

**TRƯỜNG ĐẠI HỌC HÀNG HẢI VIỆT NAM  
KHOA CÔNG NGHỆ THÔNG TIN**



**THUYẾT MINH  
ĐỀ TÀI NCKH CẤP TRƯỜNG**

**ĐỀ TÀI  
XÂY DỰNG HỆ THỐNG NHẬN DẠNG MẶT TỰ ĐỘNG SỬ  
DỤNG LPQ (LOCAL PHASE QUANTIZATION)**

**Chủ nhiệm đề tài: TS. NGUYỄN HỮU TUÂN**

**Thành viên tham gia: ThS. NGUYỄN VĂN THỦY**

**Hải Phòng, tháng 4/2016**

# MỤC LỤC

MỞ ĐẦU .....	1
1. Tính cấp thiết của vấn đề nghiên cứu .....	1
2. Tổng quan về tình hình nghiên cứu thuộc lĩnh vực đề tài .....	1
3. Mục tiêu, đối tượng, phạm vi nghiên cứu.....	1
4. Phương pháp nghiên cứu, kết cấu của công trình nghiên cứu .....	2
5. Kết quả đạt được của đề tài.....	2
CHƯƠNG 1: CƠ SỞ LÝ THUYẾT CỦA NHẬN DẠNG MẶT NGƯỜI .....	4
1. Bài toán nhận dạng mặt người – những vấn đề cơ bản.....	4
1.1. Giới thiệu .....	4
1.2. Thách thức trong nhận dạng mặt người.....	5
1.3. Các cách tiếp cận cho bài toán nhận dạng mặt người.....	6
CHƯƠNG 2: CÁC THÀNH PHẦN CỦA MỘT HỆ THỐNG NHẬN DẠNG MẶT TỰ ĐỘNG .....	10
1. Hệ thống nhận dạng mặt tự động .....	10
2. Phát hiện mặt người sử dụng đặc trưng HOG.....	11
3. Chuẩn hóa ánh sáng với bộ lọc Retinal filter.....	15
4. Phương pháp trích chọn đặc trưng LPQ (Local Phase Quantization) .....	17
5. Bộ phân lớp k-NN .....	20
CHƯƠNG 3: XÂY DỰNG HỆ THỐNG .....	23
1. Thư viện OpenCV và dlib .....	23
1.1. Thư viện OpenCV .....	23
1.2. Thư viện dlib.....	26
2. Dữ liệu và giao thức đánh giá hiệu năng của hệ thống.....	30

3. Kết quả nhận dạng.....	32
KẾT LUẬN.....	34
TÀI LIỆU THAM KHẢO.....	36

## DANH SÁCH BẢNG BIỂU

Bảng 3.1: So sánh tỉ lệ nhận dạng của hệ thống đề xuất với các kết quả đã công bố khác. ....	32
--	----

## DANH SÁCH HÌNH ẢNH

Hình 1.1: Các thông tin có trong ảnh mặt người.....	5
Hình 1.2: Các bước trong một hệ thống nhận dạng mặt .....	8
Hình 2.1: Các bước trong hệ thống nhận dạng mặt tự động.....	10
Hình 2.2: Ảnh input và hai đạo hàm của nó.....	12
Hình 2.3: Ảnh input và kết quả phát hiện và căn chỉnh ảnh mặt với HOG.....	14
Hình 2.4: Các bước trong bộ lọc Retinal filter.....	15
Hình 2.5: Kết quả của việc tiền xử lý với bộ lọc Retina filter.....	17
Hình 2.6: Một số ảnh LPQ thu được từ các toán tử LPQ.....	19
Hình 2.7: Các bước trong phương pháp trích chọn đặc trưng LPQ.....	20
Hình 2.8: Minh họa bộ phân lớp kNN .....	21
Hình 3.1: Cấu trúc của thư viện OpenCV.....	23
Hình 3.2: Cấu trúc của thư viện dlib.....	27
Hình 3.3: Một số ảnh mặt người từ cơ sở dữ liệu FERET.....	31
Hình 3.4: Một số kết quả về phát hiện mặt người trong CSDL FERET.....	32

## DANH SÁCH THUẬT NGỮ, CHỮ VIẾT TẮT

<b>Chữ viết tắt</b>	<b>Trang</b>
HOG – Histogram of Oriented Gradient	2
kNN – k-nearest neighbors	2
LPQ – Local Phase Quantization	2
FERET – Face Recognition Technology	3
PCA – Principal Components Analysis	6
LDA – Linear Discriminant Analysis	7
LBP – Local Binary Pattern	7
SVM – Support Vector Machine	8

## MỞ ĐẦU

### 1. Tính cấp thiết của vấn đề nghiên cứu

Nhận dạng mặt là bài toán có nhiều ứng dụng trong thực tế và nhận được sự quan tâm lớn từ cộng đồng các nhà khoa học nghiên cứu về Thị giác máy tính trong thời gian gần đây. Một hệ thống nhận dạng mặt có nhiều bước và mỗi bước lại là một lĩnh vực nghiên cứu với các cách tiếp cận khác nhau. Do vậy, hầu hết các nghiên cứu đều thực hiện các thực nghiệm trên các cơ sở dữ liệu chuẩn, có sẵn các dữ liệu cần thiết, mà không tập trung vào việc xây dựng một hệ thống *tự động* nhận dạng từ các dữ liệu thô ban đầu (các ảnh mặt chưa qua xử lý).

### 2. Tổng quan về tình hình nghiên cứu thuộc lĩnh vực đề tài

Hiện nay các nghiên cứu về nhận dạng mặt được chia thành các hướng tiếp cận khác nhau cho từng khâu của một hệ thống hoàn chỉnh vốn gồm có nhiều bước, mỗi bước có một chức năng và ảnh hưởng khác nhau tới hiệu năng chung của hệ thống. Các thử nghiệm về tỉ lệ nhận dạng của các cách phương pháp đề xuất thường được tiến hành trên các tập dữ liệu được thu nhận trong điều kiện hạn chế về ánh sáng, hướng, ..., do đó khi áp dụng vào thực tế thì kết quả thu được thường kém xa so với môi trường thử nghiệm. Chính vì vậy, trong đề tài này, chúng tôi tập trung vào việc xây dựng một hệ thống tự động nhận dạng và thử nghiệm với các điều kiện ảnh thu nhận được trong các điều kiện thực tế.

### 3. Mục tiêu, đối tượng, phạm vi nghiên cứu

Mục tiêu của đề tài là xây dựng một hệ thống nhận dạng mặt hoàn toàn tự động với dữ liệu đầu vào là các ảnh mặt người sử dụng LPQ, một phương pháp trích chọn đặc trưng ảnh mặt có khả năng làm việc tốt với các ảnh bị mờ hay bị ảnh hưởng bởi ánh sáng. Hệ thống sẽ gồm các module thực hiện: phát hiện vùng ảnh mặt, tiền xử lý, trích chọn đặc trưng, học và nhận dạng.

**4. Phương pháp nghiên cứu, kết cấu của công trình nghiên cứu**

Nhằm xây dựng một chương trình nhận dạng mặt tự động, đề tài thực hiện nghiên cứu các vấn đề sau:

+ Nghiên cứu kỹ thuật phát hiện ảnh mặt dựa trên đặc trưng HOG (Histogram of Oriented Gradients).

+ Nghiên cứu kỹ thuật tiền xử lý để chuẩn hóa ảnh mặt

+ Nghiên cứu áp dụng LPQ (Local Phase Quantization) cho bài toán nhận dạng mặt.

+ Nghiên cứu thư viện mã nguồn mở OpenCV.

+ Xây dựng chương trình

**5. Kết quả đạt được của đề tài**

Đã nghiên cứu khảo sát các bước của một hệ thống nhận dạng mặt tự động và các kỹ thuật khác nhau có thể áp dụng cho từng bước. Đề xuất và xây dựng thành công một hệ thống nhận dạng mặt tự động với các bước cụ thể như sau:

+ Tự động phát hiện và căn chỉnh khuôn mặt bằng phương pháp HOG (Histograms of Oriented Gradients).

+ Chuẩn hóa điều kiện ánh sáng của các ảnh mặt người sau bước phát hiện và căn chỉnh bằng kỹ thuật retina filter.

+ Trích chọn đặc trưng của ảnh mặt nhằm biểu diễn các bức ảnh thành các vector đặc trưng sử dụng phương pháp trích chọn đặc trưng LPQ, một phương pháp cho kết quả nhận dạng tốt trong nhiều điều kiện khó khăn, chẳng hạn như thay đổi của ánh sáng và ảnh bị mờ.

+ Sử dụng phương pháp phân lớp kNN (k-nearest neighbors) cho bước nhận dạng để đưa ra danh tính của bức ảnh cần nhận dạng.

Hệ thống được cài đặt bằng ngôn ngữ C++ sử dụng bộ công cụ Visual Studio 2015 update 2 và hai thư viện mã nguồn mở OpenCV và dlib. Hiệu năng nhận dạng của hệ thống được thử nghiệm đánh giá bằng protocol chuẩn trên cơ



sở dữ liệu ảnh mặt người công cộng FERET (FacE Recognition Technology) và so sánh với các kết quả đã công bố của các phương pháp có liên quan. Kết quả so sánh cho thấy rằng hệ thống đề xuất có hiệu năng khá tốt đối với các ảnh mặt có góc nhìn thẳng được thu nhận trong các điều kiện trong nhà và có kiểm soát.

Mặc dù vậy khi so sánh với các hệ thống bán tự động thì kết quả nhận dạng của hệ thống vẫn chưa cao bằng. Điều này là hợp lý vì việc căn chỉnh ảnh trong các hệ thống bán tự động dựa trên các tọa độ của mắt người được xác định bằng tay vẫn cho kết quả chính xác cao hơn so với việc phát hiện và căn chỉnh tự động.

## **CHƯƠNG 1:**

### **CƠ SỞ LÝ THUYẾT CỦA NHẬN DẠNG MẶT NGƯỜI**

#### **1. Bài toán nhận dạng mặt người – những vấn đề cơ bản**

##### ***1.1. Giới thiệu***

Nhận dạng mặt người (Face recognition) là một chủ đề nghiên cứu thuộc lĩnh vực thị giác máy tính (Computer Vision) đã được phát triển từ đầu những năm 90 của thế kỷ trước [1]. Cho tới hiện nay, đây vẫn là một chủ đề nghiên cứu mở nhận được sự quan tâm của nhiều nhà nghiên cứu từ nhiều lĩnh vực nghiên cứu khác nhau như nhận dạng mẫu (Pattern Recognition), học máy (Machine Learning), thống kê (Statistics), sinh trắc học (Biometrics). Điều này là do có rất nhiều ứng dụng thực tế cần tới một hệ thống nhận dạng mặt, từ các hệ thống quản lý đăng nhập đơn giản cho tới các ứng dụng giám sát tại các địa điểm công cộng (public areas surveillance) hoặc quản lý dân số (population management) và pháp lý (fornensics). Bên cạnh đó, so với các hệ thống nhận dạng dựa trên các đặc điểm sinh trắc học khác của con người, như nhận dạng móng mắt và vân tay (fingerprint and iris recognitions), dáng đi (gait recognition), nhận dạng mặt có nhiều ưu điểm:

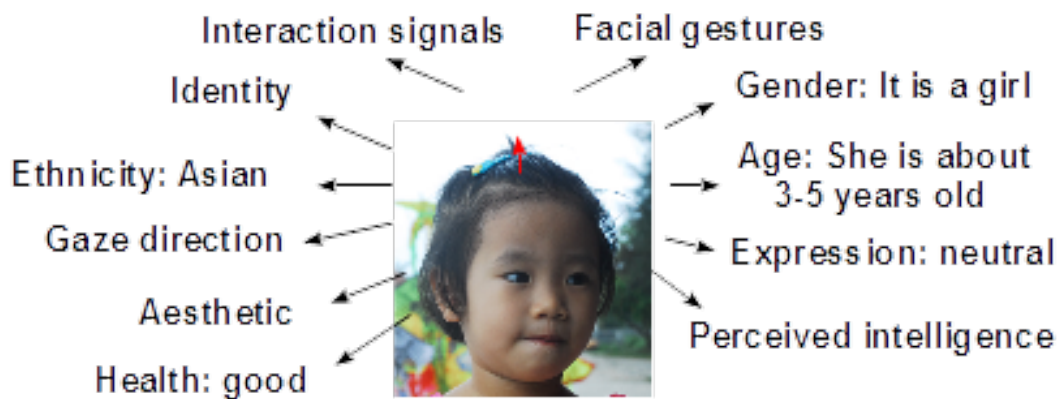
+ Một hệ thống nhận dạng mặt không đòi hỏi có sự tương tác trực tiếp giữa đối tượng được nhận dạng và hệ thống.

+ Việc thu nhận dữ liệu (ảnh mặt) cho quá trình nhận dạng một con người dễ thực hiện hơn so với thu nhận các đặc điểm sinh trắc học khác (như thu nhận dấu vân tay và móng mắt).

+ Dữ liệu về mặt người phổ biến hơn so với các đặc trưng khác do sự bùng nổ các mạng xã hội (facebook, twitter ...), các dịch vụ chia sẻ dữ liệu đa phương tiện (youtube, vimeo ...) và sự phát triển mạnh mẽ của các thiết bị thu nhận hình ảnh.

+ Từ ảnh khuôn mặt của một người ta có thể khai thác nhiều thông tin liên quan chứ không chỉ là danh tính, chẳng hạn như giới tính (gender), màu da (skin

color), hướng nhìn (gaze direction), chủng tộc, hành vi, sức khỏe, độ tuổi, cảm xúc và mức độ thông minh ... (xem thêm trong hình minh họa 1.1).



*Hình 1.1: Các thông tin có trong ảnh mặt người*

### ***1.2. Thách thức trong nhận dạng mặt người***

Tuy nhiên việc xây dựng một hệ thống nhận dạng mặt hoàn toàn tự động với khả năng nhận dạng chính xác cao thực sự là một thách thức đối với các nhà nghiên cứu. Điều này là do các yếu tố (chủ quan và khách quan) ảnh hưởng tới quá trình thu nhận ảnh và tạo ra các bức ảnh có độ khác biệt rất lớn của cùng một khuôn mặt. Có thể liệt kê ra đây các yếu tố chủ yếu ảnh hưởng tới độ chính xác của một hệ thống nhận dạng mặt [2]:

+ ánh sáng (light conditions). Các bức ảnh mặt thu nhận ở các điều kiện sáng khác nhau sẽ rất khác nhau và làm giảm sự chính xác trong quá trình nhận dạng.

+ thay đổi về tuổi (aging changes). Khuôn mặt người có các thay đổi lớn khi tuổi thay đổi và khó nhận dạng hơn ngay cả đối với hệ thống thị giác của con người.

+ các vấn đề về hướng (pose variations). Việc nhận dạng với các ảnh có góc chụp thẳng (frontal) có kết quả tốt hơn rất nhiều so với các ảnh được chụp ở góc nghiêng lớn hơn 45°. Giải pháp thường thấy đối với các ảnh có hướng chụp lớn là sử dụng các thuật toán nội suy để cố gắng bù đắp phần khuôn mặt bị che khuất.

+ cảm xúc (facial expression variations). Ở các trạng thái cảm xúc khác nhau, các đặc điểm quan trọng cho nhận dạng mặt (như mắt, mũi, mồm) có thể bị biến dạng (deformed) và dẫn tới các kết quả nhận dạng sai.

+ che khuất (occlusions). Các ảnh mặt có thể bị che khuất bởi các yếu tố khách quan như vật chắn ở trước mặt hoặc chủ quan như các phụ kiện trên khuôn mặt (khăn, kính mắt) và làm cho quá trình nhận dạng bị sai.

Các hệ thống nhận dạng mặt được chia thành hai loại: xác định danh tính (face identification) và xác thực (face verification). Bài toán xác định danh tính là bài toán dạng 1-N trong đó hệ thống sẽ đưa ra kết quả là danh tính của ảnh được nhận dạng dựa trên sự tương đồng của ảnh input với một danh sách N ảnh đã biết danh tính chính xác. Trong khi đó, ở bài toán xác thực danh tính, hệ thống sẽ đưa ra câu trả lời đúng hoặc sai dựa vào việc xác định xem 2 bức ảnh có thuộc về cùng một người hay không. Trong phạm vi của đề tài này, chúng tôi chỉ tập trung vào bài toán xác định danh tính.

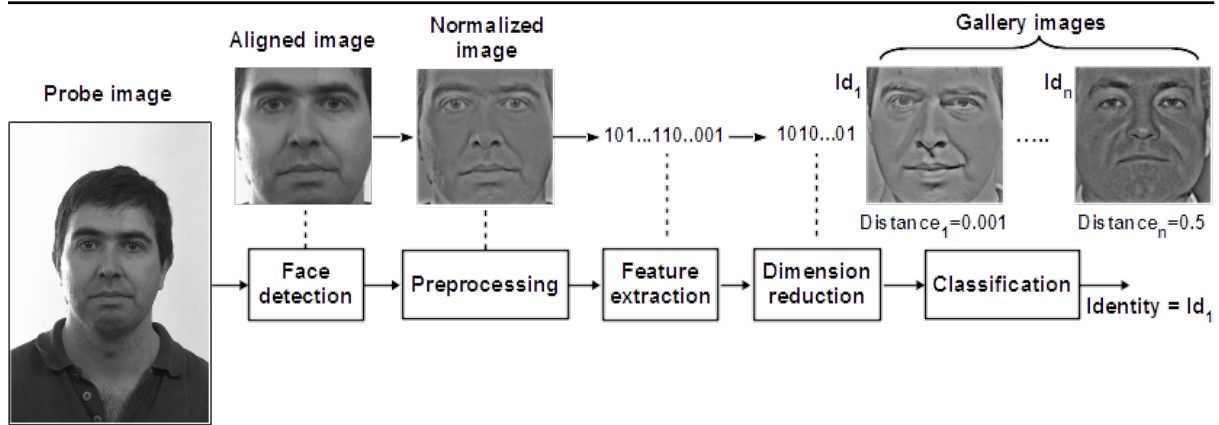
### ***1.3. Các cách tiếp cận cho bài toán nhận dạng mặt người***

Dựa vào việc sử dụng các đặc trưng của khuôn mặt trong khi nhận dạng, các hệ thống nhận dạng mặt được chia thành hai hướng tiếp cận chính: hướng tiếp cận toàn cục (global approaches) và hướng tiếp cận cục bộ (local features based approaches) [2]. Các phương pháp theo hướng tiếp cận toàn cục sẽ sử dụng các đặc điểm toàn cục của khuôn mặt (hình dạng, màu sắc, các đường nét chính ...) trong khi các hệ thống theo cách tiếp cận dựa vào các đặc trưng cục bộ sử dụng các đặc điểm cục bộ của khuôn mặt (các điểm ảnh, các chi tiết như mắt, mũi, mồm, lông mày ...) để nhận dạng. Trong số các hệ thống nhận dạng mặt dựa trên các đặc điểm toàn cục Eigengaces [3] và Fisherfaces [4] là các hệ thống tiêu biểu nhất. Eigenfaces sử dụng phép phân tích thành phần chính PCA (Principal Components Analysis) để biểu diễn mỗi ảnh mặt như là một tổ hợp tuyến tính của các vector riêng thu được từ phép phân giải ma trận hiệp phương sai được tính từ các ảnh mặt sau khi đã chuẩn hóa. Do PCA là một thuật toán học không có giám sát nên không tận dụng được các thông tin về lớp khi tập ảnh huấn luyện có nhiều hơn 1

mẫu cho mỗi lớp nên trong Fisherfaces phương pháp phân tích tuyến tính khác biệt LDA (Linear Discriminant Analysis) đã được sử dụng nhằm khai thác tốt hơn các thông tin này. Về các hệ thống nhận dạng mặt dựa trên các đặc điểm cục bộ thì phương pháp mẫu nhị phân cục bộ LBP (Local Binary Pattern) [5] và biến đổi sóng nhỏ Gabor (Gabor wavelets) [6] là các kỹ thuật điển hình dùng để trích xuất các đặc điểm cục bộ từ ảnh mặt.

Các nghiên cứu cho thấy các hệ thống dựa trên các đặc điểm cục bộ cho kết quả tốt hơn các hệ thống theo cách tiếp cận toàn cục, đặc biệt là khi làm việc với các ảnh bị ảnh hưởng bởi các điều kiện như đã đề cập ở trên. Một cách tiếp cận khác là kết hợp cả hai đặc trưng cục bộ và toàn cục để nhận được một hệ thống lai ghép (hybrid system) nhằm đạt được hiệu quả cao hơn. Hệ thống được xây dựng trong tài liệu này dựa trên các đặc trưng cục bộ.

Về cơ bản, một hệ thống nhận dạng mặt người làm việc giống như hệ thống thị giác của con người khi cần nhận dạng (nhận ra) một ai đó khi nhìn vào một bức ảnh. Ở bước đầu tiên hệ thống cần xác định xem có mặt người trong bức ảnh input hay không, sau đó nếu có thì chỉ vùng ảnh đó sẽ được quan tâm, xử lý, bước này gọi là phát hiện khuôn mặt (face detection). Tiếp đến, các đặc điểm là đặc trưng nhất của khuôn mặt (giúp phân biệt ảnh mặt người này với người khác và các ảnh thuộc về cùng một người) sẽ được trích chọn (extract) để nhận dạng, bước này gọi là trích chọn đặc trưng (feature extraction). Trong bước tiếp theo, gọi là nhận dạng (recognition), hệ thống sẽ đối sánh các đặc trưng vừa nhận được với một kho dữ liệu các đặc trưng của những người đã biết rõ danh tính để xác định xem các đặc trưng đó là của khuôn mặt này và đưa ra kết luận về danh tính tương ứng với khuôn mặt có tỉ lệ tương đồng lớn nhất. Tỉ lệ các ảnh nhận dạng đúng danh tính (so với dữ liệu thực tế) trên tổng số ảnh cần nhận dạng gọi là tỉ lệ nhận dạng (recognition rate) và được dùng để đánh giá hiệu năng (độ chính xác) của một hệ thống nhận dạng.



Hình 1.2: Các bước trong một hệ thống nhận dạng mặt

Trên thực tế, việc xây dựng một hệ thống nhận dạng mặt sẽ gồm nhiều bước chi tiết hơn như được chỉ ra trong hình 1.2. Trước bước trích chọn đặc trưng, một thuật toán tiền xử lý (preprocessing) sẽ được sử dụng để thực hiện lọc nhiễu và chuẩn hóa các điều kiện ánh sáng của các bức ảnh sao cho chúng có cùng điều kiện ánh sáng nhất có thể. Tiếp đó, sau bước trích chọn đặc trưng, mỗi ảnh mặt sẽ được biểu diễn dưới dạng một vector đặc trưng có số chiều lớn (từ vài nghìn cho tới hàng trăm nghìn) và chứa nhiều thông tin dư thừa, không cần thiết cho quá trình nhận dạng. Do đó, một thuật toán học sẽ được sử dụng để loại bỏ các đặc điểm dư thừa và đồng thời làm tăng tính phân biệt (discriminative) của các thuộc tính được giữ lại nhằm làm tăng hiệu năng của hệ thống về cả tốc độ xử lý (do chỉ cần làm việc với các vector có số chiều nhỏ hơn rất nhiều) và độ chính xác. Cuối cùng, ở bước nhận dạng (classification stage), có nhiều thuật toán có thể sử dụng, chẳng hạn như bộ phân lớp k-láng giềng gần nhất (k Nearest Neighbors – k-NN), SVM (Support Vector Machine) hoặc mạng nơron (neural network).

Do một hệ thống nhận dạng mặt gồm nhiều bước nên các nghiên cứu về nhận dạng mặt cũng chia thành các lĩnh vực nhỏ hơn chỉ tập trung vào các phương pháp để giải quyết từng bước trong cả hệ thống. Vì thế, mặc dù đã có nhiều nghiên cứu về nhận dạng mặt, song việc xây dựng hoàn chỉnh một hệ thống hoàn toàn tự động từ bước đầu tiên tới cuối cùng vẫn còn chưa nhiều. Trong đề tài này, chúng tôi hướng tới việc xây dựng một hệ thống nhận dạng mặt hoàn toàn tự động với các kỹ thuật được áp dụng cho các bước như sau:

+ phát hiện khuôn mặt: sử dụng các đặc trưng HOG (Histogram of Oriented Gradients) [7].

+ tiền xử lý: sử dụng phương pháp retinal filter [8].

+ trích chọn đặc trưng: nghiên cứu áp dụng kỹ thuật LPQ (Local Phase Quantization) [9].

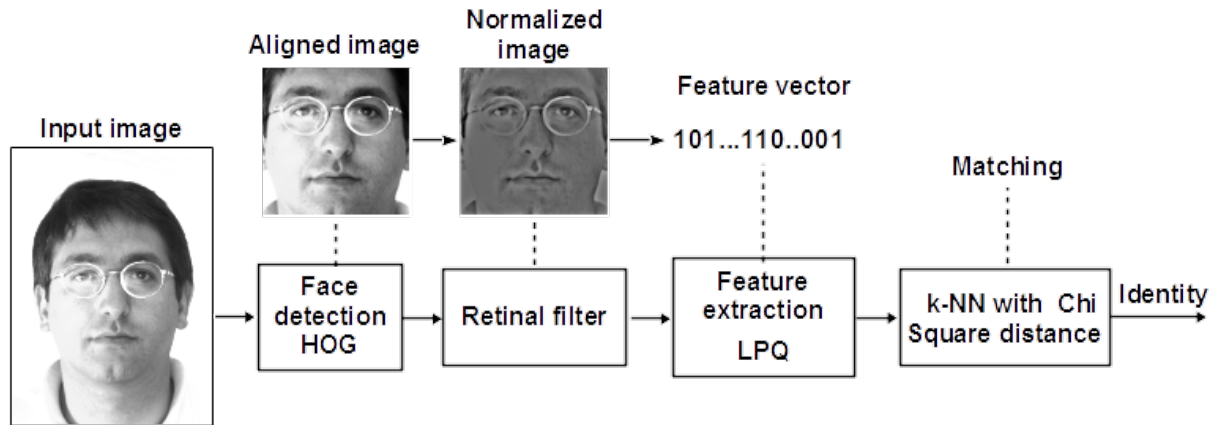
+ nhận dạng đối tượng với bộ phân lớp k-NN.

Trong chương này, các vấn đề liên quan tới bài toán nhận dạng mặt người và các bước cơ bản để xây dựng một hệ thống nhận dạng mặt tự động cùng với các cách tiếp cận liên quan đã được trình bày. Chương tiếp theo của tài liệu này sẽ trình bày chi tiết về các kỹ thuật được áp dụng khi triển khai hệ thống.

## CHƯƠNG 2:

# CÁC THÀNH PHẦN CỦA MỘT HỆ THỐNG NHẬN DẠNG MẶT TỰ ĐỘNG

### 1. Hệ thống nhận dạng mặt tự động



Hình 2.1: Các bước trong hệ thống nhận dạng mặt tự động.

Mục tiêu của đề tài này là xây dựng một hệ thống nhận dạng mặt tự động với các bước được mô tả như trong hình 2.1. Từ ảnh input, các ảnh mặt người sẽ được phát hiện bằng cách sử dụng các đặc trưng HOG (Histogram of Oriented Gradient). Sau đó phương pháp chuẩn hóa ánh sáng Retinal filter sẽ được sử dụng để giảm bớt các ảnh hưởng xấu của ánh sáng tới hiệu năng nhận dạng. Ở bước trích chọn đặc trưng, phương pháp Local Phase Quantization (LPQ) sẽ được sử dụng để trích xuất các đặc trưng quan trọng nhất của mỗi bức ảnh mặt. Bộ phân lớp kNN (k=1) sẽ được sử dụng ở bước phân lớp trong đó hàm Chi Square sẽ được dùng để đánh giá độ tương đồng của các ảnh mặt theo công thức sau:

$$d(X, Y) = \sum_{i=0}^M \frac{(x_i - y_i)^2}{x_i + y_i},$$

trong đó  $X = [x_1, x_2, \dots, x_M]$  và  $Y = [y_1, y_2, \dots, y_M]$  là hai vector đặc trưng tương ứng với 2 bức ảnh mặt.

Trong các phần tiếp theo của chương này chúng tôi sẽ trình bày chi tiết các bước của các thuật toán được sử dụng cho hệ thống nhận dạng mặt tự động.



## **2. Phát hiện mặt người sử dụng đặc trưng HOG**

Các đặc trưng HOG [7] ban đầu được thiết kế để phát hiện người (human detection) trong dữ liệu ảnh nhưng sau đó được mở rộng và được sử dụng rộng rãi cho các bài toán phát hiện đối tượng nói chung (object detection). Phương pháp HOG dựa trên việc đếm số lần xuất hiện của các hướng đạo hàm (gradient orientation) trong các vùng cục bộ của ảnh.

Bản chất của phương pháp HOG là các thông tin về hình dáng và vẻ bề ngoài của các đối tượng cục bộ trong ảnh có thể được mô tả bằng cách sử dụng thông tin về sự phân bố của các cường độ gradient (intensity gradients) hoặc của các hướng biên (edge directions). Các toán tử HOG được cài đặt bằng cách chia nhỏ một bức ảnh thành các vùng con được gọi là các “*tế bào*” (cells) và với mỗi cell đó một histogram về các hướng của gradients sẽ được tính cho các điểm nằm trong cell. Ghép các histogram này lại ta sẽ có một biểu diễn cho bức ảnh ban đầu. Để tăng cường hiệu năng nhận dạng, các histogram cục bộ có thể được chuẩn hóa về độ tương phản bằng cách tính một ngưỡng cường độ trong một vùng lớn hơn của ảnh, gọi là các khối (blocks) và sử dụng giá trị ngưỡng đó để chuẩn hóa tất cả các cell trong khối. Kết quả của bước chuẩn hóa này là vector đặc trưng sẽ có tính bất biến cao hơn đối với các thay đổi về điều kiện ánh sáng.

Các bước cơ bản của việc tính một vector HOG cho một ảnh gồm có: Tính gradient, gán hướng vào các vùng giá trị khác nhau, tính histogram của các khối và cuối cùng là chuẩn hóa các khối.

**Tính gradient:** Đây là bước đầu tiên trong phương pháp HOG và thường được thực hiện bằng hai phép nhân chập ảnh gốc với 2 nhân 1 chiều tương ứng với các toán tử lấy đạo hàm ảnh theo hai hướng  $O_x$  và  $O_y$ , cụ thể là:

$$D_x = [-1 \ 0 \ 1] \text{ và } D_y = [1 \ 0 \ -1]^T,$$

trong đó  $^T$  là ký hiệu của phép chuyển vị ma trận.

Với một ảnh input  $I$ , ta sẽ có 2 ảnh đạo hàm riêng theo 2 hướng được tính bằng các công thức:  $I_x = I * D_x$  và  $I_y = I * D_y$ .

Thành phần cường độ (magnitude) và hướng (orientation) sẽ được tính theo các công thức:  $|G| = \sqrt{I_x^2 + I_y^2}$  và  $\theta = \arctan(I_y, I_x)$ . Ví dụ về kết quả của phép lấy đạo hàm có thể xem ở hình sau:



*Hình 2.2: Ảnh input và hai đạo hàm của nó.*

**Gán hướng:** Sau khi tính đạo hàm, ta có thành phần hướng ở các điểm ảnh trong mỗi cell của ảnh input. Các giá trị hướng này được chia thành các vùng hướng đều nhau trong khoảng từ 0 tới 360°. Theo các tác giả của HOG, thì 9 vùng chia cho kết quả tốt nhất đối với các bài toán nhận dạng. Có nghĩa là các giá trị hướng nằm trong khoảng  $[0, 360/9)$  sẽ được gán bằng 1 (coi là như nhau), các giá trị hướng trong khoảng  $[360/9, 2*360/9)$  sẽ được gán bằng 2, vân vân. Sau đó histogram của mỗi cell sẽ được tính độc lập.

**Tính histogram của các khối:** Để đáp ứng tốt hơn với các thay đổi về điều kiện ánh sáng và độ tương phản, các giá trị gradient cần phải được chuẩn hóa cục bộ, điều này đòi hỏi việc nhóm các cell thành các vùng liên kết không gian với nhau gọi là các khối. Vector HOG do đó sẽ là tổng hợp các histogram của các cell đã được chuẩn hóa từ tất cả các khối. Các khối sẽ được chia và không tách biệt nhau, nói cách khác mỗi cell sẽ đóng góp thông tin về các hướng (tại các điểm ảnh của nó) nhiều hơn 1 lần.

**Chuẩn hóa khối:** Có nhiều phương pháp có thể được dùng cho việc chuẩn hóa khối. Gọi  $\nu$  là vector cần chuẩn hóa chứa tất cả các histogram của một khối,

$\|v_k\|$  là giá trị chuẩn của nó theo các chuẩn  $k=1, 2$  và  $e$  là một hằng số nhỏ, khi đó các giá trị chuẩn hóa có thể tính bằng 1 trong các công thức sau:

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

$$\text{L1-norm: } f = \frac{v}{\|v\|_1 + e}$$

$$\text{L1-sqrt: } f = \sqrt{\frac{v}{\|v\|_1 + e}}$$

**Phát hiện mặt người với các đặc trưng HOG:** để phát hiện mặt người với các đặc trưng HOG ta tiến hành các bước như sau:

**Bước 1:** Chuẩn bị P mẫu là ảnh mặt người và trích xuất các vector đặc trưng HOG từ các bức ảnh này.

**Bước 2:** Chuẩn bị N mẫu không phải là ảnh mặt người (N rất lớn so với P) và trích xuất các vector HOG từ các ảnh này.

**Bước 3:** Sử dụng một bộ phân loại SVM tuyến tính để học với các vector của các mẫu tích cực (là ảnh mặt người) và tiêu cực (các ảnh không phải mặt người) đã chuẩn bị.

**Bước 4:** Đối với mỗi bức ảnh trong bộ ảnh tiêu cực, sử dụng một cửa sổ trượt di chuyển qua tất cả các vị trí có thể của ảnh input. Tại mỗi vị trí của cửa sổ trượt, tính vector HOG của cửa sổ và đưa vào bộ phân lớp. Nếu bộ phân lớp phân lớp sai một cửa sổ là ảnh mặt thì ghi lại vector tương ứng cùng với xác suất phân lớp.

**Bước 5:** Lấy các mẫu nhận dạng sai ở bước 4 và sắp xếp chúng theo mức xác suất nhận dạng sai và cho bộ phân lớp học lại sử dụng các mẫu sai này.

**Bước 6:** Áp dụng bộ phân lớp đã được học lại với các ảnh cần phát hiện mặt người.

Trong đề tài này, toàn bộ phân phát hiện mặt người sử dụng các dữ liệu phân lớp đã được học trước được cung cấp bởi thư viện dlib. Kết quả của bước này có thể xem ở các hình minh họa sau:

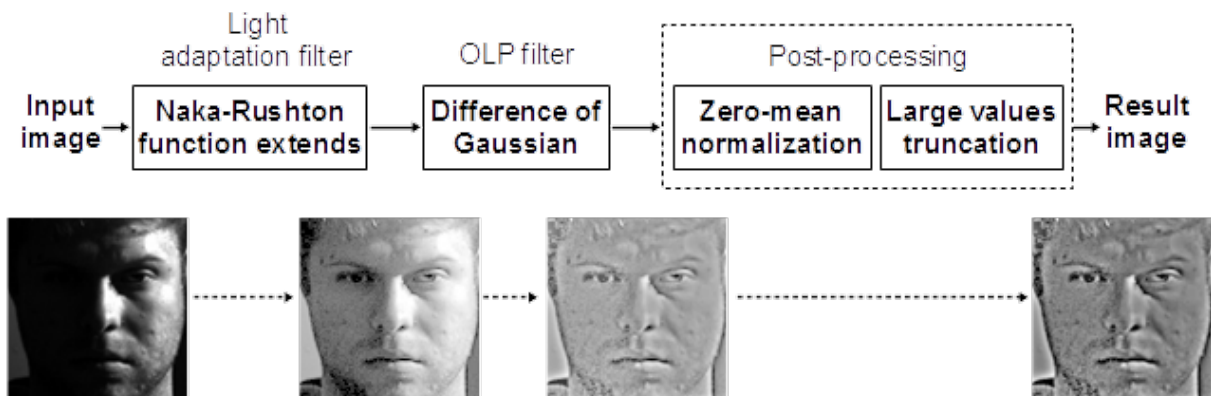


*Hình 2.3: Ảnh input và kết quả phát hiện và căn chỉnh ảnh mặt với HOG.*

### 3. Chuẩn hóa ánh sáng với bộ lọc Retinal filter

Bộ lọc Retinal filter ban đầu được phát triển như một bộ lọc ảnh thông thường dựa vào việc mô phỏng các bước trong hệ thống nhận dạng hình ảnh của con người [8]. Sau đó nó được áp dụng cho bài toán nhận dạng mặt với mục đích chuẩn hóa ánh sáng ở bước tiền xử lý và đã chứng tỏ được tính hiệu quả của mình.

Về cơ bản bộ lọc thực hiện mô phỏng hoạt động của hai lớp chính của hệ thống thị giác của con người là lớp Photoreceptor và Outer Plexiform bằng cách áp dụng liên tiếp hai hàm phi tuyến, bộ lọc DoG và một hàm cắt hậu xử lý như được minh họa trong hình 3.



*Hình 2.4: Các bước trong bộ lọc Retinal filter*

Hàm Naka-Rushton [10] được sử dụng để tăng cường độ tương phản cục bộ của bức ảnh theo công thức sau:

$$Y = \frac{X}{X + X_0}$$

Trong đó X là ảnh input, X<sub>0</sub> là tác nhân điều chỉnh và có giá trị thay đổi với mỗi điểm ảnh còn Y là ảnh output.

Từ ảnh input I ban đầu, bộ lọc Retinal filter sử dụng hai bộ lọc thông thấp liên tiếp với công thức (xuất phát từ công thức của hàm Naka-Rushton ở trên) như sau:

$$I_{la1} = (\max(I) + F_1(p)) \frac{I(p)}{I(p) + F_1(p)}$$

$$I_{la2} = (\max(I_{la1}) + F_2(p)) \frac{I_{la1}(p)}{I_{la1}(p) + F_2(p)}$$

Trong đó các biểu thức  $(\max(I)+F_1(p))$  và  $(\max(I_{la1})+F_2(p))$  đóng vai trò là các nhân tố chuẩn hóa và hàm max là hàm trả về giá trị cường độ sáng lớn nhất của ảnh. Hai biểu thức  $F_1(p)$  và  $F_2(p)$  được tính theo các công thức sau:

$$F_1(p) = I(p) * G_1 + \frac{\bar{I}}{2}$$

$$F_2(p) = I_{la1}(p) * G_2 + \frac{\overline{I_{la1}}}{2}$$

Ký hiệu ‘\*’ là ký hiệu của phép nhân chập,  $\bar{\phantom{x}}$  là ký hiệu của hàm tính trung bình,  $G_1, G_2$  là các bộ lọc Gaussian tương ứng với hai độ lệch chuẩn  $\sigma_1 = 1$  và  $\sigma_2 = 3$  như sau:

$$G_1(x, y) = \frac{1}{2\pi\sigma_1^2} e^{-\frac{x^2+y^2}{2\sigma_1^2}}, \text{ và}$$

$$G_2(x, y) = \frac{1}{2\pi\sigma_2^2} e^{-\frac{x^2+y^2}{2\sigma_2^2}}.$$

Ảnh  $I_{la2}$  sau đó được xử lý bằng một bộ lọc DoG để cải thiện các thuộc tính sắc nét như sau:

$$I_{dog} = DoG * I_{la2},$$

trong đó DoG được tính theo công thức:

$$DoG = \frac{1}{2\pi\sigma_{Ph}^2} e^{-\frac{x^2+y^2}{2\sigma_{Ph}^2}} - \frac{1}{2\pi\sigma_H^2} e^{-\frac{x^2+y^2}{2\sigma_H^2}}.$$

Với  $\sigma_{Ph} = 0.5$  và  $\sigma_H = 4$ . Vì bộ lọc DoG làm giảm độ tương phản toàn cục của bức ảnh nhận được nên một bước cắt bỏ các giá trị quá lớn sẽ được thực hiện sau khi ảnh được chuẩn hóa để có giá trị trung bình bằng 0 như sau:

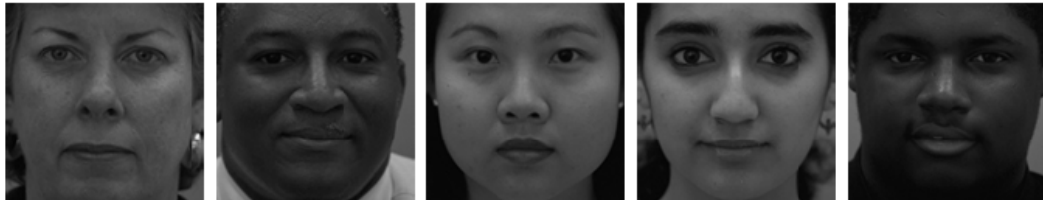
$$I_{norm} = \frac{I_{dog} - \overline{I_{dog}}}{std(I_{dog})} = \frac{I_{dog}}{std(I_{dog})}.$$

$$I_{result}(p) = \begin{cases} \max(Th, |I_{norm}(p)|) & \text{nếu } I_{norm}(p) \geq 0 \\ -\max(Th, |I_{norm}(p)|) & \text{nếu ngược lại} \end{cases}$$

Kết quả của bộ lọc Retina filter trong việc chuẩn hóa ánh sáng của các ảnh mặt được thu nhận trong các điều kiện sáng khác nhau có thể thấy trong hình minh họa 2.3. Có thể thấy rằng kết quả của bộ lọc này khá tốt và đạt được 2 mục đích:

+ đưa các bức ảnh mặt có điều kiện ánh sáng khác nhau về cùng một vùng giá trị khá gần nhau.

+ tăng cường các đường nét quan trọng của khuôn mặt khi áp dụng cho bài toán nhận dạng mặt người, chẳng hạn như lông mày, mắt, mũi và miệng.



(a) Input images



(b) Output images

*Hình 2.5: Kết quả của việc tiền xử lý với bộ lọc Retina filter.*

#### **4. Phương pháp trích chọn đặc trưng LPQ (Local Phase Quantization)**

Phương pháp LPQ dựa trên tính bất biến đối với độ mờ của thành phần phổ pha trong biểu diễn Fourier của ảnh. Phương pháp này sử dụng các thông tin cục bộ về pha được trích xuất bằng cách sử dụng các biến đổi DFT 2 chiều, hay cụ thể hơn, một biến đổi Fourier “ngắn hạn” (short-term), ký hiệu là STFT, được tính trên một vùng lân cận kích thước  $M \times M$  ký hiệu là  $N_x$  tại mỗi vị trí điểm ảnh  $x$  của ảnh  $f(x)$  và được xác định bởi công thức:

$$F(\mathbf{u}, \mathbf{x}) = \sum_{\mathbf{y} \in N_x} f(\mathbf{x} - \mathbf{y}) e^{-j2\pi \mathbf{u}^T \mathbf{y}} = \mathbf{w}_u^f \mathbf{f}_x$$

trong đó  $w_u$  là vector cơ sở của biến đổi DFT 2 chiều tại tần số  $u$ , và  $f_x$  là một vector khác chứa tất cả  $M^2$  mẫu từ  $N_x$ .

Có thể thấy được từ công thức trên một cách hiệu quả khác để cài đặt kỹ thuật STFT là sử dụng các phép nhân chập 2 chiều  $f(x) * e^{-2\pi ju^T x}$  với tất cả các giá trị tần số  $u$ . Vì các hàm cơ sở là có thể tách biệt nên việc tính toán có thể được thực hiện hiệu quả hơn nữa bằng cách sử dụng các phép nhân chập 1 chiều cho các hàng và các cột một cách liên tiếp.

Trong LPQ chỉ có 4 hệ số phức được sử dụng tương ứng với 4 tần số 2 chiều là  $u_1 = [a, 0]^T$ ,  $u_2 = [0, a]^T$ ,  $u_3 = [a, a]^T$ ,  $u_4 = [a, -a]^T$ , trong đó  $a$  là một tần số vô hướng thỏa mãn điều kiện bất biến với độ mờ của ảnh.

Gọi:

$$F_x^c = [F(u_1, x), F(u_2, x), F(u_3, x), F(u_4, x)] , \text{ và} \\ F_x = [\text{Re}\{F_x^c\}, \text{Im}\{F_x^c\}] ,$$

với  $\text{Re}\{\cdot\}$  và  $\text{Im}\{\cdot\}$  là các hàm trả về phần thực và ảo của một số phức tương ứng. Ma trận biến đổi có kích thước  $8xM^2$  sẽ là:

$$W = [\text{Re}\{w_{u1}, w_{u2}, w_{u3}, w_{u4}\}, \text{Im}\{w_{u1}, w_{u2}, w_{u3}, w_{u4}\}]^T .$$

Để các thuộc tính của phương pháp LPQ mạnh hơn cho các bài toán nhận dạng, một quá trình chuẩn hóa sẽ được sử dụng để tách sự tương quan của các hệ số  $F_x$  như sau:

$$G_x = V^T F_x ,$$

trong đó  $V$ , một ma trận trực giao, sẽ được tính bằng cách sử dụng một biến đổi SVD từ ma trận  $D$  như sau:

$$D = U \Sigma V^T .$$

$D$  là ma trận hiệp phương sai của các hệ số  $F_x$  và được tính như sau:

$$D = W C W^T ,$$

trong đó  $C$ , là ma trận hiệp phương sai kích thước  $M \times M$  của các mẫu trong  $N_x$ , được tính bằng công thức sau:

---



$$\mathbf{C} = \begin{pmatrix} 1 & \sigma_{1,2} & \dots & \sigma_{1,M^2} \\ \sigma_{2,1} & 1 & \dots & \sigma_{2,M^2} \\ \dots & \dots & \dots & \dots \\ \sigma_{M^2,1} & \sigma_{M^2,2} & \dots & 1 \end{pmatrix}.$$

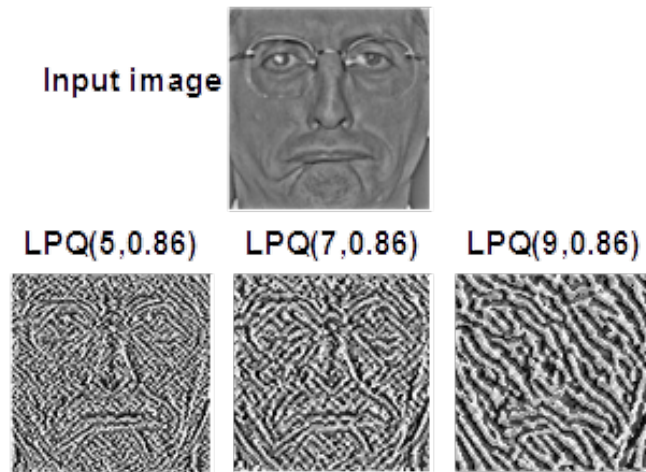
Mỗi phần tử của m trận C, được xác định bằng công thức  $\sigma_{i,j} = \rho \|x_i - x_j\|$  ( $\|\cdot\|$  là ký hiệu của hàm chuẩn hóa  $L_2$  và  $\rho$  là hệ số tương quan giữa các giá trị điểm ảnh lân cận khi giả sử rằng hàm ảnh  $f(x)$  là kết quả của một chuỗi Markov bậc 1 và phương sai của mỗi mẫu bằng 1), chính là hiệp phương sai giữa hai vị trí  $x_i$  và  $x_j$  của  $\mathcal{N}_x$ .

Từ các hệ số đã được tách biệt  $\mathbf{G}_x$ , một ảnh LPQ sẽ được sinh bằng cách lượng tử hóa các thành phần của nó theo công thức:

$$\mathbf{LPQ}_{image} = \sum_{k=1}^8 q_k 2^{k-1},$$

trong đó  $q_k$  là một toán tử lượng tử hóa thực hiện trên thành phần thứ k của  $\mathbf{G}_x$  theo công thức:

$$q_k = \begin{cases} 1 & \text{nếu } g_k \geq 0 \\ 0 & \text{nếu ngược lại} \end{cases}$$

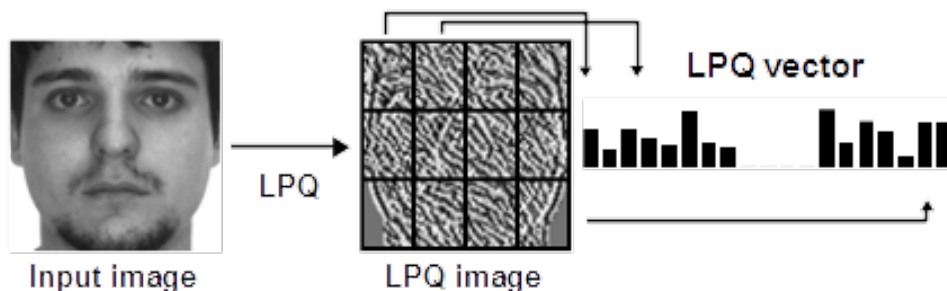


*Hình 2.6: Một số ảnh LPQ thu được từ các toán tử LPQ.*

Trên hình 2.2 là một số ảnh LPQ thu được từ các toán tử LPQ với các tham số khác nhau.

Tiếp đến, từ ảnh LPQ nhận được, các vùng con không giao nhau của ảnh sẽ được sử dụng để tính các histogram cục bộ. Các histogram cục bộ sẽ được ghép

lại với nhau để tạo thành vector LPQ cho ảnh input ban đầu. Toàn bộ quá trình này được minh hoạt trong hình sau:



*Hình 2.7: Các bước trong phương pháp trích chọn đặc trưng LPQ.*

Trong đề tài này, các ảnh mặt trong hệ thống (ảnh huấn luyện, tham chiếu và kiểm thử) đều được áp dụng phương pháp LPQ để biểu diễn dưới dạng các vector đặc trưng và tiến hành nhận dạng.

Về mặt chi tiết, phương pháp LPQ gồm các bước như sau:

Bước 1: Sinh các nhân (kernels) tương ứng với 4 tần số sử dụng các tham số đầu vào là kích thước một cửa sổ STFT (một số lẻ) và số  $\rho$ .

Bước 2: Thực hiện 4 phép nhân chập ảnh input với 4 nhân vừa tạo ra ở bước 1. Sau bước này ta sẽ có 8 thành phần (4 phần thực và 4 phần ảo) hệ số của biến đổi STFT.

Bước 3: Thực hiện chuẩn hóa 8 thành phần nhận được ở bước 3.

Bước 4: Lượng tử hóa 8 thành phần ở bước 3 để nhận được một ảnh LPQ.

Bước 5: Chia ảnh LPQ thành các vùng con hình chữ nhật không giao nhau và tính histogram của các vùng con này.

Bước 6: Ghép các dãy histogram của các vùng con của ảnh LPQ thành một vector duy nhất là biểu diễn LPQ cho ảnh input ban đầu.

## **5. Bộ phân lớp k-NN**

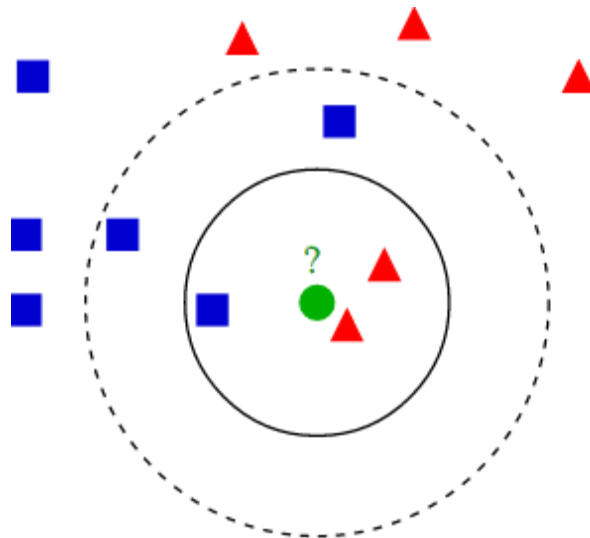
Để xác định danh tính của một bức ảnh mặt đưa vào nhận dạng có thể sử dụng các phương pháp phân lớp khác nhau. Trong đề tài này, chúng tôi đề xuất sử dụng bộ phân lớp kNN (với  $k=1$ ) vì các lý do sau:

+ kNN đơn giản và dễ cài đặt.

+ kNN cho tỉ lệ nhận dạng chấp nhận được.

+ dữ liệu dành cho việc thử nghiệm là dữ liệu thuộc kiểu single sample per person, nghĩa là mỗi người chỉ có 1 ảnh mặt duy nhất trong tập tham chiếu nên việc sử dụng kNN (k=1) là phù hợp.

+ kNN có tốc độ thực hiện nhanh.



*Hình 2.8: Minh họa bộ phân lớp kNN*

So với các phương pháp khác (chẳng hạn như SVM) kNN là một bộ phân lớp không có bước học (xem minh họa ở hình 2.7 ở trên). Khi cần xác định danh tính của một đối tượng cần nhận dạng, kNN sẽ tính độ tương đồng của đối tượng đó so với tất cả các đối tượng có trong tập tham chiếu và sắp xếp theo chiều giảm dần của độ tương đồng. Nếu có đối tượng nào đó trong tập tham chiếu có  $(1 + k/2)$  độ tương đồng nằm trong số k độ tương đồng lớn nhất của danh sách sắp xếp thì ta sẽ gán nhãn (danh tính) của đối tượng đó cho đối tượng nhận dạng.

Trong trường hợp k=1, ta chỉ cần tìm đối tượng trong tập tham chiếu có độ tương đồng lớn nhất so với đối tượng cần nhận dạng và gán danh tính của đối tượng đó cho đối tượng cần nhận dạng.

Độ tương đồng của các bức ảnh mặt trong đề tài này được tính dựa trên khoảng cách Chi Square giữa các vector LPQ biểu diễn cho các bức ảnh. Khoảng cách càng lớn tức là độ tương đồng càng nhỏ.

Trong chương này, các chi tiết về các thuật toán được sử dụng cho hệ thống nhận dạng mặt tự động đã được trình bày. Trong chương sau, đề tài sẽ tập trung vào các vấn đề liên quan tới việc cài đặt, kiểm thử hiệu năng của hệ thống.

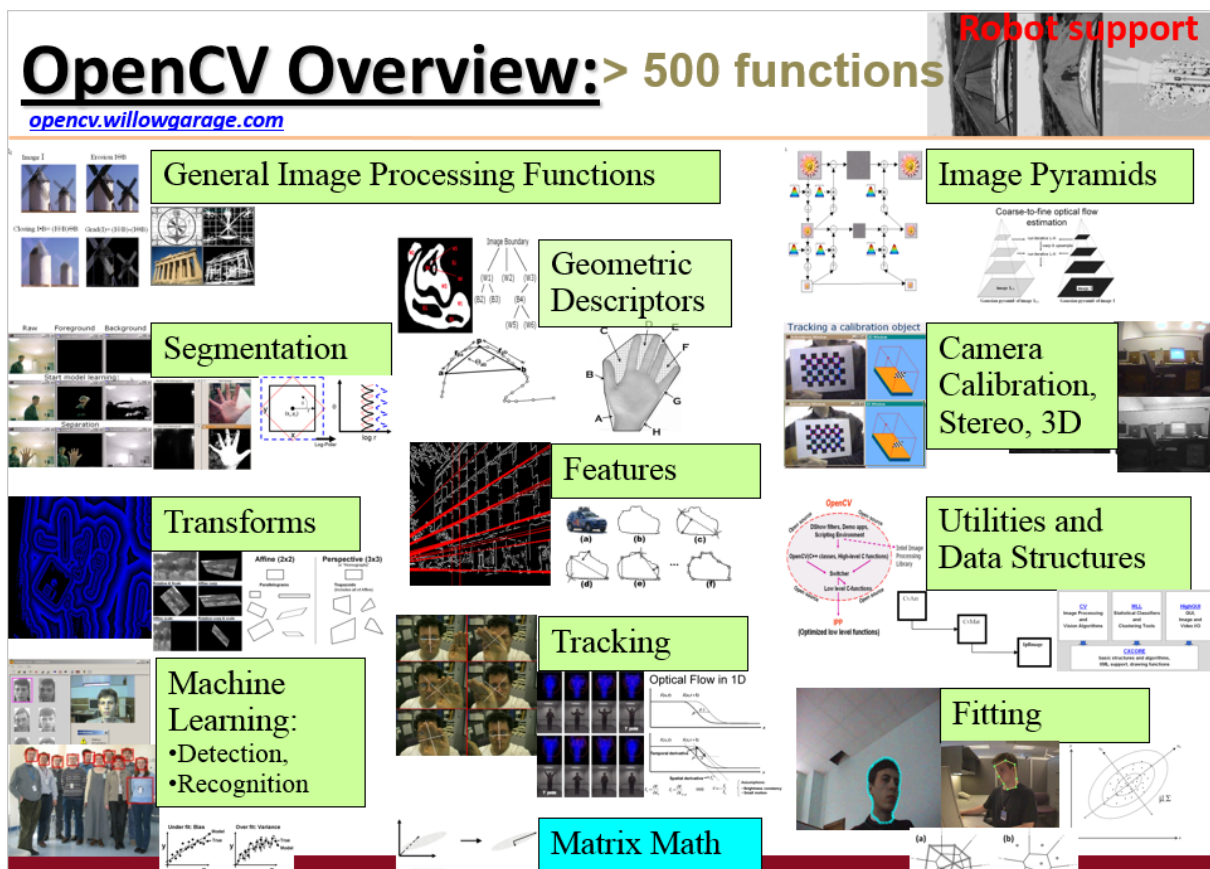
CHƯƠNG 3:

XÂY DỰNG HỆ THỐNG

1. Thư viện OpenCV và dlib

Để cài đặt hệ thống, trong đề tài này chúng tôi lựa chọn ngôn ngữ C++ với bộ công cụ Visual Studio 2015 Update 2 và sử dụng hai thư viện mã nguồn mở là OpenCV và dlib.

1.1. Thư viện OpenCV



Hình 3.1: Cấu trúc của thư viện OpenCV.

OpenCV (<http://opencv.org>) là một thư viện mã nguồn mở cài đặt các thuật toán xử lý ảnh, thị giác máy tính và học máy. OpenCV được viết bằng ngôn ngữ C/C++ và có thể chạy trên hầu hết các nền tảng phần cứng (Intel, Mac, Raspberry Pi, Aduino ..) và phần mềm (Windows, Mac OS, Linux, iOS, Android). Các ngôn ngữ và công cụ lập trình phổ biến như Python, Java, Ruby, Matlab, Objective C,

cũng có thể dễ dàng sử dụng OpenCV trong việc viết và triển khai các ứng dụng thực tế.

Ban đầu thư viện OpenCV do hãng Intel phát triển và bảo trợ (từ những năm cuối của thế kỷ trước – bản alpha được phát hành chính thức vào tháng 2 năm 1999). Sau đó Intel quyết định chuyển OpenCV thành một thư viện mã nguồn mở và từ bỏ dự án này. Ngày nay OpenCV được phát triển và hỗ trợ bởi đông đảo các nhà phát triển, các nhà nghiên cứu trên toàn thế giới. Mỗi ngày lại có một bản vá lỗi với các thuật toán mới được thêm vào. Phiên bản mới nhất của OpenCV là 3.1 và công ty hiện đang bảo trợ cho OpenCV là một công ty của Nga có tên là Itseez.

Mục đích thiết kế của OpenCV là nhắm tới các ứng dụng thời gian thực đòi hỏi các thao tác tính toán phải hết sức hiệu quả về tốc độ thực hiện nên các kỹ thuật tối ưu hóa được sử dụng ở tất cả các mức độ: từ việc tận dụng các kiến trúc đa nhân cho tới việc sử dụng các tập lệnh của CPU mới nhất như SSE 2, AVX, AVX 512 ... OpenCV cũng tận dụng các thư viện lập trình song song mới nhất như TBB (Thread Building Blocks) của Intel, OpenMP và nền tảng CUDA của Nvidia để có thể triệt để khai thác các tài nguyên của hệ thống mà nó chạy trên.

Một mục tiêu quan trọng của OpenCV là cung cấp một hạ tầng các thuật toán thị giác máy tính có giao diện đơn giản và dễ dàng sử dụng để có thể trợ giúp cho các lập trình viên, các nhà nghiên cứu nhanh chóng xây dựng được (hay hiện thực hóa được các ý tưởng của họ) các ứng dụng thực tế. Thư viện OpenCV hiện đã có tới hơn 500 hàm khác nhau trải dài từ các hàm xử lý ảnh cơ bản, giao diện, phân đoạn ảnh, trích chọn đặc trưng, học máy cho tới các hàm hỗ trợ an toàn bảo mật và robotics.

Trên thực tế có khá nhiều các thư viện thị giác máy tính và xử lý ảnh (mã nguồn mở hoặc thương mại) khác nhưng việc lựa chọn OpenCV trong đề tài là do các nguyên nhân sau:

+ OpenCV có hiệu năng cao. Do được tối ưu ở nhiều mức và sử dụng các thư viện tốt nhất (TBB và IPP) của hãng Intel nên các ứng dụng viết bằng OpenCV

cho tốc độ thực hiện rất nhanh và tận dụng được hết các tài nguyên phần cứng của máy tính (chạy song song).

+ OpenCV có nhiều tài liệu hỗ trợ và một cộng đồng các nhà phát triển sẵn sàng hỗ trợ các lỗi debug ([answers.opencv.org](http://answers.opencv.org)). So với các thư viện khác, hiện nay số lượng các sách và tài liệu tham khảo viết về OpenCV chiếm ưu thế vượt trội về số lượng và chất lượng.

+ Cú pháp linh hoạt và đơn giản, dễ sử dụng. Ở các phiên bản đầu tiên, muốn sử dụng OpenCV, các nhà phát triển phải viết các đoạn mã C dài dòng và khó nhớ. Nhưng hiện tại, cú pháp của OpenCV đã được cải tiến rất nhiều (do học tập từ Matlab). Điều này làm giảm đáng kể công sức của các lập trình viên khi không cần phải nhớ các cú pháp phức tạp như trước.

Về mặt cấu trúc, OpenCV là một tập hợp các thư viện (xem thêm ở hình 3.1), sau đây là danh sách các thư viện cốt lõi:

+ core: thư viện con chứa tất cả các kiểu dữ liệu cơ sở và các hàm thực hiện các thao tác cơ sở.

+ imgproc: thư viện xử lý ảnh chứa các thao tác xử lý ảnh cơ bản, chẳng hạn như các phép nhân chập và các bộ lọc không gian.

+ highgui: thư viện chứa các hàm hỗ trợ giao diện đồ họa chẳng hạn như việc hiển thị các ảnh hoặc nhận các dữ liệu input đơn giản.

+ video: thư viện chứa các hàm thực hiện việc đọc dữ liệu, ghi dữ liệu đối với các file video.

+ features2d: các thuật toán phát hiện, mô tả và đối sánh các điểm bất biến (keypoints).

+ objdetect: thư viện chứa các thuật toán để phát hiện các đối tượng đặc thù chẳng hạn như mặt người hoặc người đi bộ. Chúng ta cũng có thể sử dụng thư viện này để xây dựng các bộ huấn luyện để phát hiện các đối tượng khác trong các ứng dụng cụ thể.





*Hình 3.2: Cấu trúc của thư viện dlib.*

Ngoài thư viện OpenCV, chúng tôi còn sử dụng dlib (<http://dlib.net>), một thư viện mã nguồn mở khác cho việc cài đặt hệ thống. Khác với mục đích của OpenCV là cung cấp một hạ tầng thuật toán cho các ứng dụng xử lý ảnh và thị giác máy tính, dlib được thiết kế cho các ứng dụng học máy và trí tuệ nhân tạo với các thư viện con chính sau:

- + classification: các kỹ thuật phân lớp chủ yếu dựa trên hai phương pháp cơ sở là kNN và SVM.

- + data transformation: các thuật toán biến đổi dữ liệu nhằm giảm số chiều, loại bỏ các dữ liệu dư thừa và tăng cường tính khác biệt (discriminant) của các đặc điểm được giữ lại.

- + clustering: các kỹ thuật phân cụm.

- + regression: các kỹ thuật hồi qui

- + structured prediction: các thuật toán dự đoán có cấu trúc

- + Markov Random Fields: các thuật toán dựa trên các trường Markov ngẫu nhiên.

Cụ thể trong đề tài này, phần phát hiện và căn chỉnh ảnh mặt người một cách tự động sẽ được cài đặt bằng cách sử dụng thư viện dlib với dữ liệu huấn luyện đã được cung cấp trước.

Đoạn mã phát hiện mặt người và căn chỉnh cụ thể như sau:

```
#include <dlib/image_processing/frontal_face_detector.h>
#include <dlib/image_processing/render_face_detections.h>
#include <dlib/image_processing.h>
#include <dlib/gui_widgets.h>
#include <dlib/image_io.h>
#include <iostream>
```

```
using namespace dlib;

int main(int argc, char** argv)
{
    try
    {
        if (argc == 1)
        {
            cout << "Error while loading files for face detection." << endl;
            return 0;
        }

        // Use a face detector to detect faces in input image and draw bounds
        // for every face in the given image.
        frontal_face_detector detector = get_frontal_face_detector();
        // Loading a shape_predictor for detecting landmark positions for each face
        // region inside the input image.
        shape_predictor sp;
        deserialize(argv[1]) >> sp;

        image_window win, win_faces;
        // make an iteration for every input image
        for (int i = 2; i < argc; ++i)
        {
```

```
cout << "processing image " << argv[i] << endl;

array2d<rgb_pixel> img;

load_image(img, argv[i]);

// To detect small faces, we enlarge the input image

pyramid_up(img);

// get and process the list of detected faces.

// make boudary around each face

std::vector<rectangle> dets = detector(img);

cout << "Number of faces detected: " << dets.size() << endl;

// Now we will go ask the shape_predictor to tell us the pose of

// each face we detected.

std::vector<full_object_detection> shapes;

for (unsigned long j = 0; j < dets.size(); ++j)

{

    full_object_detection shape = sp(img, dets[j]);

    cout << "number of parts: " << shape.num_parts() << endl;

    cout << "pixel position of first part: " << shape.part(0) << endl;

    cout << "pixel position of second part: " << shape.part(1) << endl;

    // You get the idea, you can get all the face part locations if

    // you want them. Here we just store them in shapes so we can

    // put them on the screen.

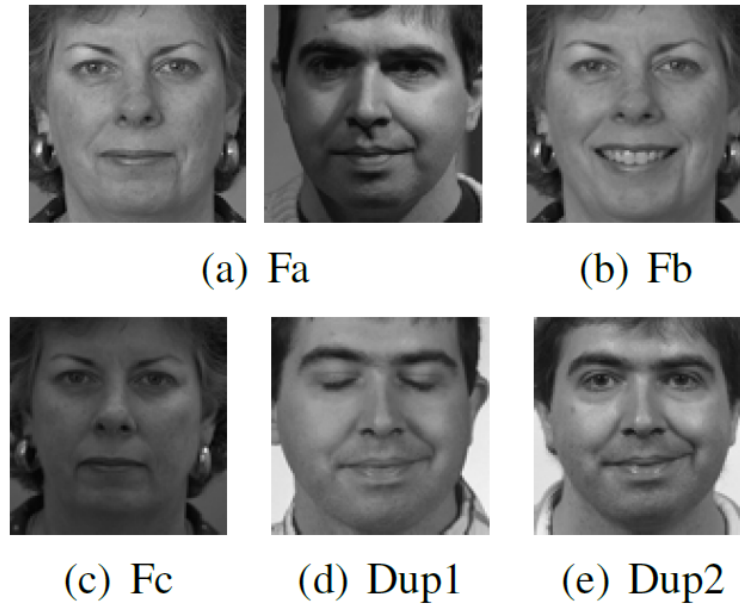
    shapes.push_back(shape);

}
```

```
    }  
  
    // display face on the output screen  
  
    win.clear_overlay();  
  
    win.set_image(img);  
  
    win.add_overlay(render_face_detections(shapes));  
  
    // extract each face and frontalize it  
  
    dlib::array<array2d<rgb_pixel> > face_chips;  
  
    extract_image_chips(img, get_face_chip_details(shapes), face_chips);  
  
    win_faces.set_image(tile_images(face_chips));  
  
  
    cout << "Press enter for processing the next image..." << endl;  
  
    cin.get();  
  
    }  
  
    }  
  
catch (exception& e)  
  
{  
  
    cout << "\nError!" << endl;  
  
    cout << e.what() << endl;  
  
    }  
  
}
```

## **2. Dữ liệu và giao thức đánh giá hiệu năng của hệ thống**

Để đánh giá hiệu năng (tỉ lệ nhận dạng) của hệ thống, trong đề tài này chúng tôi sử dụng cơ sở dữ liệu ảnh mặt FERET [11], một trong những cơ sở dữ liệu công cộng được sử dụng rộng rãi nhất trên thế giới cho bài toán nhận dạng mặt.



*Hình 3.3: Một số ảnh mặt người từ cơ sở dữ liệu FERET.*

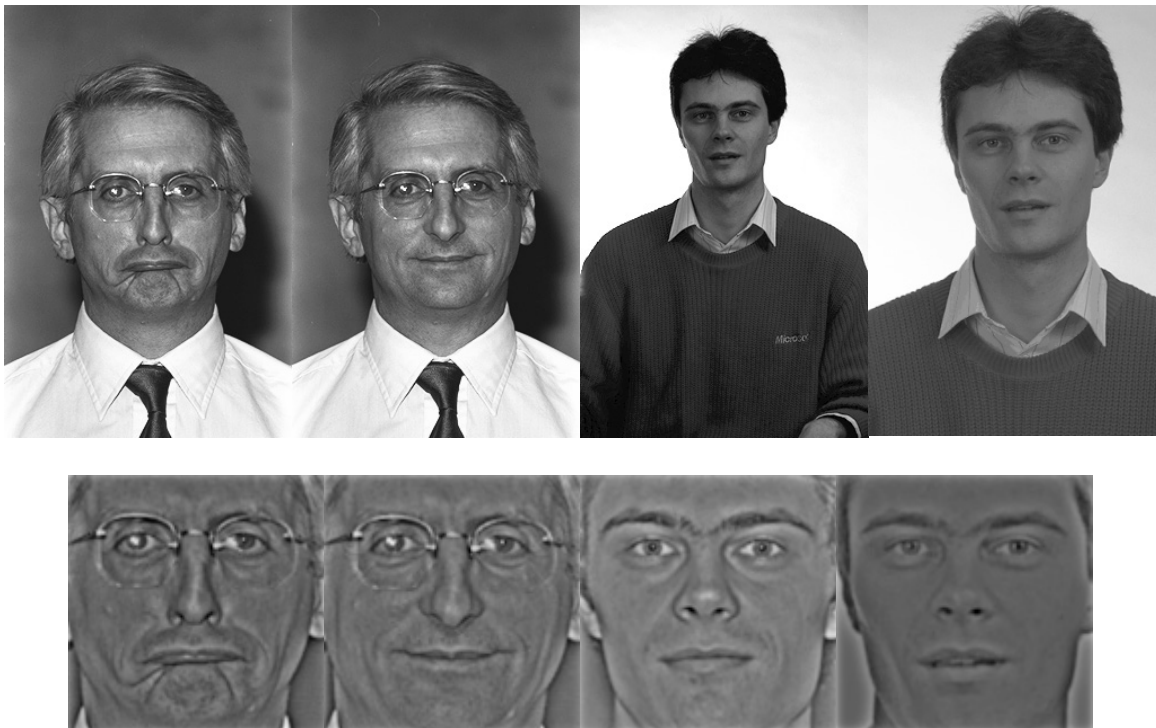
Với số ảnh mặt của hơn 1000 người được thu nhận ở các điều kiện khác nhau, FERET có năm tập ảnh mặt người ở góc nhìn thẳng (frontal face images) được đặt tên lần lượt là Fa (hình 3.3-a), Fb (hình 3.3-b), Fc (hình 3.3-c), Dup 1 (hình 3.3-d) và Dup 2 (hình 3.3-e). Tập Fa gồm 1196 ảnh được sử dụng làm tập tham chiếu. Các tập Fb, Fc, Dup 1 và Dup 2 được dùng làm các tập kiểm thử và gồm có lần lượt 1195, 194, 722 và 234 ảnh của các đối tượng khác nhau. Các ảnh của tập Fb được thu nhận với các biểu hiện khác nhau của khuôn mặt trong khi các ảnh của tập Fc được chụp ở các điều kiện ánh sáng khác nhau. Các ảnh trong tập Dup 1 và Dup 2 được chụp sau các ảnh trong tập Fa các khoảng thời gian lần lượt là 1 và 2 năm. Trong số 4 tập ảnh test, hai tập Dup 1 và Dup 2 là khó nhất vì chúng tương ứng với các thay đổi của ảnh khuôn mặt khi bị tác động bởi yếu tố thời gian, một trong những thách thức lớn nhất của nhận dạng mặt người.

Trong đề tài này, chúng tôi sử dụng giao thức chuẩn của cơ sở dữ liệu ảnh FERET và so sánh kết quả với các công bố gần đây nhất trên cùng bộ dữ liệu. Các ảnh sau khi được căn chỉnh qua bước phát hiện mặt tự động sẽ được điều chỉnh về kích thước 128x128 và được chuẩn hóa ánh sáng bằng bộ lọc Retina filter. Tiếp đến chúng sẽ được biểu diễn thành các vector đặc trưng LPQ bằng cách áp dụng kỹ thuật LPQ được trình bày trong chương 2. Cuối cùng bộ phân lớp kNN sẽ được

sử dụng để đưa ra định danh tương ứng của ảnh cần nhận dạng. Tỷ lệ nhận dạng được tính bằng số ảnh nhận dạng đúng trên tổng số ảnh cần nhận dạng và so sánh với kết quả của các phương pháp liên quan.

### 3. Kết quả nhận dạng

**Kết quả phát hiện mặt người:** sử dụng các đặc trưng HOG, chương trình đã phát hiện được ảnh mặt trong tất cả ảnh dữ liệu của cả năm tập ảnh của cơ sở dữ liệu FERET. Trong hình sau là một số kết quả tiêu biểu:



Hình 3.4: Một số kết quả về phát hiện mặt người trong CSDL FERET.

**Kết quả nhận dạng:** Kết quả nhận dạng của hệ thống được so sánh với kết quả của các phương pháp khác trong bảng 3.1.

Method	Fb	Fc	Dup 1	Dup 2	Avg
LBP[12]	97.0	79.0	66.0	64.0	82.7
LGBPHS[13]	98.0	97.0	74.0	71.0	87.8
GEWC[14]	96.3	99.5	78.8	77.8	89.3
<b>Hệ thống đề xuất</b>	<b>96.3</b>	<b>98.5</b>	<b>79.4</b>	<b>78.6</b>	<b>89.5</b>
LPQ	97.7	97.9	79.5	78.6	90.2
G-LQP[15]	99.9	100	93.2	91.0	97.0
MBC-F[16]	99.7	99.5	93.6	91.5	97.0
SLK-RKR[17]	99.7	99.5	96.3	94.4	98.1

Bảng 3.1: So sánh tỉ lệ nhận dạng của hệ thống đề xuất với các kết quả đã công bố khác.

Kết luận đầu tiên có thể rút ra từ bảng so sánh 3.1 là hệ thống nhận dạng mặt tự động mà đề tài xây dựng có hiệu năng khá tốt so với hệ thống bán tự động (với bước phát hiện và căn chỉnh ảnh mặt được thực hiện thủ công dựa trên các tọa độ mắt của người – dòng có tiêu đề **LPQ**).

Xét về ảnh hưởng của các điều kiện khác nhau, có thể thấy trong bảng 3.1 hệ thống đề xuất làm việc tốt với các ảnh trong tập Fb và Fc nhưng tỉ lệ nhận dạng đối với các tập Dup 1 và Dup 2 chưa tốt lắm. Điều này chứng tỏ hệ thống có thể làm việc tốt đối với các thay đổi về cảm xúc khuôn mặt và ánh sáng nhưng chưa giải quyết được vấn đề về thời gian. So sánh với một số phương pháp bán tự động đã công bố, kết quả của hệ thống cao hơn (trong các bài báo [12] - [14]).

Cũng từ bảng so sánh 3.1 ta cũng có thể quan sát được rằng khi so sánh với các hệ thống nhận dạng mặt hàng đầu ([15]-[17], đều là bán tự động) thì hệ thống đề xuất có kết quả thua kém khá xa. Điều này là hợp lý vì các hệ thống hàng đầu đó được xây dựng dựa trên 2 kỹ thuật quan trọng khác:

+ Phương pháp trích xuất đặc trưng sử dụng các ảnh đa phân giải (multiresolution) hoặc đa tỉ lệ (multiscale) nên sẽ cho một biểu diễn mặt có tính đặc trưng cao hơn, dẫn tới kết quả nhận dạng tốt hơn.

+ Các hệ thống này đều sử dụng một kỹ thuật học để giảm số chiều, loại bỏ thông tin dư thừa trong các vector đặc trưng nên tỉ lệ nhận dạng cao hơn hệ thống mà đề tài xây dựng.

## KẾT LUẬN

Bài toán nhận dạng mặt người tự động là một bài toán còn nhiều thách thức, đặc biệt là trong các điều kiện ảnh mặt bị tác động bởi các yếu tố như ánh sáng, thời gian, hướng ... Trong đề tài này, một hệ thống nhận dạng mặt tự động đã được đề xuất, xây dựng và đánh giá hiệu năng đối với cơ sở dữ liệu ảnh mặt người FERET.

Hệ thống đề xuất sử dụng các đặc trưng HOG cho bước phát hiện mặt người trong ảnh input. Kỹ thuật Retinal filter được sử dụng để chuẩn hóa ánh sáng cho các ảnh đã được căn chỉnh nhằm giảm các ảnh hưởng xấu của ánh sáng. Ở bước trích chọn đặc trưng, phương pháp LPQ được sử dụng để trích xuất các đặc trưng quan trọng nhất cho việc nhận dạng. Cuối cùng bộ phân lớp kNN và hàm Chi Square được dùng để xác định danh tính cho bức ảnh cần nhận dạng.

Các kết quả so sánh cho thấy hệ thống đề xuất cho kết quả khá tốt đối với các ảnh có hướng thẳng. Tuy nhiên so với các hệ thống bán tự động sử dụng các tọa độ của mắt (được xác định một cách thủ công) thì tỉ lệ nhận dạng của hệ thống vẫn còn chưa bằng. Điều này không quá khó hiểu vì đối với các hệ thống bán tự động, các bức ảnh được căn chỉnh tốt hơn và do đó sẽ cho tỉ lệ nhận dạng cao hơn so với một hệ thống tự động hoàn toàn.

Trong tương lai, các tác giả mong muốn thử nghiệm hệ thống với các bộ dữ liệu được thu nhận trong các điều kiện không có kiểm soát (về ánh sáng, cảm xúc của khuôn mặt, hướng ...) để đánh giá được chính xác hơn hiệu năng của hệ thống trong các tình huống khác nhau. Một hướng phát triển nữa của đề tài là áp dụng một thuật toán giảm số chiều sau bước trích chọn đặc điểm để có thể giảm số chiều của các vector đặc điểm xuống nhằm đạt được tốc độ nhận dạng nhanh hơn và có thể cho tỉ lệ nhận dạng tốt hơn.

Một hướng khác có thể giúp cải thiện hiệu năng nhận dạng cho hệ thống là áp dụng các phương pháp kết hợp LPQ với các đặc trưng khác nhau Local Binary Patterns hoặc áp dụng LPQ với các phép phân tích hình ảnh đa phân giải (multi



resolution) hoặc đa tỉ lệ (Multiscale) như các phép biến đổi Gabor wavelets, Log Gabor wavelets hoặc Monogenic Filter.

**TÀI LIỆU THAM KHẢO**

- [1] S. Z. Li and A. K. Jain, Eds., *Handbook of Face Recognition*, 2nd Edition. Springer, 2011.
- [2] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, “Face recognition: A literature survey,” *Acm Comput. Surv. CSUR*, vol. 35, no. 4, pp. 399–458, 2003.
- [3] M. A. Turk and A. P. Pentland, “Face recognition using eigenfaces,” 1991, pp. 586–591.
- [4] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, “Eigenfaces vs. fisherfaces: Recognition using class specific linear projection,” *Pattern Anal. Mach. Intell. IEEE Trans. On*, vol. 19, no. 7, pp. 711–720, 1997.
- [5] T. Ahonen, A. Hadid, and M. Pietikainen, “Face description with local binary patterns: Application to face recognition,” *Pattern Anal. Mach. Intell. IEEE Trans. On*, vol. 28, no. 12, pp. 2037–2041, 2006.
- [6] J. G. Daugman, “Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters,” *Opt. Soc. Am. J. Opt. Image Sci.*, vol. 2, pp. 1160–1169, 1985.
- [7] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 2005, vol. 1, pp. 886–893.
- [8] N. S. Vu and A. Caplier, “Illumination-robust face recognition using retina modeling,” in *Image Processing (ICIP), 2009 16th IEEE International Conference on*, 2009, pp. 3289–3292.
- [9] T. Ahonen, E. Rahtu, V. Ojansivu, and J. Heikkila, “Recognition of blurred faces using local phase quantization,” in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, 2008, pp. 1–4.
- [10] K.-I. Naka and W. A. Rushton, “S-potentials from luminosity units in the retina of fish (Cyprinidae),” *J. Physiol.*, vol. 185, no. 3, pp. 587–599, 1966.
- [11] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, “The FERET evaluation methodology for face-recognition algorithms,” *Pattern Anal. Mach. Intell. IEEE Trans. On*, vol. 22, no. 10, pp. 1090–1104, 2000.
- [12] T. Ahonen, A. Hadid, and M. Pietikäinen, “Face recognition with local binary patterns,” *Comput. Vis.-ECCV 2004*, pp. 469–481, 2004.
- [13] W. Zhang, S. Shan, W. Gao, X. Chen, and H. Zhang, “Local Gabor binary pattern histogram sequence (LGBPHS): A novel non-statistical model for face representation and recognition,” 2005, vol. 1, pp. 786–791.
- [14] W. Deng, J. Hu, and J. Guo, “Gabor-Eigen-Whiten-Cosine: a robust scheme for face recognition,” *Anal. Model. Faces Gestures*, pp. 336–349, 2005.

- [15] S. U. Hussain, T. Napoléon, and F. Jurie, “Face Recognition using Local Quantized Patterns.”
- [16] M. Yang, L. Zhang, S. C.-K. Shiu, and D. Zhang, “Monogenic Binary Coding: An Efficient Local Feature Extraction Approach to Face Recognition,” *IEEE Trans. Inf. Forensics Secur.*, vol. 7, no. 6, pp. 1738–1751, Dec. 2012.
- [17] M. Yang, L. Zhang, S. C.-K. Shiu, and D. Zhang, “Robust Kernel Representation With Statistical Local Features for Face Recognition,” *IEEE Xplore*, 2013.