етапів обробки алгоритму LBPН для розпізнавання обличчя .....	41

РОЗДІЛ 3. ІМПЛЕМЕНТАЦІЯ РІШЕННЯ ДЛЯ ОПТИМІЗАЦІЇ ТА НАЛАШТУВАННЯ МЕТОДІВ РОЗПІЗНАВАННЯ ОБРАЗІВ .....

3.1. Процедура інтеграції бібліотеки OpenCV із середовищем розробки Visual Studio за допомогою CMake .....	53
3.2. Процедура компіляції, підготовки даних та тестування прикладів бібліотеки OpenCV для розпізнавання облич .....	56
3.3. Оцінка продуктивності та порівняння методів розпізнавання обличчя .....	61
3.4. Удосконалення алгоритму просторової гістограми локальних бінарних шаблонів контексті аналізу зображень обличчя.....	69

ВИСНОВКИ..... 75

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ .....

БІБЛІОГРАФІЧНА ДОВІДКА

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

CV - Computer Vision - Комп'ютерний зір — галузь, що вивчає, як комп'ютери "бачать" та інтерпретують зображення.

PR - Pattern Recognition - Розпізнавання шаблонів — методи для класифікації даних на основі виявлених закономірностей.

FR - Face Recognition - Розпізнавання облич — визначення або ідентифікація людей за їхніми обличчями.

FD - Face Detection - Виявлення облич — знаходження облич на зображенні без ідентифікації.

ROI - Region of Interest - Область інтересу — виділений фрагмент зображення для аналізу.

CNN - Convolutional Neural Network - Згортова нейронна мережа — популярна архітектура для аналізу зображень.

RNN - Recurrent Neural Network - Рекурентна мережа — використовується для послідовних даних.

GAN - Generative Adversarial Network - Генеративно-змагальна мережа — для створення штучних зображень.

DNN - Deep Neural Network - Глибока нейронна мережа — загальний термін для багаторівневих НМ.

LFW - Labeled Faces in the Wild - Популярний датасет для тестування розпізнавання облич.

MS-Celeb-1M - Microsoft Celebrity 1 Million - Великий датасет облич.

VGGFace - Visual Geometry Group Face Dataset - Датасет і модель для розпізнавання облич.

YOLO - You Only Look Once - Реальна-time модель для виявлення об'єктів.

MTCNN - Multi-task Cascaded Convolutional Networks - Архітектура для виявлення та вирівнювання облич.

					БР.ІП – 13.00.00.000 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

HOG - Histogram of Oriented Gradients - Метод для опису локальних ознак зображень.

SIFT - Scale-Invariant Feature Transform - Метод виявлення та опису локальних ознак.

SURF - Speeded-Up Robust Features - Поліпшення SIFT з кращою швидкістю.

					БР.ІП – 13.00.00.000 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

## ВСТУП

У сучасному світі, де автоматизація та інтелектуальні системи відіграють дедалі важливішу роль у повсякденному житті, розпізнавання образів, зокрема обличчя, стало однією з найактуальніших задач у галузі комп'ютерного зору. Системи розпізнавання обличчя широко застосовуються в системах безпеки, контролю доступу, відеоспостереженні, фінансових та мобільних застосунках, а також у сфері розваг та маркетингу. Ефективне розпізнавання вимагає не лише точності, а й швидкодії, масштабованості та стійкості до зовнішніх чинників, таких як варіації освітлення, виразу обличчя, кута зйомки або часткового закриття обличчя.

Протягом останніх десятиліть було запропоновано безліч алгоритмів для вирішення цієї задачі. Серед них — методи, що базуються на статистичних характеристиках (Eigenfaces, Fisherfaces), локальних дескрипторах (Local Binary Patterns, LBPН), а також сучасні моделі глибокого навчання, зокрема згорткові нейронні мережі (CNN). Кожен із підходів має свої переваги й обмеження. У реальних умовах часто важливо знайти компроміс між обчислювальною складністю та точністю, що зумовлює актуальність досліджень, спрямованих на вдосконалення класичних методів розпізнавання.

### Актуальність роботи

У сучасному світі, де ідентифікація особистості має вирішальне значення для забезпечення безпеки, доступу до сервісів, персоналізації та аналізу поведінки користувачів, автоматичне розпізнавання обличчя стало одним з найважливіших напрямів у сфері комп'ютерного зору та штучного інтелекту. Значний прогрес у машинному навчанні, особливо в глибоких нейронних мережах, дозволив досягти високої точності в задачах розпізнавання, проте залишається низка викликів, пов'язаних з варіативністю освітлення, ракурсів, частковим перекриттям облич та віковими змінами.

					БР.ІП – 13.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

Особливої уваги заслуговують методи, що поєднують точність та швидкість обробки, як-от Local Binary Patterns Histograms (LBPН), які мають потенціал для оптимізації та практичного впровадження в реальних умовах. Актуальність дослідження обумовлена потребою в удосконаленні таких методів для підвищення ефективності систем розпізнавання обличчя.

Розпізнавання обличчя залишається однією з ключових задач у галузі комп'ютерного зору та машинного навчання, що має широке практичне застосування. Попри значний прогрес у розробці алгоритмів глибокого навчання, вони часто вимагають великих обчислювальних ресурсів і об'ємів навчальних даних. У цьому контексті класичні методи, такі як Eigenfaces, Fisherfaces і особливо LBPН, залишаються актуальними завдяки своїй простоті та високій ефективності в реальних умовах. Актуальність дослідження зумовлена необхідністю подальшого вдосконалення цих методів, їх оптимізації та адаптації до нових викликів — таких як варіативність зображень, шум, освітлення та інші фактори. Робота спрямована на розробку рішення, яке поєднує ефективність класичних підходів і гнучкість сучасних вимог до систем розпізнавання облич.

Особливу увагу в роботі приділено методу LBPН, який вирізняється високою швидкістю, простотою реалізації та здатністю працювати з невеликим набором тренувальних даних. Проте стандартна реалізація LBPН має певні обмеження щодо стійкості до змін освітлення та масштабування. У зв'язку з цим у межах роботи було запропоновано вдосконалений підхід до формування просторових гістограм локальних бінарних шаблонів, що дозволяє підвищити точність класифікації облич на зображеннях.

**Мета роботи** полягає в аналізі існуючих алгоритмів розпізнавання обличчя, розробці та імплементації оптимізованого варіанту методу LBPН, а також оцінці його ефективності на практичних прикладах із використанням бібліотеки OpenCV.

					БР.ІП – 13.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

Для цього було поєднано теоретичні аспекти машинного навчання з практичними методами програмної реалізації та тестування.

### **Завдання дослідження**

- Провести аналіз сучасних методів машинного навчання та алгоритмів розпізнавання облич.
- Дослідити сильні та слабкі сторони класичних та сучасних підходів (Eigenfaces, Fisherfaces, LBPН, CNN).
- Розробити покращену версію алгоритму LBPН.
- Реалізувати рішення із застосуванням OpenCV та Visual Studio.
- Провести порівняльне тестування продуктивності та точності методів.
- Надати рекомендації щодо практичного застосування оптимізованого методу.

**Об'єкт дослідження** - процеси та технології автоматичного розпізнавання облич на основі методів машинного навчання.

**Предмет дослідження** - алгоритми та моделі машинного навчання, що застосовуються для розпізнавання облич, зокрема метод LBPН та його вдосконалення.

### **Методи дослідження**

- Теоретичний аналіз літературних джерел.
- Порівняльний аналіз алгоритмів.
- Математичне моделювання.
- Експериментальне тестування.
- Програмна реалізація за допомогою OpenCV та C++.
- Візуалізація та статистична оцінка результатів.

### **Наукова новизна**

У роботі запропоновано вдосконалений підхід до побудови просторових гістограм у межах алгоритму LBPН, що дозволяє підвищити якість розпізнавання обличчя при збереженні обчислювальної ефективності, що є важливим для застосування в реальному часі.

					БР.ІП – 13.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

## Практичне застосування

Результати дослідження можуть бути використані при розробці систем відеоспостереження, контролю доступу, в мобільних додатках для розпізнавання користувачів, а також в освітніх і дослідницьких цілях для вивчення методів комп'ютерного зору.

Бакалаврська робота містить 79 сторінок, 25 рисунків, 3 розділи список використаних джерел із 45 найменуваннями.

					БР.ІП – 13.00.00.000 ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

# РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ МАШИННОГО НАВЧАННЯ ТА МЕТОДІВ РОЗПІЗНАВАННЯ ОБРАЗІВ

## 1.1. Порівняльний аналіз алгоритмів та покращення методу локальних бінарних шаблонів для автоматичного розпізнавання обличчя

Розпізнавання обличчя є однією з ключових задач у галузі комп'ютерного зору та біометричної ідентифікації. Цей процес полягає в автоматичній ідентифікації або верифікації особи шляхом аналізу зображень її обличчя. Зростаючі вимоги до систем безпеки, відеоспостереження та контролю доступу обумовлюють нагальну потребу у розробці та вдосконаленні надійних та високоточних алгоритмів розпізнавання обличчя.

Критично важливим аспектом є мінімізація показника помилкової ідентифікації. Незважаючи на існування значної кількості методів розпізнавання обличчя, універсально "найкращий" алгоритм відсутній, що стимулює дослідження та порівняльний аналіз їхньої ефективності за різних умов.

В даній роботі проведено порівняльне дослідження трьох відомих алгоритмів розпізнавання обличчя: Eigenfaces, Fisherfaces та Local Binary Patterns (LBP). Метою дослідження є визначення методу з найвищою точністю прогнозування на стандартних наборах даних та розробка підходу до покращення продуктивності одного з досліджуваних алгоритмів.

Для проведення експериментального порівняння алгоритмів та оцінки їхньої ефективності було використано програмні інструменти OpenCV (для обробки зображень та реалізації алгоритмів), CMake (як система збірки проекту) та Visual Studio (як інтегроване середовище розробки).

Експериментальна процедура складалася з наступних нижчеперелічених етапів:

					БР.ІП – 13.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

- Формування набору даних. Збір колекції зображень обличч різних осіб, які слугували експериментальними суб'єктами. Набір даних був розділений на тренувальний (для навчання системи) та тестовий (для оцінки точності).

- Етап навчання. Для кожного з досліджуваних алгоритмів проводилося навчання на тренувальному наборі даних. На цьому етапі система аналізувала зображення, виділяла характерні ознаки (наприклад, власні обличчя для Eigenfaces/Fisherfaces, локальні текстурні шаблони для LBP) та формувала відповідні моделі або простори ознак для кожного суб'єкта.

- Етап тестування. Нові зображення обличчя з тестового набору подавалися на вхід навченої системи. Система порівнювала виділені ознаки тестового зображення з навченими моделями і намагалася ідентифікувати особу, присвоюючи їй ідентифікатор одного з суб'єктів з тренувального набору.

Оцінка точності. Точність розпізнавання (prediction rate) визначалася як відсоток правильно ідентифікованих обличч на тестовому наборі.

За результатами проведених експериментів було встановлено, що серед трьох досліджених базових алгоритмів найвищий показник точності розпізнавання продемонстрував метод Eigenfaces. Алгоритм Local Binary Patterns (LBP) показав найнижчу точність, а Fisherfaces – проміжний результат.

Враховуючи потенціал методу LBP для ефективного представлення текстурних ознак обличчя, незважаючи на його початково нижчу продуктивність порівняно з методами, що базуються на просторах ознак (Eigenfaces, Fisherfaces), було прийнято рішення зосередити подальшу роботу на його вдосконаленні.

Запропонована модифікація алгоритму LBP полягає у застосуванні адаптивного зважування локальних гістограм LBP. На етапі навчання, для кожного суб'єкта аналізуються гістограми LBP, отримані з різних локальних регіонів обличчя. Визначаються найбільш інформативні або дискримінантні

регіони, тобто ті, які найкраще дозволяють відрізнити дану особу від інших. Вагові коефіцієнти для кожного регіону призначаються пропорційно його інформативності або на основі аналізу контрастності пікселів у відповідній області.

На етапі розпізнавання, при обробці нового зображення обличчя, гістограми LBP з різних регіонів також обчислюються, але при порівнянні з навченими моделями застосовуються попередньо визначені адаптивні вагові коефіцієнти. Надання більшої ваги більш значущим регіонам має на меті підвищити точність ідентифікації, особливо в умовах, коли деякі частини обличчя можуть бути спотворені (наприклад, через освітлення, часткове перекриття або вираз обличчя). Додатково, такий підхід потенційно може сприяти прискоренню процесу розпізнавання в реальному часі за рахунок оптимізації процедури порівняння.

Проведені експерименти з модифікованим алгоритмом LBP підтвердили ефективність запропонованого підходу. Порівняно з базовою версією LBP, модифікований алгоритм продемонстрував статистично значуще покращення показника точності розпізнавання на тестовому наборі даних.

В рамках даної роботи було проведено порівняльний аналіз трьох алгоритмів розпізнавання обличчя (Eigenfaces, Fisherfaces, LBP) та показано, що на використаному наборі даних Eigenfaces має найвищу точність. Запропоновано та реалізовано модифікацію алгоритму LBP шляхом адаптивного зважування локальних гістограм на основі інформативності регіонів обличчя. Експериментально підтверджено, що запропонована модифікація значно покращує точність розпізнавання порівняно з базовим алгоритмом LBP, що свідчить про перспективність подальших досліджень у напрямку адаптивних методів обробки ознак для підвищення робастності та точності систем розпізнавання обличчя.

					БР.ІП – 13.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

## 1.2. Постановка задачі та актуальність розпізнавання обличчя

Основною метою даної роботи є порівняльний аналіз, експериментальна оцінка та вдосконалення існуючих алгоритмів автоматичного розпізнавання обличчя.

Важливо чітко розрізнити два фундаментальні завдання у сфері комп'ютерного зору, пов'язані з обробкою зображень обличчя: детекцію обличчя та розпізнавання (ідентифікацію або верифікацію) обличчя.

Детекція обличчя (Face Detection) – це задача визначення наявності та точного місцезнаходження (локалізації) одного або кількох облич на зображенні чи у відеопотоці. Як правило, результат детекції представлений у вигляді обмежувальних прямокутників (bounding boxes) навколо виявлених облич. По суті, детекція відповідає на бінарне питання: "Чи є обличчя на цьому зображенні?" або "Де знаходяться обличчя на зображенні?". Це часто є початковим етапом у складніших системах обробки обличчя.

Розпізнавання обличчя (Face Recognition, Face Identification, Face Verification) – це задача встановлення особистості людини за зображенням її обличчя. На відміну від детекції, розпізнавання відповідає на запитання: "Кому належить це обличчя?" (ідентифікація, пошук відповідності у базі даних) або "Чи є це обличчя тією ж особою, що і обличчя Y?" (верифікація, підтвердження особи). Задача розпізнавання є значно складнішою, оскільки вимагає не лише знайти обличчя, а й екстрагувати з нього унікальні біометричні ознаки та зіставити їх із раніше збереженими шаблонами.

У багатьох критично важливих застосуваннях, таких як біометрична аутентифікація та системи безпеки, саме розпізнавання обличчя, а не просто його детекція, має вирішальне значення. Системи біометричної аутентифікації використовують унікальні фізіологічні або поведінкові характеристики особи для підтвердження її ідентичності. Розпізнавання обличчя є неінвазивним та зручним методом біометричної аутентифікації.

					БР.ІП – 13.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

З точки зору безпеки, точна ідентифікація особи є фундаментальною вимогою. Це стосується систем контролю фізичного доступу (наприклад, вхід до приміщень обмеженого доступу, зони аеропортів), логічного контролю доступу (наприклад, вхід до комп'ютерних систем, мобільних пристроїв), а також систем відеоспостереження для ідентифікації осіб у натовпі або пошуку розшукуваних осіб. У цих сценаріях правильне розпізнавання особи є критично важливим для надання відповідних прав доступу або вжиття необхідних заходів безпеки, при цьому помилки ідентифікації (як помилкові позитивні, так і помилкові негативні спрацьовування) можуть мати серйозні наслідки.

Сучасні мобільні пристрої (смартфони, планшети) широко використовують розпізнавання обличчя як зручний та безпечний спосіб аутентифікації користувача для розблокування пристрою, доступу до встановлених додатків або підтвердження онлайн-платежів. Це забезпечує додатковий рівень захисту персональних даних та фінансової інформації користувача від несанкціонованого доступу.

Таким чином, актуальність досліджень у галузі розпізнавання обличчя зумовлена його ключовою роллю у створенні надійних та ефективних систем біометричної безпеки та персональної ідентифікації в широкому спектрі застосувань.

### **1.3. Огляд машинного навчання, розпізнавання об'єктів та його застосування до розпізнавання обличчя**

#### *1.3.1. Вступ до машинного навчання та розпізнавання об'єктів*

Машинне навчання є підгалуззю штучного інтелекту, що зосереджується на розробці алгоритмів, які дозволяють комп'ютерним системам навчатися без явного програмування. Машинне навчання стосується вивчення алгоритмів, які аналізують дані для того, щоб допомогти

					БР.ІП – 13.00.00.000 ПЗ	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

комп'ютерним системам ставати точнішими з часом при виконанні завдання. Ключовою метою машинного навчання є надання системам здатності перманентно покращувати свою продуктивність у певному завданні шляхом автоматичного виявлення закономірностей у даних. Цей процес реалізується через конструювання та оптимізацію алгоритмів, що дозволяють системам адаптуватися та вчитися. Класичним прикладом застосування машинного навчання є фільтрація спаму, де система постійно навчається розрізняти небажані повідомлення на основі зворотного зв'язку від користувача.

Машинне навчання відіграє фундаментальну роль у розвитку систем розпізнавання об'єктів. Розпізнавання об'єктів – це задача комп'ютерного зору, що полягає не лише у визначенні наявності об'єкта на зображенні або у відеопотоці (що є задачею детекції), а й у класифікації цього об'єкта за певною категорією та його точній локалізації. Іншими словами, система розпізнавання об'єктів відповідає на запитання: "Що це за об'єкт?" та "Де він знаходиться?". Одним із найбільш відомих і активно досліджуваних підзавдань розпізнавання об'єктів є розпізнавання обличчя.

### *1.3.2. Огляд розпізнавання обличчя*

Розпізнавання обличчя є високоактуальним напрямом у галузі комп'ютерного зору та біометричної ідентифікації, що стрімко розвивається завдяки прогресу у алгоритмічній базі, зростанню обчислювальної потужності та зниженню вартості обладнання. Це завдання полягає у встановленні особистості людини на основі аналізу зображень її обличчя.

Ключовою причиною інтенсивного дослідження та впровадження систем розпізнавання обличчя є їхній потенціал для забезпечення безпеки та ефективного спостереження. Крім того, розпізнавання обличчя має широкий спектр застосувань, що охоплюють різноманітні сфери життєдіяльності. Основні напрямки застосування включають:

					БР.ІП – 13.00.00.000 ПЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

- Контроль фізичного доступу (будівлі, аеропорти, морські порти, банкомати, контрольні-пропускні пункти); забезпечення комп'ютерної та мережевої безпеки; аутентифікація користувачів на мультимедійних робочих станціях (наприклад, вхід до системи електронної пошти).

- Автоматичне відстеження та ідентифікація осіб зі списків розшукуваних осіб (наприклад, злочинців) у системах відеоспостереження (ССТV).

- Ідентифікація в процесах реєстрації виборців, банківському секторі, електронній комерції; ідентифікація новонароджених; використання у національних посвідченнях особи, паспортах, водійських посвідченнях, перепустках співробітників.

- Системи формування та аналізу фотороботів/реєстраційних фотографій; аналіз відеоматеріалів після подій; використання у криміналістичних експертизах.

- Пошук осіб у великих базах даних (наприклад, ліцензованих водіїв, отримувачів соціальних виплат, зниклих безвісти дітей, іммігрантів, осіб з поліцейських дос'є).

- Збереження біометричного шаблону обличчя (а не самого зображення) на смарт-карті, у штрих-кодів або на магнітній смужці для подальшої аутентифікації шляхом порівняння "живого" зображення з цим шаблоном.

- Мультимедійні середовища та адаптивні інтерфейси користувача-комп'ютера. Частина контекстно-залежних систем; моніторинг поведінки (наприклад, у дитячих садках чи закладах для літніх людей); розпізнавання клієнтів та оцінка їхніх потреб.

- Автоматичне маркування та пошук обличчя у відеозаписах.

- Побудова зображення обличчя на основі словесного опису свідків.

У контексті розпізнавання обличчя прийнято виділяти дві основні категорії задач, які, хоча й схожі, мають суттєві відмінності. Це верифікація

					БР.ІП – 13.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

обличчя та ідентифікація обличчя. Перевірка обличчя – це зіставлення типу 1:1, яке порівнює зображення обличчя з еталонним зображенням обличчя, чия особистість заявляється. Навпаки, ідентифікація обличчя – це задача типу 1:N, яка порівнює запитове зображення обличчя з усіма еталонними зображеннями в базі даних обличчя, щоб визначити особистість запитового обличчя.

На сучасному ринку існує велика кількість програмних продуктів, що використовують розпізнавання обличчя. Більшість із них у задачах аутентифікації, зокрема для забезпечення безпеки пристроїв (наприклад, смартфонів або комп'ютерів), реалізують саме верифікацію обличчя (1:1 зіставлення). Пристрій надає доступ лише в тому випадку, якщо поточне зображення обличчя відповідає зареєстрованому еталону для заявленої особи. Прикладом такої системи є Windows Hello, інтегрована в операційну систему Windows 10.

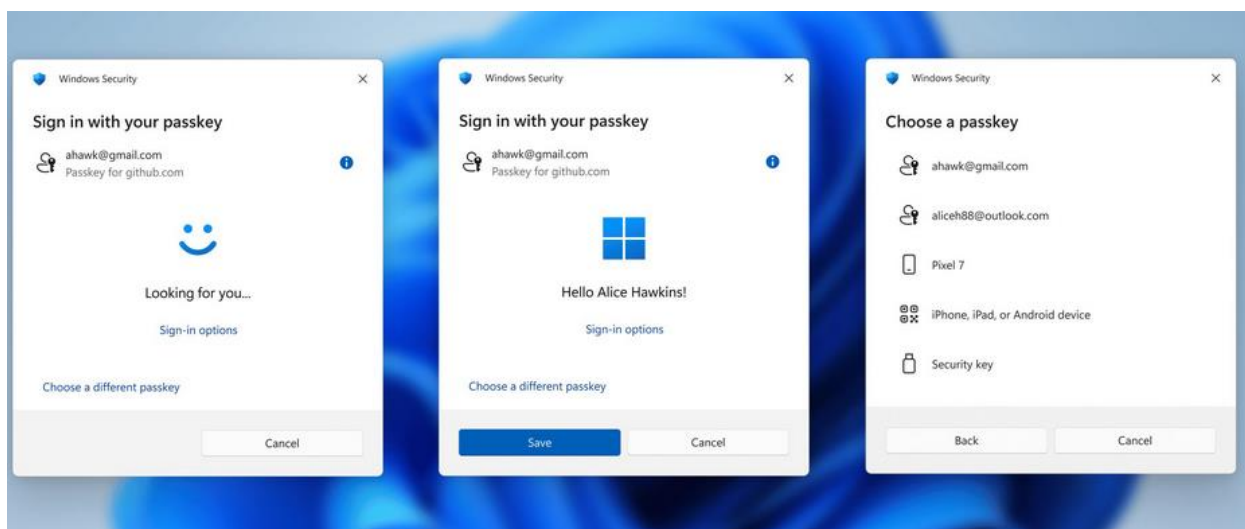


Рисунок 1.1 – Система розпізнавання обличчя Windows Hello

Важливою передумовою для успішного розпізнавання обличчя є наявність якісного та робастного класифікатора об'єктів. Для створення такого класифікатора система машинного навчання повинна бути навчена на

					БР.ІП – 13.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		21

значному обсязі позначених даних (зображень облич), щоб алгоритми могли автоматично виділяти дискримінантні ознаки, характерні для обличчя.

### *1.3.3. Виклики у розпізнаванні обличчя*

Розробка ефективних систем розпізнавання обличчя стикається з рядом суттєвих викликів. Однією з ключових проблем є створення робастного класифікатора, здатного долати варіації зображень. "Слабкий" класифікатор демонструє високий рівень помилок (наприклад, понад 50% за певних розподілів даних).

Для побудови робастного класифікатора необхідний достатньо великий та різноманітний навчальний набір даних з позначеними зображеннями. Недостатній обсяг або репрезентативність навчальних даних може призвести до надмірного узагальнення або, навпаки, недоузагальнення, спричиняючи помилкові спрацювання.

Зокрема, може виникнути проблема хибних позитивних спрацювань (False Positives), коли система неправильно ідентифікує об'єкт, якого насправді немає (наприклад, "знаходить" обличчя на зображенні з деревами). З іншого боку, надмірно великий навчальний набір або навчання на надто специфічних ознаках може призвести до низького показника виявлення (False Negatives), коли система не може розпізнати обличчя через незначні відхилення від навчених шаблонів (наприклад, не розпізнає обличчя з окулярами, якщо навчальний набір переважно містив обличчя без них). Оптимальний розмір та склад навчального набору переважно визначається емпірично.

Крім того, значним викликом, особливо для ідентифікації обличчя в реальних умовах, є вплив різноманітних зовнішніх та внутрішніх факторів, що змінюють вигляд обличчя на зображенні. Існує п'ять ключових факторів, що впливають на продуктивність розпізнавання обличчя, включають:

					БР.ІП – 13.00.00.000 ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

1. Варіації освітлення. Різні умови освітлення суттєво змінюють розподіл яскравості та тіней на обличчі, що може значно погіршувати продуктивність 2D методів, особливо в поєднанні зі змінами пози.

2. Зміни пози (ракурсу). Обертання голови призводить до проєкційних спотворень та самозатінення. Хоча існують методи, що справляються з обмеженими кутами повороту (наприклад, до 32 градусів), кути огляду камер спостереження можуть виходити за ці межі.

3. Вирази обличчя. Різноманітні емоційні стани людини (усмішка, здивування тощо) змінюють геометрію обличчя. Більшість сучасних алгоритмів є відносно стійкими до помірних виразів, але екстремальні вирази (наприклад, крик) можуть викликати проблеми.

4. Затримка часу (старіння). Зміни обличчя з віком є нелінійними та довготривалими, що становить серйозний виклик. Це одна з найскладніших проблем у розпізнаванні обличчя, яка досі недостатньо вирішена.

5. Часткові перекриття (Occlusion): Наявність об'єктів, що закривають частину обличчя (шарфи, головні убори, руки, інші люди), значно ускладнює процес розпізнавання, особливо якщо перекриті ключові дискримінантні області.

Крім того, схожість рис обличчя серед різних осіб у навчальній або тестовій вибірці (наприклад, у членів однієї сім'ї) також може призводити до помилок ідентифікації.

#### *1.3.4. Огляд поточних робіт*

На сучасному етапі розпізнавання обличчя є надзвичайно активною дослідницькою галуззю. Розробляється та тестується велика кількість різноманітних алгоритмів, що базуються на різних принципах, як показано, наприклад, в таблиці 1.1. Серед широко відомих та досліджуваних методів можна виділити Eigenfaces, Fisherfaces та алгоритми на основі Local Binary Patterns (LBP).

					БР.ІП – 13.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

Таблиця 1.1 - Алгоритми розпізнавання обличчя

Назва	Бази даних	Розмір зображення	Проміжок часу	Показник розп. (%)	Експ.	ВЛ	Поза	Вік
PCA	AR-Faces	85×6085×60	Hi	70		Hi	Hi	Hi
LDA	AR-Faces	85×6085×60	Hi	88		Hi	Hi	Hi
Fisharfaces	VALE		Hi	99,6	Так	Так	Hi	Hi
Direct LDA	ORL	112 x 92	Hi	90,8	Так	Так	Так	Hi
DF-LDA	ORL	112 x 92	Так	96		Так	Hi	Hi
	UMIST	112 x 92	Hi	98		Hi	Hi	Hi
DCV	Yale	126 x 152	Hi	97,33		Так	Hi	Hi
	AR-Faces	229 x 299	Так	99,35				
ICA	FERET	60 x 50	Так	89	Так	Hi	Hi	Hi
PDB/NN	SCR	80×2080×20	Hi	100	Так	Так	Так	Hi
	FERET		Hi	99	Так	Так	Hi	Hi
	ORL		Hi	96		Так	Так	Hi
RBF	ORL	160 x 120		98,1	Так		Так	Hi
	PropertyDB			100				
HMM	FERET	128 x 128	Hi	97	Так	Hi	Hi	Hi
DLA	PropertyDB	128 x 128	Hi	90,3	Так		Так	Hi
Gabor EFM	FERET	128 x 128	Hi	99	Так	Hi	Hi	Hi
	ORL	128 x 128	Hi	100	Так	Hi	Так	Hi
EGM	FERET	256 x 384	Hi	80	Так		Так	Hi
	PropertyDB			90	Так		Так	Hi
WPA	MIT	480 x 640		80,5	Так	Так		no
	FERET	256 x 384		89				
IFS	PropertyDB	64 x 64		100		Hi	Hi	Hi
IFS	ORL	92×11292×112	Hi	95				Hi
IFS	MIT	480 x 640		90			Так	Hi
Th-Infrared	PropertyDB		Hi	98	Так	Так	Hi	Hi
Thermal	PropertyDB	99×13299×132	Так	93	Так	Так	Hi	Hi
Th-Spectrum	Equinox			86,8	Так		Так	Hi
Hyperspectral	PropertyDB		Так	92	Hi	Так	Hi	Hi
LEM	Bern		Hi	72,09	Так		Так	Hi
	AR-Faces			86,03		Так	Hi	Hi
	Yale			85,45		Так	Hi	Hi
ICA	Subset of AR	46×5646×56		98		Так	Так	Hi
	Faces, Yale, ORL, Bern and FERET							

Назва	Бази даних	Розмір зображення	Проміжок часу	Показник розп. (%)	Експ.	ВЛ	Поза	Вік
LDA/GSVD	CMC_PIE/Pose27		Hi	100	Hi	Так	Hi	Hi
LDA/GSVD				99,53				
LDA/GS				99				
	YaleB/Pose00		Hi	99				
				98,03		Так	Hi	Hi
Cones Gen.	Yale B	36×4236×42	Hi	97	Hi	Так	Так	Hi
Linear	ATR-Database		Hi	98,7	n	Hi	Так	Hi
Subspaces								
Eigen Lights	CMC-PIE		Hi	36	Hi	Так	Так	Hi

Значний обсяг досліджень у цій галузі став можливим завдяки доступності великих публічних баз даних зображень облич, скомпільованих дослідницькими організаціями.

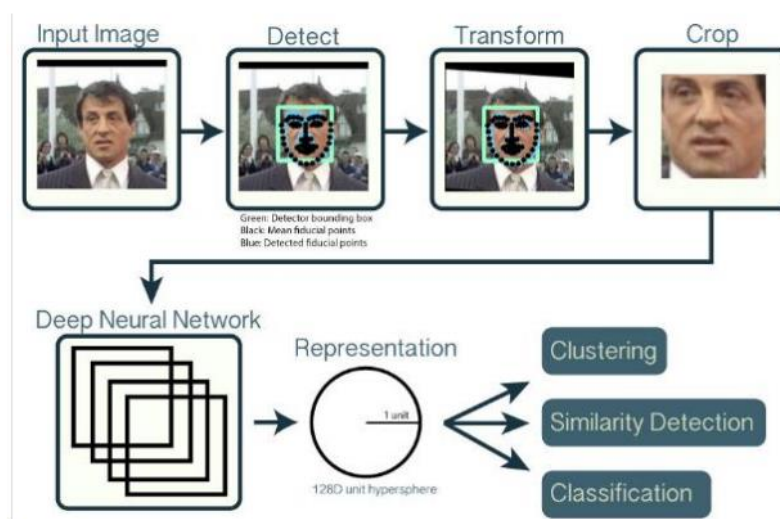


Рисунок 1.2 – Алгоритм роботи системи автоматичного розпізнавання обличчя

На рисунку 1.2 зображено типовий конвеєр (pipeline) системи автоматичного розпізнавання обличчя, заснований на методах глибокого навчання. Процес включає кілька послідовних етапів:

- Вхідне зображення (Input Image). Процес починається з отримання вхідного зображення, яке потенційно містить одне або кілька облич.

- Детекція (Detect). На цьому етапі система виконує задачу детекції обличчя. Метою є визначення наявності обличчя на зображенні та його точна локалізація. Результатом є визначення області, що містить обличчя, яка зазвичай позначається обмежувальною рамкою (Bounding Box), як показано на рисунку зеленим кольором. Одночасно або як наступний крок детекції може здійснюватися локалізація ключових точок обличчя (Fiducial Points) – характерних міток на обличчі (наприклад, куточки очей, кінчик носа, куточки рота). На рисунку показані середні (референсні) фідуціальні точки чорним кольором та виявлені точки синім кольором.

- Трансформація/Вирівнювання (Transform). Використовуючи координати виявлених фідуціальних точок, обличчя на зображенні піддається геометричним перетворенням. Цей етап, який також називається алайнментом або вирівнюванням (Alignment), має на меті стандартизувати положення та орієнтацію обличчя (наприклад, зробити його фронтальним, вирівняти по очах). Це необхідно для зменшення впливу варіацій ракурсу на подальші етапи.

- Вирізання (Crop). Після вирівнювання стандартизована область, що містить обличчя, вирізається з оригінального зображення. Це забезпечує, що для подальшої обробки подається лише релевантна частина зображення, що містить обличчя, у стандартизованому розмірі та положенні.

- Глибока нейронна мережа (Deep Neural Network). Вирізане та вирівняне зображення обличчя подається на вхід глибокої нейронної мережі (наприклад, Convolutional Neural Network - CNN). Ця мережа виконує функцію екстракції ознак – вона обробляє зображення через багато шарів згортки, пулінгу та активації, автоматично виділяючи ієрархічні та високодискримінантні ознаки обличчя. Це є ключовим етапом, де сирі піксельні дані перетворюються на більш абстрактне та інформативне представлення.

					БР.ІП – 13.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

- Представлення (Representation). Виходом глибокої нейронної мережі є компактний вектор ознак, який часто називають ембедінгом (embedding). На рисунку він символічно зображений як точка на гіперсфері (наприклад, 1280-вимірна одинична гіперсфера), що підкреслює ідею стандартизованого представлення. Цей вектор числових значень є унікальним "відбитком" обличчя в багатовимірному просторі ознак. Важливо, що вектори ознак для зображень однієї особи знаходяться близько один до одного у цьому просторі, тоді як вектори різних осіб – далеко.

- Подальші задачі (Downstream Tasks). Отриманий вектор ознак може бути використаний для вирішення різних задач розпізнавання обличчя:

Кластеризація (Clustering). Групування векторів ознак схожих облич (наприклад, для пошуку всіх зображень однієї особи у великій колекції).

- Визначення подібності (Similarity Detection): Обчислення міри подібності (наприклад, косинусної подібності або евклідової відстані) між двома векторами ознак. Ця задача є основою для верифікації обличчя (порівняння 1:1).

- Класифікація (Classification): Зіставлення вектора ознак тестового обличчя з векторами ознак зареєстрованих осіб у базі даних для визначення особистості (ідентифікація, пошук 1:N) або присвоєння класу (імені).

Таким чином, рисунок 1.2 ілюструє послідовний процес: від отримання вхідного зображення, через етапи попередньої обробки (детекція, вирівнювання, вирізання), до екстракції дискримінантних ознак за допомогою глибокого навчання, і завершуючи використанням отриманого векторного представлення для вирішення конкретної задачі розпізнавання (ідентифікація, верифікація, кластеризація).

Популярні бібліотеки комп'ютерного зору, такі як OpenCV, вже містять реалізації алгоритмів Eigenfaces, Fisherfaces та LBP для розпізнавання обличчя. Саме ці реалізації алгоритмів були обрані та використані в рамках даного проекту для проведення експериментального тестування, порівняння

					БР.ІП – 13.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		27

їхньої продуктивності та подальшого вдосконалення алгоритму Local Binary Patterns.

#### 1.4. Огляд методів та алгоритмів розпізнавання обличчя

У галузі автоматичного розпізнавання обличчя розроблено значну кількість алгоритмів, які можна класифікувати за різними ознаками (наприклад, за підходом до представлення обличчя, за методом класифікації тощо). Нижче наведено опис кількох відомих та широко застосовуваних алгоритмів.

##### 1.4.1. Метод Eigenfaces (Власні обличчя)

Метод Eigenfaces є одним з ранніх та найбільш відомих прикладів застосування підходу, заснованого на глобальному представленні зовнішнього вигляду (appearance-based method) та аналізі лінійного підпростору. Він використовує метод головних компонент (Principal Component Analysis - PCA) для зменшення розмірності простору зображень обличчя та виділення найбільш значущих варіацій.

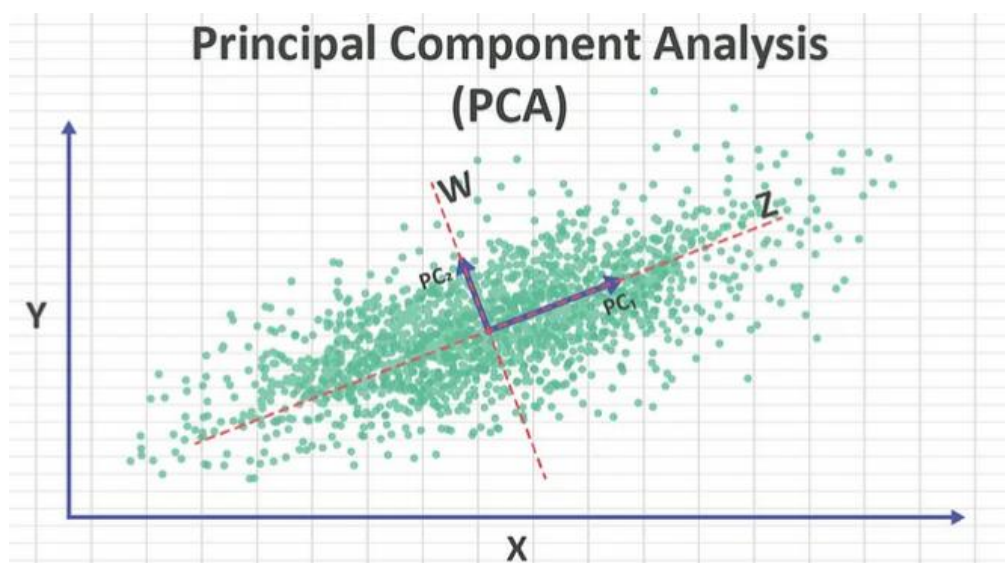


Рисунок 1.3 – Метод Principal Component Analysis - PCA

Набір тренувальних зображень обличчя перетворюється у вектори. Обчислюється коваріаційна матриця цих векторів. Знаходяться власні вектори (eigenvectors) цієї матриці, які відповідають найбільшим власним значенням (eigenvalues). Ці власні вектори у просторі зображень інтерпретуються як "власні обличчя" (eigenfaces). Вони формують ортонормований базис "простору обличчя". Кожне тренувальне або нове зображення обличчя проектується на цей підпростір, отримуючи вектор вагових коефіцієнтів. Розпізнавання здійснюється шляхом порівняння вектора вагових коефіцієнтів тестового зображення з векторами коефіцієнтів тренувальних зображень (наприклад, за евклідовою відстанню) у просторі обличчя.

Перевагою є відносна простота реалізації, висока швидкість на етапі розпізнавання (після побудови простору обличчя), ефективне стиснення даних.

#### 1.4.2. Метод Fisherfaces (Обличчя Фішера)

Метод Fisherfaces також є підходом, заснованим на зовнішньому вигляді та аналізі підпростору. Він використовує лінійний дискримінантний аналіз (Linear Discriminant Analysis - LDA) для пошуку проекції, яка краще розділяє різні класи (особи) порівняно з PCA. Часто застосовується після попереднього зменшення розмірності за допомогою PCA (PCA+LDA).

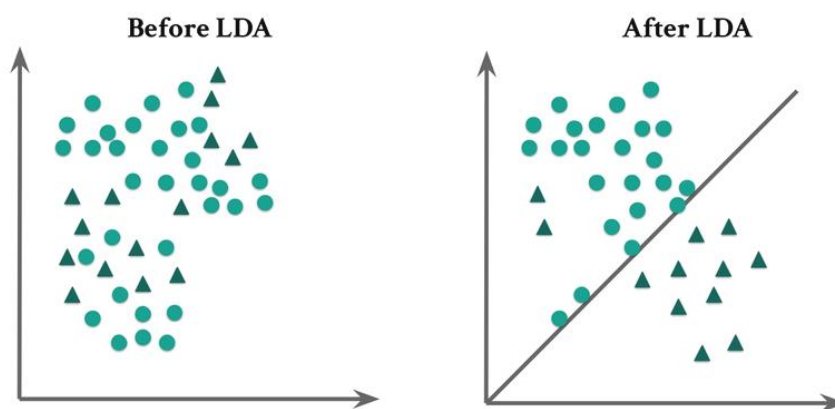


Рисунок 1.4 – Приклад роботи Linear Discriminant Analysis

LDA шукає таку лінійну трансформацію даних, яка максимізує відношення міжкласової дисперсії (scatter between classes) до внутрішньокласової дисперсії (scatter within classes). Це означає, що проєкція LDA намагається зблизити зображення одного класу (різні зображення однієї особи) та віддалити зображення різних класів (зображення різних осіб). Базисні вектори цього простору називаються "обличчями Фішера" (fisherfaces). Розпізнавання здійснюється шляхом проєктування зображень на цей "простір Фішера" та порівняння відстаней між проєкціями.

Даний метод є більш робастний до варіацій у межах одного класу (таких як освітлення та вирази обличчя) порівняно з Eigenfaces, оскільки явно враховує інформацію про класи при побудові простору ознак.

Але метод все ще чутливий до значних змін ракурсу. Потребує наявності достатньої кількості тренувальних зображень для кожної особи для ефективного застосування LDA.

### 1.4.3. Метод Local Binary Patterns Histograms (LBP)

Метод LBP базується на екстракції локальних текстурних ознак. Він аналізує локальні бінарні шаблони (Local Binary Patterns - LBP) – мікротекстурні описувачі, стійкі до монотонних змін освітлення.

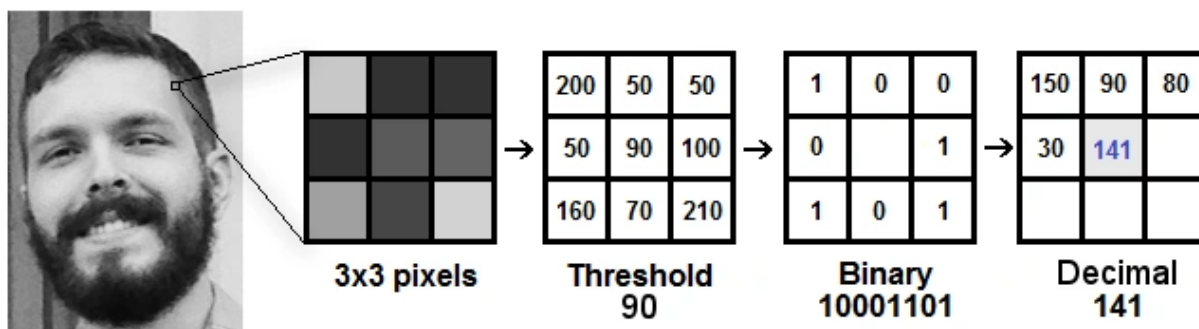


Рисунок 1.5 – Принцип роботи Local Binary Patterns

Зображення обличчя ділиться на невеликі прямокутні області (регіони). Для кожного пікселя в зображенні (або тільки для пікселів сітки)

обчислюється локальний бінарний шаблон шляхом порівняння інтенсивності центрального пікселя з інтенсивністю його сусідів. Результат порівняння (більший чи менший) кодується бінарною цифрою, формуючи бінарне число для групи сусідів. Для кожного регіону обчислюється гістограма розподілу таких локальних бінарних шаблонів. Ці локальні гістограми конкатенуються в один глобальний вектор ознак для всього обличчя. Розпізнавання здійснюється шляхом порівняння гістограм ознак тестового зображення з гістограмами тренувальних зображень (наприклад, використовуючи метрику перетину гістограм або  $\chi^2$ -квадрат відстань).

Перевагою є висока обчислювальна ефективність на етапі екстракції ознак, відносна стійкість до монотонних змін освітлення, добре захоплює локальну текстуру. Але даний метод є менш ефективний наодинці для глобальних варіацій порівняно з методами підпростору. Чутливий до змін масштабу. Базова версія не враховує глобальну структуру обличчя.

#### *1.4.4. Сучасні методи на основі глибокого навчання (CNN-based)*

Сучасні підходи до розпізнавання обличчя переважно базуються на архітектурах глибоких згорткових нейронних мереж (Convolutional Neural Networks - CNN). Ці мережі автоматично навчаються виділяти складні, ієрархічні та високодискримінантні ознаки безпосередньо із сирих піксельних даних.

Тренувальний процес включає навчання багат шарової CNN на надзвичайно великих наборах даних зображень обличчя (мільйони або десятки мільйонів зображень). Мережа оптимізується таким чином, щоб останні шари генерували компактний вектор ознак (embedding) для кожного обличчя, де вектори для зображень однієї особи є близькими, а для різних осіб – віддаленими. Розпізнавання полягає у пропуску тестового зображення через навчену мережу для отримання його вектора ознак та порівнянні цього

					БР.ІП – 13.00.00.000 ПЗ	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дата		

вектора з векторами ознак, попередньо збереженими для зареєстрованих осіб (зазвичай, за косинусною подібністю).

Демонструють найвищу, сучасну точність розпізнавання, значна робастність до варіацій освітлення, ракурсу, виразів обличчя та часткових перекриттів.

Але потребують величезних обсягів розмічених тренувальних даних та значних обчислювальних ресурсів (GPU) для навчання. Архітектура та внутрішня логіка роботи можуть бути складними для інтерпретації ("чорний ящик").

Вибір конкретного алгоритму розпізнавання обличчя залежить від специфічних вимог застосування, доступності обчислювальних ресурсів, обсягу та якості тренувальних даних, а також від умов, в яких буде працювати система (наприклад, контрольоване освітлення vs. природні умови).

					БР.ІП – 13.00.00.000 ПЗ	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

## РОЗДІЛ 2. ПРЕДСТАВЛЕННЯ МЕТОДОЛОГІЙ ПОБУДОВИ РІШЕННЯ ДЛЯ ОПТИМІЗАЦІЇ ТА НАЛАШТУВАННЯ МЕТОДІВ РОЗПІЗНАВАННЯ ОБРАЗІВ

В рамках даного проекту було проведено порівняльне дослідження та оцінку ефективності трьох алгоритмів автоматичного розпізнавання обличчя: Eigenfaces, Fisherfaces та методу, заснованого на гистограмах локальних бінарних шаблонів (Local Binary Patterns Histograms - LBPН). Оцінка та порівняння здійснювались за допомогою бібліотеки комп'ютерного зору OpenCV. Експериментальна процедура включала формування навчального набору даних, що складався із зображень обличчя декількох суб'єктів, з кількістю зразків від 20 до 50 фотографій на кожную навчальну особу. Після завантаження та обробки цих зображень для тренування відповідних моделей, проводилась оцінка точності розпізнавання кожного методу шляхом ідентифікації суб'єктів на тестовій вибірці, яка могла включати зображення або відеопотік.

### 2.1. Опис методу Eigenfaces

Метод Eigenfaces базується на застосуванні аналізу головних компонент (Principal Component Analysis - PCA) до набору тренувальних зображень обличчя. Основна ідея полягає у знаходженні підпростору, що найкраще описує розподіл даних зображень у багатовимірному просторі пікселів [4]. Процес реалізації включає наступні етапи:

1. Формування вибірки. Збирається набір тренувальних зображень обличчя. Рекомендовано, щоб зображення мали схожі умови освітлення, ракурс та були попередньо вирівняні (наприклад, за положенням очей та рота).

					БР.ІП – 13.00.00.000 ПЗ	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дата		

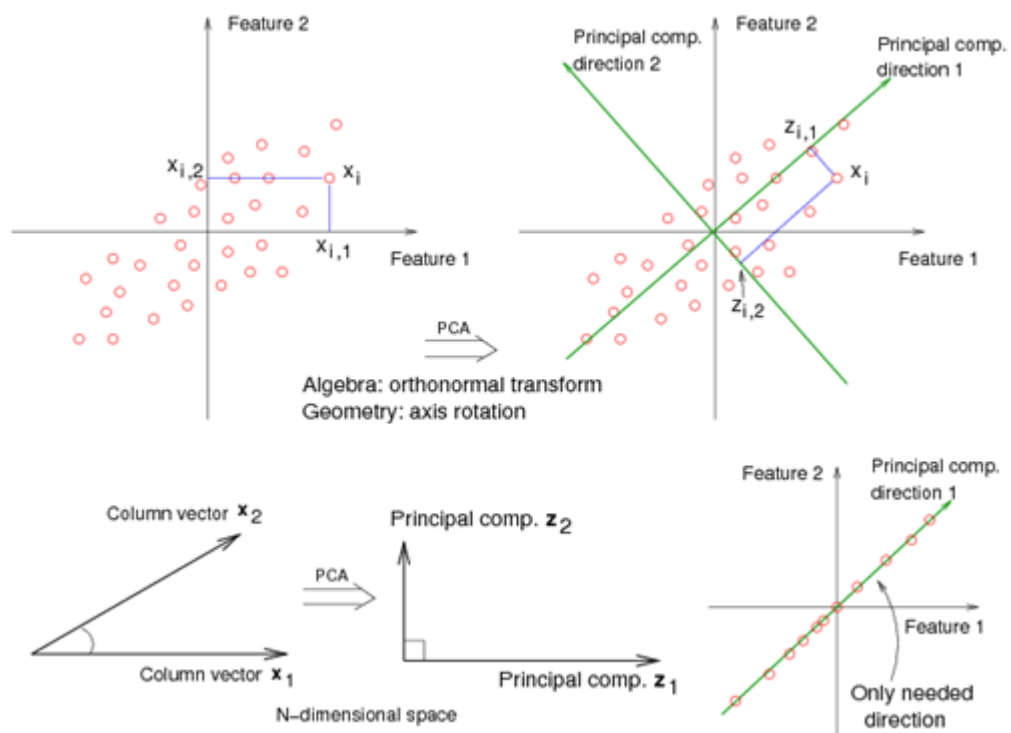


Рисунок 2.1 - Зменшення розмірності за допомогою аналізу головних КОМПОНЕНТ

2. Обчислення середнього обличчя. Розраховується середнє значення інтенсивності пікселів для всіх зображень тренувальної вибірки. Це "середнє обличчя" потім віднімається від кожного зображення для центрування даних.

3. Обчислення коваріаційної матриці. Будується коваріаційна матриця центрованих векторів зображень.

4. Знаходження власних векторів та власних значень. Обчислюються власні вектори та відповідні власні значення коваріаційної матриці. Власні вектори, які мають ту ж розмірність, що й вихідні зображення, називаються "власними обличчями" (Eigenfaces).

5. Відбір головних компонент. Власні вектори сортуються у порядку спадання відповідних власних значень. Відбираються  $k$  власних векторів з найбільшими власними значеннями, які формують базис підпростору обличчя. Вибір  $k$  визначає розмірність простору ознак.

6. Представлення у просторі обличчя. Кожне зображення обличчя (тренувальне або тестове) проектується на цей  $k$ -вимірний підпростір,

отримуючи вектор вагових коефіцієнтів, який є його представленням у просторі Eigenfaces. Розпізнавання здійснюється шляхом порівняння цих векторів ознак.

## 2.2. Опис методу Fisherfaces

Метод Fisherfaces поєднує техніки PCA та лінійного дискримінантного аналізу Фішера (Fisher's Linear Discriminant - FLD), який по суті є LDA (Linear Discriminant Analysis). На відміну від Eigenfaces (PCA), який шукає напрямки максимальної дисперсії даних, Fisherfaces прагне знайти проекцію, що максимізує співвідношення міжкласової дисперсії (між різними особами) до внутрішньокласової дисперсії (варіації одного обличчя).

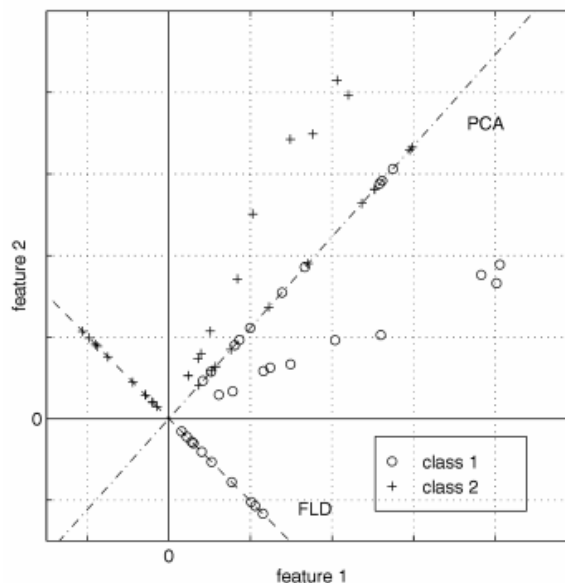


Рисунок 2.2 - Аналіз головних компонент і лінійний дискримінант Фішера

Спочатку PCA може використовуватись для початкового зменшення розмірності простору (наприклад, з  $N$  до  $N-s$ , де  $N$  - кількість зображень,  $s$  - кількість класів/осіб). Далі застосовується LDA для подальшого зменшення розмірності до щонайбільше  $s-1$ . Оскільки PCA не враховує приналежність

до класів, він може відкинути деякі дані, що є важливими для дискримінації класів. LDA, навпаки, явно оптимізує відокремлюваність класів.

LDA шукає лінійні комбінації ознак, що найкраще розділяють класи. Цей метод максимізує співвідношення дисперсії між класами до дисперсії всередині класів. PCA може проектувати дані так, що класи залишаються перекритими, тоді як LDA знаходить проекцію, де класи більш чітко розділені, що демонструє максимізацію співвідношення розсіювання класів.

Метод Fisherfaces є менш чутливим до варіацій освітлення та деяких змін ракурсу порівняно з методом Eigenfaces, оскільки він спеціально навчається розрізняти різні особи, мінімізуючи при цьому варіації всередині однієї особи.

### 2.3. Дослідження методу Local Binary Patterns Histograms (LBPН)

Метод LBPН є ефективним підходом, що базується на аналізі локальної текстури обличчя та використанні гістограм як векторів ознак.

Процес включає наступні кроки:

- Розбиття зображення. Вхідне зображення обличчя ділиться на сітку невеликих, можливо, перекриваючихся регіонів.

- Обчислення LBP-шаблонів. Для кожного пікселя (або для пікселів на рівномірній сітці) у кожному регіоні обчислюється локальний бінарний шаблон. Це робиться шляхом порівняння інтенсивності центрального пікселя з інтенсивністю його  $P$  сусідів, розташованих на радіусі  $R$  навколо центру. Якщо інтенсивність сусіда більша або дорівнює інтенсивності центру, йому присвоюється значення '1', інакше – '0'. Ці  $P$  бінарних значень, розташовані в певному порядку (наприклад, за годинниковою стрілкою), формують  $P$ -бінарний код. Цей бінарний код зазвичай перетворюється на десяткове число.

- Побудова локальних гістограм. Для кожного регіону обчислюється гістограма розподілу отриманих десяткових LBP-значень.

					БР.ІП – 13.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36

- Конкатенація гістограм. Локальні гістограми з усіх регіонів об'єднуються (конкатенуються) в один глобальний вектор ознак, який представляє все обличчя.

- Розпізнавання. При ідентифікації тестового зображення для нього обчислюється аналогічний глобальний вектор ознак (гістограма). Потім обчислюється відстань (міра схожості) між цією гістограмою та гістограмами всіх тренувальних зображень або моделей. Найменша відстань (найбільша схожість) вказує на ідентифіковану особу.

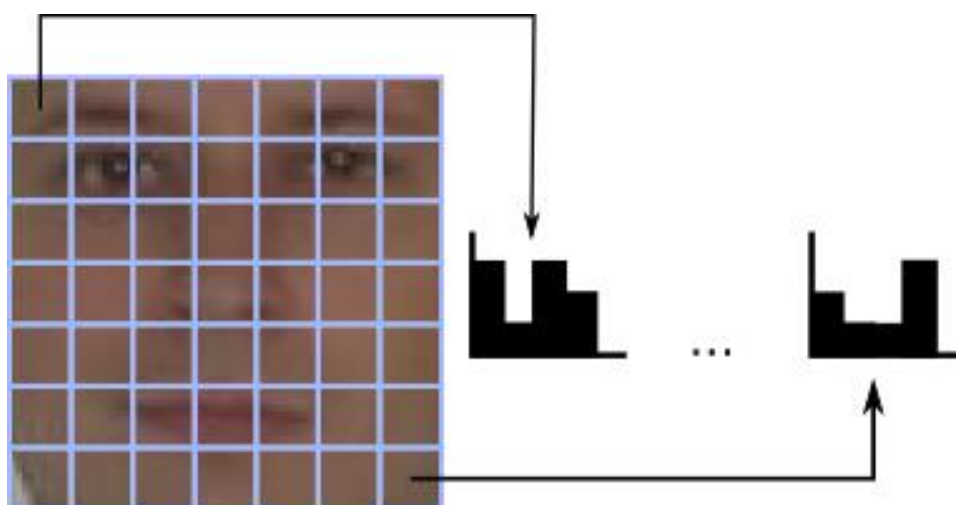


Рисунок 2.3 – Розбиття зображення на просторові гістограми

Метод LBPН забезпечує послідовно точні результати розпізнавання, поєднуючи прийнятну швидкість та хорошу дискримінаційну здатність. Метод є досить стійким до варіацій виразів обличчя, монотонних змін освітлення та деяких обертань зображення.

Існують різні варіанти LBPН, зокрема "оригінальний" LBP (LBP8,1), що використовує 8 сусідів на радіусі 1. Обчислювальна складність залежить від роздільної здатності зображення та кількості оброблюваних пікселів. Наприклад, для зображення 200x200 пікселів обчислення шаблону для кожного пікселя потребує порівняння з 8 сусідами.

## 2.4. Оригінальні та розширені локальні бінарні шаблонні гістограми

Оригінальний локальний бінарний шаблон (olbp) використовує вісім сусідів і використовує радіус один. Це означає, що кожен піксель порівнюється з кожним із восьми сусідніх пікселів, які його торкаються. Єдиним недоліком цього методу є те, що його обчислювальний час може бути великим, залежно від кількості пікселів на зображенні. Наприклад, зображення розміром  $200 \times 200$  пікселів матиме 40 000 пікселів, які потрібно порівняти з вісьмома сусідами.

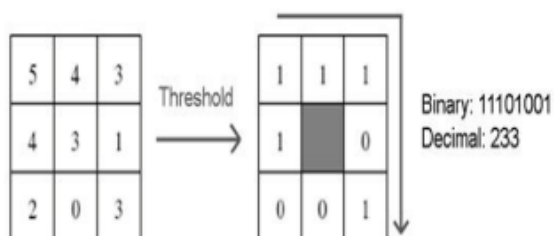


Рисунок 2.4 - Отримання бінарних значень порівнянням із сусідами

Нижче наведено код OpenCV для обчислення бінарних значень восьми сусідів залежно від значення сірого кольору.

Лістинг 2.1. Код оригінального локального бінарного шаблону

```
for(int i=1;i<src.rows-1;i++) {
    for(int j=1;j<src.cols-1;j++) {
        _Tp center = src.at<_Tp>(i,j);
        unsigned char code = 0;
        code |= (src.at<_Tp>(i-1,j-1) >= center) << 7;
        code |= (src.at<_Tp>(i-1,j) >= center) << 6;
        code |= (src.at<_Tp>(i-1,j+1) >= center) << 5;
        code |= (src.at<_Tp>(i, j+1) >= center) << 4;
        code |= (src.at<_Tp>(i+1,j+1) >= center) << 3;
        code |= (src.at<_Tp>(i+1,j) >= center) << 2;
        code |= (src.at<_Tp>(i+1,j-1) >= center) << 1;
        code |= (src.at<_Tp>(i,j-1) >= center) << 0;
        dst.at<unsigned char>(i-1,j-1) = code;
    }
}
```

Розширений локальний бінарний шаблон (elbp) трохи відрізняється від оригінального тим, що дозволяє користувачеві вказати кількість сусідів для використання, а також радіус.

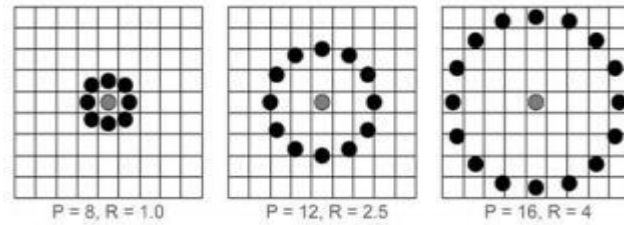


Рисунок 2.5 - Обчислення сусідів для розширеного локального бінарного шаблону

Коли стандартна кількість сусідів і радіус змінюються, необхідно використовувати нову формулу для обчислення координат  $x$  і  $y$  сусідніх пікселів. Білінійна інтерполяція потрібна в тому випадку, якщо координати  $x$  і  $y$  сусідніх пікселів не знаходяться в центрі пікселя.

$$LBP_{p,r} = \sum_{n=0}^{r-1} (x_{r,n} - x_{0,0}) 2^n, s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

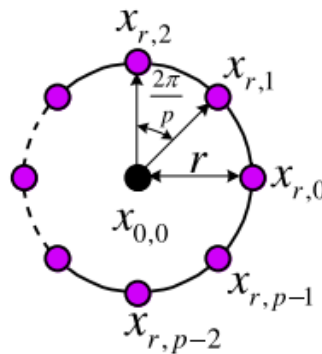


Рисунок 2.6 - X та Y координати розширеного локального бінарного шаблону

Формула має такий вигляд:

$$\left( -r \sin \left( \frac{2\pi n}{p} \right), r \cos \left( \frac{2\pi n}{p} \right) \right)$$

де  $r$  — радіус, а  $p$  — кількість сусідів.

Нижче наведено код OpenCV для обчислення бінарних значень заданого набору сусідів.

Лістинг 2.2. Вихідний код розширеного локального бінарного шаблону

```
for(int n=0; n<neighbors; n++) {
    // sample points
    float x = static_cast<float>(-radius *
sin(2.0*CV_PI*n/static_cast<float>(neighbors)));
    float y = static_cast<float>(radius *
cos(2.0*CV_PI*n/static_cast<float>(neighbors)));
    // relative indices
    int fx = static_cast<int>(floor(x));
    int fy = static_cast<int>(floor(y));
    int cx = static_cast<int>(ceil(x));
    int cy = static_cast<int>(ceil(y));
    // fractional part
    float ty = y - fy;
    float tx = x - fx;
    // set interpolation weights
    float w1 = (1 - tx) * (1 - ty);
    float w2 = tx * (1 - ty);
    float w3 = (1 - tx) * ty;
    float w4 = tx * ty;
    // iterate through your data
    for(int i=radius; i < src.rows-radius;i++) {
        for(int j=radius; j < src.cols-radius;j++) {
            // calculate interpolated value
            float t = static_cast<float>(w1*src.at<_Tp>(i+fy,j+fx) +
w2*src.at<_Tp>(i+fy,j+cx) + w3*src.at<_Tp>(i+cy,j+fx) + w4*src.at<_Tp>(i+cy,j+cx));
            // floating point precision, so check some machine-dependent epsilon
            dst.at<int>(i-radius,j-radius) += ((t > src.at<_Tp>(i,j)) || (std::abs(t
src.at<_Tp>(i,j)) < std::numeric_limits<float>::epsilon())) << n;
        }
    }
}
```

## 2.5. Представлення переваг і недоліків шаблонів

Кожен із трьох згаданих алгоритмів має свої переваги та недоліки. Існує багато різних алгоритмів для розпізнавання обличчя через те, що жоден один метод не є найбільш оптимальним в одній області, не жертвуючи перевагою в іншій області. У таблиці 2.1 наведено різні переваги та недоліки трьох перевірених алгоритмів.

					БР.ІП – 13.00.00.000 ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 2.1 - Переваги та недоліки алгоритмів

Алгоритм	Переваги	Недоліки
Eigenfaces	<ul style="list-style-type: none"> <li>- Агностичний до того, чи є об'єкт обличчям</li> <li>- Адекватно зменшує статистичну надмірність у представленні зображення обличчя</li> </ul>	<ul style="list-style-type: none"> <li>- Чутливий до світла</li> <li>- Чутливий до пози та виразів обличчя</li> <li>- Чутливий до незбігу пікселів</li> </ul>
Fisherfaces	<ul style="list-style-type: none"> <li>- Досягає більшої міжкласової розсіяності, тим самим полегшуючи класифікацію порівняно з Eigenfaces</li> <li>- Нечутливий до світла</li> </ul>	<ul style="list-style-type: none"> <li>- Якщо міжкласова розсіяність велика, то внутрікласова розсіяність також може бути великою</li> </ul>
Локальні бінарні шаблони	<ul style="list-style-type: none"> <li>- Стійкість до змін освітлення</li> <li>- Низька обчислювальна складність</li> <li>- Здатність кодувати дрібні деталі</li> <li>- Один із найкращих описових текстур дескрипторів</li> </ul>	<ul style="list-style-type: none"> <li>- Генерують довгі гістограми, що може сповільнити швидкість розпізнавання, особливо на великій навчальній базі даних</li> <li>- Можуть пропустити локальну структуру, не враховуючи ефект центрального пікселя</li> <li>- Мала просторова підтримка, у своїй базовій формі не може належним чином виявляти великомасштабні текстурні структури (olb)</li> </ul>

## 2.6. Розробка алгоритмів та представлення етапів обробки алгоритму LBRH для розпізнавання обличчя

Загальна послідовність етапів автоматичного розпізнавання обличчя, що формує типовий конвеєр обробки, проілюстровано в лістингу 2.2. Кожен алгоритм розпізнавання потребує етапу навчання для побудови внутрішнього представлення або "еталону" для кожного зареєстрованого суб'єкта, з яким будуть порівнюватися нові вхідні зображення.

У даній роботі детально розглянуто та реалізовано метод розпізнавання обличчя, заснований на гістограмах локальних бінарних шаблонів (Local Binary Patterns Histograms - LBRH). Специфіка етапу навчання для алгоритму LBRH представлена на рисунку 2.7. У межах навчального процесу алгоритму LBRH, два етапи, що вимагають найбільших обчислювальних ресурсів, це

обчислення розширених локальних бінарних шаблонів (Extended Local Binary Pattern - ELBP) та побудова просторових гістограм (Spatial Histogram).

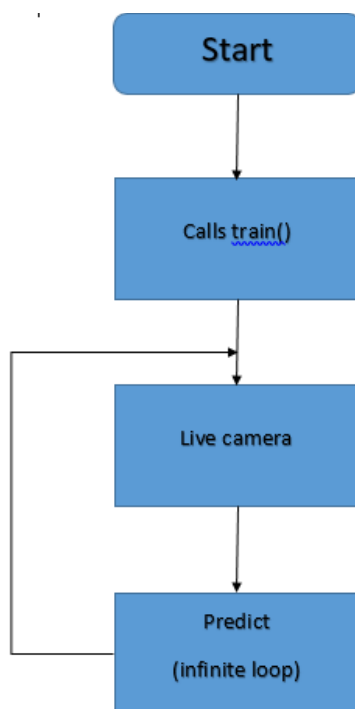


Рисунок 2.7 – Спрощений алгоритм розпізнавання обличчя

#### Етап навчання алгоритму LBPН

1. Обчислення ELBP. Для кожного центрального пікселя вхідного зображення обличчя обчислюються координати (X та Y) восьми сусідніх пікселів (як ілюстровано на рис. 2.6, що стосується обчислення сусідів для розширеного LBPН). Ці координати використовуються для вибірки значень інтенсивності сусідніх пікселів. Бінарний шаблон для центрального пікселя генерується шляхом порівняння значень сірого кольору восьми сусідів зі значенням сірого кольору центрального пікселя. Якщо значення інтенсивності сусіда більше або дорівнює значенню центрального пікселя, відповідному біту присвоюється значення '1'; в іншому випадку присвоюється значення '0'. Отримані вісім бінарних значень, розташовані послідовно (наприклад, за годинниковою стрілкою, див. рис. 2.4), формують бінарний код, який потім конвертується у десяткове число (як показано на

					БР.ІП – 13.00.00.000 ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

рис. 2.8). Ця операція виконується для кожного центрального пікселя зображення, формуючи зображення, де кожен піксель замінений його LBP-КОДОМ.

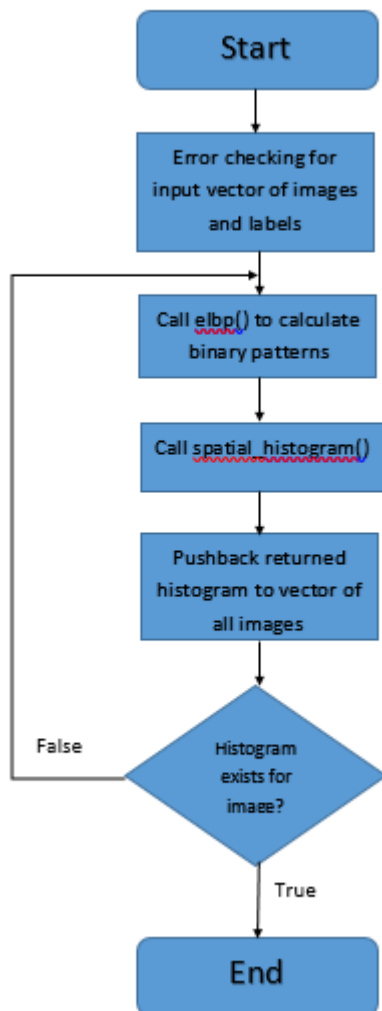


Рисунок 2.8 – Алгоритм навчання локального бінарного шаблону

2. Побудова просторових гістограм. Функція `spatial_histogram` обчислює гістограму розподілу LBP-кодів для кожного тренувального зображення. Спочатку зображення ділиться на фіксовану кількість прямокутних регіонів. Розміри регіонів визначаються на основі заздалегідь заданих параметрів ширини та висоти сітки поділу. Після визначення просторових меж кожного регіону, десяткові LBP-значення, отримані на попередньому етапі для кожного пікселя, групуються відповідно до регіону,

в якому знаходиться піксель. Далі для кожного окремого регіону обчислюється гістограма розподілу LBP-кодів. Ці гістограми можуть бути нормалізовані. Нарешті, локальні гістограми з усіх регіонів конкатенуються (об'єднуються послідовно) в одну велику глобальну гістограму, яка слугує вектором ознак для всього зображення обличчя (як представлено на рис. 2.9).

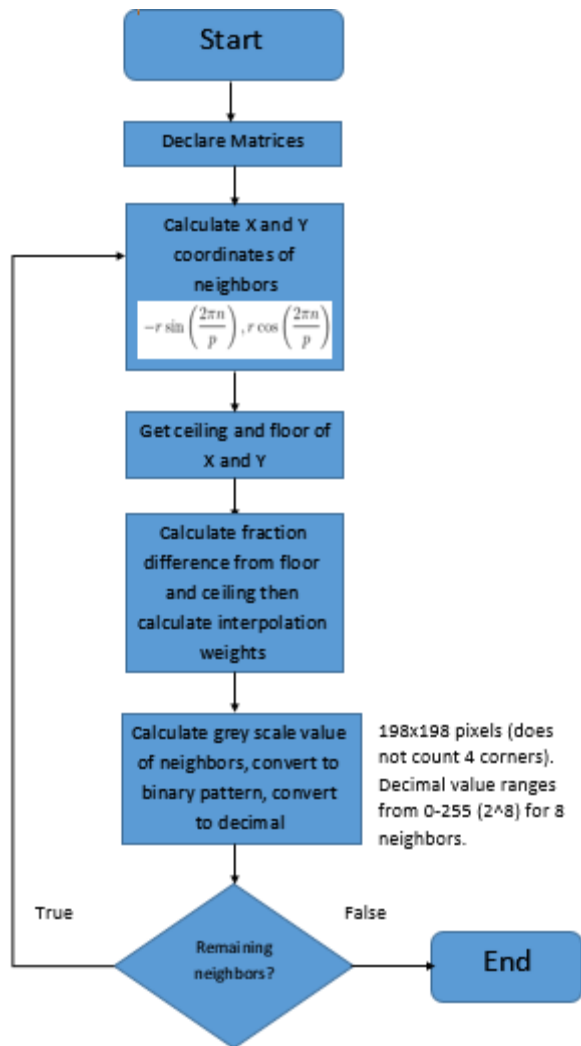


Рисунок 2.9 – Алгоритм розширеного локального бінарного шаблону

Ці гістограми зберігаються у вигляді окремих векторів для кожного тренувального зображення, формуючи базу "еталонів" для подальшого порівняння. Цей процес повторюється для всіх зображень у навчальному наборі, і отримані вектори гістограм зберігаються у відповідній структурі даних (наприклад, векторі матриць). На цьому завершується етап навчання.

Етап передбачення для нового (тестового) зображення обличчя включає виконання тих самих кроків обчислення ELBP та побудови просторової гістограми, що і на етапі навчання. Однак ці обчислення застосовуються лише до одного вхідного зображення (наприклад, кадру з "живого" відеопотоку), а не до всього навчального набору.

Отримана гістограма тестового зображення потім порівнюється з гістограмами всіх зображень, збережених у базі даних з навчального набору. Порівняння гістограм здійснюється за допомогою метрики відстані Хі-квадрат ( $\chi^2$ ), яка визначається наступним рівнянням:

$$\chi^2(H_1, H_2) = \sum_i \frac{(H_{1,i} - H_{2,i})^2}{H_{1,i} + H_{2,i}}$$

де  $H_1$  та  $H_2$  – гістограми, що порівнюються, а  $H_{1,i}$  та  $H_{2,i}$  – значення  $i$ -го біна в кожній гістограмі. У деяких варіантах використовується зважена версія, як у згаданому рівнянні (в оригіналі), що може мати вигляд:

$$\chi_w^2(H_1, H_2) = \sum_i \omega_i \frac{(H_{1,i} - H_{2,i})^2}{H_{1,i} + H_{2,i}}$$

де  $\omega_i$  – ваговий коефіцієнт для  $i$ -го біна або регіону гістограми.

Обчислюється відстань між гістограмою тестового зображення та гістограмою кожного зображення з навчального набору. Ці відстані послідовно порівнюються. Змінна `min_dist` зберігає мінімальну знайдену відстань. Початково `min_dist` встановлюється як відстань до першого зображення навчального набору. При порівнянні з кожним наступним зображенням, якщо обчислена відстань менша за поточне значення `min_dist`, `min_dist` оновлюється, і запам'ятовується ідентифікатор відповідного тренувального зображення. Якщо нова відстань більша, поточний `min_dist` та

відповідний ідентифікатор залишаються незмінними, і процес переходить до наступного зображення навчального набору.

Після того, як гістограма тестового зображення порівняна з гістограмами всіх зображень у навчальному наборі, ідентифікатор тренувального зображення, що відповідає мінімальній обчисленій відстані Хі-квадрат, визначається як результат передбачення (розпізнавання) особи на тестовому зображенні (як ілюстровано на рис. 2.10).

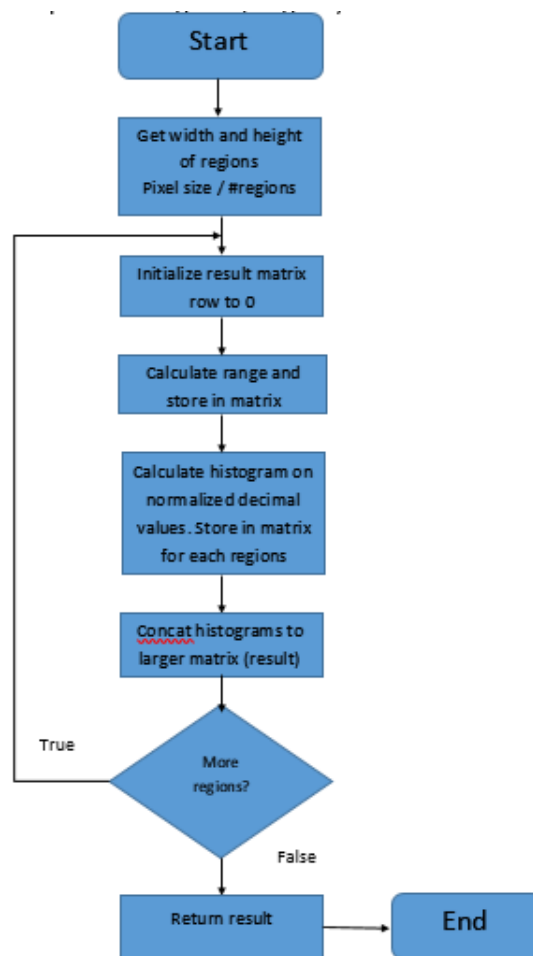


Рисунок 2.10 – Алгоритм просторової гістограми локальних бінарних шаблонів (оригінальний)

Рисунок 2.10 являє собою блок-схему алгоритму, який обчислює просторову гістограму, локальних бінарних шаблонів (Local Binary Patterns - LBP). Ось опис кроків алгоритму згідно з блок-схемою:

1. Start (Початок): Алгоритм розпочинається.

2. Get width and height of regions. Pixel size / #regions (Отримати ширину та висоту регіонів. Розмір пікселя / Кількість регіонів): На цьому кроці визначаються розміри підрегіонів, на які буде розділено зображення. Зображення ділиться на сітку, і розмір кожного регіону обчислюється на основі загального розміру зображення та бажаної кількості регіонів.

3. Initialize result matrix. Set to 0 (Ініціалізувати результуючу матрицю. Встановити в 0): Створюється матриця або структура даних для зберігання кінцевого результату (об'єднаних гістограм). Вона ініціалізується нулями.

4. Calculate range and store in matrix (Обчислити діапазон та зберегти в матриці): Цей крок дещо неоднозначний без додаткового контексту. У стандартних просторових гістограмах LBP прямого кроку "обчислення діапазону" зазвичай немає. Це стосується обчислення діапазону значень LBP в кожному регіоні або іншої статистичної міри, специфічної для цієї реалізації. Цей діапазон зберігається в матриці.

5. Calculate histogram on normalized decimal values. Store in matrix for each regions (Обчислити гістограму на нормалізованих десяткових значеннях. Зберегти в матриці для кожного регіону): Це ключовий крок.

- Для кожного пікселя в кожному регіоні зображення обчислюються локальні бінарні шаблони (LBP).

- Отримані бінарні коди, ймовірно, перетворюються на десяткові значення.

- Вказується "нормалізовані десяткові значення", що може означати нормалізацію LBP-значень або нормалізацію отриманої гістограми.

- Для кожного окремого регіону обчислюється гістограма цих (нормалізованих) значень.

- Ці гістограми для кожного регіону зберігаються в ініціалізованій результуючій матриці.

					БР.ІП – 13.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		47

6. Concat histograms to larger matrix (result) (Об'єднати гістограми у більшу матрицю (результат)): Гістограми, обчислені для кожного окремого регіону, об'єднуються (конкатенуються) в один великий вектор ознак. Ця об'єднана гістограма представляє просторовий розподіл LBP-шаблонів по всьому зображенню. Це і є сутність "просторової гістограми".

7. More regions? (Decision) (Більше регіонів? (Рішення)): Перевіряється, чи залишилися регіони для обробки.

- True (Так): Якщо є ще необроблені регіони, процес повертається до кроку обчислення гістограми для наступного регіону.

- False (Ні): Якщо всі регіони оброблені, алгоритм переходить до наступного кроку.

8. Return result (Повернути результат): Кінцева об'єднана гістограма (результуюча матриця) повертається як вихід алгоритму.

9. End (Кінець): Виконання алгоритму завершується.

Таким чином, алгоритм розділяє зображення на регіони, обчислює гістограму локальних характеристик (LBP) для кожного регіону, а потім об'єднує ці регіональні гістограми для формування кінцевого вектора ознак, який враховує просторове розташування шаблонів. Крок з обчисленням та збереженням "діапазону" є специфічним для цієї схеми і відсутній у стандартних описах просторових гістограм LBP.

На рисунку 2.11 надана блок-схема описує алгоритм передбачення або класифікації зображення на основі порівняння його просторової гістограми (локальних бінарних шаблонів - LBP) з набором вже відомих гістограм. Ось детальний опис кроків:

1. Start (Початок): Алгоритм розпочинає свою роботу.

2. Vector empty? (Перевірка на порожність вектора?): На цьому кроці перевіряється, чи є порожнім вектор `_histogram`. Цей вектор містить попередньо обчислені просторові гістограми для різних відомих класів або зразків зображень (тобто, "навчальні" гістограми).

					БР.ІП – 13.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		48

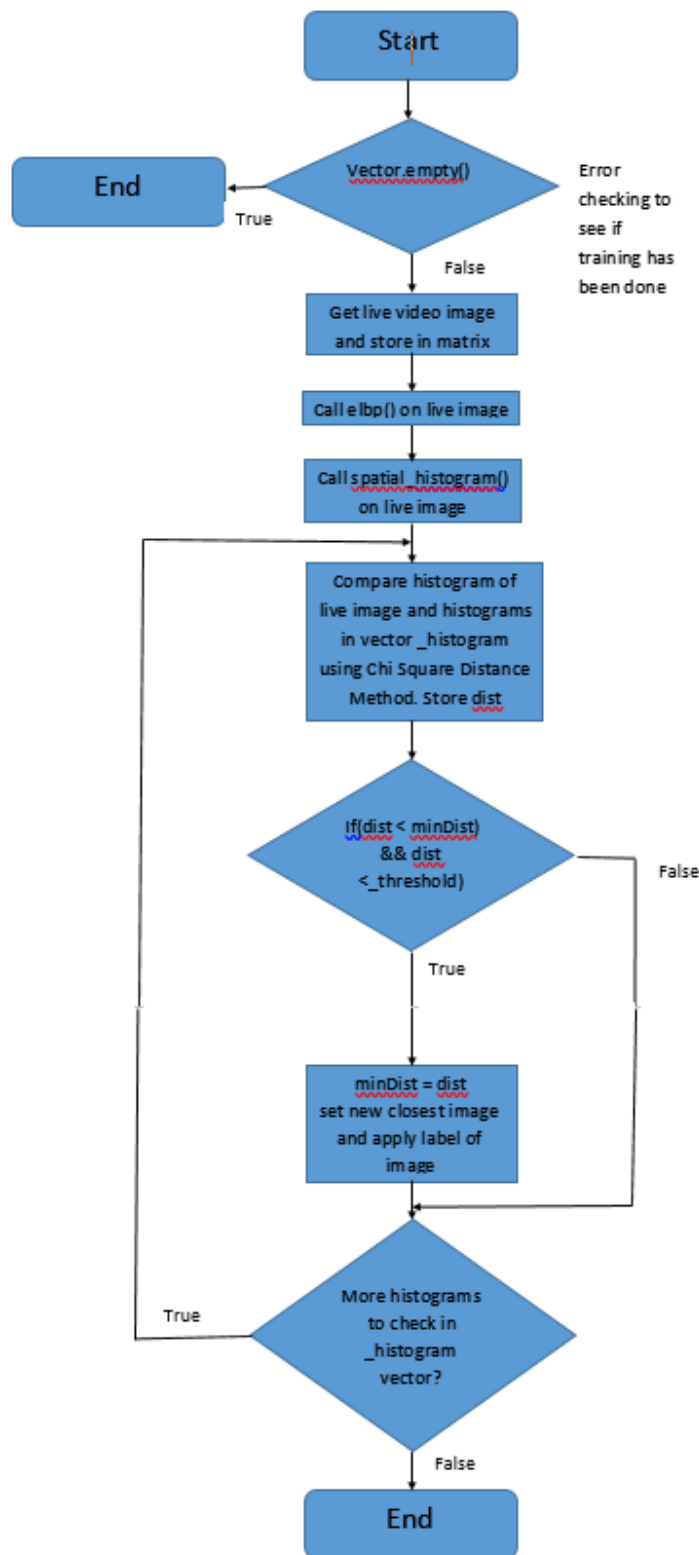


Рисунок 2.11 – Алгоритм передбачення локального бінарного шаблону

- Якщо вектор порожній (True), це означає, що немає даних для порівняння (не було етапу навчання). Блок-схема веде до End (Кінець) з приміткою "Error checking to see if training has been done" (Перевірка

помилки, щоб побачити, чи було проведено навчання). Алгоритм не може виконувати передбачення без еталонних гістограм.

- Якщо вектор не порожній (False), алгоритм продовжує виконання.

3. Get live video image and store in matrix (Отримати живе відеозображення та зберегти в матриці): Алгоритм отримує вхідне зображення (назване "живе відеозображення", що може вказувати на обробку кадрів відео або просто на вхідне зображення для класифікації) і зберігає його у вигляді матриці пікселів.

4. Call lbp() on live image (Викликати lbp() для живого зображення): Для отриманого вхідного зображення обчислюються локальні бінарні шаблони (LBP). Це призводить до створення LBP-зображення, де кожен піксель кодує локальну текстурну інформацію.

5. Call spatial\_histogram() on live image (Викликати spatial\_histogram() для живого зображення): Для LBP-зображення (або для оригінального зображення, а функція сама обчислює LBP) обчислюється просторова гістограма. Цей крок використовує алгоритм, подібний до описаного у попередньому запиті, де зображення ділиться на регіони, і для кожного регіону обчислюється гістограма LBP-значень, а потім ці гістограми об'єднуються в один вектор ознак.

6. Compare histogram of live image and histograms in vector \_histogram using Chi Square Distance Method. Store dist (Порівняти гістограму живого зображення та гістограми у векторі \_histogram, використовуючи метод відстані Хі-квадрат. Зберегти dist): Обчислена просторова гістограма вхідного зображення порівнюється з кожною з еталонних гістограм, що зберігаються у векторі \_histogram. Для порівняння використовується метод відстані Хі-квадрат (Chi-Square Distance), який є поширеною метрикою для вимірювання відмінностей між гістограмами. Обчислене значення відстані зберігається у змінній dist.

					БР.ІП – 13.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		50

7. If (dist < minDist) && (dist <= threshold) (Якщо (dist < minDist) && (dist <= поріг)): На цьому кроці порівнюється обчислена відстань dist з поточною мінімальною знайденою відстанню minDist (яка ініціалізується до дуже великого значення перед циклом порівняння) та з заданим порогом threshold.

- Якщо умова істинна (True) (тобто, поточна відстань менша за мінімальну знайдену раніше і менша або дорівнює порогу), це означає, що знайдено кращий (ближчий) збіг, який є прийнятним.

- Якщо умова хибна (False), поточна еталонна гістограма не є кращим збігом або відстань надто велика.

8. minDist = dist. set new closest\_image and apply label of image (minDist = dist. встановити нове closest\_image та застосувати мітку зображення): (Виконується, якщо умова в кроці 7 істинна). Змінна minDist оновлюється значенням dist. Ідентифікатор або мітка, пов'язана з поточною еталонною гістограмою (яка є найкращим збігом на даний момент), зберігається, ймовірно, оновлюючи змінну closest\_image. "apply label of image" означає, що мітка, відповідна цьому найкращому збігу, призначається як передбачена мітка для вхідного "живого" зображення.

9. More histograms to check in \_histogram vector? (Більше гістограм для перевірки у векторі \_histogram?): Алгоритм перевіряє, чи залишилися ще еталонні гістограми у векторі \_histogram для порівняння.

- Якщо так (True), процес повертається до кроку порівняння гістограми вхідного зображення з наступною еталонною гістограмою.

- Якщо ні (False), всі еталонні гістограми були перевірені, і алгоритм переходить до наступного кроку.

10. End (Кінець): Виконання алгоритму завершується. Передбачена мітка для вхідного зображення визначається на основі найкращого знайденого збігу (змінна closest\_image або застосована мітка).

					БР.ІП – 13.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		51

Загалом, цей алгоритм реалізує метод класифікації (передбачення) зображень, де вхідне зображення порівнюється з набором відомих зразків на основі схожості їхніх просторових гістограм LBP, використовуючи відстань Хі-квадрат як міру схожості. Зображенню присвоюється мітка того зразка, гістограма якого виявилася найбільш схожою (в межах певного порогу).

					БР.ІП – 13.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		52

## РОЗДІЛ 3. ІМПЛЕМЕНТАЦІЯ РІШЕННЯ ДЛЯ ОПТИМІЗАЦІЇ ТА НАЛАШТУВАННЯ МЕТОДІВ РОЗПІЗНАВАННЯ ОБРАЗІВ

### 3.1. Процедура інтеграції бібліотеки OpenCV із середовищем розробки Visual Studio за допомогою CMake

В даному підрозділі описується послідовність дій, необхідних для налаштування та компіляції проєктів, що використовують бібліотеку комп'ютерного зору OpenCV, у інтегрованому середовищі розробки Microsoft Visual Studio. Процес передбачає використання системи збірки CMake для генерації файлів проєкту.

#### 1. Інсталяція необхідного програмного забезпечення.

Необхідно інсталювати відповідну версію середовища розробки Microsoft Visual Studio, сумісну з вимогами проєкту. Слід зазначити, що вибір версії Visual Studio (наприклад, 2010, 2012, 2015) впливає на процес конфігурації за допомогою CMake, оскільки різні версії можуть використовувати несумісні налаштування компіляції або версії бібліотек. Це може вимагати специфічних модифікацій під час генерації проєкту. Актуальна версія Visual Studio доступна для завантаження з офіційного веб-сайту: <https://www.visualstudio.com/downloads/>.

Отримання вихідних файлів бібліотеки OpenCV здійснюється шляхом завантаження відповідного дистрибутиву з офіційного веб-сайту проєкту: <http://opencv.org/releases.html>.

Необхідно завантажити та інсталювати систему збірки CMake. Цей інструмент буде використовуватись для конфігурації вихідних файлів OpenCV та генерації файлів проєкту для Visual Studio. Дистрибутив CMake доступний для завантаження за посиланням: <https://cmake.org/download/>.

#### 2. Підготовка вихідних файлів OpenCV.

					БР.ІП – 13.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		53

Перед початком процесу генерації проекту за допомогою CMake важливо переконатися у наявності всіх необхідних файлів прикладів у відповідних директоріях. Зокрема, файли `facerec_eigenfaces.cpp`, `facerec_fisherfaces.cpp`, `facerec_lbph.cpp`, `facerec_save_load.cpp` та `facerec_video.cpp`, які є частиною прикладів розпізнавання облич, можуть бути відсутні у стандартній директорії `samples/cpp` у деяких версіях дистрибутиву OpenCV. Відсутність цих файлів у відповідному місці призведе до некоректної генерації виконуваних файлів прикладів. Для забезпечення успішної збірки необхідно скопіювати зазначені п'ять файлів з директорії `C:\opencv\sources\modules\contrib\doc\facerec\src` до директорії `C:\opencv\sources\samples\cpp` (шляхи можуть варіюватися залежно від місця розпакування або встановлення OpenCV).

### 3. Генерація файлів проекту за допомогою CMake.

Після підготовки вихідних файлів бібліотеки OpenCV можна перейти до генерації файлів проекту для Visual Studio за допомогою CMake.

Запустіть графічний інтерфейс програми CMake.

У полі *Where is the source code:* вкажіть шлях до директорії з вихідними файлами OpenCV (наприклад, `C:/opencv/sources`).

У полі *Where to build the binaries:* вкажіть шлях до директорії, де будуть збережені згенеровані файли проекту Visual Studio (рекомендовано створити окрему директорію, наприклад, `C:/opencv/build`).

Натисніть кнопку *Configure*. CMake проаналізує вихідні файли та систему. Вам буде запропоновано обрати генератор для проекту. Оскільки метою є інтеграція з Visual Studio, необхідно обрати відповідне нативне середовище (наприклад, "Visual Studio 10 2010" для Visual Studio 2010). Вибір генератора має відповідати інсталюваній версії Visual Studio.

Після завершення конфігурації можуть з'явитися додаткові параметри, які можна налаштувати (специфічні параметри конфігурації, що виявились

					БР.ІП – 13.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		54

оптимальними для даного проєкту, можуть бути представлені на рисунках 3.1 – 3.4).

Натисніть кнопку Generate. CMake згенерує файли проєкту Visual Studio у директорії, вказаній у полі Where to build the binaries:.

Після успішної генерації ви отримаєте файл рішення Visual Studio (зазвичай з розширенням .sln), який можна відкрити у Visual Studio для подальшої компіляції бібліотеки та проєктів прикладів.

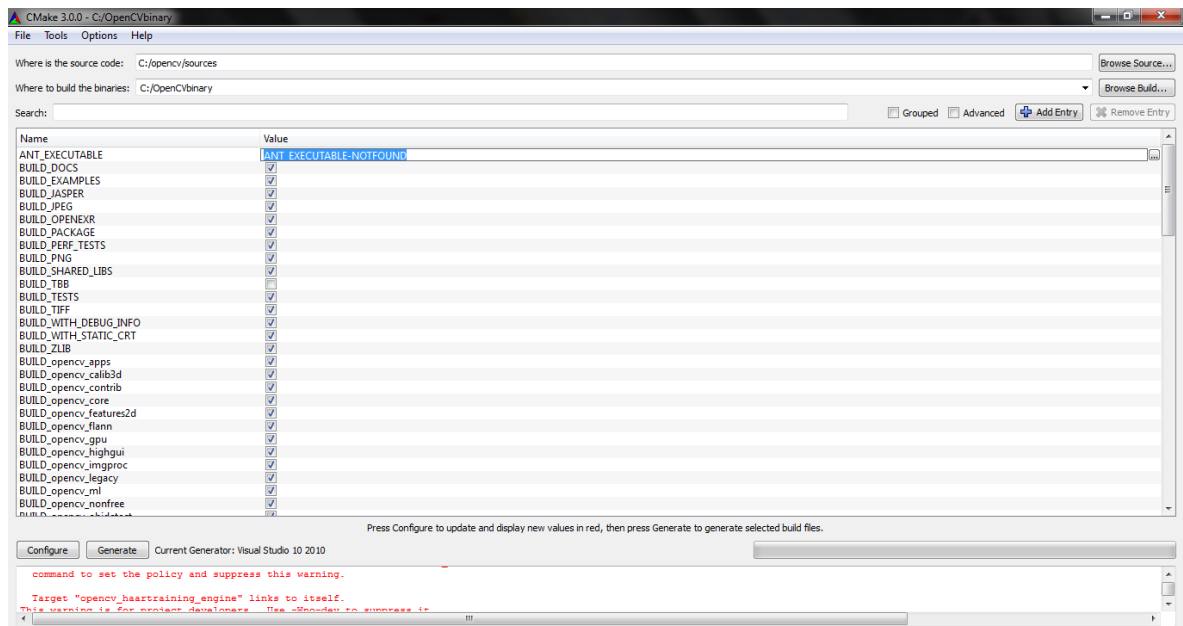


Рисунок 3.1 – Налаштування CMake (крок 1)

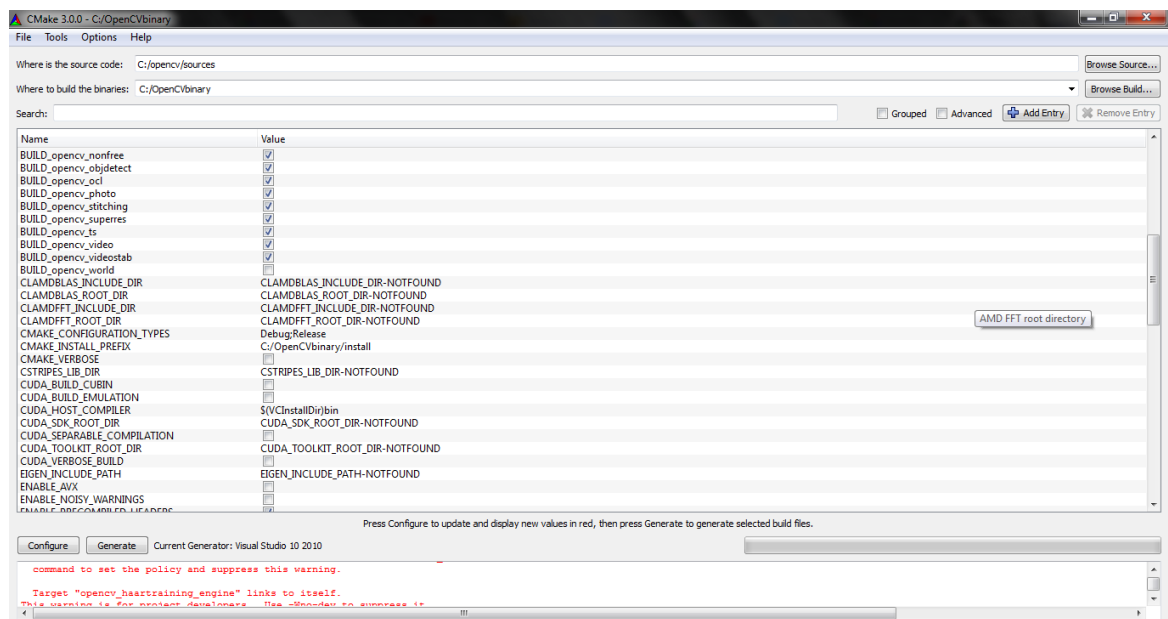


Рисунок 3.2 – Налаштування CMake (крок 2)

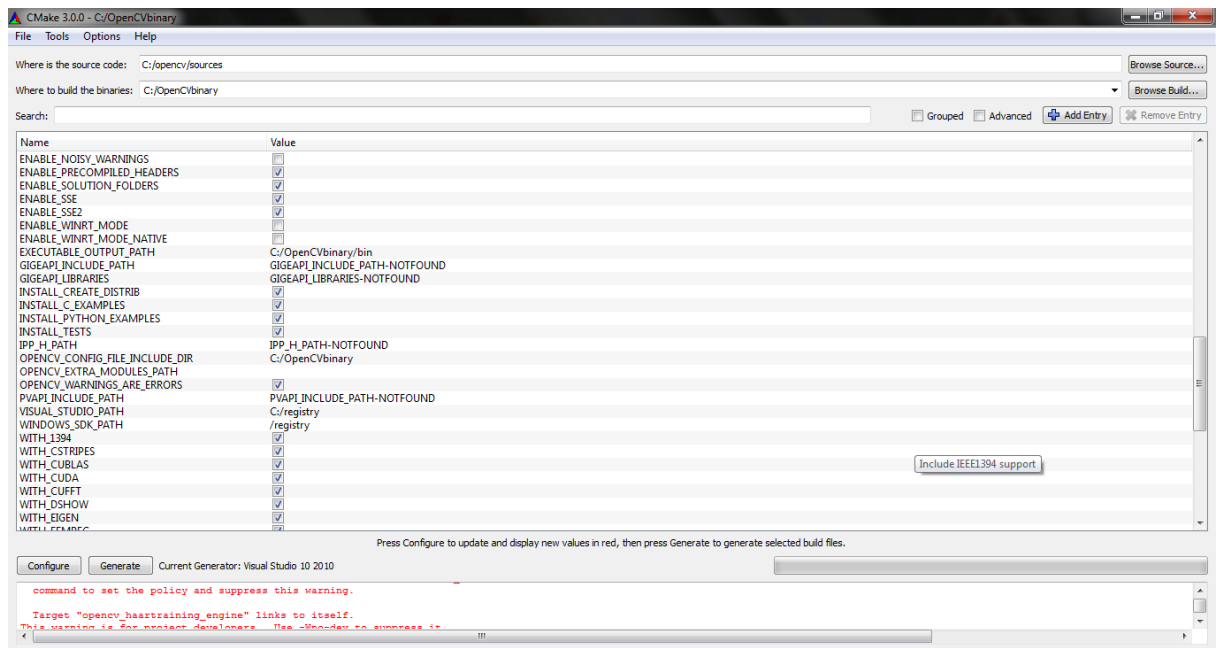


Рисунок 3.3 – Налаштування CMake (крок 3)

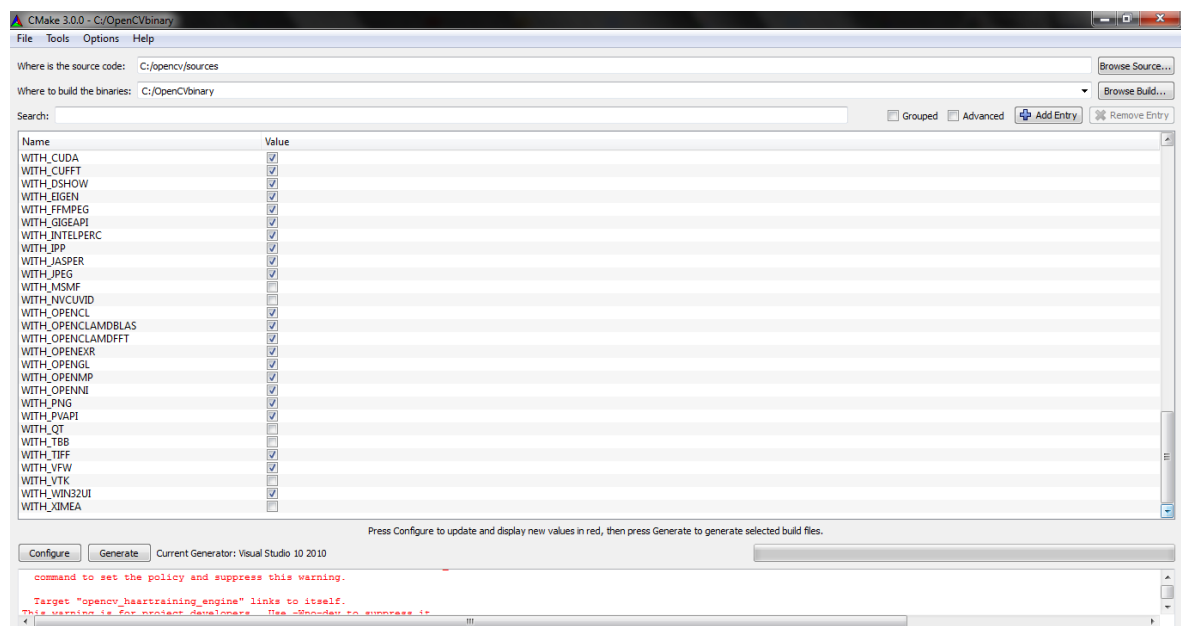


Рисунок 3.4 – Налаштування CMake (крок 4)

## 3.2. Процедура компіляції, підготовки даних та тестування прикладів бібліотеки OpenCV для розпізнавання облич

Цей розділ деталізує процедуру компіляції згенерованого проекту OpenCV у середовищі Visual Studio, методику підготовки набору даних для

навчання, а також кроки для виконання та тестування прикладу програми для розпізнавання облич.

## 1. Компіляція проекту у Visual Studio

Після успішної генерації файлів проекту за допомогою CMake, необхідно відкрити отриманий файл рішення (Solution) у Microsoft Visual Studio. Процес компіляції може вимагати внесення специфічних коригувань у налаштування проекту.

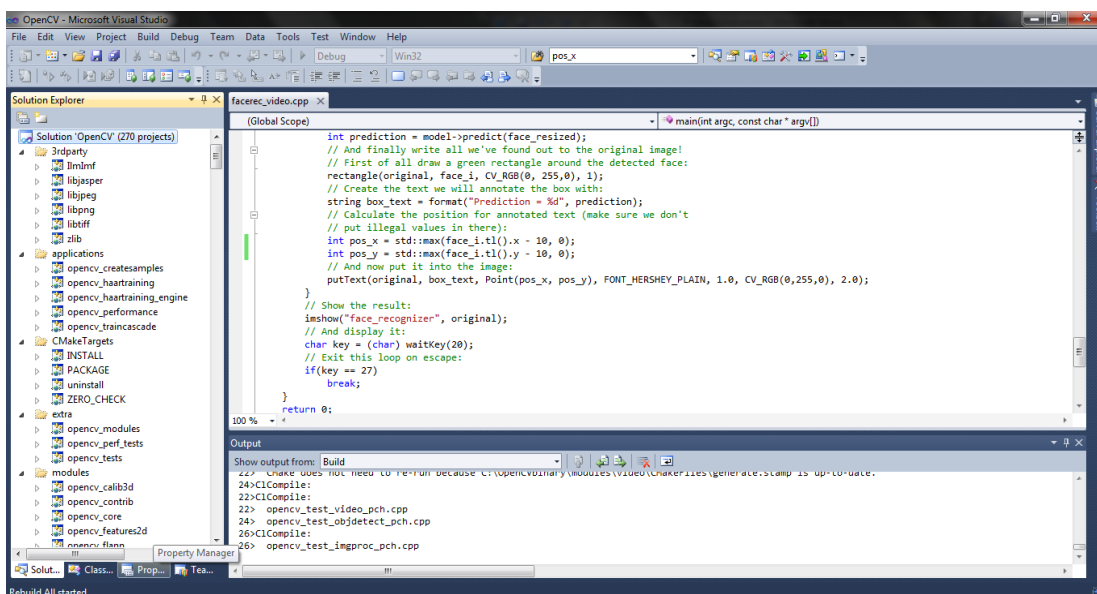


Рисунок 3.5 – Налаштування Visual Studios (крок 1)

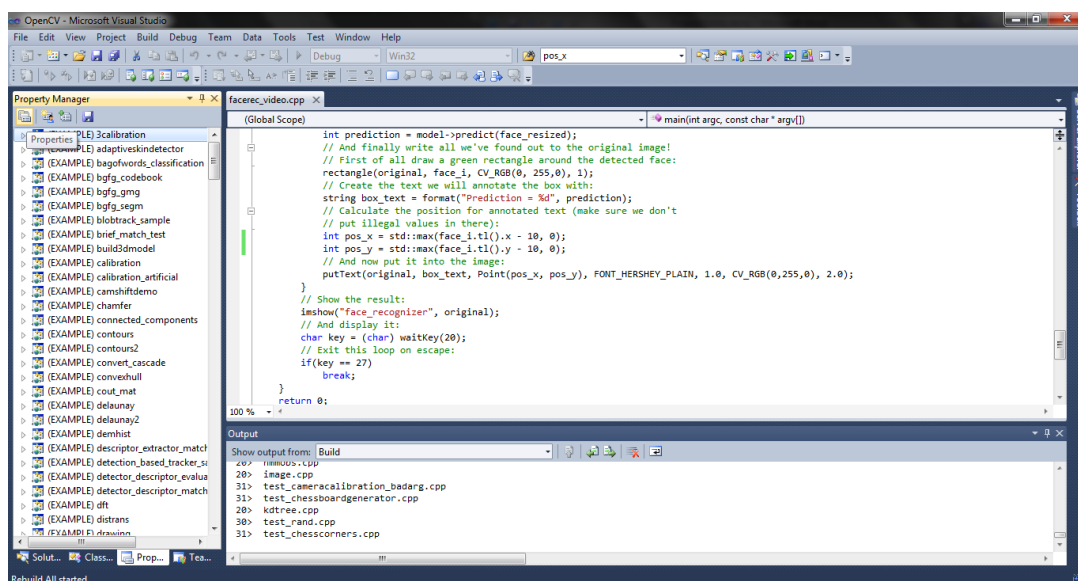


Рисунок 3.6 – Налаштування Visual Studios (крок 2)

Змн.	Арк.	№ докум.	Підпис	Дата

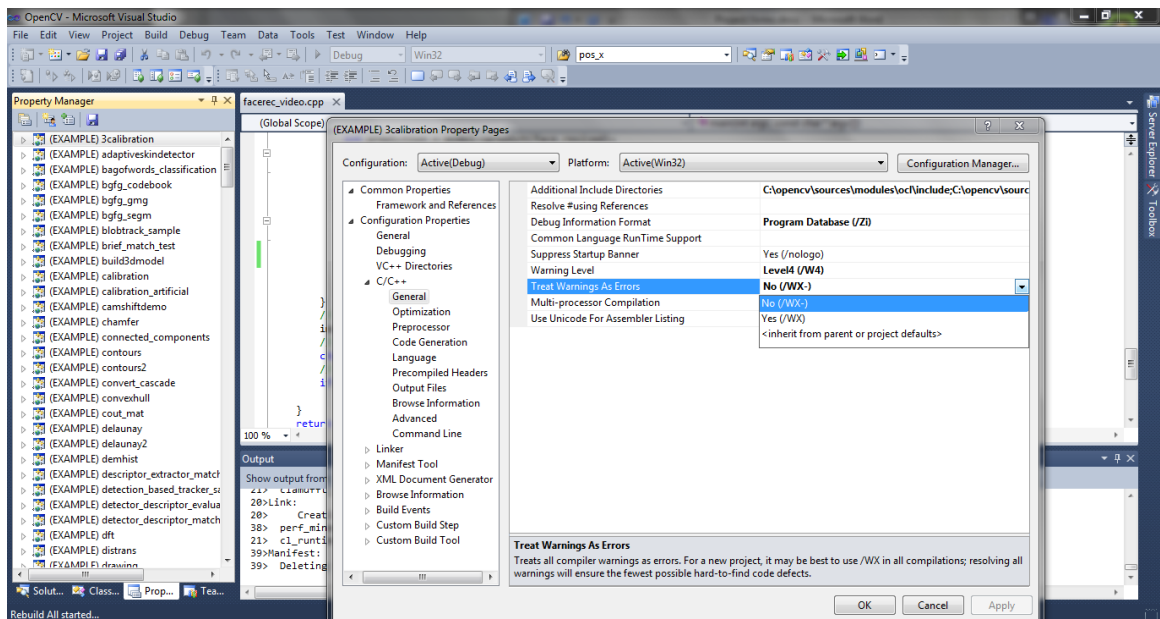


Рисунок 3.7 – Налаштування Visual Studios (крок 3)

Зокрема, для спрощення збірки було визначено необхідність деактивації опції "Тракувати попередження як помилки". Це налаштування можна змінити, виконавши наступні дії:

- Відкрийте вікно "Property Manager".
- Виберіть файл проєкту, що підлягає налаштуванню (наприклад, facerec\_video).
- Перейдіть до розділу Configuration Properties -> C/C++ -> General.
- Змініть значення параметра "Тракувати попередження як помилки" (Treat Warnings as Errors) на "Ні" (No).
- Поверніться до вікна "Solution Explorer".
- Виконайте процедуру побудови рішення (Build Solution).

Деталі конфігурації вікна властивостей проєкту відображено на рисунках 3.5 – 3.7.

## 2. Підготовка набору даних для навчання

Для ефективного навчання моделі розпізнавання обличчя необхідний підготовлений набір зображень. Методика підготовки даних включала наступні етапи:

									Арк.
									58
Змн.	Арк.	№ докум.	Підпис	Дата	БР.ІП – 13.00.00.000 ПЗ				

- Збір зображень. Використовувалися як зображення, отримані з відкритих джерел (наприклад, зображення відомих осіб), так і фотографії тестових суб'єктів, зроблені спеціально для цього дослідження.

- Передоробка зображень. Зображення були оброблені з метою ізоляції обличчя від фону шляхом обрізання. Для стандартизації розмірів зображень до 200x200 пікселів та їх квадратизації використовувалося програмне забезпечення IrfanView.

- Створення файлу міток. Для кожного зображення в наборі даних було створено відповідний запис у текстовому файлі з розширенням .txt. Кожен рядок у цьому файлі містить шлях до зображення та відповідну числову мітку, розділені символом крапки з комою. Приклад формату рядка: C:\Pictures\Project2\Brad\_Pitt\_01.jpg;0.

- Визначення обсягу даних. Для досягнення задовільних результатів навчання та точного розпізнавання особистостей рекомендовано використовувати достатню кількість зображень для кожного суб'єкта. Необхідний обсяг даних визначається емпірично шляхом проб і помилок. У рамках даного проєкту було використано 50 зображень на кожного суб'єкта.

### 3. Виконання та тестування програми розпізнавання облич

Після успішної компіляції та підготовки даних, додаток для розпізнавання обличчя готовий до виконання та тестування. Запуск програми здійснюється через командний рядок із передачею необхідних параметрів:

```
Bash
C:\OpenCVbinary5\bin\Debug\cpp-example-facerec_video.exe
C:\opencv\sources\data\haarcascades\haarcascade_frontalface_default.xml C:\celebrities2.txt 1
```

Структура команди та значення параметрів:

- C:\OpenCVbinary5\bin\Debug\cpp-example-facerec\_video.exe. Шлях до скомпільованого виконуваного файлу програми. Цей шлях залежить від

					БР.ІП – 13.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		59

директорії, обраної для збереження бінарних файлів під час конфігурації в CMake.

- C:\opencv\sources\data\haarcascades\haarcascade\_frontalface\_default.xml.

Шлях до файла XML, що містить дані каскадного класифікатора Хаара для виявлення фронтальних облич. Розташування цього файла залежить від директорії вихідних файлів OpenCV.

- C:\celebrities2.txt: Шлях до текстового файла з мітками та шляхами до навчальних зображень, створеного на етапі підготовки даних.

- 1: Індекс пристрою веб-камери, який буде використовуватися для захоплення відео.

Для спрощення багаторазового запуску команди її можна зберегти у скриптовому файлі та виконувати його замість ручного введення команди щоразу.

#### 4. Вибір алгоритму розпізнавання

За замовчуванням у прикладі програми може бути активований певний метод розпізнавання обличчя (наприклад, Fisherfaces).

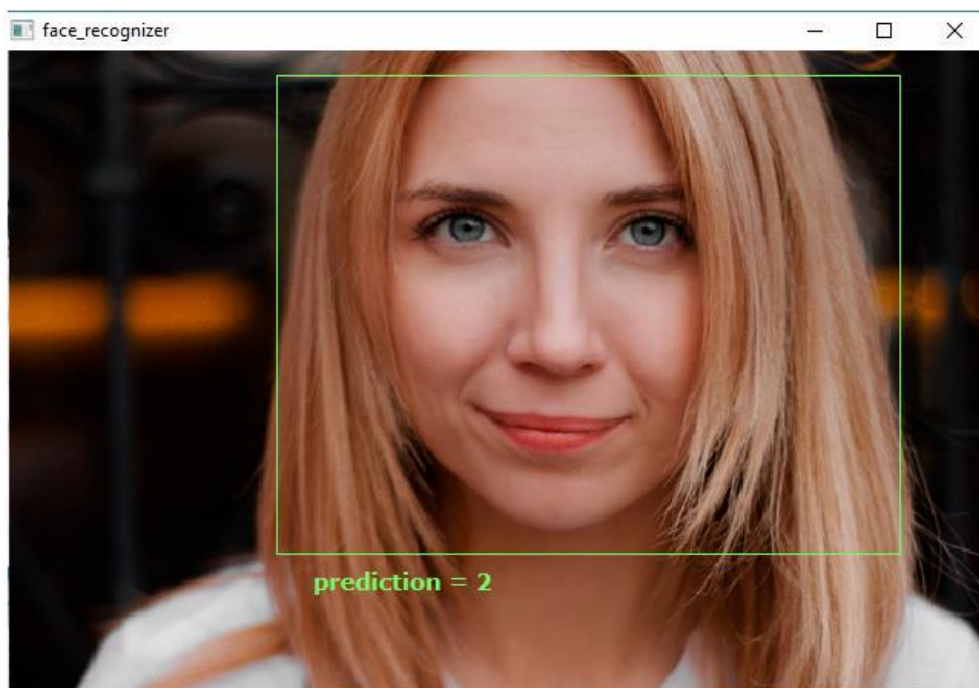


Рисунок 3.8 - Зображення для розпізнавання

Зміна використовуваного алгоритму здійснюється шляхом редагування вихідного коду файла `facerec_video.cpp` у Visual Studio. Необхідно розкоментувати рядок, що відповідає бажаному алгоритму, та закоментувати інші варіанти. Типові рядки для вибору алгоритму виглядають наступним чином (рис. 3.9)

```
// Створіть FaceRecognizer і навчіть його на заданих зображеннях:  
Ptr<FaceRecognizer> model = createFisherFaceRecognizer(); // Активний алгоритм Fisherfaces  
//Ptr<FaceRecognizer> model = createEigenFaceRecognizer(); // Закоментовано (Eigenfaces)  
//Ptr<FaceRecognizer> model = createLBPHFaceRecognizer(); // Закоментовано (LBPH)
```

Рисунок 3.9 – Код вибору алгоритму

Слід активувати лише один рядок з функцією `create*FaceRecognizer()`.

Детальна інструкція та додаткова інформація щодо використання модулів розпізнавання обличчя в OpenCV доступна за посиланням: [https://www.google.com/search?q=http://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec\\_tutorial.html%23introduction](https://www.google.com/search?q=http://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html%23introduction).

### 3.3. Оцінка продуктивності та порівняння методів розпізнавання обличчя

Для оцінки точності та ефективності трьох різних методів розпізнавання обличчя (ймовірно, Eigenfaces, Fisherfaces та LBPH, виходячи з попереднього контексту) було проведено серію експериментів з використанням стандартизованої методики тестування.

#### 1. Методика експерименту

Процедура тестування включала оцінку кожного методу шляхом проведення трьох випробувань для кожного користувача. Тестування проводилося за різних конфігурацій просторового розташування суб'єкта:

- Відстань від камери: близько 0.5 м, 0.76 м та 1 м.

- Положення обличчя (поза): Прямо (фронтальне), ліворуч (лівий профіль), праворуч (правий профіль).

- Кожен суб'єкт тестувався протягом фіксованого періоду часу - десяти секунд, перебуваючи у статичному положенні.

## 2. Метрики оцінювання

Оцінка продуктивності методів базувалася на наступних метриках:

- Очікуваний результат передбачення: Цілочисельна мітка, що відповідає ідентичності особи, яка проходить тестування (істина мітка).

- Фактичний результат передбачення: Цілочисельна мітка, прогнозована програмою розпізнавання обличчя для особи, що тестується.

- Коефіцієнт передбачення (%): Відсоток часу протягом десятисекундного випробування, коли фактичний результат передбачення збігався з очікуваним результатом передбачення.

- Рівень хибних позитивних спрацювань (%): Відсоток часу протягом десятисекундного випробування, коли програма помилково ідентифікувала або виявляла обличчя в областях зображення, де обличчя фактично не було присутнє (наприклад, на одязі, фоні тощо). Цей показник визначався кількістю секунд хибного виявлення, переведеною у відсотки від загальної тривалості випробування (наприклад, 8 секунд хибних спрацювань відповідає 80%).

Відповідність між цілочисельними мітками та іменами суб'єктів, використаних у навчальному наборі, наведено в таблиці 3.1.

Таблиця 3.1 - Відповідність між мітками та суб'єктами

Цілочисельна мітка	Ім'я людини
0	Бред Пітт (знаменитість)
1	Барбір Людмила (ведуча 1+1)
2	Марія Чумак (ведуча ICTV)
3	Анджеліна Джолі (знаменитість)
4	Алла Мазур (ведуча 1+1)
5	Богдан Бенюк (знаменитість)

### 3. Сценарій верифікації 1:1

Окрім загального тестування розпізнавання (ідентифікації), було проведено специфічний тест на верифікацію обличчя за схемою 1:1. У цьому сценарії використовувався один суб'єкт, дані якого були включені до навчального набору (Суб'єкт А), і проводилося тестування проти трьох різних суб'єктів.

Тестування із Суб'єктом А (навченим): Метою було визначити тривалість часу, протягом якого програма могла правильно верифікувати (ідентифікувати) Суб'єкта А протягом 10-секундного тестового періоду. Довший час правильної верифікації відповідав вищому коефіцієнту передбачення у цьому сценарії.

Тестування з іншими суб'єктами (ненавченими): Для двох інших суб'єктів, дані яких не були включені до навчального набору, верифікація як Суб'єкта А не повинна відбуватися. Метою було мінімізувати будь-який час помилкової верифікації (ідентифікації як Суб'єкта А) протягом тестового періоду. У цьому випадку менший час помилкової верифікації свідчив про вищий коефіцієнт передбачення (у сенсі правильного відхилення).

Важливо зазначити, що метод Fisherfaces не міг бути оцінений у цьому сценарії з одним навченим користувачем, оскільки він вимагає наявності даних щонайменше двох різних користувачів для навчання.

Результати порівняльного тестування за цими методиками представлені у таблиці 3.2.

У цих результатах показано, що Eigenfaces і оригінальний локальний бінарний шаблонний гістограма працювали схоже щодо як показника виявлення, так і показника хибних позитивних результатів. Під час тестування зміненого локального бінарного шаблонного гістограма порівняно з оригінальним показники прогнозування були схожі; однак показники хибних позитивних результатів зменшилися для зміненого методу (див. таблицю 3.3).

					БР.ІП – 13.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		63

Таблиця 3.2 – Аналіз Eigenfaces і Локального бінарного шаблону  
(Фрагмент таблиці)

Eigenfaces vs Local Binary Pattern Histogram						
Person 1:	Trial 1:	Position	Prediction Rate (%)		False Positive Rate (%)	
			Eigenfaces	LBPB	Eigenfaces	LBPB
		1.5 ft. Straight	100	100	20	5
		1.5 ft. Right Side	100	100	60	0
		1.5 ft. Left Side	40	100	30	0
		2.5 ft. Straight	100	100	70	100
		2.5 ft. Right Side	70	100	100	100
		2.5 ft. Left Side	10	100	80	70
		3.5 ft. Straight	100	100	20	80
		3.5 ft. Right Side	100	100	10	20
		3.5 ft. Left Side	10	100	70	80
	<b>Trial 2:</b>	1.5 ft. Straight	100	100	80	20
		1.5 ft. Right Side	95	100	10	5
		1.5 ft. Left Side	10	20	50	90
		2.5 ft. Straight	100	100	40	20
		2.5 ft. Right Side	100	100	60	80
		2.5 ft. Left Side	30	100	40	10
		3.5 ft. Straight	100	100	100	100
		3.5 ft. Right Side	90	100	10	20
		3.5 ft. Left Side	95	100	10	100
	<b>Trial 3:</b>	1.5 ft. Straight	100	100	100	0
		1.5 ft. Right Side	100	100	20	10
		1.5 ft. Left Side	0	80	10	80
		2.5 ft. Straight	100	100	100	70
		2.5 ft. Right Side	95	100	100	80
		2.5 ft. Left Side	0	100	100	10
		3.5 ft. Straight	100	100	10	30
		3.5 ft. Right Side	100	100	10	70
		3.5 ft. Left Side	80	100	20	50
<b>Person 2:</b>	<b>Trial 1:</b>	<b>Position</b>	<b>Eigenfaces</b>	<b>LBPB</b>	<b>Eigenfaces</b>	<b>LBPB</b>

Таблиця 3.3 - Оригінальний локальний бінарний шаблон проти  
 модифікованого локального бінарного шаблону  
 (Фрагмент таблиці)

Local Binary Pattern Histogram Original vs Local Binary Pattern Modified						
Person 1:	Trial 1:	Position	Prediction Rate (%)		False Positive Rate (%)	
			LBP Original	LBP Modified	LBP Original	LBP Modified
		1.5 ft. Straight	100	100	5	5
		1.5 ft. Right Side	100	100	0	0
		1.5 ft. Left Side	100	100	0	0
		2.5 ft. Straight	100	100	100	0
		2.5 ft. Right Side	100	100	100	80
		2.5 ft. Left Side	100	100	70	20
		3.5 ft. Straight	100	100	80	20
		3.5 ft. Right Side	100	100	20	20
		3.5 ft. Left Side	100	60	80	30
	<b>Trial 2:</b>	1.5 ft. Straight	100	100	20	100
		1.5 ft. Right Side	100	10	5	60
		1.5 ft. Left Side	20	100	90	60
		2.5 ft. Straight	100	100	20	95
		2.5 ft. Right Side	100	100	80	20
		2.5 ft. Left Side	100	100	10	90
		3.5 ft. Straight	100	100	100	90
		3.5 ft. Right Side	100	80	20	80
		3.5 ft. Left Side	100	95	100	10
	<b>Trial 3:</b>	1.5 ft. Straight	100	100	0	100
		1.5 ft. Right Side	100	100	10	100
		1.5 ft. Left Side	80	95	80	70
		2.5 ft. Straight	100	100	70	95
		2.5 ft. Right Side	100	100	80	100
		2.5 ft. Left Side	100	70	10	90
		3.5 ft. Straight	100	100	30	5

Процес ідентифікації обличчя включав сім суб'єктів у навчальному наборі та тестувався проти чотирьох суб'єктів. Тестування включало

перевірку, чи може кожен суб'єкт бути правильно ідентифікований у фактичних результатах прогнозування, будучи правильно ідентифікованим протягом якомога більшого часу під час десяти секунд тестування. Це призведе до вищого показника прогнозування. Результати порівняння тестування можна побачити в таблиці 3.4.

Таблиця 3.4 - 1:N Eigenfaces vs Fisherfaces vs Local Binary Pattern  
(Фрагмент таблиці)

Eigenfaces vs Fisherfaces vs Local Binary Pattern Histogram								
Person 1:	Trial 1:	Position	Prediction Rate (%)			False Positive Rate (%)		
			Eigenface s	Fisherface s	LBP H	Eigenface s	Fisherface s	LBP H
		1.5 ft. Straight	100	30	0	10	20	0
		1.5 ft. Right Side	0	-	-	90	20	10
		1.5 ft. Left Side	80	-	0	30	10	10
		2.5 ft. Straight	90	20	0	20	50	0
		2.5 ft. Right Side	0	60	-	70	10	30
		2.5 ft. Left Side	100	-	-	30	5	0
		3.5 ft. Straight	100	80	30	90	100	10
		3.5 ft. Right Side	95	0	-	40	95	20
		3.5 ft. Left Side	10	95	100	20	10	10
	<b>Trial 2:</b>	1.5 ft. Straight	95	0	30	10	10	5
		1.5 ft. Right Side	100	-	-	90	10	10
		1.5 ft. Left Side	90	-	0	60	10	10
		2.5 ft. Straight	100	50	20	70	5	30
		2.5 ft. Right Side	70	60	-	30	80	60
		2.5 ft. Left Side	20	-	20	40	20	10
		3.5 ft. Straight	95	50	0	20	30	10
		3.5 ft. Right Side	100	30	0	10	90	10
		3.5 ft. Left Side	10	100	50	10	20	0

Аналіз результатів порівняльного тестування методів Eigenfaces, Fisherfaces та Оригінального Локального бінарного шаблону (LBP) щодо показників коефіцієнта передбачення та рівня хибних позитивних спрацювань дозволив зробити наступні висновки. Метод Eigenfaces продемонстрував найвищу ефективність за показником коефіцієнта передбачення. Однак, цей метод також характеризувався найвищим рівнем хибних позитивних спрацювань.

Оригінальний метод LBP показав найнижчий коефіцієнт передбачення, але при цьому відзначився найнижчим рівнем хибних позитивних спрацювань серед трьох протестованих методів.

На підставі отриманих даних, зокрема низького рівня хибних позитивних спрацювань Оригінального методу LBP, було прийнято рішення сфокусувати подальші дослідження на його покращенні. Після реалізації модифікацій, алгоритм Локального бінарного шаблону продемонстрував підвищення як коефіцієнта передбачення, так і рівня хибних позитивних спрацювань порівняно з його оригінальною версією. Зокрема, було зафіксовано збільшення відсотка коректних передбачень для трьох із чотирьох тестованих користувачів. Також спостерігалось зменшення кількості випадків, коли коефіцієнт передбачення дорівнював 0%. Детальні результати порівняння продуктивності Оригінального та Модифікованого Локального бінарного шаблону у сценарії 1:N представлено в таблиці 3.5.

Таким чином, модифікація алгоритму Локального бінарного шаблону призвела до покращення здатності коректно ідентифікувати відомих користувачів, але ціною зростання ймовірності хибної ідентифікації або виявлення.

Детальний кількісний аналіз впливу модифікацій на показники коефіцієнта передбачення та рівня хибних позитивних спрацювань у сценарії 1:N може бути виконаний на основі даних, представлених у таблиці 3.5.

					БР.ІП – 13.00.00.000 ПЗ	Арк.
						67
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 3.5 - 1:N Оригінальний Локальний бінарний шаблон проти  
Модифікованого Локального бінарного шаблону

Local Binary Pattern Histogram Original vs Local Binary Pattern Modified						
Person 1:	Trial 1:	Position	Prediction Rate (%)		False Positive Rate (%)	
			LBP Original	LBP Modified	LBP Original	LBP Modified
		1.5 ft. Straight	0	40	0	20
		1.5 ft. Right Side	-	20	10	40
		1.5 ft. Left Side	0	0	10	80
		2.5 ft. Straight	0	50	0	30
		2.5 ft. Right Side	-	40	30	10
		2.5 ft. Left Side	-	0	0	40
		3.5 ft. Straight	30	0	10	20
		3.5 ft. Right Side	-	0	20	10
		3.5 ft. Left Side	100	5	10	30
	<b>Trial 2:</b>	1.5 ft. Straight	30	10	5	60
		1.5 ft. Right Side	-	0	10	40
		1.5 ft. Left Side	0	30	10	20
		2.5 ft. Straight	20	10	30	40
		2.5 ft. Right Side	-	20	60	30
		2.5 ft. Left Side	20	0	10	20
		3.5 ft. Straight	0	30	10	80
		3.5 ft. Right Side	0	0	10	20
		3.5 ft. Left Side	50	30	0	10
	<b>Trial 3:</b>	1.5 ft. Straight	30	40	0	60
		1.5 ft. Right Side	-	20	10	60
		1.5 ft. Left Side	-	0	20	70
		2.5 ft. Straight	5	0	30	30
		2.5 ft. Right Side	-	40	60	20
		2.5 ft. Left Side	40	0	20	40
		3.5 ft. Straight	0	10	20	80
		3.5 ft. Right Side	-	30	80	20
		3.5 ft. Left Side	5	0	40	70

Змн.	Арк.	№ докум.	Підпис	Дата

### 3.4. Удосконалення алгоритму просторової гистограми локальних бінарних шаблонів контексті аналізу зображень обличчя

В контексті аналізу зображень обличчя, різні анатомічні області можуть мати різну інформативність для ідентифікації особи. Наприклад, такі риси, як очі, ніс та рот, зазвичай несуть більше унікальної інформації, ніж області чола чи щік. Відповідно, не всі ділянки обличчя мають однакову дискримінаційну здатність при самостійному аналізі.

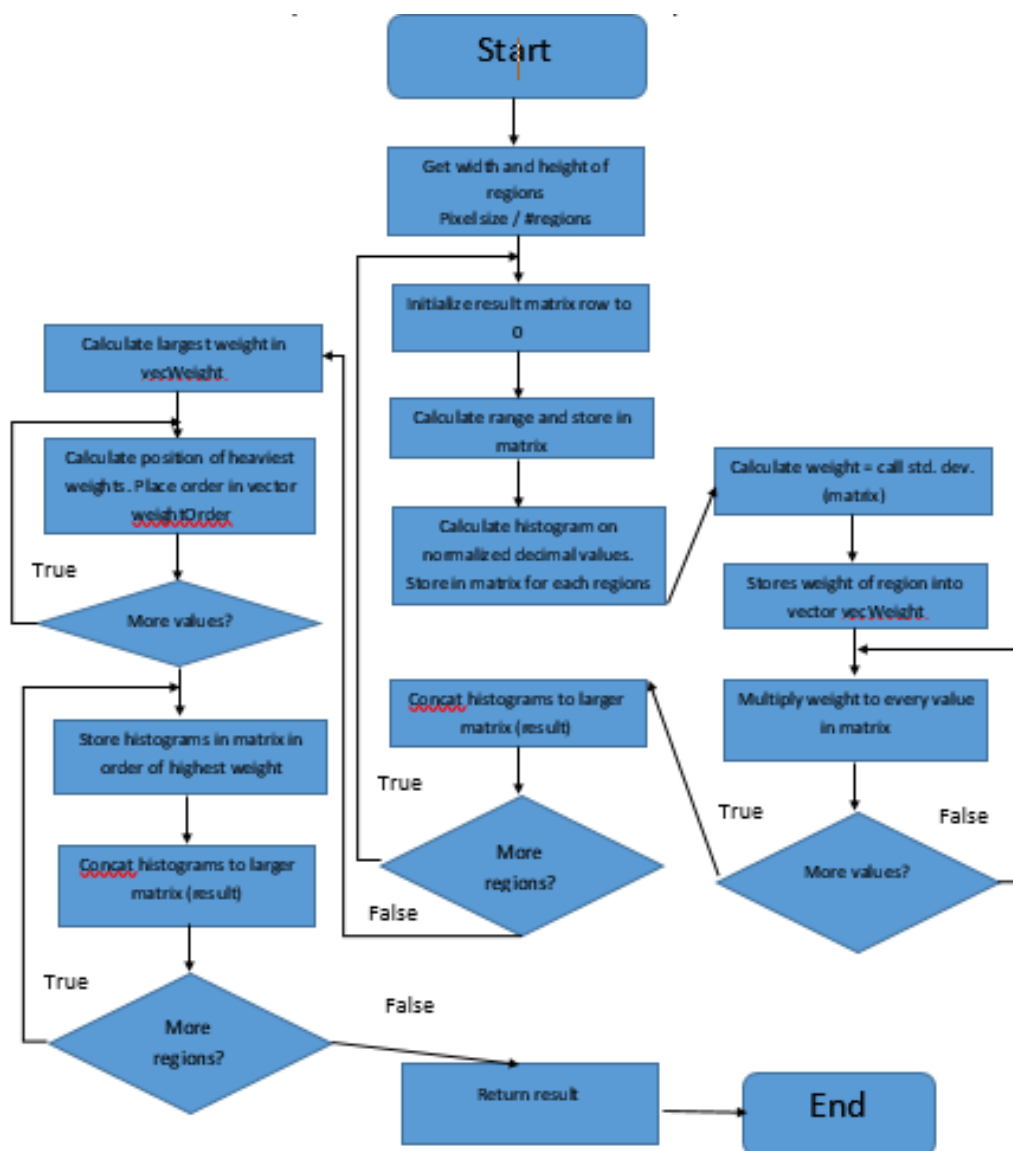


Рисунок 3.10 - Блок-схема модифікованої просторової гистограми локального бінарного шаблону

Змн.	Арк.	№ докум.	Підпис	Дата

Враховуючи цю нерівномірність інформативності, було запропоновано метод удосконалення алгоритму просторової гістограми локальних бінарних шаблонів (LBP) шляхом застосування ваг до специфічних регіонів обличчя. Кожна вага призначається певному регіону та диференціюється на основі міри контрасту або варіативності значень інтенсивності (градацій сірого) для кожного пікселя та його сусідів у межах цього регіону. Розрахунок ваг здійснюється з використанням методу стандартного відхилення.

Обчислені ваги застосовуються до регіональних гістограм LBP. Замість простого конкатенації гістограм регіонів, їхній внесок у фінальний вектор ознак (загальну просторову гістограму) масштабується відповідно до обчислених ваг.

Подальша побудова фінального вектора передбачає впорядковану конкатенацію регіональних гістограм: першими додаються гістограми регіонів з найвищими вагами, за ними — регіони з наступними за величиною вагами, і так далі, доки гістограма регіону з найменшою вагою не буде додана останньою (процес візуалізовано на блок-схемі Модифікованої просторової гістограми LBP, рисунок 3.10).

Блок-схема (рис. 3.11) ілюструє алгоритм обчислення стандартного відхилення для набору значень, представлених у "регіональній матриці". Зважаючи на попередній контекст, ця матриця, імовірно, містить значення інтенсивності пікселів або LBP-значення для конкретного регіону зображення, і обчислене стандартне відхилення використовується як вага цього регіону.

Ця блок-схема точно відображає стандартну процедуру статистичного розрахунку стандартного відхилення для заданого набору даних.

Такий підхід спрямований на підвищення точності розпізнавання шляхом надання більшого впливу на процес ідентифікації ознакам з регіонів, які мають вищі ваги та, відповідно, вважаються більш інформативними. Реалізація описаного методу вимагала додавання та модифікації певних

					БР.ІП – 13.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		70

функцій у програмному коді проекту, що відображено на рисунках 3.12, 3.13. Внесений новий або змінений код супроводжується коментарями, що починаються з // New:, з описом функціональності відповідних ділянок коду.

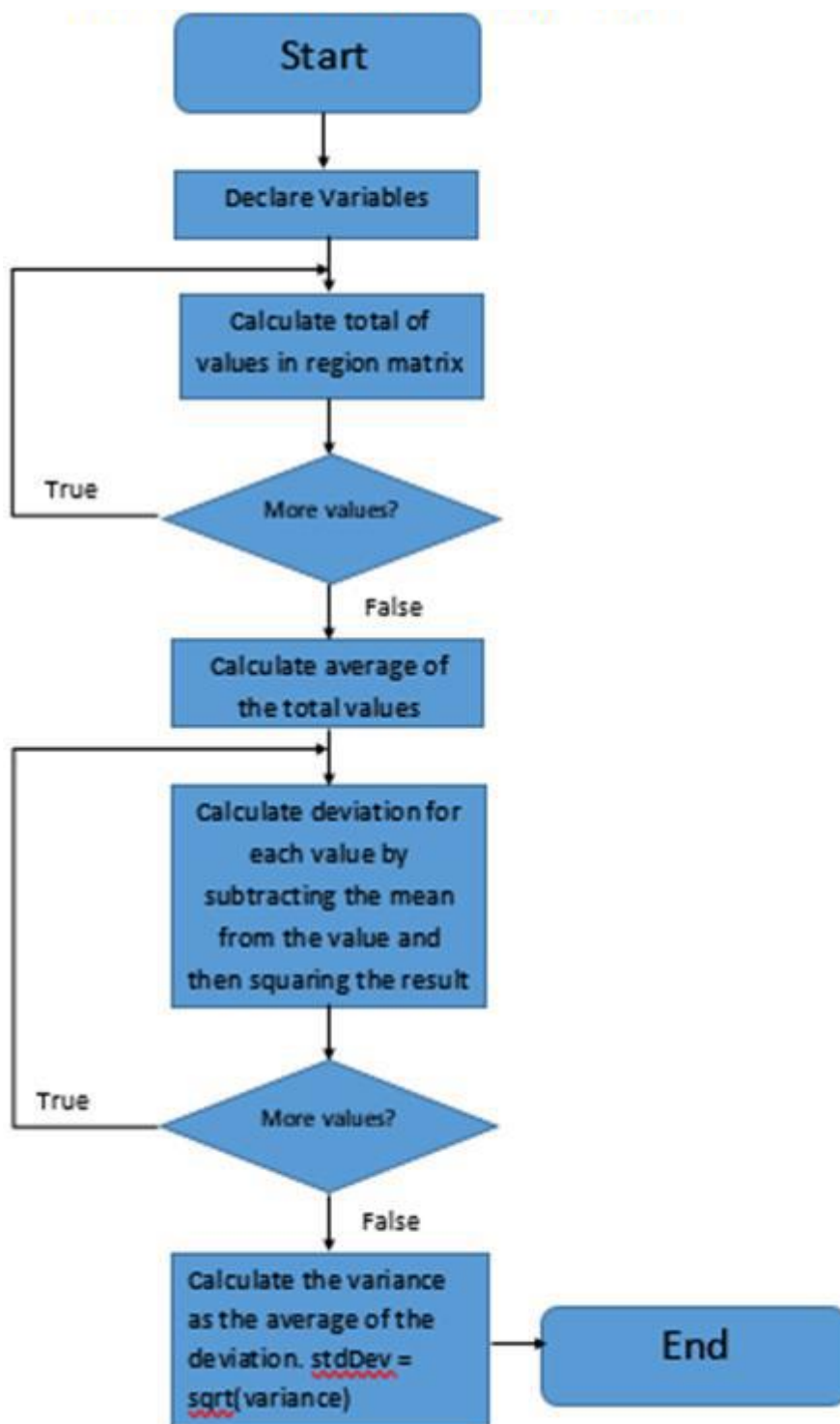


Рисунок 3.11 - Блок-схема розрахунку стандартного відхилення для локального бінарного шаблону

```

/*****
*NEW FUNCTION *
*Calculates the standard deviation for each region matrix and *
*assigns the value as a weight. *
*****/
static float standard_deviation(Mat input)
{
    float total;
    float totalDev;
    float mean;
    int i;
    float stdDev;

    for(i = 0; i < input.cols; i++)
    {
        total = input.at<float>(i);
    }

    mean = total / i;

    for(i = 0; i < input.cols; i++)
    {
        totalDev = (input.at<float>(i) - mean) * (input.at<float>(i) - mean);
    }

    stdDev = totalDev / i;

    return stdDev;
}

static Mat spatial_histogram(InputArray _src, int numPatterns,
                             int grid_x, int grid_y, bool /*normed*/)
{
    Mat src = _src.getMat();
    Mat result_row; //new
    Mat cell_hist; //new
    float weight; //new
    vector<float> vecWeight; //new

    // calculate LBP patch size
    int width = src.cols/grid_x;
    int height = src.rows/grid_y;
    // allocate memory for the spatial histogram
    Mat result = Mat::zeros(grid_x * grid_y, numPatterns, CV_32FC1);
    // return matrix with zeros if no data was given
    if(src.empty())
        return result.reshape(1,1);
    // initial result_row
    int resultRowIdx = 0;
    // iterate through grid
    for(int i = 0; i < grid_y; i++) {
        for(int j = 0; j < grid_x; j++) {
            Mat src_cell = Mat(src, Range(i*height,(i+1)*height), Range(j*width,(j+1)*width));

            cell_hist = histc(src_cell, 0, (numPatterns-1), true);
        }
    }
}

```

Рисунок 3.12 - Модифікований Локальний бінарний шаблон Вихідний

код 1

					БР.ІП – 13.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		72

```

// New: Calls the fuction to caculate each regions weight using standard deviation
weight = standard_deviation(cell_hist);
// New: The weight is added to a vector to keep track of the weight for all of the
regions in an image
vecWeight.push_back(weight);

// New: Uses scalar multiplication to multiply the regions weight to the region
matrix
for(int i = 0; i < cell_hist.cols; i++)
{
    cell_hist.at<float>(i) = cell_hist.at<float>(i) * weight;
}

// copy to the result matrix
result_row = result.row(resultRowIndex);
cell_hist.reshape(1,1).convertTo(result_row, CV_32FC1);
// increase row count in result matrix
resultRowIndex++;
}
}

// New: Calculates the maximum weight in the vecWeight vector
float it = *max_element(vecWeight.begin(), vecWeight.end());
vector<int> weightOrder;

// New: Calculates the position of the heaviest weights, and then places this order in a
vector.
// After the heaviest weight is used, it is assigned a -1 so that it is not used again.
while(it != -1)
{
    for(int i = 0; i < vecWeight.size(); i++)
    {
        if(vecWeight[i] == it)
        {
            weightOrder.push_back(i);
            vecWeight[i] = -1;
        }
    }
    it = *max_element(vecWeight.begin(), vecWeight.end());
}

// New: Places the regions in the result matrix based on the weight of the region
for(int i = 0; i < result.rows; i++)
{
    //copy to the result matrix
    resultRowIndex = weightOrder[i];
    result_row = result.row(resultRowIndex);
    cell_hist.reshape(1,1).convertTo(result_row, CV_32FC1);
}

//return result as reshaped feature vector
return result.reshape(1,1);

```

Рисунок 3.13 - Модифікований Локальний бінарний шаблон Вихідний

код 2

Удосконалення методів розпізнавання обличчя, зокрема алгоритму просторової гістограми локальних бінарних шаблонів, є актуальним завданням, що обумовлено наявністю певних переваг та недоліків у існуючих підходів. Висока точність передбачень у системах розпізнавання облич є

						БР.ІП – 13.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			73

критично важливою, що вимагає безперервної розробки нових алгоритмів та вдосконалення вже існуючих. Це особливо актуально в контексті зростання популярності технологій розпізнавання обличчя як форми біометричної аутентифікації для завдань верифікації та ідентифікації осіб. Пошук та розробка оптимальних методів розпізнавання обличчя потенційно може сприяти експоненційному зростанню їх використання, а також суттєвому підвищенню рівня безпеки та ефективності процесів ідентифікації.

Як один з потенційних напрямків подальшого удосконалення алгоритму просторової гистограми Локального бінарного шаблону в рамках даного проєкту, може бути розглянута можливість ігнорування регіонів з вагою, нижчою за встановлене порогове значення. При побудові загальної гистограми зображення шляхом конкатенації регіональних гистограм, впорядкованих за вагою, можливим є виключення гистограм тих регіонів, чий обчислені ваги не перевищують визначеного порогу. Реалізація такого підходу може призвести до зменшення обчислювальної складності, зокрема часу, необхідного для етапу навчання моделі. Ключовим аспектом для дослідження є визначення потенційного впливу виключення низько вагових регіонів на точність передбачення (коефіцієнт передбачення) при одночасному скороченні часу навчання.

					БР.ІП – 13.00.00.000 ПЗ	Арк.
						74
Змн.	Арк.	№ докум.	Підпис	Дата		

## ВИСНОВКИ

В дипломній роботі було здійснено глибокий аналіз предметної області машинного навчання та методів розпізнавання образів, зосереджений, зокрема, на розпізнаванні обличчя. У першому розділі було охарактеризовано основні підходи та алгоритми, що застосовуються в цій галузі, зокрема методи Eigenfaces, Fisherfaces, LBPН та сучасні моделі на основі глибокого навчання. Проведений огляд актуальних досліджень та викликів підтвердив високу значущість завдання розпізнавання обличчя для сучасних інформаційних систем.

У другому розділі було детально представлено методології побудови рішень для розпізнавання облич, здійснено порівняльний аналіз класичних і вдосконалених підходів, а також обґрунтовано вибір методу LBPН як базового для подальшої оптимізації. Запропоновано модифікації базового алгоритму, зокрема через удосконалення просторової гістограми, що дозволило підвищити стійкість методу до варіацій зображень.

Третій розділ було присвячено реалізації та тестуванню запропонованих рішень. Було описано процедуру інтеграції бібліотеки OpenCV у середовище Visual Studio, підготовку експериментальних даних, а також проведено серію експериментів з оцінки точності та швидкодії методів. Отримані результати підтверджують ефективність покращеного алгоритму LBPН у контексті розпізнавання облич на практичних прикладах.

Загалом, робота поєднує теоретичний огляд сучасних підходів і практичну імплементацію ефективного рішення для задач розпізнавання образів, що може бути корисним у розробці інтелектуальних систем безпеки, контролю доступу та інших застосунків комп'ютерного зору.

					БР.ІП – 13.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		75

## ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Duda, R. O., Hart, P. E., & Stork, D. G. (2001). Pattern Classification (2nd ed.). Wiley-Interscience.
2. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
3. Szeliski, R. (2022). Computer Vision: Algorithms and Applications (2nd ed.). Springer.
4. Jain, A. K., Duin, R. P. W., & Mao, J. (2000). "Statistical Pattern Recognition: A Review". IEEE Transactions on Pattern Analysis and Machine Intelligence.
5. Boyd, S., & Vandenberghe, L. (2004). Convex Optimization. Cambridge University Press.
6. Bengio, Y. (2012). "Practical Recommendations for Gradient-Based Training of Deep Architectures". In Neural Networks: Tricks of the Trade.
7. Snoek, J., Larochelle, H., & Adams, R. P. (2012). "Practical Bayesian Optimization of Machine Learning Algorithms". NeurIPS.
8. Hutter, F., Kotthoff, L., & Vanschoren, J. (Eds.). (2019). Automated Machine Learning: Methods, Systems, Challenges. Springer.
9. Bergstra, J., & Bengio, Y. (2012). "Random Search for Hyper-Parameter Optimization". JMLR.
10. Gonzalez, R. C., & Woods, R. E. (2018). Digital Image Processing (4th ed.). Pearson.
11. Zhang, D. (2012). Image Feature Extraction and Recognition: Pattern Recognition Techniques. Springer.
12. Turk, M., & Pentland, A. (1991). "Eigenfaces for Recognition". Journal of Cognitive Neuroscience.
13. Dalal, N., & Triggs, B. (2005). "Histograms of Oriented Gradients for Human Detection". CVPR.

					БР.ІІІ – 13.00.00.000 ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		76

14. Lowe, D. G. (2004). "Distinctive Image Features from Scale-Invariant Keypoints". IJCV.
15. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). "ImageNet Classification with Deep Convolutional Neural Networks". NeurIPS.
16. He, K., Zhang, X., Ren, S., & Sun, J. (2016). "Deep Residual Learning for Image Recognition". CVPR.
17. Simonyan, K., & Zisserman, A. (2014). "Very Deep Convolutional Networks for Large-Scale Image Recognition". arXiv.
18. Linear Discriminant Analysis (LDA) - The Decision Lab - <https://thedeisionlab.com/reference-guide/statistics/linear-discriminant-analysis>
19. Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K. Q. (2017). "Densely Connected Convolutional Networks". CVPR.
20. Taigman, Y., Yang, M., Ranzato, M., & Wolf, L. (2014). "DeepFace: Closing the Gap to Human-Level Performance in Face Verification". CVPR.
21. Schroff, F., Kalenichenko, D., & Philbin, J. (2015). "FaceNet: A Unified Embedding for Face Recognition and Clustering". CVPR.
22. Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2015). "Deep Face Recognition". BMVC.
23. Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). "Joint Face Detection and Alignment Using Multi-task Cascaded Convolutional Networks". IEEE SPL.
24. Deng, J., Guo, J., Xue, N., & Zafeiriou, S. (2019). "ArcFace: Additive Angular Margin Loss for Deep Face Recognition". CVPR.
25. Srivastava, N., et al. (2014). "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". JMLR.
26. Ioffe, S., & Szegedy, C. (2015). "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". ICML.
27. Loshchilov, I., & Hutter, F. (2019). "Decoupled Weight Decay Regularization". ICLR.

28. Kingma, D. P., & Ba, J. (2015). "Adam: A Method for Stochastic Optimization". ICLR.
29. Zhang, H., et al. (2018). "mixup: Beyond Empirical Risk Minimization". ICLR.
30. Principal Component Analysis (PCA) In Depth | by Fraidoon Omarzai | Medium - <https://medium.com/@fraidoonomarzai99/principal-component-analysis-pca-in-depth-93c871f25dfa>
31. Powers, D. M. W. (2011). "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation". Journal of Machine Learning Technologies.
32. Bradski, G. (2000). "The OpenCV Library". Dr. Dobb's Journal of Software Tools.
33. Viola, P., & Jones, M. (2001). "Rapid Object Detection using a Boosted Cascade of Simple Features". CVPR.
34. Zhou, E., et al. (2015). "Face Recognition with Deeper Neural Networks". arXiv.
35. Paszke, A., et al. (2019). "PyTorch: An Imperative Style, High-Performance Deep Learning Library". NeurIPS.
36. Abadi, M., et al. (2016). "TensorFlow: A System for Large-Scale Machine Learning". OSDI.
37. Redmon, J., & Farhadi, A. (2018). "YOLOv3: An Incremental Improvement". arXiv.
38. Understanding Face Recognition Using LBPH Algorithm - <https://www.analyticsvidhya.com/blog/2021/07/understanding-face-recognition-using-lbph-algorithm/>
39. Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). "YOLOv4: Optimal Speed and Accuracy of Object Detection". arXiv.
40. Ren, S., He, K., Girshick, R., & Sun, J. (2015). "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". NeurIPS.

41. Wang, M., & Deng, W. (2021). "Deep Face Recognition: A Survey". Neurocomputing.
42. Liu, W., et al. (2020). "A Survey of Deep Learning-Based Face Recognition". Pattern Recognition.
43. Zhao, W., et al. (2003). "Face Recognition: A Literature Survey". ACM Computing Surveys.
44. Li, S. Z., & Jain, A. K. (Eds.). (2011). Handbook of Face Recognition (2nd ed.). Springer.
45. Zhang, Y., & Ling, H. (2021). "Learning to Optimize: A Primer and A Benchmark". IEEE Signal Processing Magazine.

					БР.ІП – 13.00.00.000 ПЗ	Арк.
						79
Змн.	Арк.	№ докум.	Підпис	Дата		

## БІБЛІОГРАФІЧНА ДОВІДКА

**Тема дипломної роботи:** “Побудова рішення для оптимізації та налаштування методів розпізнавання образів ”

Обсяг пояснювальної записки: 79 аркушів.

Дата закінчення роботи: 10 червня 2025 р.

Підпис студента \_\_\_\_\_