

BỘ GIÁO DỤC VÀ ĐÀO TẠO
ĐẠI HỌC HUẾ
TRƯỜNG ĐẠI HỌC KHOA HỌC

PHẠM HỒNG NGUYỄN

**NHẬN DẠNG ĐỐI TƯỢNG
SỬ DỤNG THUẬT TOÁN ADABOOST**

EBOOKBKMT.COM
TÀI LIỆU KỸ THUẬT MIỄN PHÍ

CHUYÊN NGÀNH: KHOA HỌC MÁY TÍNH
MÃ SỐ: 60.48.01

**LUẬN VĂN THẠC SĨ KHOA HỌC
CÔNG NGHỆ THÔNG TIN**

NGƯỜI HƯỚNG DẪN KHOA HỌC
TS. NGUYỄN ĐĂNG BÌNH

Huế, 2009

MỞ ĐẦU

Nhận dạng hay nhận biết một đối tượng là một khả năng tự nhiên của con người cũng như các loài vật, theo một cách bản năng nhất mọi loài vật qua cảm nhận từ các cơ quan cảm giác như: mắt, mũi, miệng, tay,... bằng hành động nhìn, nghe, ngửi,... có thể dễ dàng cảm nhận ngay được đối tượng đang tiếp xúc với mình là cái gì, quen hay lạ. Chính vì vậy, có thể nói rằng cơ thể mỗi một loài vật chính là một hệ thống nhận dạng tối ưu nhất.

Với sự phát triển của khoa học công nghệ và nhất là khoa học về robot thì càng ngày mong ước tạo ra một hệ thống nhận dạng máy học có khả năng tương tự và trội hơn hệ thống nhận dạng sinh học là một khát khao cháy bỏng của các nhà khoa học.

Chính vì vậy từ những năm 70 đến nay các nhà khoa học đã không ngừng nghiên cứu phát triển bài toán nhận dạng đối tượng, từ nhận dạng ký tự sơ khai cho đến hệ thống nhận dạng cho các robot thông minh hiện nay khoa học đã tiến những bước dài. Ứng dụng của bài toán này trong cuộc sống, khoa học kỹ thuật càng ngày càng rộng rãi trên nhiều lĩnh vực. Tuy nhiên, dù đã có nhiều thành công nhưng chỉ giới hạn do đây là bài toán có rất nhiều vấn đề hóc búa và đòi hỏi tri thức tổng hợp của nhiều lĩnh vực.

Một hệ thống nhận dạng đối tượng là hệ thống nhận vào một ảnh hoặc một đoạn video (một chuỗi các ảnh). Qua xử lý tính toán hệ thống xác định được vị trí đối tượng trong ảnh (nếu có) và xác định là đối tượng nào trong số những đối tượng hệ thống đã được biết (qua quá trình học) hay là đối tượng mới. Yêu cầu đặt ra đối với hệ thống nhận dạng đối tượng là độ chính xác cao vì vậy hệ thống đòi hỏi phải có các đặc trưng tốt. Hệ thống phải biết chọn đặc trưng như thế nào để có thể biểu diễn tốt được thông tin đối tượng cần nhận dạng. Đồng thời, đặc trưng phải được tính toán nhanh để không làm chậm công việc nhận dạng.Thêm vào đó, hệ thống phải có phương pháp học hiệu quả, có khả năng nhận dạng tốt các mẫu mới chứ không chỉ làm tốt trên các mẫu đã học.

Để có thể đạt được các mục tiêu trên, đã có nhiều cách tiếp cận được đưa ra: William T. Freeman và Michal Roth [26] sử dụng đặc trưng biểu đồ định hướng, Bowden và Sarhadi [18] sử dụng mô hình phân tán điểm phi tuyến,...Nhìn chung, trong các cách tiếp cận trên đều có chung một hạn chế là không thể đạt được sự cân đối giữa khả năng nhận dạng và thời gian xử lý.

AdaBoost (Adaptive Boosting) là một giải thuật máy học được đề xuất bởi Yoav Freund và Robert Schapire [23] vào năm 1995. Nó là một tiếp cận boosting và hoạt động trên nguyên tắc kết hợp tuyến tính các bộ phân loại yếu để có một bộ phân loại mạnh. Do những ưu điểm đó nên đã có rất nhiều nghiên cứu về nhận dạng đối tượng đã chọn AdaBoost để làm công cụ thực hiện.

Với những lý do trên tôi chọn đề tài ***“Nhận dạng đối tượng sử dụng thuật toán AdaBoost”***

Luận văn chia thành ba chương, nội dung chủ yếu của Chương 1 tìm hiểu những kiến thức tổng quan về nhận dạng đối tượng và việc ứng dụng AdaBoost để nhận dạng đối tượng, cụ thể là nhận dạng khuôn mặt người.

Trong Chương 2 tìm hiểu về thuật toán AdaBoost và mô hình áp dụng thuật toán vào việc nhận dạng đối tượng

Chương 3 tập trung về vấn đề thực nghiệm nhận dạng khuôn mặt, bao gồm chuẩn bị bộ dữ liệu, các bước tiến hành, lượng hóa kết quả huấn luyện và kết quả kiểm thử

Phần cuối của luận văn là kết luận và hướng phát triển của đề tài.

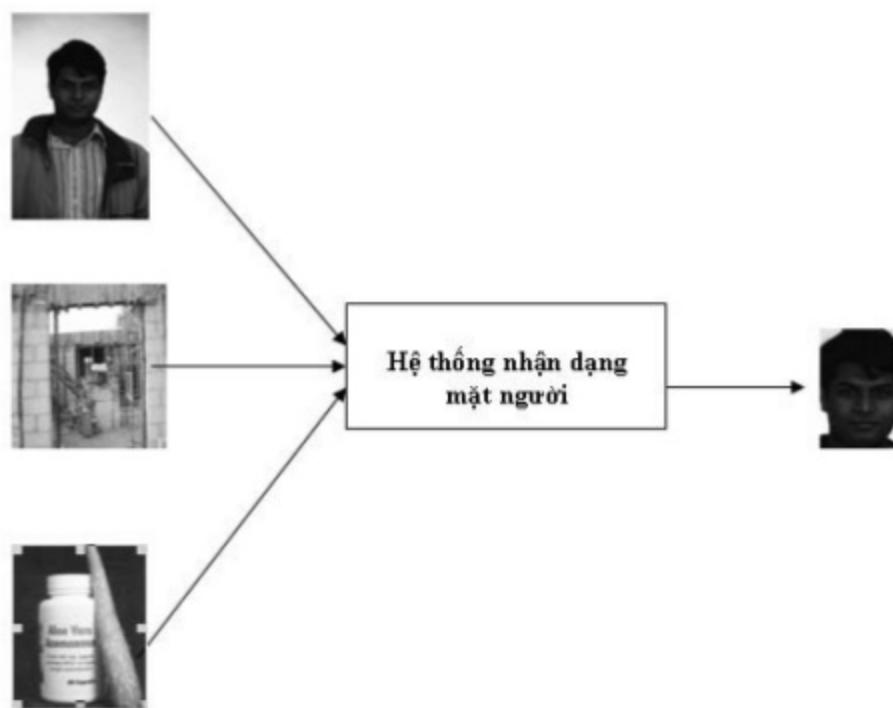
Chương 1

TỔNG QUAN

Chương này giới thiệu tổng quan về nhận dạng đối tượng, các hướng nghiên cứu về lĩnh vực này, sơ lược về những công trình nghiên cứu về nhận dạng có sử dụng thuật toán AdaBoost

1.1 Định nghĩa bài toán nhận dạng đối tượng

Nhận dạng đối tượng (Object recognition) lĩnh vực thuộc “machine learning” nghiên cứu việc tìm một đối tượng trong một ảnh hay video cho trước, đó là việc phát hiện lớp đối tượng cụ thể với các lớp đối tượng khác của hệ thống.



Hình 1.1: Minh họa về nhận dạng mặt người

Sự phát triển khoa học kỹ thuật công nghệ đã thúc đẩy cho việc đưa bài toán nhận dạng đối tượng vào ứng dụng thực tiễn với mức độ tinh vi ngày càng cao, ta có thể thấy rất nhiều ứng dụng trong nhận dạng đối tượng phổ biến hiện nay như: các camera theo dõi trong các trung tâm lớn, hệ thống nhận dạng tội phạm, nhận dạng vân tay, công nghệ robot,... Vì vậy, nhận dạng đối tượng là một lĩnh vực đang rất được quan tâm và có tính ứng dụng cao.

1.2 Các khó khăn của việc nhận dạng

Đối với con người thì việc nhận dạng các đối tượng trong ảnh là việc không phải là phức tạp; tuy nhiên, đối với một hệ thống nhân tạo thì nhận ra một đối tượng từ một ảnh đòi hỏi phải giải quyết được rất nhiều vấn đề hóc búa. Chính vì thế vấn đề này vẫn đang được nhiều nhóm trên thế giới quan tâm nghiên cứu. Khó khăn của bài toán nhận dạng có thể kể ra như sau:

a. Tư thế, góc chụp: Ảnh chụp khuôn mặt có thể thay đổi rất nhiều bởi vì góc chụp giữa camera và khuôn mặt. Chẳng hạn như: chụp thẳng, chụp xéo bên trái 45° hay xéo bên phải 45°, chụp từ trên xuống, chụp từ dưới lên, v.v...). Với các tư thế khác nhau, các thành phần trên khuôn mặt như mắt, mũi, miệng có thể bị khuất một phần hoặc thậm chí khuất hết.



Hình 1.2: Minh họa sự thay đổi góc chụp

b. Sự xuất hiện hoặc thiếu một số thành phần: Các thành phần biểu tả một đối tượng có thể xuất hiện hoặc không trong ảnh làm cho bài toán nhận dạng càng trở nên khó hơn rất nhiều.



Hình 1.3: Minh họa sự thiếu thành phần

c. Sự biến dạng của đối tượng: Biến dạng của đối tượng có thể làm ảnh hưởng đáng kể lên các thông số của đối tượng đó. Chẳng hạn, cùng một khuôn mặt một người, nhưng có thể sẽ rất khác khi họ cười hoặc sợ hãi,v.v...



Hình 1.4: Minh họa sự biến dạng

d. Sự che khuất: Đối tượng có thể bị che khuất bởi các đối tượng khác.



Hình 1.5: Minh họa sự che khuất

e. Sự phức tạp của hình nền: Hình nền phức tạp sẽ khiến việc nhận dạng trở nên khó khăn.



Hình 1.6: Minh họa hình nền phức tạp

f. Điều kiện của ảnh: Ảnh được chụp trong các điều kiện khác nhau về: chiếu sáng, về tính chất camera (máy kỹ thuật số, máy hồng ngoại,v.v...) ảnh hưởng rất nhiều đến chất lượng ảnh.



Hình 1.7: Minh họa độ sáng khác nhau

1.3 Các ứng dụng nhận dạng đối tượng

Bài toán nhận dạng đối tượng có thể áp dụng rộng rãi trong nhiều ứng dụng thực tế khác nhau. Đó chính là lý do mà bài toán này hấp dẫn rất nhiều nhóm nghiên cứu trong thời gian dài. Các ứng dụng liên quan đến nhận dạng đối tượng có thể kể ra như:

- Hệ thống phát hiện tội phạm: camera được đặt tại một số điểm công cộng như: siêu thị, nhà sách, trạm xe buýt, sân bay, v.v... Khi phát hiện được sự xuất hiện của các đối tượng là tội phạm, hệ thống sẽ gửi thông điệp về cho trung tâm xử lý.

- Hệ thống theo dõi nhân sự trong một đơn vị: giám sát giờ ra vào của từng nhân viên và chấm công.

- Hệ thống giao tiếp người máy: thay thế việc tương tác giữa người và máy theo những cách truyền thống như: bàn phím, chuột, v.v... Thay vào đó là sử dụng các giao tiếp trực quan: biểu cảm khuôn mặt, dấu hiệu, cử chỉ bằng tay.

- Hệ thống tìm kiếm thông tin trên ảnh, video dựa trên nội dung (chỉ mục theo người). Chẳng hạn như: đài truyền hình Việt Nam (VTV) có một kho dữ liệu video tin tức khá lớn cần tìm kiếm nhanh những đoạn video nào có G. Bush hoặc Bin Laden.

- Các hệ thống bảo mật dựa trên thông tin trắc sinh học: mặt người, vân tay, v.v... thay vì xác nhận mật khẩu, khóa, v.v...

- Phần mềm nhận dạng chữ in, đặc biệt dành riêng cho ngôn ngữ tiếng Việt, dịch vụ Nhận dạng chữ in.

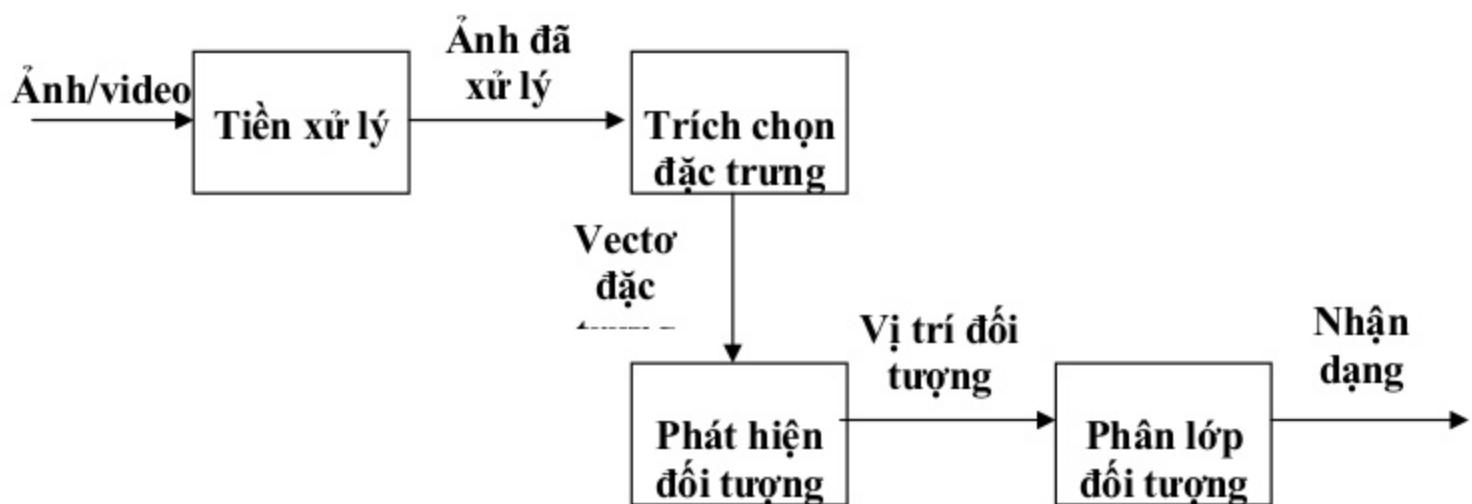
- Công nghệ nhận dạng quản lý giao thông quản lý hình ảnh của xe và biển số xe thay đổi (không khớp với cơ sở dữ liệu), giám sát bãi đỗ xe, phương tiện giao thông tại các khu vực nhạy cảm như các sân bay, bến cảng.

1.4 Tổng quan kiến trúc một hệ thống nhận dạng đối tượng

Một hệ thống nhận dạng đối tượng thông thường xử lý bốn bước sau:

- Thu nhận tín hiệu (hình ảnh) và tiền xử lý

- Trích chọn đặc trưng (Đặc trưng extraction)
- Phát hiện đối tượng (detection)
- Phân lớp đối tượng (classification)



Hình 1.8: Các bước trong hệ thống nhận dạng mặt người

Ảnh đầu vào được thu nhận và tiền xử lý sau đó các ảnh được qua công đoạn tiền xử lý nhằm tăng độ chính xác cho hệ thống. Các ảnh sau đó được trích chọn đặc trưng để tạo ra các vectơ đặc trưng trong bước rút trích đặc trưng. Những véc-tor đặc trưng này sẽ là dữ liệu đầu vào cho một mô hình đã được huấn luyện trước. Phát hiện đối tượng: dò tìm và định vị những vị trí đối tượng xuất hiện trong ảnh hoặc trên các chuỗi ảnh của video. Những đối tượng thu được qua bước phát hiện sẽ tiếp tục được phân lớp thành từng lớp riêng biệt để nhận dạng.

1.5 Tổng quan các phương pháp trong bài toán nhận dạng đối tượng

Có rất nhiều hướng tiếp cận và phương pháp khác nhau liên quan đến vấn đề nhận dạng. Theo Ming-Hsuan Yang [11], có thể phân loại thành bốn hướng tiếp cận

chính: Hướng tiếp cận dựa trên cơ sở tri thức, hướng tiếp cận dựa trên các đặc trưng bất biến, hướng tiếp cận dựa trên đối sánh mẫu, và Hướng tiếp cận dựa vào diện mạo xuất hiện, phương pháp này thường dùng một mô hình máy học nên còn được gọi là phương pháp dựa trên cơ sở máy học (machine learning-based).

a. Hướng tiếp cận dựa trên cơ sở tri thức: Mã hóa các hiểu biết của con người về đối tượng thành các luật. Thông thường các luật mô tả quan hệ của các đặc trưng.

Trong hướng tiếp cận này, các luật sẽ phụ thuộc rất lớn vào tri thức của những tác giả nghiên cứu. Đây là hướng tiếp cận dạng từ trên xuống. Dễ dàng xây dựng các luật cơ bản để mô tả các đặc trưng của đối tượng và các quan hệ tương ứng. Ví dụ, một khuôn mặt thường có hai mắt đối xứng nhau qua trực thăng đứng ở giữa khuôn mặt và có một mũi, một miệng. Các quan hệ của các đặc trưng có thể được mô tả như quan hệ về khoảng cách và vị trí. Thông thường các tác giả sẽ trích đặc trưng của khuôn mặt trước tiên để có được các ứng viên, sau đó các ứng viên này sẽ được nhận dạng thông qua các luật để biết ứng viên nào là khuôn mặt (face) và ứng viên nào không phải khuôn mặt (none-face). Thường áp dụng quá trình xác định để giảm số lượng nhận dạng sai.

Một vấn đề khá phức tạp khi dùng hướng tiếp cận này là làm sao chuyển từ tri thức con người sang các luật một cách hiệu quả. Nếu các luật này quá chi tiết thì khi nhận dạng có thể nhận dạng thiếu các đối tượng có trong ảnh, vì những đối tượng này không thể thỏa mãn tất cả các luật đưa ra. Nhưng các luật tổng quát quá thì có thể chúng ta sẽ nhận dạng lầm một vùng nào đó không phải là đối tượng mà lại nhận dạng là đối tượng và cũng khó khăn mở rộng yêu cầu từ bài toán để nhận dạng các đối tượng có nhiều tư thế khác nhau. Có một số nghiên cứu từ rất sớm đã áp dụng phương pháp này như của Kanade [17].

b. Hướng tiếp cận dựa trên đặc trưng bất biến: Mục tiêu các thuật toán đi tìm các đặc trưng mô tả cấu trúc đối tượng, các đặc trưng này sẽ không thay đổi khi vị trí đối tượng, vị trí đặt thiết bị thu hình hoặc điều kiện ánh sáng thay đổi.

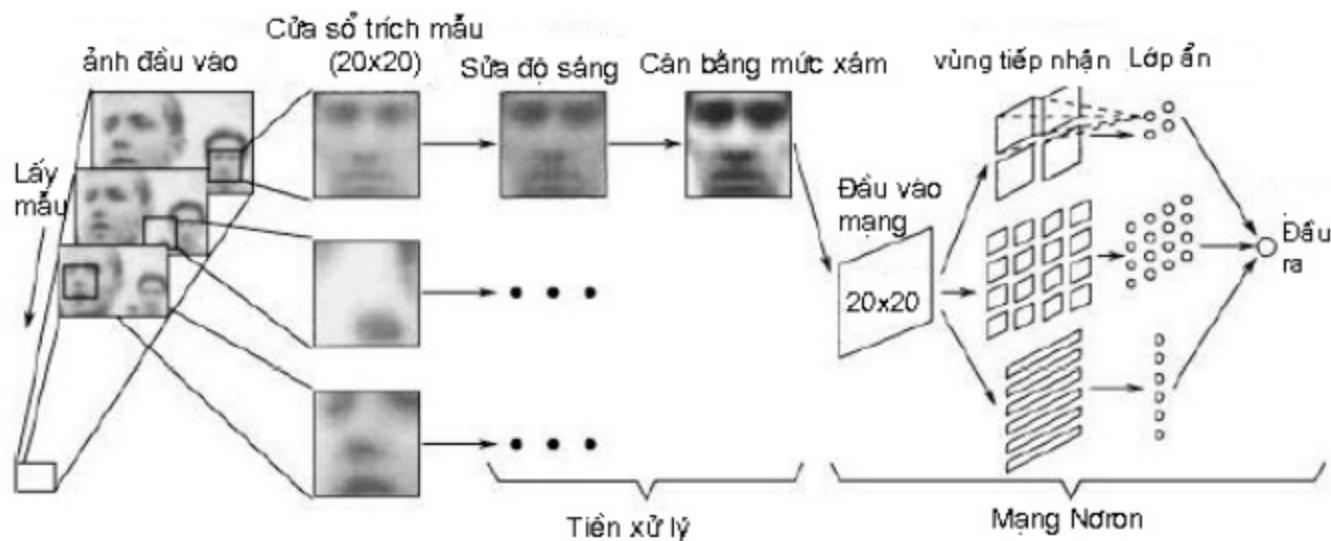
Đây là hướng tiếp cận theo kiểu bottom-up. Các tác giả cố gắng tìm các đặc trưng không thay đổi của đối tượng để nhận dạng đối tượng. Dựa trên nhận xét thực tế, con người dễ dàng nhận biết các đối tượng trong các tư thế khác nhau và điều kiện ánh sáng khác nhau, thì phải tồn tại các thuộc tính hay đặc trưng không thay đổi. Có nhiều nghiên cứu đầu tiên nhận dạng các đặc trưng đối tượng rồi chỉ ra có đối tượng trong ảnh hay không. Ví dụ: Các đặc trưng như: lông mày, mắt, mũi, miệng, và đường viền của tóc được trích bằng phương pháp xác định cạnh. Trên cơ sở các đặc trưng này, thực hiện việc xây dựng một mô hình thống kê để mô tả quan hệ của các đặc trưng này và nhận dạng sự tồn tại của khuôn mặt trong ảnh.

Một vấn đề của các thuật toán theo hướng tiếp cận đặc trưng cần phải điều chỉnh cho phù hợp điều kiện ánh sáng, nhiễu, và bị che khuất. Đôi khi bóng của đối tượng sẽ tạo thêm cạnh mới, mà cạnh này lại rõ hơn cạnh thật sự của nó, vì thế nếu dùng cạnh để nhận dạng sẽ gặp khó khăn. Những công trình sử dụng hướng tiếp cận này có thể kể đến là: K. C. Yow và R. Cipolla 1997 [8], T. K. Leung 1995 [22].

c. Hướng tiếp cận dựa trên so khớp mẫu: Trong so khớp mẫu, các mẫu chuẩn của đối tượng (thường được chụp thẳng) sẽ được nhận dạng trước hoặc nhận dạng các tham số thông qua một hàm. Từ một ảnh đưa vào, tính các giá trị tương quan so với các mẫu chuẩn. Thông qua các giá trị tương quan này mà các tác giả quyết định có hay không có tồn tại đối tượng trong ảnh. Hướng tiếp cận này có lợi thế là rất dễ cài đặt, nhưng không hiệu quả khi tỷ lệ, tư thế, và hình dáng thay đổi. Nhiều độ phân giải, đa tỷ lệ, các mẫu con, và các mẫu biến dạng được xem xét thành bát biến về tỷ lệ và hình dáng. I. Craw 1992 [6] đã áp dụng một mẫu cứng trong khi A. Lanitis 1995[1] sử dụng một mẫu có thể biến dạng trong bước phát hiện khuôn mặt.

d. Hướng tiếp cận dựa trên diện mạo: Trái ngược với các phương pháp so khớp mẫu với các mẫu đã được định nghĩa trước bởi những chuyên gia, các mẫu trong hướng tiếp cận này được học từ các ảnh mẫu. Một cách tổng quát, các phương pháp theo hướng tiếp cận này áp dụng các kỹ thuật theo hướng xác suất thống kê và

máy học để tìm những đặc tính liên quan của đối tượng và không phải là đối tượng. Các đặc tính đã được học ở trong hình thái các mô hình phân bố hay các hàm biệt số nên dùng có thể dùng các đặc tính này để nhận dạng đối tượng. Đồng thời, bài toán giảm số chiều thường được quan tâm để tăng hiệu quả tính toán cũng như hiệu quả nhận dạng.



Hình 1.9: Hệ thống của Rowley [17] và các đồng sự

Các tiếp cận khác trong hướng tiếp cận dựa trên diện mạo là tìm một hàm biệt số (mặt phẳng quyết định, siêu phẳng để tách dữ liệu, hàm ngưỡng) để phân biệt hai lớp dữ liệu: đối tượng và không phải là đối tượng. Bình thường, các mẫu ảnh được chiếu vào không gian có số chiều thấp hơn, rồi sau đó dùng một hàm biệt số (dựa trên các độ đo khoảng cách) để phát hiện, hoặc xây dựng mặt quyết định phi tuyến bằng mạng neural đa tầng. Hoặc dùng SVM (Support Vector Machine) và các phương thức kernel, chiếu hoàn toàn các mẫu vào không gian có số chiều cao hơn để dữ liệu bị rời rạc hoàn toàn và ta có thể dùng một mặt phẳng quyết định phát hiện các mẫu đối tượng và không phải là đối tượng. Có nhiều mô hình máy học được áp dụng trong hướng tiếp cận này: Eigenface (M. Turk và A. Pentland 1991 [10]), Mô hình dựa trên phân phối (K. Sung and T. Poggio 1998 [9]), Mạng Nơ-ron (H. Rowley 1998 [20]), Support Vector Machine (E. Osuna et al 1997 [12]), Phân lớp Bayes (H. Schneiderman và T. Kanade 1998 [19]), Mô hình Markov ẩn (A.

Rajagopalan et al 1998 [16]), và các mô hình tăng cường AdaBoost của P. Viola và M. Jones [13],[14]; FloatBoost do Stan Z. Li và Zhen Qiu Zhang 2004 [20],[21].

1.6 Tổng quan về nhận dạng đối tượng với thuật toán AdaBoost

Nhận dạng đối tượng bằng thuật toán theo kỹ thuật boosting với thuật toán AdaBoost đã và đang được coi là một phương pháp hiệu quả, vừa đảm bảo tốc độ nhận dạng vừa có độ chính xác cao.

Violas và Jones [15],[16] áp dụng thành công trong việc xây dựng một bộ phát hiện khuôn mặt nhanh bằng cách dùng thuật toán AdaBoost kết hợp các bộ phân loại yếu sử dụng các đặc trưng Haar-like theo mô hình cascade. Đầu tiên, một hình ảnh được chia nhỏ thành các ảnh con (sub-image), mỗi ảnh được biểu thị bởi một vector đặc trưng. Violas và Jones đã trích xuất bốn đặc trưng bằng cách sử dụng bộ mặt nạ như hình dưới để phát hiện khuôn mặt. Với mỗi mặt nạ, tổng cấp xám của các điểm ảnh trong vùng sáng trừ tổng cấp xám trong cùng tối được coi như là một đặc trưng. Vì vậy với cách phân chia thành các ảnh con sẽ có nhiều hơn một triệu các đặc trưng và mỗi đặc trưng được tính rất nhanh.

Phương pháp của Viola và Jones có tốc độ xử lý khá nhanh, gấp khoảng từ 10-15 lần so với các phương pháp hiện tại và tỷ lệ chính xác hơn 80% trên ảnh xám. Có thể nói, phương pháp này được công nhận là một trong những bước đột phá lớn của lĩnh vực máy học tính chung và phát hiện khuôn mặt nói riêng.

Schneiderman và Kanade [19] cải tiến thành công bằng cách dùng wavelet để trích đặc trưng rồi xây dựng hệ thống học với thuật toán Adaboost, dựa trên xác suất và histogram để xác định khuôn mặt người. Tỷ lệ chính xác trên 90%. Chen ước lượng tham số ảnh để điều chỉnh ánh sáng cho phù hợp với các mẫu bằng SVM, sau cùng dùng Adaboost để xác định khuôn mặt người với tư thế chụp thẳng, tỷ lệ chính xác là 89,7%.

Ngoài Violas và Jones, Schneiderman và Kanade thì Agarwal và Roth [2] thành công trong việc kết hợp sử dụng kỹ thuật học nhận dạng Winnow để nhận dạng xe hơi.

Một cách tiếp cận nhận dạng đối tượng khác đã được giới thiệu bởi Dorko và Schmid [5] đó là dựa trên việc xây dựng và lựa chọn các thành phần bất biến và áp dụng AdaBoost để học cách phân loại.

Một phương pháp phân loại đối tượng mới đã được giới thiệu bởi Thureson và Carlsson trong [7] là dựa trên histogram của ảnh. Yếu tố này được tính toán dựa trên việc kết hợp của bộ gồm vị trí và hướng gradient của mẫu. Cách tiếp cận này dựa trên việc kết hợp các ảnh đã được phân loại từ các ảnh huấn luyện, nó đòi hỏi phải được phân đoạn trước để giảm độ phức tạp.

Carbonetto và các đồng sự [3] tiếp cận nhận dạng đối tượng theo ngũ cành dựa trên một hình ảnh phân đoạn kết hợp thuật toán huấn luyện AdaBoost. Họ đính kèm nhãn đến các vùng hình ảnh và học một mô hình không gian của các mối quan hệ giữa chúng.

Chương 2

NGHIÊN CỨU THUẬT TOÁN ADABOOST VÀ ÁP DỤNG VÀO NHẬN DẠNG ĐỐI TƯỢNG

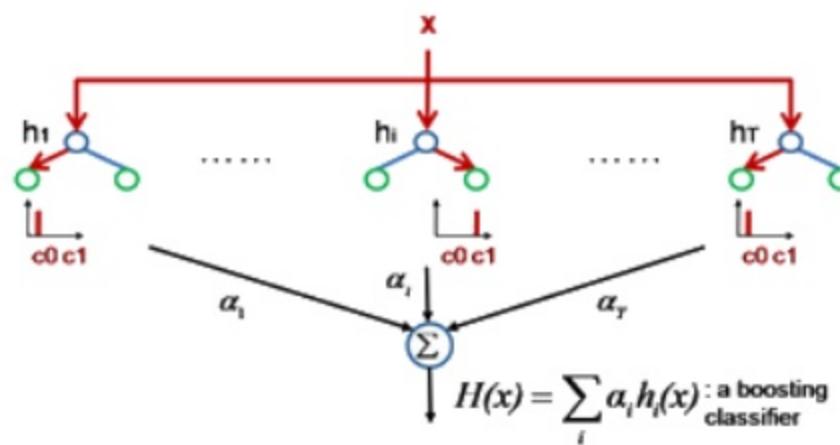
Chương này trình bày về thuật toán AdaBoost là thuật toán ứng dụng trong nhiều trong bài toán nhận dạng vì có nhiều ưu điểm là nhanh chóng, đơn giản và dễ dàng cài đặt. Nó không có các tham số để điều chỉnh và không cần tri thức về bộ học yếu và do đó có thể được linh hoạt kết hợp với phương pháp nào cho việc tìm kiếm bộ phân loại yếu. Với những ưu điểm đó, ta nghiên cứu áp dụng vào việc phát hiện và nhận dạng đối tượng.

2.1 Thuật toán AdaBoost

2.1.1 Tiếp cận Boosting

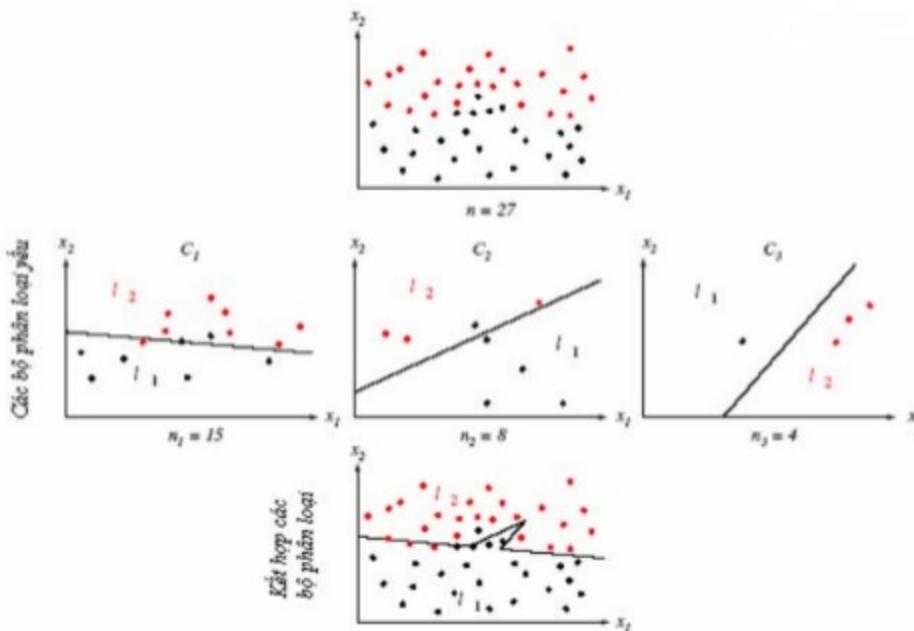
Boosting là kỹ thuật dùng để tăng độ chính xác cho các thuật toán học (Learning algorithm). Nguyên lý cơ bản của nó là kết hợp các bộ phân loại yếu

thành một bộ phân loại mạnh. Trong đó, bộ phân loại yếu là các bộ phát hiện đơn giản chỉ cần có độ chính xác trên 50%. Bằng cách này, chúng ta nói bộ phát hiện đã được “boost”.



Hình 2.1: Minh họa hoạt động theo kỹ thuật boosting

Xét một bài toán phát hiện 2 lớp (mẫu cần nhận dạng sẽ được phân vào 1 trong 2 lớp) với D là tập huấn luyện gồm có n mẫu. Trước tiên, chúng ta sẽ chọn ngẫu nhiên ra n_1 mẫu từ tập D ($n_1 < n$) để tạo tập D_1 . Sau đó, chúng ta sẽ xây dựng bộ phân loại yếu đầu tiên C_1 từ tập D_1 . Tiếp theo, chúng ta xây dựng tập D_2 để huấn luyện bộ phát hiện C_2 . D_2 sẽ được xây dựng sao cho một nửa số mẫu của nó được phát hiện đúng bởi C_1 và nửa còn lại bị phát hiện sai bởi C_1 . Bằng cách này, D_2 chứa đựng những thông tin bổ sung cho C_1 . Nay giờ chúng ta sẽ xây dựng bộ huấn luyện C_2 từ D_2 . Tiếp theo, chúng ta sẽ xây dựng tập D_3 từ những mẫu không được phát hiện tốt bởi sự kết hợp giữa C_1 và C_2 : những mẫu còn lại trong D mà C_1 và C_2 cho kết quả khác nhau. Như vậy, D_3 sẽ gồm những mẫu mà C_1 và C_2 hoạt động không hiệu quả. Sau cùng, chúng ta sẽ huấn luyện bộ phát hiện C_3 từ D_3 . Nay giờ chúng ta đã có một bộ phân loại mạnh: sự kết hợp C_1, C_2 và C_3 . Khi tiến hành nhận dạng một mẫu X , kết quả sẽ được quyết định bởi sự thỏa thuận của 3 bộ C_1, C_2 và C_3 : Nếu cả C_1 và C_2 đều phân X vào cùng một lớp thì lớp này chính là kết quả phát hiện của X ; ngược lại, nếu C_1 và C_2 phân X vào 2 lớp khác nhau, C_3 sẽ quyết định X thuộc về lớp nào.



Hình 2.2: Boosting

2.1.2 Thuật toán AdaBoost

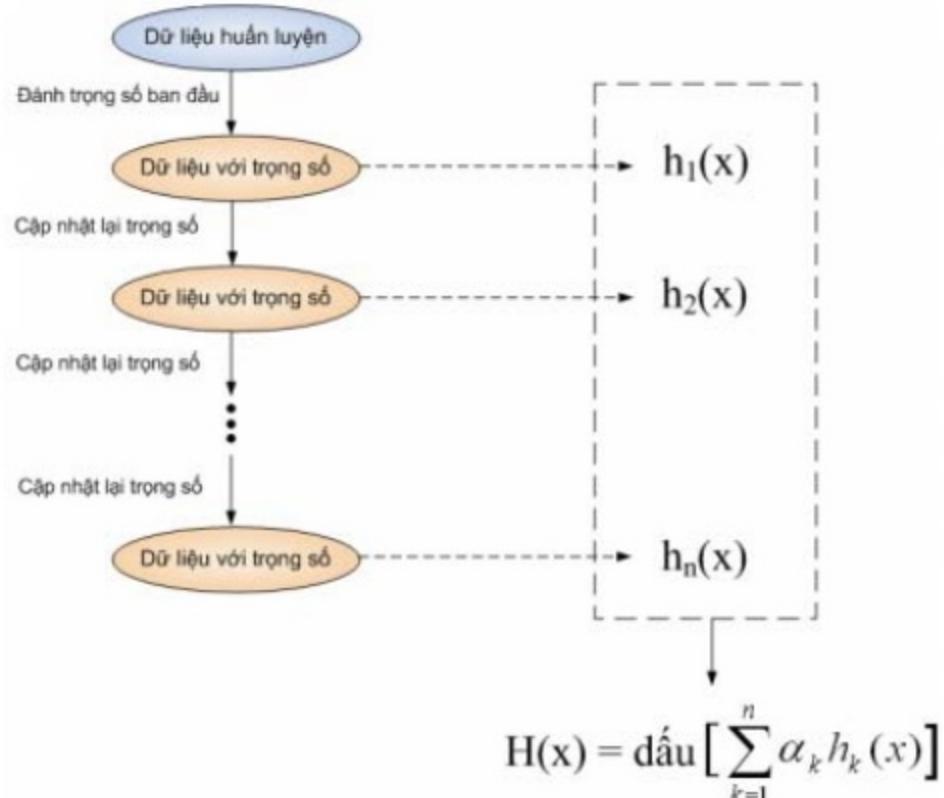
Adaboost (Adaptive Boost) là một tiếp cận boosting được Freund và Schapire [23] đưa ra vào năm 1995 đã giải quyết nhiều thực tế khó khăn của các thuật toán boosting trước đó.

Adaboost là một cải tiến của tiếp cận boosting, Adaboost sử dụng thêm khái niệm trọng số (weight) để đánh dấu các mẫu khó nhận dạng. Trong quá trình huấn luyện, cứ mỗi bộ phân loại yếu được xây dựng, thuật toán sẽ tiến hành cập nhật lại trọng số để chuẩn bị cho việc xây dựng bộ phân loại yếu kế tiếp là: tăng trọng số của các mẫu bị nhận dạng sai và giảm trọng số của các mẫu được nhận dạng đúng bởi bộ phân loại yếu vừa xây dựng. Bằng cách này, các bộ phân loại yếu tiếp theo có thể tập trung vào các mẫu mà các bộ phân loại yếu trước nó chưa làm tốt. Sau cùng, các bộ phân loại yếu sẽ được kết hợp tùy theo mức độ “tốt” của chúng để tạo dựng nên bộ phân loại mạnh.

Có thể hình dung một cách trực quan như sau: để biết một ảnh có phải là bàn tay hay không, ta hỏi T người (tương đương với T bộ phân loại yếu xây dựng từ T vòng lặp của boosting), đánh giá của mỗi người (tương đương với một bộ phân loại yếu) chỉ cần tốt hơn ngẫu nhiên một chút (tỉ lệ sai dưới 50%). Sau đó, ta sẽ đánh

trọng số cho đánh giá của từng người (thể hiện qua hệ số α), người nào có khả năng đánh giá tốt các mẫu khó thì mức độ quan trọng của người đó trong kết luận cuối cùng sẽ cao hơn những người chỉ đánh giá tốt được các mẫu dễ. Việc cập nhật lại trọng số của các mẫu sau mỗi vòng boosting chính là để đánh giá độ khó của các mẫu (mẫu càng có nhiều người đánh giá sai là mẫu càng khó).

Thuật toán có đầu vào là một tập huấn luyện $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ trong đó mỗi x_i thuộc về một số tên miền hay không gian X và mỗi nhãn y_i thuộc tập nhãn Y ta giả sử rằng $Y = \{-1, +1\}$. AdaBoost gọi một thuật toán học yếu trong một dãy vòng lặp từ $t=1, \dots, T$. Một trong những ý tưởng chính của thuật toán là để duy trì một phân phối, hoặc thiết lập của trọng số qua tập huấn luyện. Ban đầu, tất cả trọng số được thiết lập bằng nhau, nhưng trên mỗi vòng lặp, trọng số của các mẫu phát hiện không chính xác được tăng lên để bộ học yếu tập trung vào các mẫu khó trong tập huấn luyện.



Hình 2.3: Bộ phân loại mạnh $H(x)$ xây dựng bằng AdaBoost

Công việc của bộ học yếu là để tìm một bộ phân loại yếu $h_t : X \rightarrow \{-1, +1\}$ tương ứng với bộ phân phối D_t . Độ tốt của một giả thuyết yếu được đo bằng hệ số lỗi của nó:

$$\varepsilon_t = \Pr_{i \sim D_t} [h_t(x_i) \neq y_i] = \sum_{i: h_t(x_i) \neq y_i} D_t(i)$$

Lưu ý là lỗi được đo đối với phân phối D_t mà trên đó các bộ học yếu được huấn luyện. Trong thực tế, các bộ học yếu có thể là một thuật toán có thể dùng trọng số D_t trên các mẫu huấn luyện. Ngoài ra, khi điều này là không thể, một tập hợp con của các mẫu đang huấn luyện có thể được được lấy mẫu theo D_t và các mẫu được đánh lại trọng số có thể được dùng để huấn luyện bộ học yếu.

Thuật toán: AdaBoost

- **Đầu vào:** $(x_1, y_1), \dots, (x_m, y_m)$ với $x_i \in X$ và $y_i \in Y = \{-1, +1\}$

1: Khởi tạo: $D_1(i) = 1 / m$

2: For $t = 1, \dots, T$:

- Huấn luyện bộ học yếu sử dụng phân phối D_t
- Chọn bộ phân loại yếu $h_t : X \rightarrow \{-1, +1\}$ với lỗi:

$$\varepsilon_t = \Pr_{i \sim D_t} [h_t(x_i) \neq y_i]$$

- Chọn: $\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \varepsilon_t}{\varepsilon_t} \right)$
- Cập nhật: $D_{t+1,i} = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t}, & (x_k) = y_i \\ e^{\alpha_t}, & (x_k) \neq y_i \end{cases}$

$$= \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

Với Z_t là nhân tố bình thường hóa

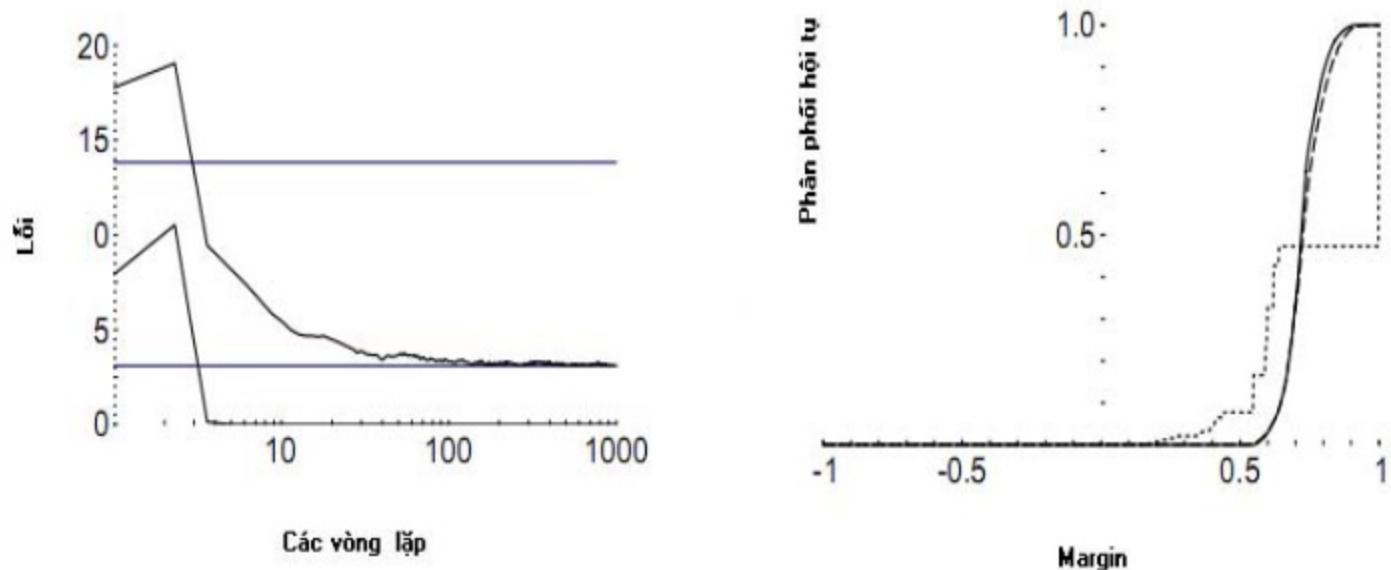
- Đầu ra: $H(x) = \text{sign} \left(\sum_{t=1}^T \alpha_t h_t(x) \right)$

Một khi bộ phân loại yếu đã được nhận, AdaBoost sẽ chọn một tham số như trên. Dễ dàng thấy α_t quan trọng được gán cho h_t . Chú ý rằng $\alpha_t \geq 0$ nếu $\varepsilon_t \leq 1/2$ (chúng ta có thể giả định mà không mất tính tổng quát).

Phân phối D_t ở bước cập nhật tiếp theo sẽ dùng các luật như trình bày giải thuật của thuật toán ở trên, tác dụng của các luật này là làm tăng trọng số của các mẫu chưa được phát hiện bởi h_t và giảm trọng số của các mẫu phát hiện chính xác vì vậy trọng số tập trung vào các mẫu khó.

Phát hiện thu được cuối cùng H là tập hợp của T bộ phân loại yếu trong đó α_t là trọng số được gán cho h_t .

Schapire và Singer[15] đã chỉ ra cách mà AdaBoost có thể được mở rộng để xử lý bộ phân loại yếu với giá trị đầu ra thực và hệ số dự đoán tin cậy. Đó là với mỗi X thì h_t sẽ cho ra $h_t(x) \in \mathbb{R}$ có gán nhãn (-1 hoặc +1) và giá trị của nó $|h_t(x)|$ sẽ đo độ tin cậy của dự đoán.



Hình 2.4: Đồ thị lỗi và phân phối hội tụ AdaBoost của Schapire

2.1.3 Phân tích lỗi huấn luyện

Lý thuyết cơ bản nhất của AdaBoost tập trung vào việc giảm lỗi huấn luyện.

Lỗi ε_t của h_t là $\frac{1}{2} - \gamma_t$; Freund và Schapire [24], [25] chứng minh rằng lỗi huấn luyện

(các mẫu lỗi trong tập huấn luyện) của phát hiện H vào khoảng:

$$\prod_t \left[2\sqrt{\varepsilon_t(1-\varepsilon_t)} \right] = \prod_t \sqrt{1-4\gamma_t^2} \leq \exp\left(-2\sum_t \gamma_t^2\right)$$

Vì vậy, nếu mỗi bộ phân loại yếu hơi tốt hơn so với ngẫu nhiên do bởi $\gamma_t > \gamma$ với mỗi $\gamma > 0$ thì lỗi huấn luyện cũng giảm theo cấp số nhân.

Đây là sự kế thừa các thuật toán boosting trước đó, tuy nhiên các thuật toán boosting trước đây thì mức độ ràng buộc là thấp hơn. AdaBoost thích nghi ở chỗ nó điều chỉnh theo tỷ lệ lỗi của những bộ phân loại yếu do vậy thuật toán có tên là AdaBoost là viết tắt của Adaptive.

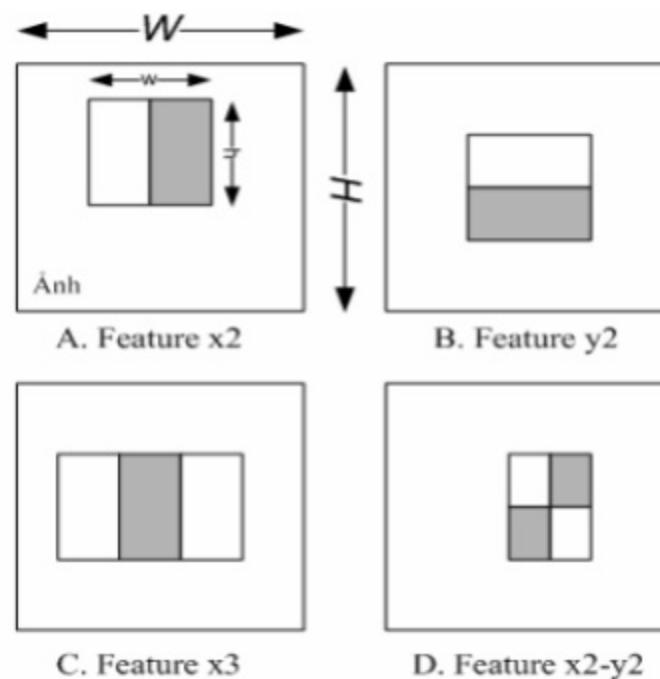
2.2 Nghiên cứu áp dụng AdaBoost vào nhận dạng đối tượng

2.2.1 Phát hiện đối tượng bằng thuật toán AdaBoost

2.2.1.1 Đặc trưng

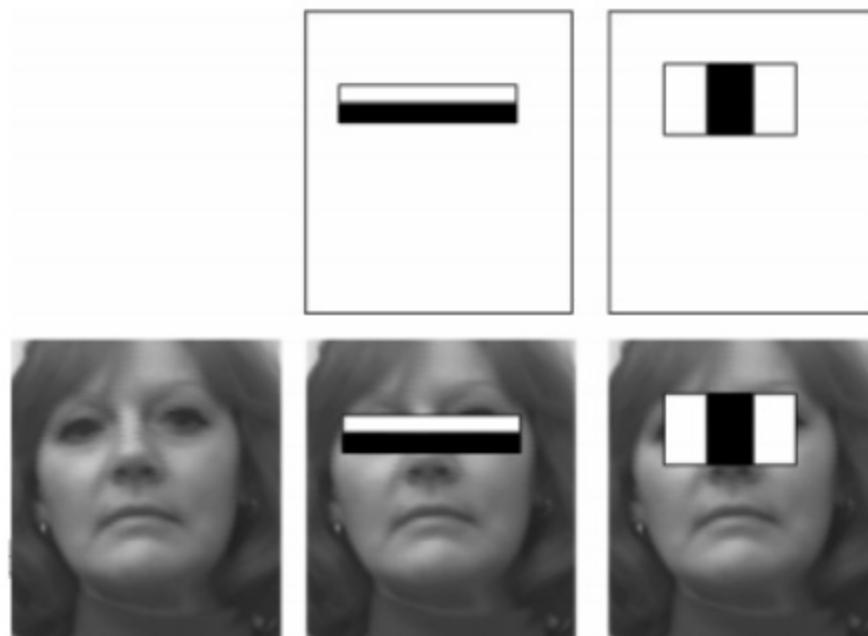
Hệ thống phát hiện đối tượng trên hình ảnh dựa trên các đặc trưng đơn giản. Có nhiều lý do để chọn việc xử lý dựa trên đặc trưng mà không thực hiện với xử lý trực tiếp trên các điểm ảnh. Lý do phổ biến nhất là các đặc trưng có thể dựa trên các tri thức hữu hạn của các dữ liệu huấn luyện để mã hóa ra các tri thức nhận dạng. Ngoài ra một hệ thống phát hiện đối tượng sử dụng đặc trưng thì xử lý sẽ nhanh hơn so với hệ thống dựa trên xử lý điểm ảnh.

Một loại đặc trưng đơn giản được sử dụng là đặc trưng Haar-like, là một loại đặc trưng thường được dùng cho bài toán nhận dạng trên ảnh. Đặc trưng Haar-like được xây dựng từ các hình chữ nhật có kích thước bằng nhau, dùng để tính độ chênh lệch giữa các giá trị điểm ảnh trong các vùng kề nhau. Trong hình a và b, giá trị của đặc trưng cho bởi 1 ảnh bằng hiệu số giữa tổng các điểm ảnh thuộc 2 vùng hình chữ nhật sáng và tối. Trong hình c thì giá trị đặc trưng bằng tổng các điểm ảnh trong 2 vùng hình chữ nhật bên ngoài trừ cho tổng các điểm ảnh trong hình chữ nhật ở giữa.



Hình 2.5: Đặc trưng Haar-like cơ bản

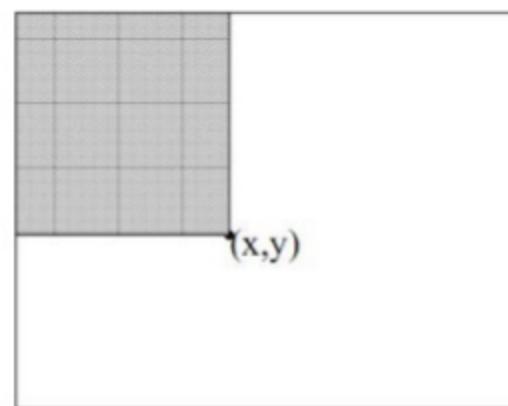
Lợi ích của đặc trưng Haar-like là nó diễn đạt được tri thức về các đối tượng trong ảnh (bởi vì nó biểu diễn mối liên hệ giữa các bộ phận của đối tượng), điều mà bản thân từng điểm ảnh không diễn đạt được.



Hình 2.6: Minh họa đặc trưng Haar-like

Trong quá trình huấn luyện, số lượng xử lý trên các đặc trưng Haar-like là rất lớn, việc tính tổng các điểm ảnh cho bởi từng đặc trưng làm cho thời gian xử lý tăng đáng kể. Để khắc phục điều này, Viola và Jones đã đưa ra khái niệm **Integral Image** [13] để tính toán nhanh cho các đặc trưng cơ bản.

Integral Image



Hình 2.7: Giá trị integral image [13] tại điểm (x,y) bằng tổng giá trị của tất cả điểm ảnh phía trên và bên trái (x,y)

Giá trị các đặc trưng có thể được tính toán nhanh chóng bằng cách sử dụng khái niệm gọi là integral Image [13], Integral Image [13] tại vị trí (x,y) bằng tổng các giá trị điểm ảnh phía trên và bên trái của (x,y) , bao gồm:

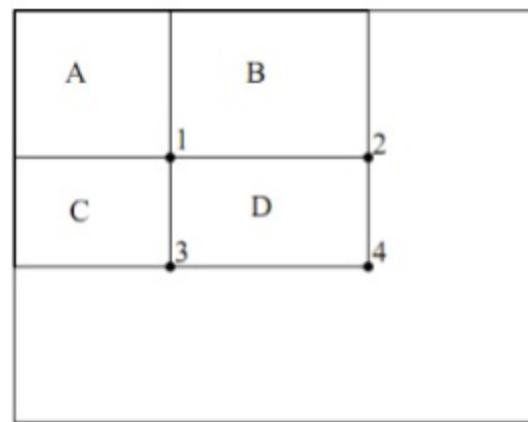
$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y')$$

Với $ii(x, y)$ là integral image và $i(x, y)$ là giá trị điểm ảnh bao gồm cặp công thức sau:

$$s(x, y) = s(x, y-1) + i(x, y)$$

$$ii(x, y) = ii(x-1, y) + s(x, y)$$

(Với $s(x, -1) = 0$ và $ii(x-1, y) = 0$)



Theo công thức trên thì integral image [13] tại điểm 1 sẽ bằng tổng giá trị điểm ảnh trong hình chữ nhật A, integral image tại điểm 2 sẽ bằng A + B, integral image tại điểm 3 sẽ bằng A + C, integral image tại điểm 4 sẽ bằng A + B + C + D.

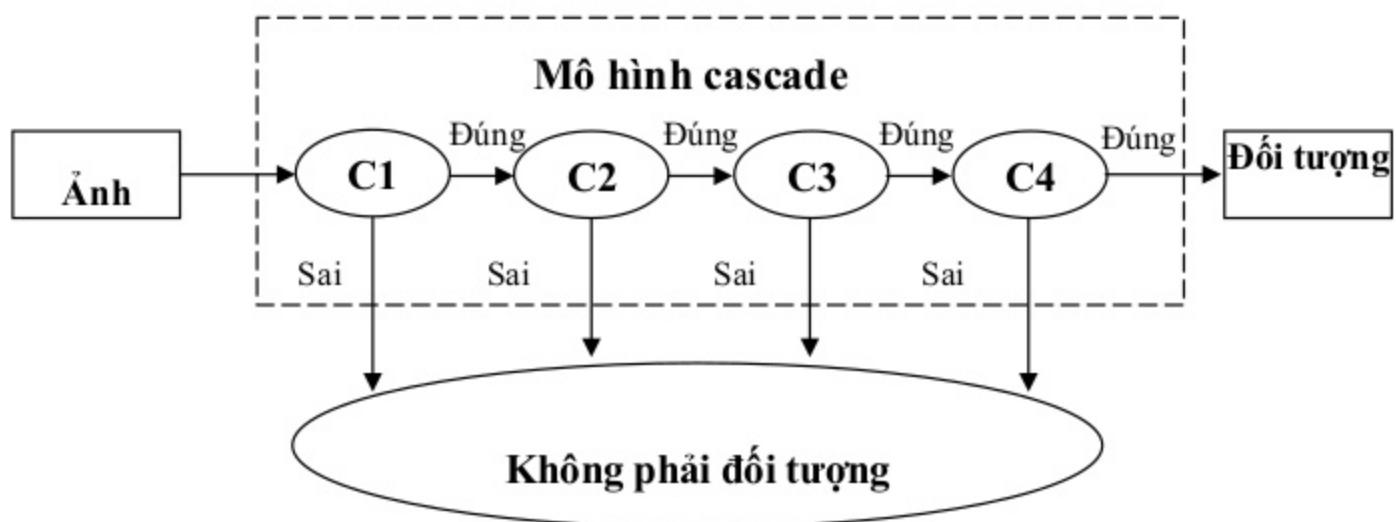
Như vậy, tổng giá trị điểm ảnh trong một hình chữ nhật bất kỳ có thể tính nhanh dựa trên integral image tại bốn đỉnh của nó, tổng giá trị điểm ảnh của hình chữ nhật D bằng $4 + 1 - (2 + 3)$.

2.2.1.2 Mô hình cascade

Một hệ thống phát hiện tốt thường tồn rất nhiều thời gian để cho ra kết quả phát hiện bởi vì nó phải xét rất nhiều đặc trưng của mẫu. Tuy nhiên, trong các mẫu

đưa vào, không phải mẫu nào cũng thuộc loại khó nhận dạng. Có những mẫu background rất dễ nhận ra (ta gọi đây là những mẫu background đơn giản). Đối với những mẫu này, ta chỉ cần xét một hay vài đặc trưng đơn giản là có thể nhận diện được chứ không cần xét tất cả các đặc trưng. Nhưng đối với các bộ phát hiện thông thường thì cho dù mẫu cần nhận dạng là dễ hay khó thì nó vẫn sẽ xét tất cả các đặc trưng mà nó rút ra được trong quá trình học. Do đó, chúng tốn thời gian xử lý một cách không cần thiết.

Mô hình cascade [10] được xây dựng chính là nhằm rút ngắn thời gian xử lý, giảm thiểu tỷ lệ nhận dạng sai của bộ nhận dạng. Mô hình cascade phân tầng theo dạng hình cây, mỗi cây gồm nhiều tầng, mỗi tầng của cây sẽ là một bộ phân loại. Một mẫu để được phát hiện là đối tượng thì nó cần phải đi qua hết tất cả các tầng của cây, các bộ phân loại ở tầng sau được huấn luyện bằng những mẫu negative mà bộ phân loại trước nó nhận dạng sai, tức là nó sẽ tập trung học từ các mẫu background khó hơn, do đó sự kết hợp các bộ phân loại này lại sẽ giúp bộ phát hiện có false alarm thấp. Với cấu trúc này, những mẫu background dễ nhận diện sẽ bị loại ngay từ những tầng đầu tiên, giúp đáp ứng tốt nhất đối với độ phức tạp gia tăng của các mẫu đưa vào, đồng thời giúp rút ngắn thời gian xử lý.



Hình 2.8: Mô hình cascade

2.2.1.3 Mô hình boost cascade

Mô hình boost cascade là mô hình mô hình cascade với mỗi bộ phân loại được xây dựng bằng AdaBoost sử dụng đặc trưng Haar-like. Mô hình này đã được Viola và Jones sử dụng rất thành công trong bài toán phát hiện khuôn mặt [14]. Với tập huấn luyện gồm 4196 hình mặt người được đưa về ảnh xám kích thước 24x24 và 9500 hình background, Viola và Jones đã xây dựng cấu trúc cây cascade gồm 38 tầng với tổng cộng 6060 đặc trưng Haar-like. Thực nghiệm đã cho thấy bộ phân loại ở tầng đầu tiên sử dụng 2 đặc trưng và loại được khoảng 50% mẫu background (không chứa khuôn mặt) và có tỷ lệ nhận dạng là 100%. Bộ phân loại ở tầng thứ 2 sử dụng 10 đặc trưng loại được 80% mẫu background vẫn với lệ nhận dạng 100%. Hệ thống này được so sánh với hệ thống của Rowley-Baluja-Kanade [17] (sử dụng mạng noron), Schneiderman-Kanade [19] (sử dụng phương pháp thống kê), và cho thấy tỉ lệ nhận dạng là ngang nhau, trong khi hệ thống của Viola và Jones chạy nhanh hơn đến 15 lần so với hệ thống của Rowley-Baluja-Kanade và nhanh hơn 600 lần hệ thống của Schneiderman-Kanade [19].

Bên cạnh đó, mô hình này cũng được Eng-Jon Ong và Richard Bowden [4] áp dụng thành công trong bài toán nhận dạng bàn tay. Do bàn tay có nhiều biến động hơn so với mặt người, Ong và Bowden đã sử dụng phương pháp học không giám sát: tiến hành phân cụm cho tất cả các mẫu trong tập huấn luyện chứa 2504 hình bàn tay chụp ở nhiều tư thế khác nhau bằng thuật toán K-mediod clustering. Cấu trúc bộ nhận dạng của Ong và Bowden gồm 2 lớp: lớp ở trên là 1 Mô hình boost cascade để nhận dạng sơ bộ bàn tay, lớp bên dưới là từng mô hình boost cascade ứng với từng cụm được chia bằng K-mediod. Kết quả thu được rất khả quan, cấu trúc cascade của bộ nhận dạng ở lớp trên gồm 11 tầng với tổng cộng 634 bộ phân loại yếu đã đạt tỉ lệ nhận dạng là 99.8% trên tập kiểm thử, còn các bộ cascade ở lớp dưới có tỉ lệ nhận dạng trung bình là 97.4%.

2.2.1.4 Đánh giá

Trước khi có hệ thống của Viola và Jones [13], hệ thống của Rowley [17] được đánh giá là bộ nhận dạng có tốc độ nhanh nhất. Hệ thống của Rowley thực

chất cũng là một cấu trúc cascade với 2 mạng noron: mạng noron thứ nhất khá đơn giản nhằm mục đích chính là loại bỏ các hình background có độ khó thấp, mạng noron thứ 2 phức tạp hơn, đảm nhiệm việc nhận dạng các mẫu đi qua neural network thứ 1.

Điều này chứng tỏ mô hình cascade đã giúp loại nhanh các mẫu có độ phức tạp thấp thực sự đẩy nhanh tốc độ của hệ thống.

Ý tưởng của Viola và Jones [13] khi đưa ra Mô hình boost cascade thật ra cũng tương tự vậy, nhưng nó mở rộng 2 tầng thành 38 tầng của cây cascade. Hệ thống của Viola và Jones càng chứng tỏ khả năng tăng tốc của mô hình cascade khi đạt tốc độ nhanh hơn hệ thống của Rowley [17] và hệ thống của Schneiderman-Kanade [19] (vốn không hề sử dụng cascade) lần lượt 15 lần và 600 lần.

Trong bài báo cáo của mình, Viola và Jones cũng đã tiến hành so sánh hệ thống sử dụng mô hình boost cascade với một hệ thống chỉ có một bộ nhận dạng duy nhất xây dựng bằng AdaBoost với tổng số đặc trưng Haar-likes sử dụng là 200. Kết quả là hệ thống theo mô hình boost cascade nhanh hơn đến 10 lần. Lý do mà cấu trúc cascade đạt tốc độ nhận dạng nhanh chính là nhờ nó sớm loại bỏ được các mẫu background đơn giản (thường có số lượng lớn hơn nhiều so với các mẫu chứa đối tượng – các mẫu thực sự cần tiến hành nhận dạng).

Bên cạnh đó, hệ thống của Viola và Jones cũng đạt được độ chính xác cao tương đương các hệ thống khác là nhờ thuật toán cấu trúc cascade các bộ nhận dạng được huấn luyện bằng AdaBoost với đặc trưng đặc trưng Haar-like mô tả tốt thông tin đối tượng, cùng với cách tính Integral Image tính nhanh các đặc trưng, không làm giảm tốc độ nhận dạng của hệ thống.

Như vậy, mô hình mô hình boost cascade [11] thật sự là một cách tiếp cận tốt cả về tốc độ lẫn khả năng nhận dạng. Tuy nhiên, bài toán cũng đặt ra một số khó khăn về số lượng mẫu và thời gian huấn luyện. AdaBoost đòi hỏi phải có số lượng mẫu rất lớn (tối thiểu phải lên đến hàng nghìn) để huấn luyện được bộ nhận dạng hiệu quả. Hệ thống nhận dạng mặt người của Viola và Jones cần đến 4916 ảnh mặt

người. Việc thu thập đủ số lượng mẫu này là một trở ngại rất lớn. Bên cạnh đó, do số lượng mẫu nhiều, đồng thời số lượng đặc trưng Haar-like xử lý lớn nên thời gian huấn luyện rất lâu. Do đó, việc xây dựng toàn bộ hệ thống nhận dạng đối tượng sẽ tốn rất nhiều thời gian (phải kể cả việc trong quá trình thử nghiệm sẽ có những lần huấn luyện thất bại, buộc phải tiến hành huấn luyện lại).

2.2.2 Nhận dạng đối tượng bằng thuật toán AdaBoost

2.2.2.1 Bộ nhận dạng một đối tượng

Để nhận dạng một đối tượng cụ thể (khuôn mặt của một người nào đó), bộ nhận dạng cho dạng có chức năng nhận dạng một mẫu có thuộc về một đối tượng đó không. Bộ nhận dạng này được xây dựng theo cấu trúc Cascade.



Hình 2.9: Bộ nhận dạng một đối tượng xác định

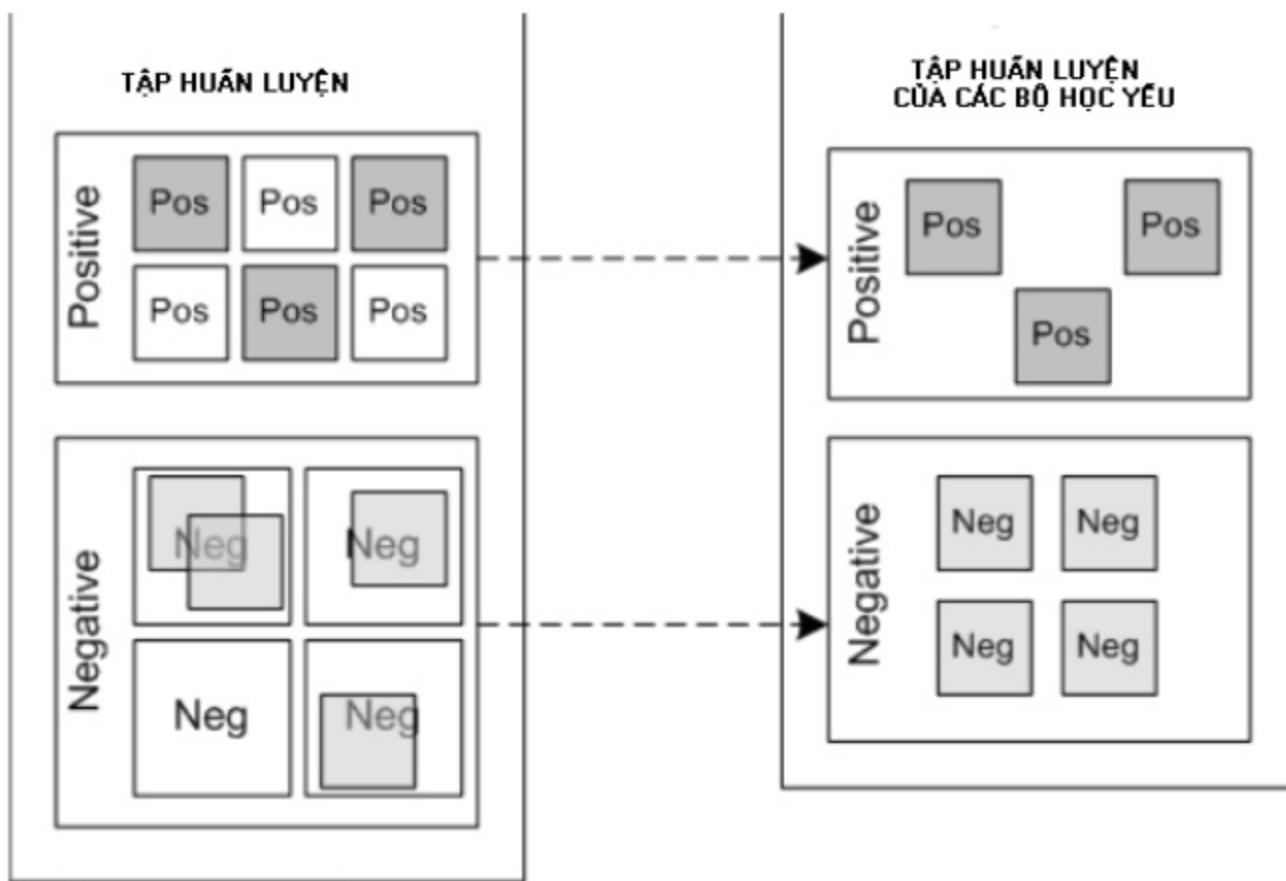
2.2.2.2 Tập huấn luyện

Tập huấn luyện bao gồm các mẫu positive (đối tượng cần nhận dạng) và negative (mẫu không chứa đối tượng). Trong bộ nhận dạng, các mẫu positive là các hình chụp của đối tượng đó đã qua chuẩn hóa: kích thước và được chuyển về ảnh xám. Các mẫu negative bao gồm tất cả các ảnh không chứa đối tượng cần nhận dạng (các hình background).

Không như các mẫu positive có kích thước cố định, các mẫu negative trong tập huấn luyện có thể có kích thước tùy ý nhưng phải lớn hơn kích thước mẫu positive. Trong quá trình huấn luyện, các bộ phân loại yếu sẽ học từ các mẫu positive trong tập huấn luyện và các mẫu negative là các vùng ảnh trích ra từ các mẫu negative trong tập huấn luyện.



Hình 2.10: Minh họa các mẫu positive dùng cho huấn luyện

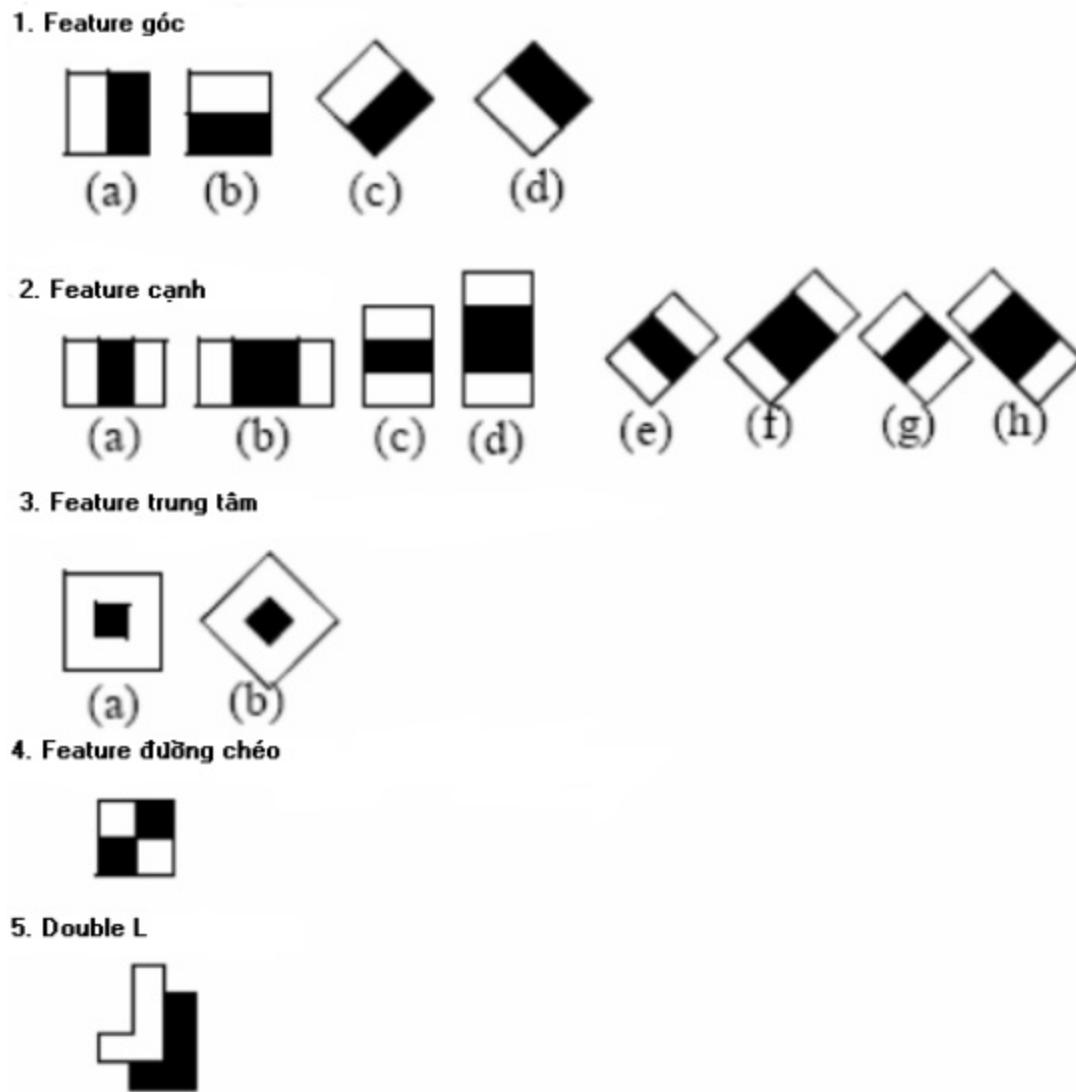


Hình 2.11: Tập huấn luyện của các bộ học yếu

2.2.2.3 Đặc trưng dùng nhận dạng

Hệ thống đặc trưng dùng để nhận dạng bao gồm cả đặc trưng Haar-like cơ bản (1a, 1b, 2a, 2b, 2c, 2d, 3a, 1, 5) và các đặc trưng Haar-like mở rộng (xoay 45° - 1c, 1d, 2e, 2f, 2g, 2h, 3b). Các đặc trưng này đều được tính toán nhanh nhờ vào khái niệm Integral Image.

Luận văn sử dụng các đặc trưng Haar-like sau:



Hình 2.12: Các đặc trưng Haar-like dùng trong bộ nhận dạng

Hệ thống đặc trưng trên bao gồm cả đặc trưng Haar-like cơ bản (1a, 1b, 2a, 2b, 2c, 2d, 3a, 1, 5) và các đặc trưng Haar-like mở rộng (xoay 45° - 1c, 1d, 2e, 2f, 2g, 2h, 3b). Các đặc trưng này đều được tính toán nhanh nhờ vào khái niệm Integral Image.

2.2.2.4 Xây dựng bộ nhận dạng với thuật toán AdaBoost

Một hệ thống nhận dạng có số lượng đặc trưng rất lớn. Tuy nhiên, không phải đặc trưng nào cũng thực hiện tốt việc nhận dạng mà chỉ có một số lượng nhỏ trong số đó là thực sự hữu dụng. Nhiệm vụ của bộ nhận dạng là phải tìm ra được các đặc trưng này.

Mỗi bộ phân loại yếu gồm có 1 đặc trưng và 1 ngưỡng, ngưỡng này chính là giá trị của một mẫu cụ thể cho bởi đặc trưng này (vấn đề của thuật toán là phải tìm

được mẫu nào dùng làm ngưỡng). Như vậy, với N mẫu và M đặc trưng thì số bộ phân loại yếu có thể có là $N \times M$, tức là với 2000 mẫu cho một cù chỉ và 160000 đặc trưng thì hệ thống sẽ phải chọn được 1 bộ phân loại yếu trong số $160000 \times 2000 = 320000000$ bộ phân loại yếu trong mỗi vòng lặp boosting.

AdaBoost được thiết kế để có thể chọn nhanh các đặc trưng, cũng là chọn nhanh các bộ phân loại yếu. Thuật toán sử dụng ở đây là Gentle AdaBoost là một cải tiến của AdaBoost. Dùng Gentle AdaBoost để xây dựng bộ nhận dạng đối tượng như sau:

Thuật toán: Tạo bộ phân loại

- Dữ liệu vào: Cho một tập huấn luyện X gồm n mẫu $\{(x_1, y_1), (x_2, y_2)\}, \dots, (x_n, y_n)\}$ với $x \in X, y \in \{-1, +1\}$

- 1: Chọn trước min detection rate và max false alarm
- 2: Xây dựng tập đặc trưng và tính toán các integral image cho tất cả các mẫu trong tập huấn luyện.
- 3: Khởi tạo trọng số ban đầu cho tất cả các mẫu

$$w_{1,k} = \frac{1}{n}$$

5: Vòng lặp khi còn thoã mãn điều kiện

- 5.1: Với mỗi đặc trưng trong tập đặc trưng, xây dựng một bộ phân loại yếu h_j với hàm nhận dạng f_j , n_g ước θ_j và tính giá trị lỗi ε_j

$$\varepsilon_j = \sum_k^n w_{t,k} |h_j(x_k) - y_k|$$

- 5.2: Từ các h_j đã có, chọn ra h_j có ε_j nhỏ nhất, ta được h_t

$$h_t(x) = g_t(f_t(x), \theta_t) : X \rightarrow R$$

5.3: Cập nhật lại trọng số:

$$\mathbf{w}_{t+1,k} = \frac{\mathbf{w}_{t,k}}{Z_t} \times e^{-\alpha_t} h_t(x_k), \text{ với } Z_t = \sum_k \mathbf{w}_{t+1,k}$$

5.4: $F_t(x) = \sum_{i=1}^t h_i(x)$, tính ngưỡng θ

$$H_t(x) = \begin{cases} 1, & F_t(x) \geq \theta \\ -1, & F_t(x) < \theta \end{cases}$$

Cho đến khi $\sum_{k=1}^n \mathbf{1}_{y_k=-1, H_t(x_k)=1} \leq \text{max_false_alarm}$

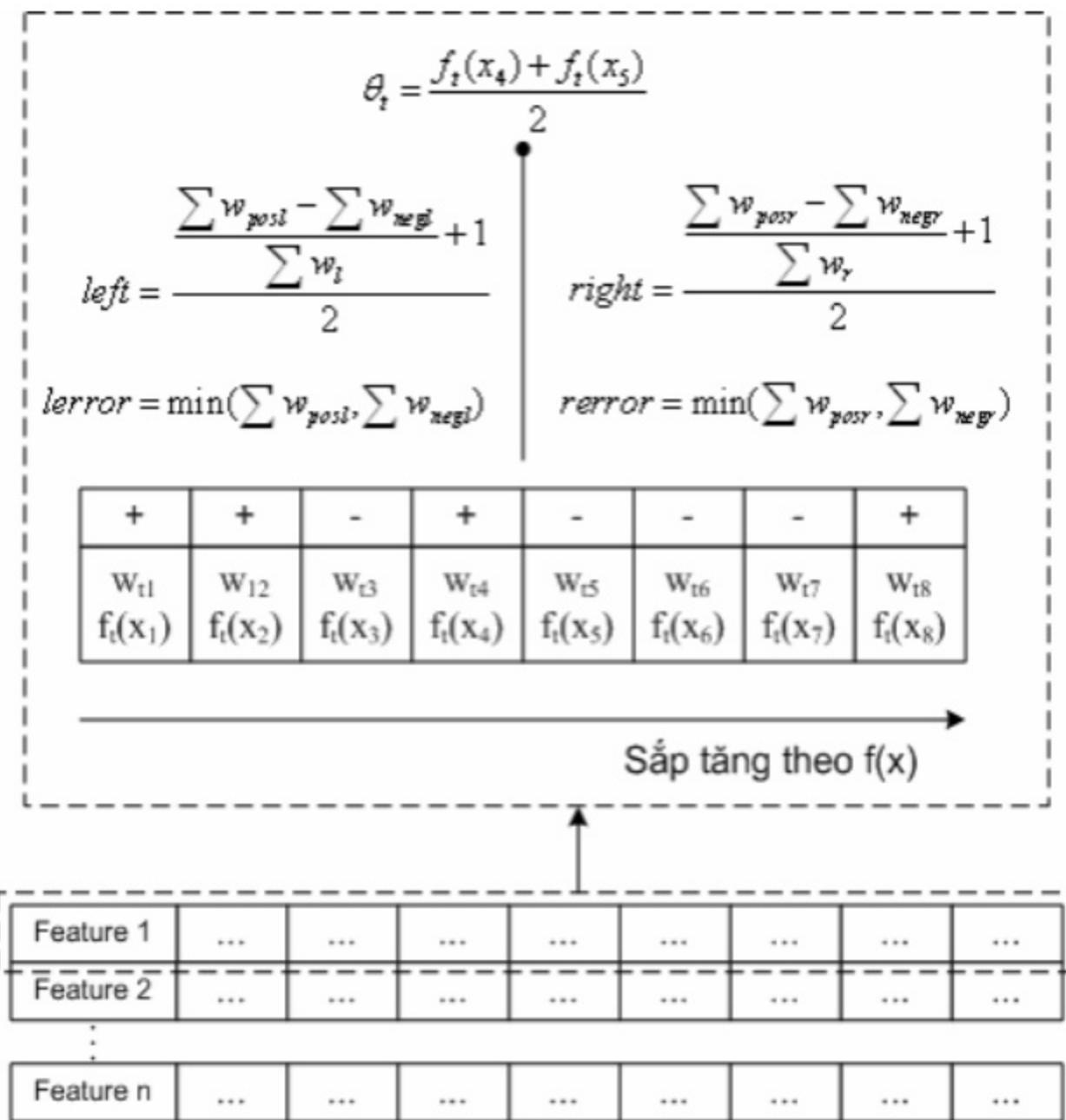
6: Kết thúc vòng lặp

7: Bộ nhận dạng $H(x)$:

$$H(x) = \begin{cases} 1, & F_t(x) \geq \theta \\ -1, & F_t(x) < \theta \end{cases}$$

- Đầu ra: Bộ phân loại mạnh $H(x)$

Mỗi bộ phân loại yếu $h_t(x)$ sẽ được chọn bằng cách tính giá trị của từng đặc trưng cho tất cả các mẫu trong tập huấn luyện $f(x_k)$ và sắp xếp các mẫu theo thứ tự tăng dần của giá trị này:



Hình 2.13: Minh họa cách chọn bộ phân loại yếu của AdaBoost

Đặc trưng được chọn là đặc trưng thực hiện phân đôi tập huấn luyện với lỗi $\varepsilon_t = lerror + rerror$ nhỏ nhất.

$$h_t(x) = \begin{cases} left, & f_t(x) < \theta_t \\ right, & f_t(x) \geq \theta_t \end{cases}$$

Sau mỗi vòng lặp của boosting, ta xây dựng thêm được một bộ phân loại yếu.

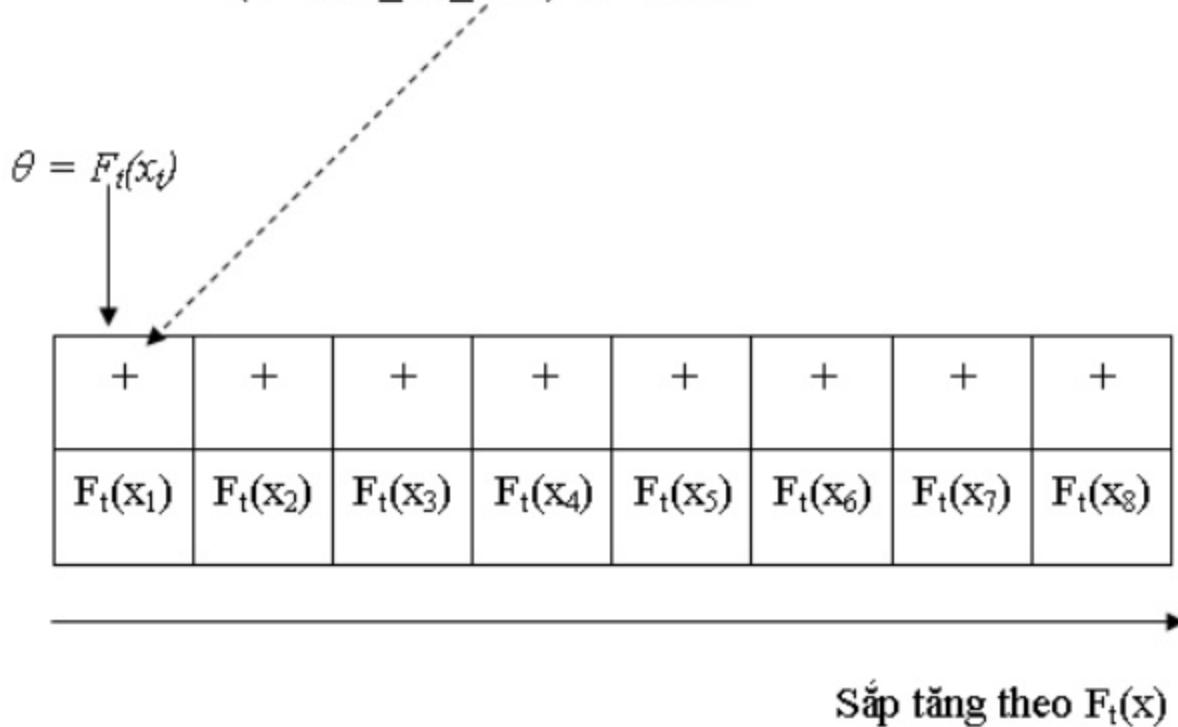
Ta có thể xây dựng được một bộ phân loại mạnh mới từ sự kết hợp của các bộ phân loại yếu có được cho tới lần boost hiện tại:

$$F_t(x) = \sum_{i=1}^t h_i(x) \text{ và } H_t(x) = \begin{cases} 1, & F_t(x) \geq \theta \\ -1, & F_t(x) < \theta \end{cases}$$

Giá trị ngưỡng θ được chọn nhờ vào giá trị tỷ lệ phát hiện đúng tối thiểu các mẫu positive (min detection rate/min hit rate) mà bộ nhận dạng phải đạt, giá trị này do chúng ta xác lập khi bắt đầu quá trình huấn luyện. Bằng cách sắp xếp các mẫu positive x_k theo $F_t(x_k)$ tăng dần:

$$\text{min_hit_rate (Tỷ lệ nhận dạng đúng tối thiểu)} = 0.997$$

$$(1 - \text{min_hit_rate}) * 8 = 0.024$$



Hình 2.14: Chọn ngưỡng θ dựa vào min _hit _ rate

Trong hình trên, min detection rate được chọn là 0,997, nghĩa là tỉ lệ nhận dạng đúng các mẫu positive không được thấp hơn 99.7%. Do đó, phần tử được chọn làm ngưỡng là mẫu đầu tiên trong danh sách sắp tăng. Giá trị ngưỡng θ này đảm bảo có ít nhất 99.7% các mẫu positive sẽ có $F_t(x) \geq \theta$. Nếu ta chọn min detection rate là 0.5, khi đó $(1 - \text{min detection rate}) * \text{số_mẫu} = 4$, phần tử thứ 4 của danh sách các mẫu sẽ được chọn làm ngưỡng, như vậy sẽ chỉ có 4 mẫu trong số 8 mẫu positive là được nhận dạng đúng.

Nguồn có ý nghĩa rất quan trọng, nguồn càng nhỏ thì “hit rate” sẽ càng cao nhưng “false alarm” cũng tăng theo và ngược lại.

Sau khi có được $H_i(x)$, vấn đề cần xem xét là liệu bộ phân loại này có đủ tốt chưa, nghĩa là giá trị lỗi của nó đã thấp hơn max false alarm (là tỷ lệ nhận dạng sai các mẫu không phải đối tượng cần nhận dạng lớn nhất) hay chưa. max false alarm là một giá trị được xác lập trước khi tiến hành huấn luyện. False alarm (hay còn gọi là false positive) là tỉ lệ nhận dạng sai các mẫu negative. Ví dụ, nếu max false alarm = 0.5 thì trên 100 mẫu negative, nó phải nhận đúng ít nhất là 50 mẫu (50 mẫu còn lại bị phân loại nhầm thành positive). Vòng lặp xây dựng bộ phân loại mạnh sẽ kết thúc khi giá trị false alarm của bộ phân loại mạnh này thấp hơn max false alarm.

2.2.2.5 Mô hình boost cascade để nhận dạng

Bộ nhận dạng đối tượng là 1 cấu trúc cascade gồm K tầng, f_i , d_i lần lượt là *max false alarm* và *min detection rate* của bộ phân loại ở tầng thứ i , *max false alarm* và *min detection rate* của cây cascade sẽ lần lượt là:

$$F = \sum_{i=1}^K f_i \text{ và } D = \sum_{i=1}^K d_i$$

Trên lý thuyết, các tầng phân loại sẽ có *max false alarm* và *min detection rate* khác nhau, các bộ phân loại ở các tầng càng sâu thì max false alarm sẽ càng lớn và min detection rate càng nhỏ do nó học trên các mẫu khó hơn. Tuy nhiên, trong thực tế cài đặt, chúng ta không thể biết chính xác số tầng mà cây cascade sẽ có trước khi tiến hành huấn luyện, dẫn đến khó khăn trong việc chọn giá trị *max false alarm* và *min detection rate* cho mỗi bộ phân loại. Do đó, trong phần cài đặt của bài toán nhận dạng đối tượng, *max false alarm* và *min detection rate* được xác lập bằng nhau ở tất cả các tầng. Khi đó, *max false alarm* và *min detection rate* của bộ nhận dạng lần lượt là: $F = f^k$ và $D = d^k$.

Thuật toán: Xây dựng cây cascade

- 1: Xác lập *max false alarm f*, *min detection rate d* cho các bộ nhận dạng ở mỗi tầng và số tầng tối đa cây cascade sẽ có
- 2: Tính *false alarm F* cho bộ phân loại chính (cây phân lớp)
- 3: $F_0 = 0$, $i = 0$
- 4: p, n là số lượng mẫu positive và negative
- 5: P_0, N_0 là tập positive và negative cho bộ phân lớp ở tầng đầu tiên
- 6: Trong khi $F_i > F$
- 7: Huấn luyện bộ phân loại H_i từ tập P_i và N_i với detection rate d và max false alarm f (thủ tục Tạo Bộ phân loại)
- 8: Thêm H_i vào cây phân lớp
- 9: Dùng cây phân lớp hiện có để tính F_{i+1} : Duyệt qua N mẫu negative cho đến khi nào tìm đủ n mẫu mà cây phân lớp hiện có phân loại sai

$$F_{i+1} = \frac{n}{N}$$

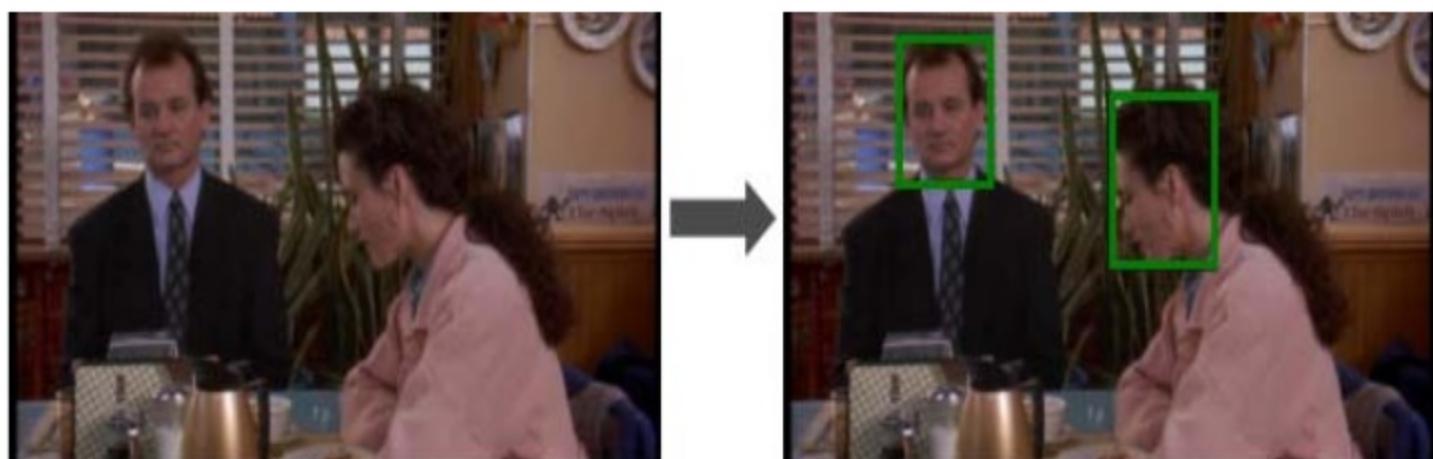
- 10: Nếu $F_{i+1} > F$, đưa n mẫu negative trên vào N_{i+1} ; xây dựng P_{i+1} với tối đa p mẫu positive mà cây phân lớp hiện có nhận dạng đúng.

- 11: $i \leftarrow i + 1$

Qua thuật toán trên ta thấy, các mẫu negative trong tập huấn luyện của bộ phân loại sau sẽ là những mẫu negative mà bộ phân loại trước nó nhận dạng sai vì thế, nó có điều kiện tập trung học những mẫu background khó. Như vậy ta thấy các bộ phân loại được xây dựng sao cho sự kết hợp của chúng xử lý rất tốt các mẫu background. Đây chính là lý do mà cấu trúc cascade giúp cho bộ nhận dạng giảm thiểu *false alarm* và cũng là tăng tỉ lệ nhận dạng cho hệ thống. Đồng thời, cấu trúc này cũng đủ thông minh để biết phái loại một mẫu background ở tầng nào tùy thuộc vào độ phức tạp của chúng, chứ không nhất thiết phái gọi đến tất cả các bộ phân loại đối với mọi mẫu background, chính điều này đã giúp tăng tốc độ nhận dạng.

2.2.2.6 Hoạt động của bộ nhận dạng

Khi đưa một ảnh vào nhận dạng, bộ nhận dạng sẽ phải xét tất cả các vùng ảnh với kích thước khác nhau trích ra được từ ảnh này để có thể đưa ra kết quả. Kích thước khởi đầu của vùng ảnh sẽ là một window bằng với kích thước của mẫu positive trong quá trình huấn luyện. Các vùng ảnh này sẽ được dịch theo chiều ngang và dọc 1 lượng từ 1 đến 2 pixel cho đến khi phủ kín ảnh cần nhận dạng. Sau đó, vùng ảnh sẽ được mở ra với tỉ lệ 1.1 (giá trị này người dùng được phép thay đổi khi tiến hành nhận dạng) và tiếp tục quá trình duyệt ảnh như trên cho đến khi vùng ảnh được mở kích thước ra bằng kích thước ảnh. Nhờ có cấu trúc cascade, các vùng ảnh không liên quan bị loại nhanh từ những tầng đầu tiên.



Hình 2.15: Các vùng ảnh không phải đối tượng sẽ bị loại ngay từ những tầng đầu tiên và kết quả thu nhận được là vùng ảnh xác định đối tượng

Qua thực nghiệm, ta thấy số lượng vùng ảnh tương tự nhau cũng có tác dụng trong việc điều chỉnh giữa hit rate và false alarm. Do tỉ lệ nhận dạng không phải 100% nên khi tiến hành nhận dạng, có thể có vài mẫu background sẽ bị nhầm là đối tượng cần nhận dạng. Do đó, ứng dụng nhận dạng sử dụng khái niệm min neighbor (số vùng ảnh tương tự nhau tối thiểu) phải có để một vùng ảnh có thể được nhận dạng là đối tượng. Giá trị này được người dùng xác lập khi tiến hành nhận dạng.

2.2.2.7 Xây dựng bộ nhận dạng đối tượng

Bộ nhận dạng đối tượng được xây dựng dựa trên sự kết hợp các bộ nhận dạng một đối tượng cụ thể, xác định đã trình bày ở trên. Đầu vào là một ảnh mẫu,

đầu ra là lớp của ảnh này (khuôn mặt của anh A, chị B, bà C,...). Các bộ nhận dạng một đối tượng có thể được kết hợp theo cách tiếp cận one against all (nhận dạng chỉ một, ví dụ chỉ nhận dạng mặt ông A chứ không nhận dạng những người khác).



Hình 2.16: Quy trình nhận dạng đối tượng

Một mẫu đưa vào sẽ đi qua tuần tự các bộ nhận dạng từ trên xuống dưới cho đến khi nào nó được phân vào một lớp cụ thể hay được cho là không thuộc loại đối tượng cần nhận dạng.

Chương 3

TRÌNH BÀY KẾT QUẢ THỰC NGHIỆM

Chương này sẽ trình bày về phương pháp thực hiện các thực nghiệm nhận dạng mặt người, bao gồm cách xây dựng bộ dữ liệu huấn luyện, bộ dữ liệu đánh giá kiểm thử và các kết quả đạt được.

3.1 Bộ dữ liệu

3.1.1 Bộ dữ liệu huấn luyện

Trong luận văn này tôi thực nghiệm trên Bộ dữ liệu ảnh Feret dùng làm bộ dữ liệu huấn luyện. Đây là Bộ dữ liệu ảnh trắng đen được thu thập trong một dự án của Bộ quốc phòng Mỹ do cơ quan nghiên cứu quốc phòng Mỹ (DARPA) nhằm nâng cao khả năng nhận dạng mặt khuôn mặt để trợ giúp an ninh, tình báo và nhân sự thi hành pháp luật trong việc thực hiện nhiệm vụ

Các hình ảnh trong bộ dữ liệu FERET đã được sử dụng để hỗ trợ chính quyền giám sát kiểm tra và đánh giá các thuật toán nhận dạng khuôn mặt bằng cách sử dụng các quy trình và kiểm thử tiêu chuẩn. Bộ dữ liệu gồm có 14051 ảnh mặt người người với nhiều góc độ chụp khác nhau.

Trước đó, Bộ dữ liệu này là một phần của Bộ dữ liệu FERET chỉ cung cấp cho những nghiên cứu và các ứng dụng mang tính công nghiệp và đã được phân phối từ năm 1998 đến tháng 2 năm 2001 gồm có 3737 hình ảnh.

Trong luận văn chỉ sử dụng 1219 ảnh của 50 người khác nhau trong Bộ dữ liệu FERET, số ảnh trên được phân thành 50 bộ dữ liệu nhỏ với tỷ lệ phân chia tập huấn luyện và tập kiểm thử là 50% để thực nghiệm, mô tả chi tiết về 50 bộ dữ liệu của 50 người được cho ở bảng dưới:

Tên	Số lượng mẫu huấn luyện	Số lượng mẫu kiểm thử	Tên	Số lượng mẫu huấn luyện	Số lượng mẫu kiểm thử
F1	26	26	F26	21	20
F2	12	12	F27	21	20
F3	11	10	F28	10	9
F4	7	6	F29	14	14
F5	8	7	F30	10	9
F6	18	17	F31	10	9
F7	8	7	F32	9	8
F8	7	6	F33	9	8
F9	15	14	F34	9	8
F10	18	17	F35	9	8
F11	9	9	F36	9	8
F12	34	34	F37	13	12
F13	7	6	F38	11	11
F14	7	6	F39	11	11
F15	12	12	F40	11	10
F16	7	6	F41	12	12
F17	13	12	F42	12	12

Tên	Số lượng mẫu huấn luyện	Số lượng mẫu kiểm thử	Tên	Số lượng mẫu huấn luyện	Số lượng mẫu kiểm thử
F18	12	11	F43	12	13
F19	9	8	F44	11	11
F20	9	8	F45	11	11
F21	11	11	F46	12	12
F22	27	26	F47	11	11
F23	17	17	F48	17	16
F24	9	9	F49	11	11
F25	14	14	F50	11	10

Bảng 3.1: Thống kê mô tả 50 bộ dữ liệu



Hình 3.1: Khuôn mặt đại diện của 50 người khác nhau đại diện của 50 tập dữ liệu ảnh của mỗi người trong bộ dữ liệu FERET.

Tập dữ liệu negative của mỗi người chính là toàn bộ ảnh của những người khác nhưng không có chứa ảnh người đó.

3.1.2 Bộ dữ liệu kiểm thử

Để tiến hành kiểm thử, luận văn sử dụng Bộ dữ liệu Face CMU là một bộ dữ liệu được tạo ra bởi dự án nhận dạng khuôn mặt của Đại học Carnegie Mellon (CMU) bao gồm 04 bộ dữ liệu là A, B, C và các tập ảnh xoay. Bộ dữ liệu B do Kah-Kay Sung và Tomaso Poggio [11] thu thập tại phòng thí nghiệm AI/CBCL của MIT và bộ dữ liệu A, C, bộ ảnh xoay được thu thập bởi Henry A. Rowley, Shumeet Baluja và Takeo Kanade [20] tại CMU.

Bộ dữ liệu thứ hai là CBCL của MIT gồm 472 khuôn mặt đa dạng của 472 người và 23573 không chứa khuôn mặt.



Hình 3.2: Một số ảnh minh họa của bộ dữ liệu CMU



Hình 3.3: Một số ảnh minh họa ảnh nền không chứa đối tượng (khuôn- mặt) trong bộ dữ liệu CBCL

3.2 Các bước cơ bản tiến hành huấn luyện mẫu

Bước 1: Để huấn luyện ảnh ta cần có các tập mẫu, tương ứng tập dữ liệu thì có hai loại là các mẫu negative và mẫu positive.

Đầu tiên ta phải tạo ra tập tin mô tả của các ảnh negative và ảnh positive. Tập tin mô tả mẫu negative có tên là list.txt chỉ chứa tên file negative:

Negative/neg-001.jpg

Negative/neg-002.jpg

Tiếp đó ta tạo ra tập tin faceList.txt liệt kê danh sách các ảnh positive, số lượng khuôn mặt và tọa độ xuất hiện khuôn mặt xuất hiện trong ảnh đó.

Cấu trúc nội dung liệt kê tập tin cụ thể như sau:

[tên tập tin ảnh] [số lượng khuôn mặt] [[x y width height] [...tọa độ khuôn mặt tiếp theo] ...]

[tên tập tin ảnh] [số lượng khuôn mặt] [[x y width height] [...tọa độ khuôn mặt tiếp theo] ...]

[tên tập tin ảnh] [số lượng khuôn mặt] [[x y width height] [...tọa độ khuôn mặt tiếp theo] ...]

...

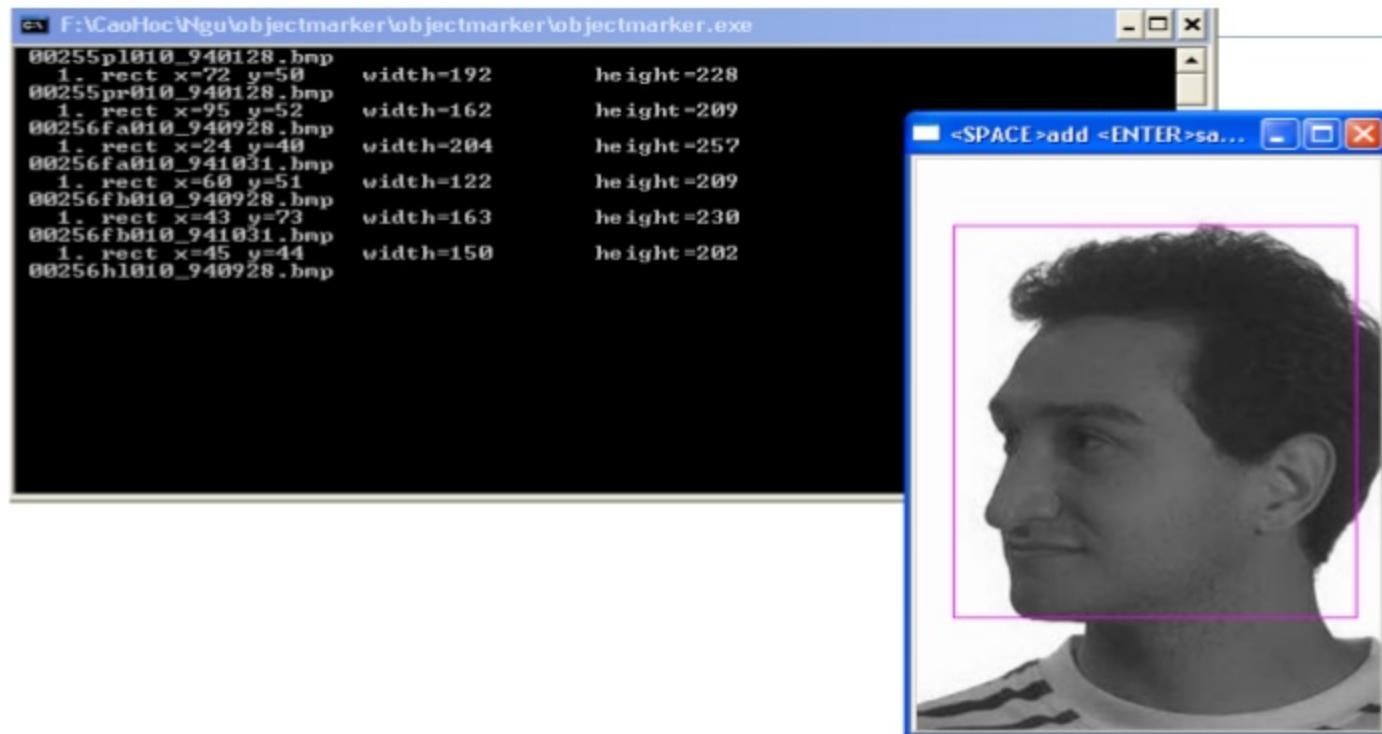
Ví dụ:

rawdata/00209hr010_940128.jpg 1 0 0 23 36

rawdata/00207fb010_940128.jpg 2 35 32 178 246 34 38 125 235

Trong đó: *rawdata* là thư mục chứa ảnh; *00209hr010_940128.jpg* và *00207fb010_940128.jpg* là hai tập tin ảnh; ảnh trên có 1 khuôn mặt, ảnh dưới có 2 khuôn mặt; 4 số ở bốn cột biểu diễn tọa độ của khuôn mặt trong ảnh.

Để xác định tọa độ này, em dùng phần mềm objectmarker.exe, giúp cho việc tạo tập tin trên đạt tốc độ cao.



Hình 3.4: Minh họa việc tạo tập tin faceList.txt bằng ObjectMarker.exe

```

faceList.txt - Notepad
File Edit Format View Help
rawdata/00207fb010_940128.jpg 1 35 32 178 246
rawdata/00207hl010_940128.jpg 1 33 34 198 254
rawdata/00207hr010_940128.jpg 1 24 33 203 267
rawdata/00207pl010_940128.jpg 1 20 23 227 256
rawdata/00207pr010_940128.jpg 1 2 45 231 259
rawdata/00208fa010_940128.jpg 1 40 34 173 244
rawdata/00208fb010_940128.jpg 1 38 23 189 268
rawdata/00208hl010_940128.jpg 1 58 23 169 247
rawdata/00208hr010_940128.jpg 1 28 30 205 247
rawdata/00208pl010_940128.jpg 1 21 22 223 248
rawdata/00208pr010_940128.jpg 1 9 34 225 250
rawdata/00209fa010_940128.jpg 1 27 28 203 263
rawdata/00209fb010_940128.jpg 1 49 40 188 282
rawdata/00209hl010_940128.jpg 1 24 23 200 269
rawdata/00209hr010_940128.jpg 1 29 26 208 267
rawdata/00209pl010_940128.jpg 1 17 22 200 278
rawdata/00209pr010_940128.jpg 1 18 25 211 278
rawdata/00210fa010_940128.jpg 1 40 45 166 241
rawdata/00210fb010_940128.jpg 1 44 47 177 236
rawdata/00210hl010_940128.jpg 1 34 70 187 206
rawdata/00210hr010_940128.jpg 1 73 44 168 268
rawdata/00210pl010_940128.jpg 1 28 72 177 239
rawdata/00210pr010_940128.jpg 1 89 55 155 271
rawdata/00211fa010_940128.jpg 1 25 34 178 267
rawdata/00211fb010_940128.jpg 1 38 43 174 263

```

Hình 3.5: Nội dung tập tin faceList.txt

Bước 2: Sau khi đã có được hai tập tin *list.txt* và *faceList.txt* thì ta tiến hành tạo mẫu bằng công cụ **createsample.bat** thực hiện một command:

```
createsamples -info faceList.txt -vec faceList.vec -num 300 -w 24 -h 36
```

Trong đó:

- - info: tập tin liệt kê danh sách ảnh (ở đây là *faceList.txt*)
- - num: chỉ số lượng ảnh
- - w và - h là kích thước ảnh
- - vec: tập tin lưu các vectơ ảnh (ở đây là tập tin *faceList.vec*)

Kết quả là xuất ra một tập tin mô tả mẫu là *faceList.vec* chứa các vectơ mẫu ảnh.

Bước 4: Sau khi hoàn thành việc tạo mẫu thì ta tiến hành huấn luyện mẫu với đầu vào là thông tin về các mẫu positive trong tập tin *faceList.txt*, số lượng ảnh huấn luyện positive là 300, ảnh negative là 230

```
Haar-like huấn luyện -data cascade/face_cascade -vec pos/faceList.vec -bg neg/list.txt -npos 300 -nneg 230 ntàng 30 -mem 800 -mode ALL -w 24 -h 36
```

Kết quả của quá trình huấn luyện cho ra bộ phân loại (bộ phân loại).

```

C:\WINDOWS\system32\cmd.exe
Precalculation time: 6.09
+---+---+---+
: N :SMP:F: ST.THR : HR : FA : EXP. ERR:
+---+---+---+
: 1:100%:-: 0.916667: 1.000000: 0.043478: 0.040000:
+---+---+---+
Stage training time: 2.91
Number of used features: 1
Parent node: 0
Chosen number of splits: 0
Total number of splits: 0
Tree Classifier
Stage
+---+---+
! 0! 1!
+---+---+
0---1

Parent node: 1
*** 1 cluster ***
POS: 20 20 1.000000
NEG: 230 0.00427176
BACKGROUND PROCESSING TIME: 0.67

```

Hình 3.6: Minh họa quá trình huấn luyện tại bộ phân loại

Để chuyển đổi bộ phân loại vừa xây dựng được thành dạng các tập tin XML, ta thực hiện lệnh sau:

```
convert_cascade ----size="" <tên bộ phân
loại> <tên file xml>
```

Ví dụ:

```
convert_cascade --size="24x36" face_cascade Haar likecascade_lan2.xml
```

3.3 Lượng hóa kết quả huấn luyện

Để đánh giá kết quả việc nhận dạng thì có hai tiêu chí được hướng tới là độ chính xác cực đại và tỷ lệ lỗi cực tiểu. Hiện nay có rất nhiều phương pháp đánh giá, mỗi một phương pháp đều cho kết quả đánh giá không giống nhau, một phương pháp được sử dụng phổ biến hiện nay là phương pháp ROC (Receiver Operating Characteristics) curve.

Phương pháp ROC curve vẽ đồ thị đánh giá độ chính xác nhận dạng dựa trên hai thông số Tỷ lệ nhận dạng mẫu đúng (True Positive Rate) và Tỷ lệ nhận dạng mẫu sai (False Positive Rate), cụ thể

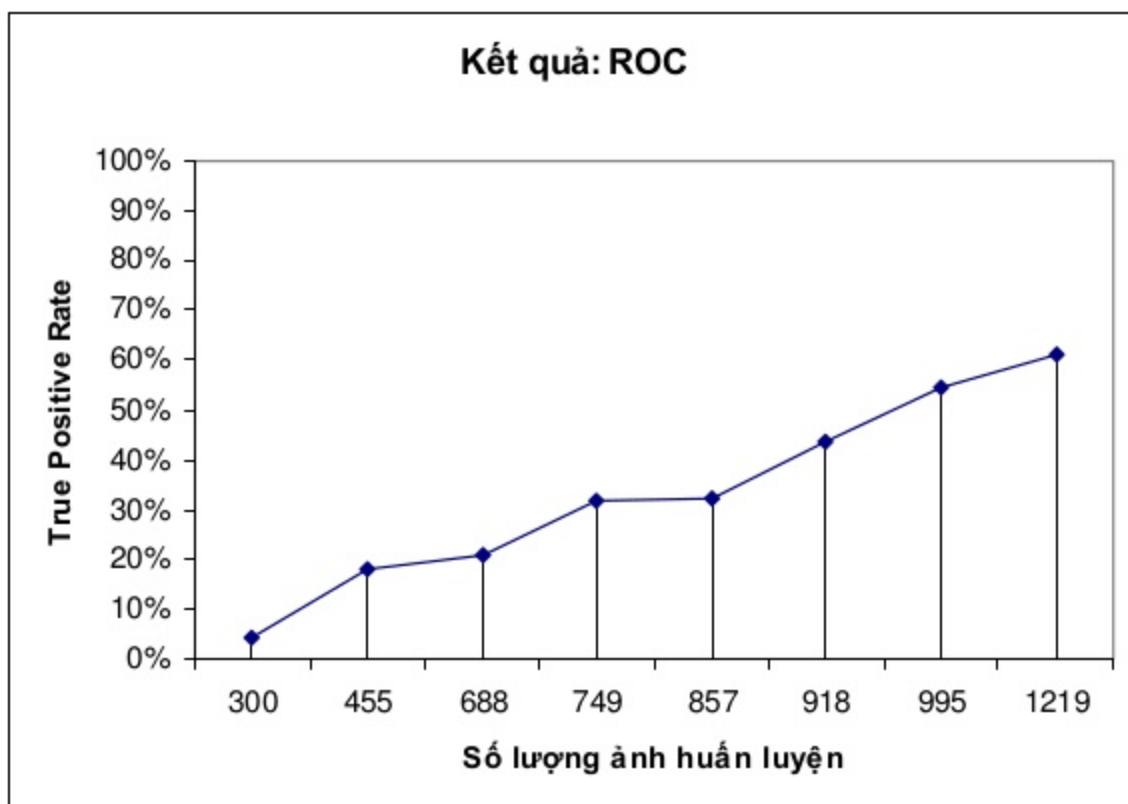
$$\text{Tỷ lệ nhận dạng mẫu (positive) đúng} = \frac{\text{Số lượng mẫu (positive) nhận dạng đúng}}{\text{Tổng số mẫu (positive) trong tập dữ liệu}}$$

$$\text{Tỷ lệ nhận dạng mẫu (positive) sai} = \frac{\text{Số lượng mẫu (positive) nhận dạng lỗi}}{\text{Tổng số mẫu (negative) trong tập dữ liệu}}$$

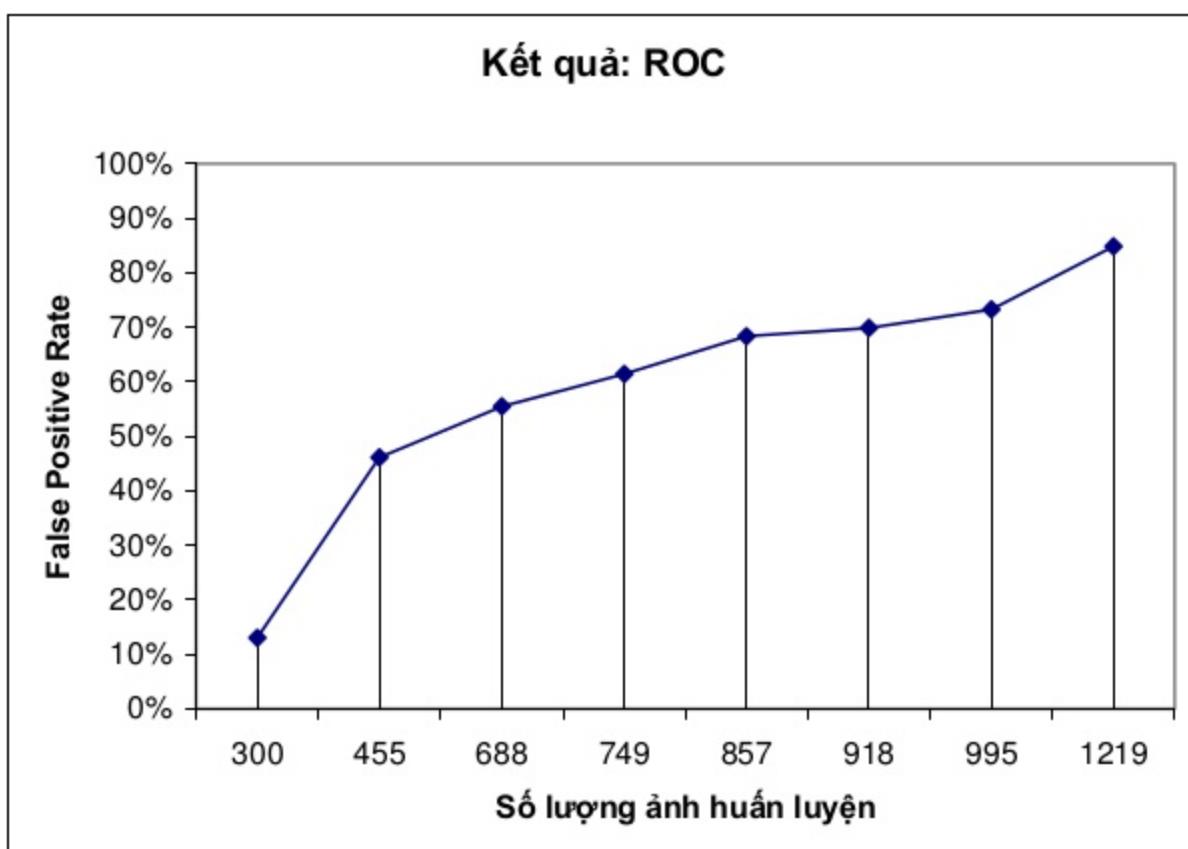
Dữ liệu dùng để đánh giá được lấy từ bộ Face CMU, lượng hóa kết quả huấn luyện theo phương pháp ROC curve thể hiện bằng biểu đồ dưới đây:

Số lượng ảnh huấn luyện	True Positive Rate	False Positive Rate
300	0.044848	0.130034
455	0.181481	0.461538
688	0.207407	0.554545
749	0.317417	0.614567
857	0.322445	0.683403
918	0.436684	0.698467
995	0.543821	0.732938
1219	0.610031	0.848475

Bảng 3.2: Kết quả huấn luyện



Hình 3.7: Biểu đồ đánh giá kết quả huấn luyện



Hình 3.9: Biểu đồ đánh giá tỷ lệ nhận dạng sai

Qua quá trình huấn luyện với số ảnh tăng dần thì tỷ lệ nhận dạng đúng cũng tăng lên từ dưới 10% lên trên 80%, đây là kết quả chấp nhận được.

3.4 Kiểm thử

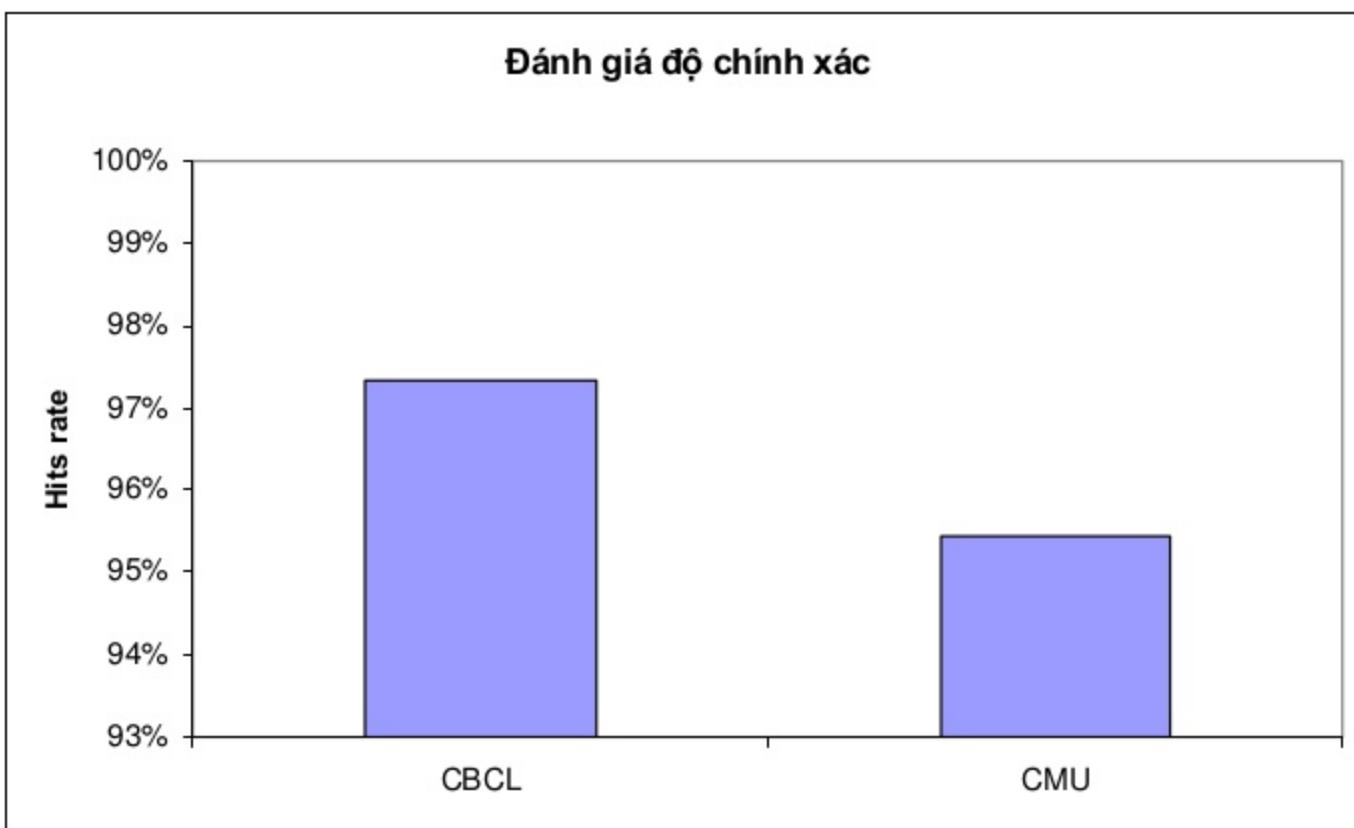
Quá trình kiểm thử thực hiện trên bộ dữ liệu kiểm thử CMU cho ra kết quả như bảng dưới đây:

Thử nghiệm	Số lượng positive	Hits	Missed	False
CBCL	472	306	694	20
CMU	256	244	384	7

Bảng 3.3: Kết quả kiểm thử hai bộ dữ liệu CBCL và CMU

Trong đó:

- Hits: là số lượng các mẫu nhận dạng đúng
- Missed: số lượng đối tượng chưa nhận dạng được (nghĩa là số lượng khuôn mặt có trong ảnh nhưng không nhận dạng được)
- False: số lượng đối tượng nhận dạng sai từ các ảnh không chứa đối tượng (tức số lượng đối tượng được nhận dạng nhưng không phải là khuôn mặt)

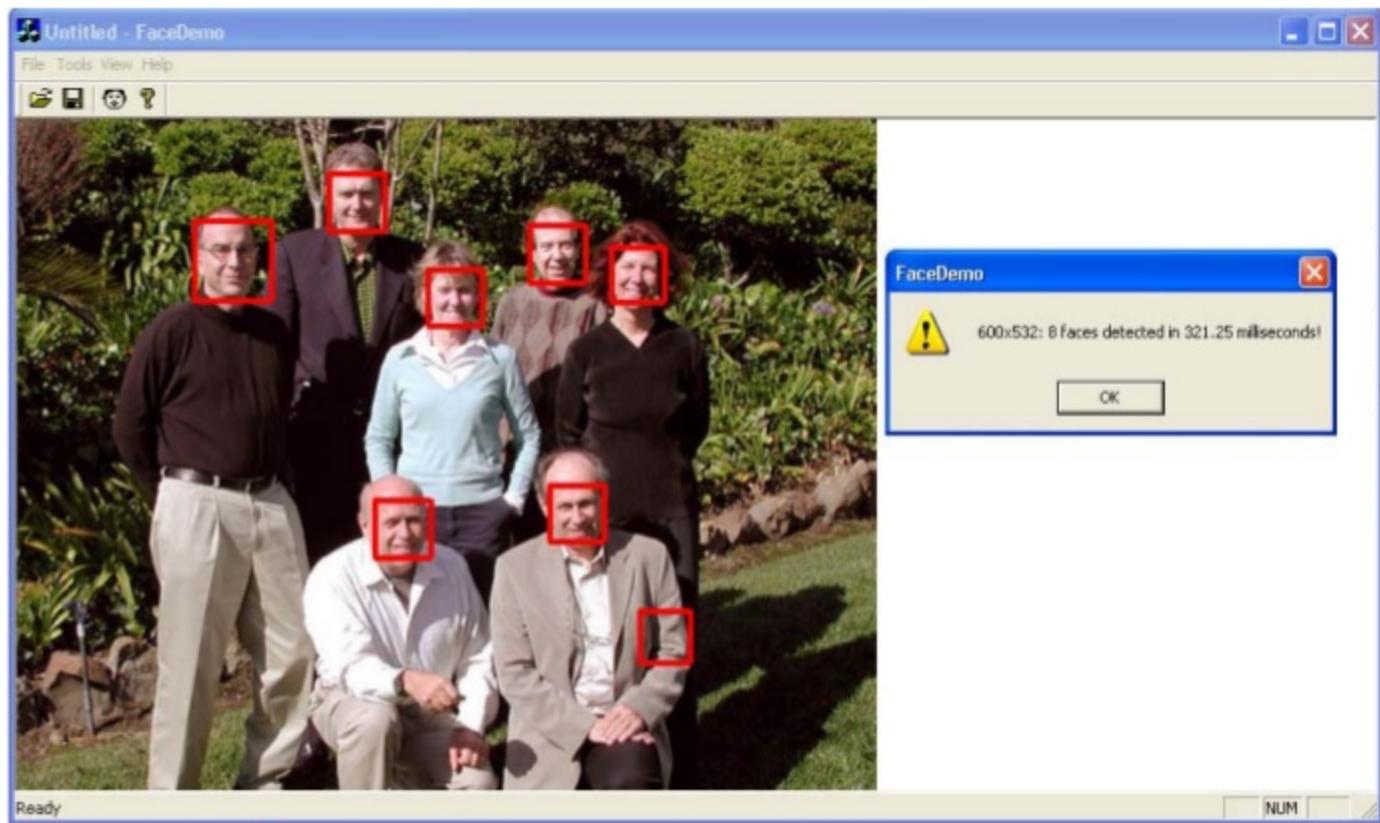


Hình 3.9: Tỉ lệ nhận dạng chính xác đối tượng của 2 bộ: CBCL và CMU

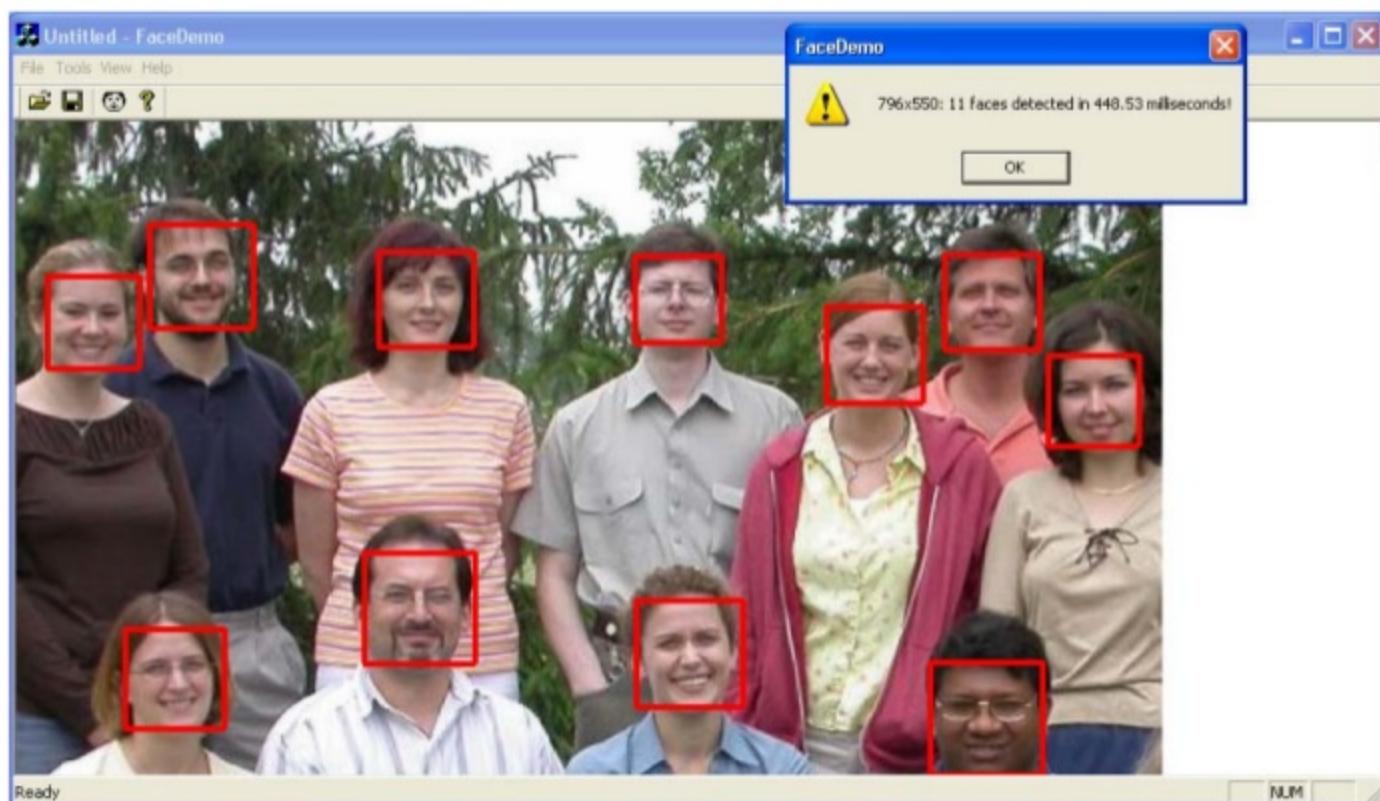
Thực nghiệm trên hai tập dữ liệu CBCL, CMU cho thấy độ chính xác khi thử nghiệm trên bộ CBCL là hơn 97% cao hơn, CMU chỉ đạt trên 95%. Kết quả trên cho thấy AdaBoost có tỷ lệ nhận dạng chính xác rất cao.

3.5 Một số hình ảnh minh họa kết quả nhận dạng

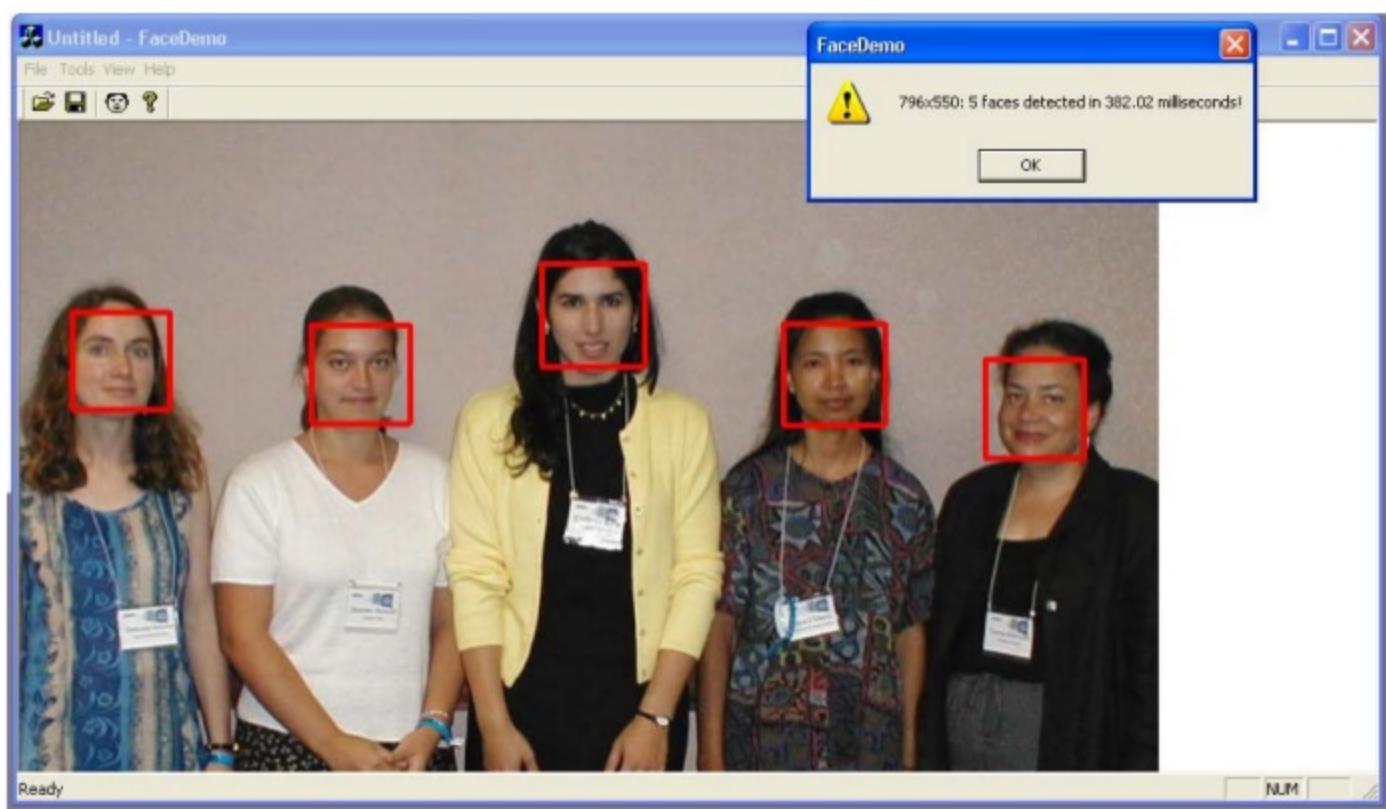
Minh họa kết quả kiểm thử trên tập dữ liệu CBCL:



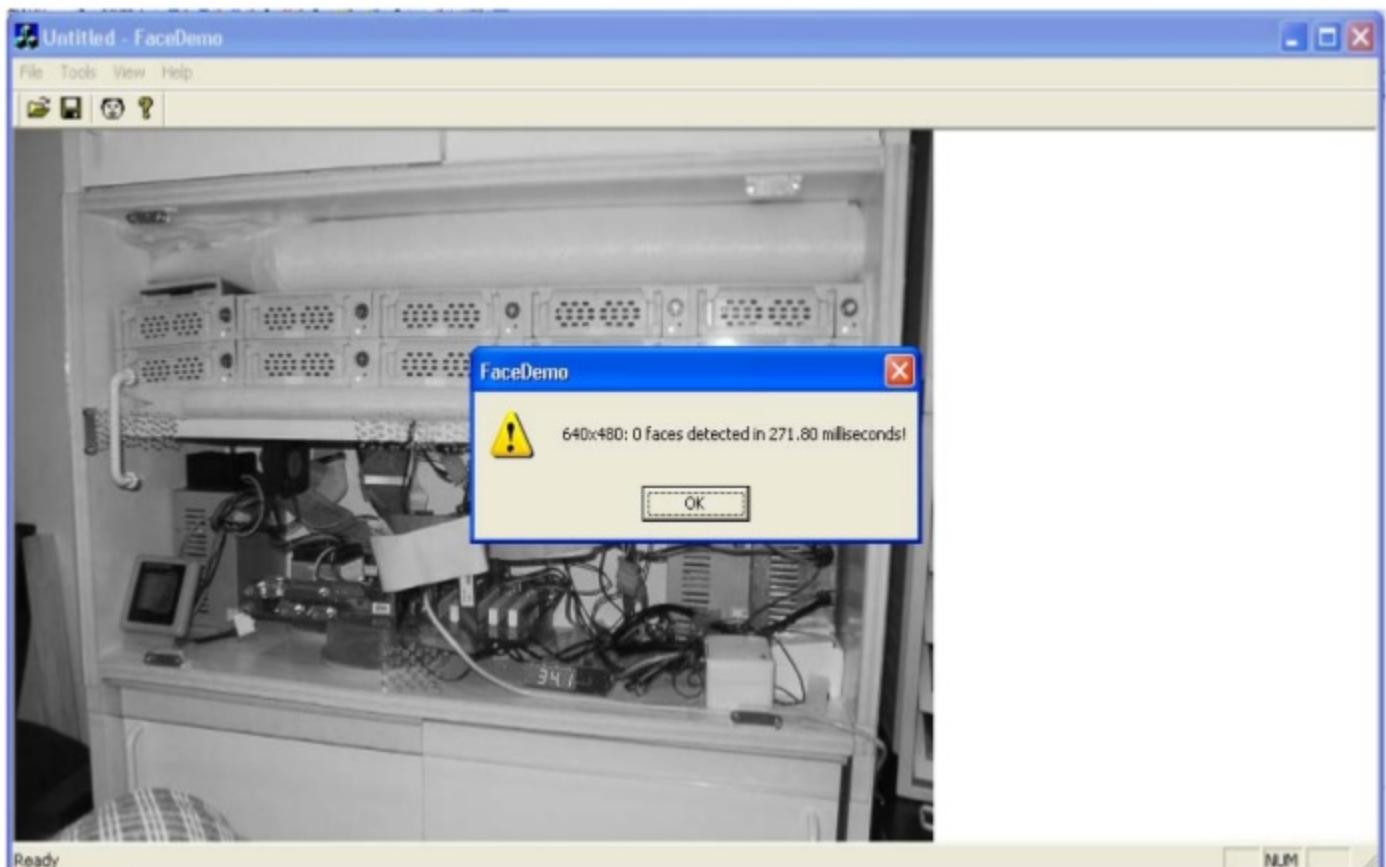
Hình 3.10: Nhận dạng 8 face trong 321.25s



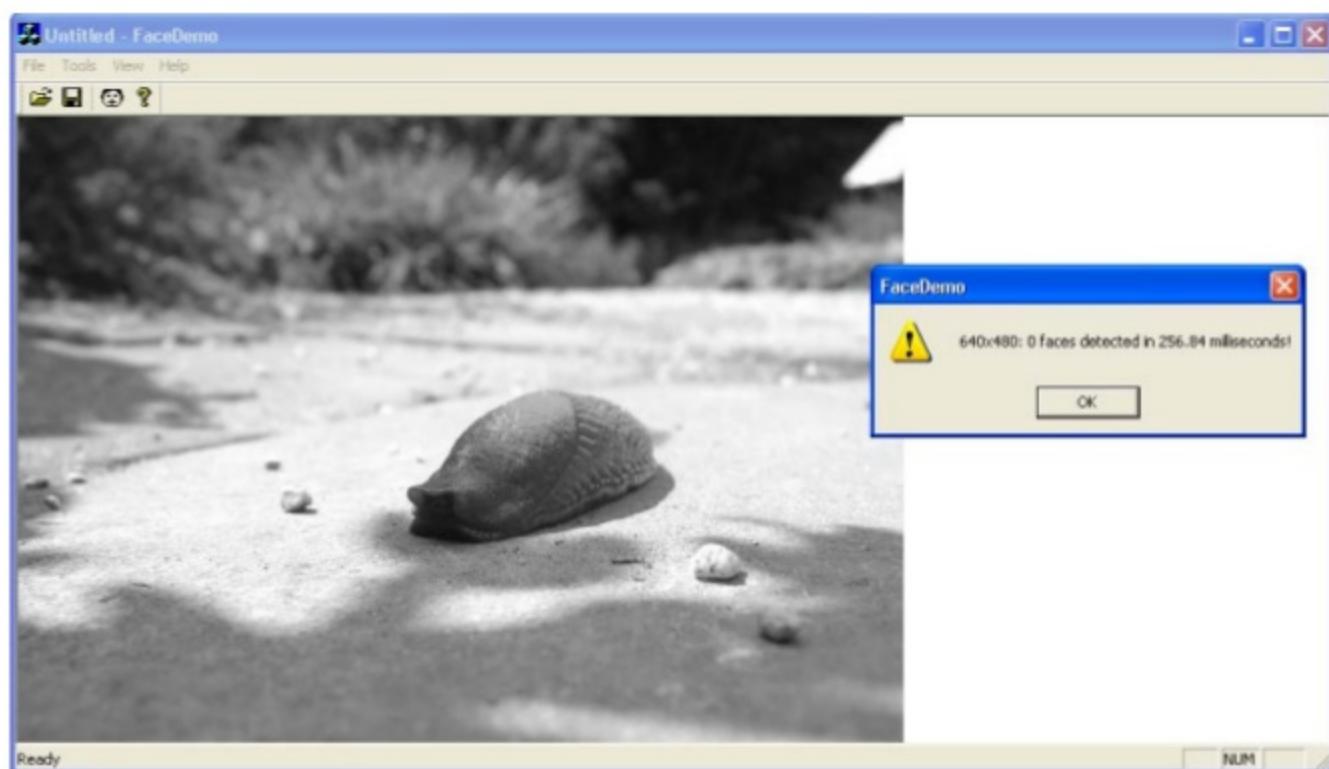
Hình 3.11: Nhận dạng 11 face trong 448.53s



Hình 3.12: Nhận dạng 7 face trong 382.02s

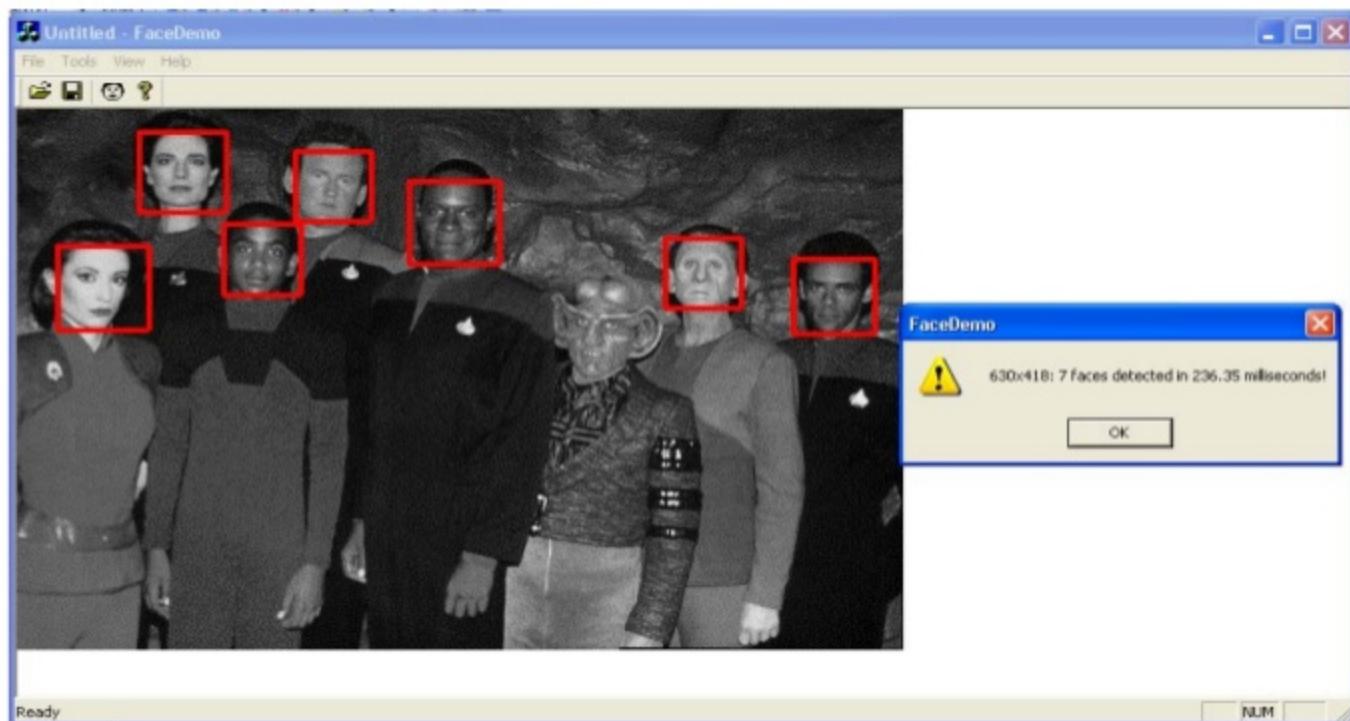


Hình 3.13: 0 face được phát hiện trong 27,80s

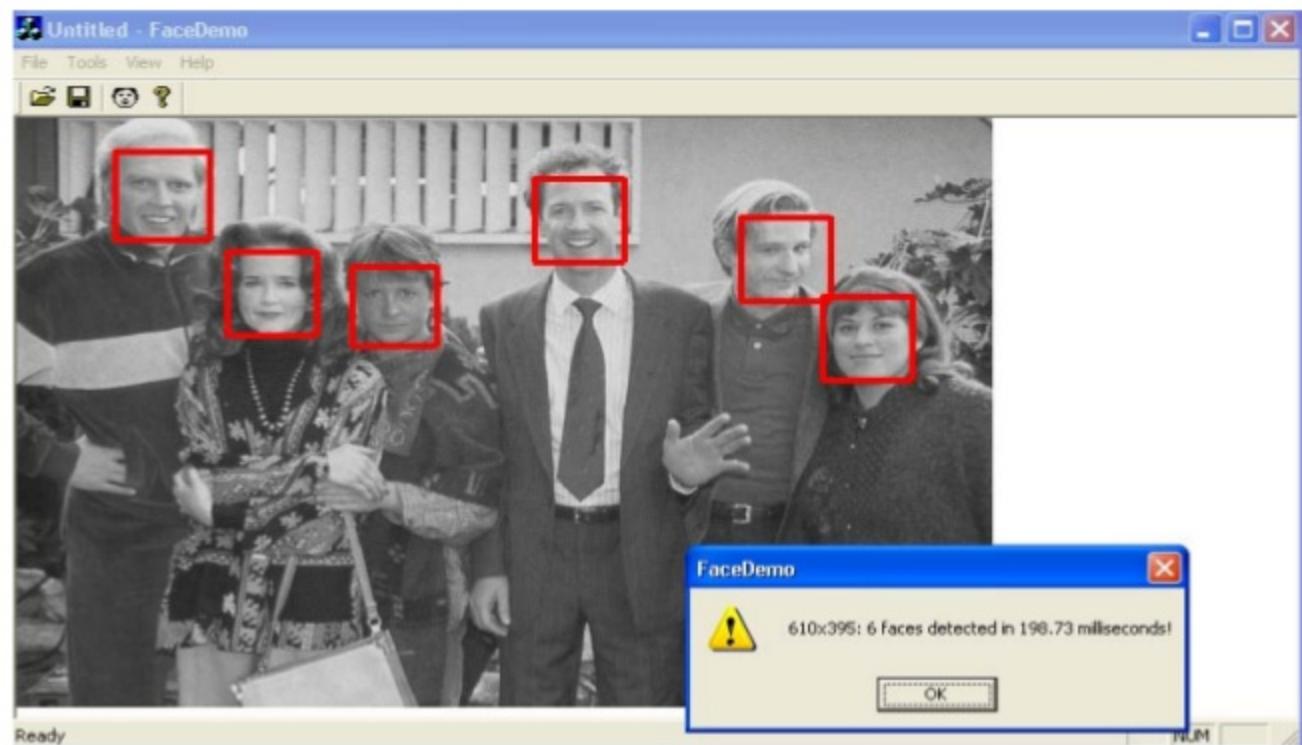


Hình 3.14: 0 face được phát hiện trong 256,84s

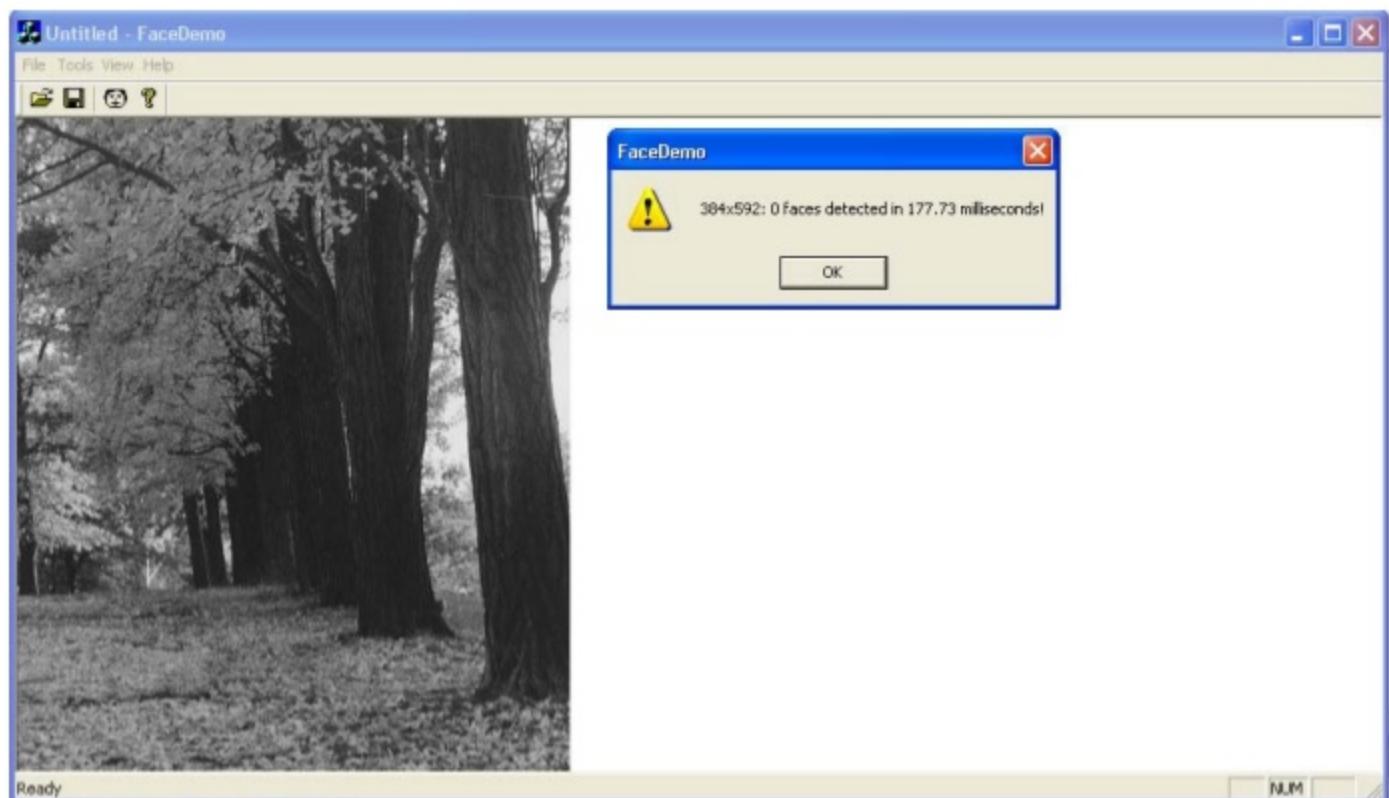
Minh họa kết quả kiểm thử trên tập dữ liệu CMU



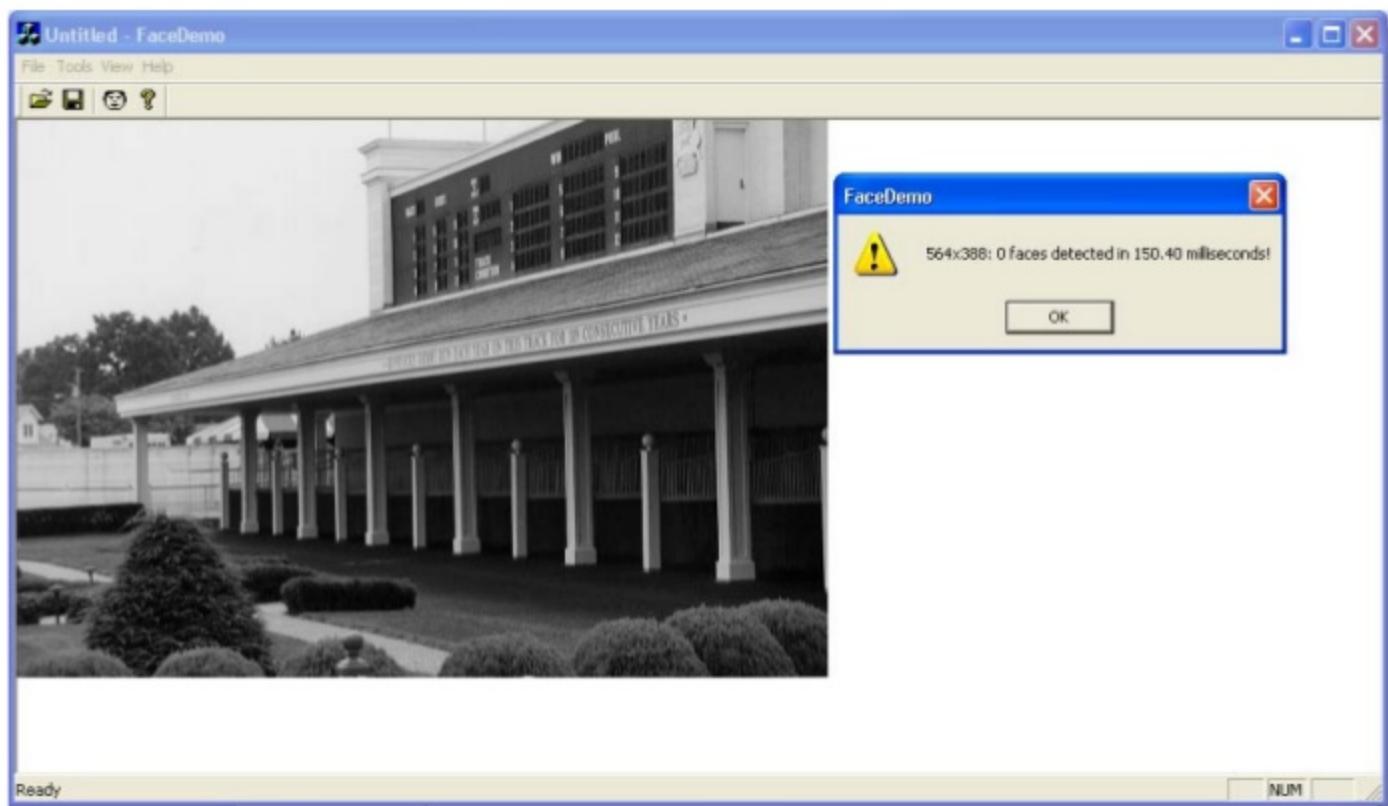
Hình 3.15: 7 face được phát hiện trong 236,35s



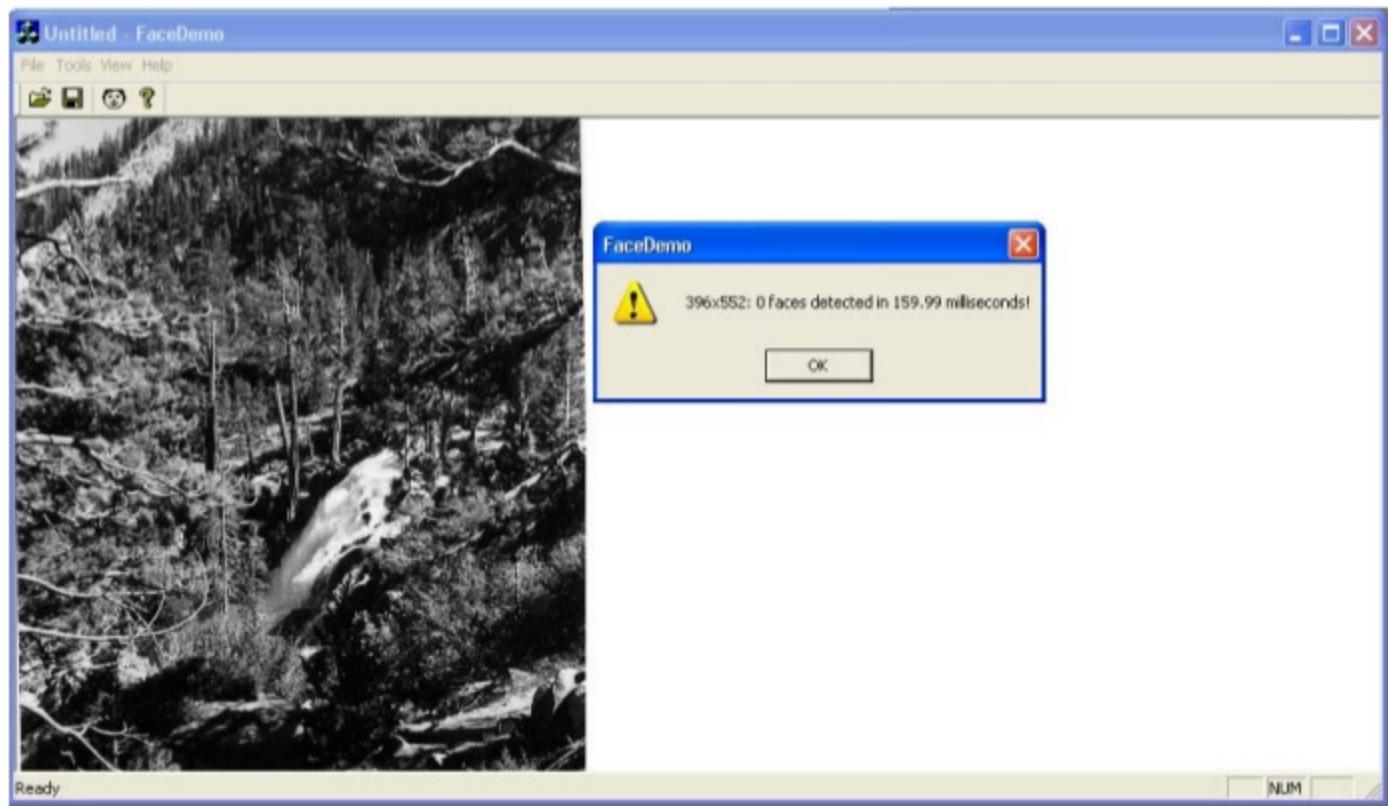
Hình 3.16: 6 face được phát hiện trong 198,73s



Hình 3.17: 0 face được phát hiện trong 177,73s

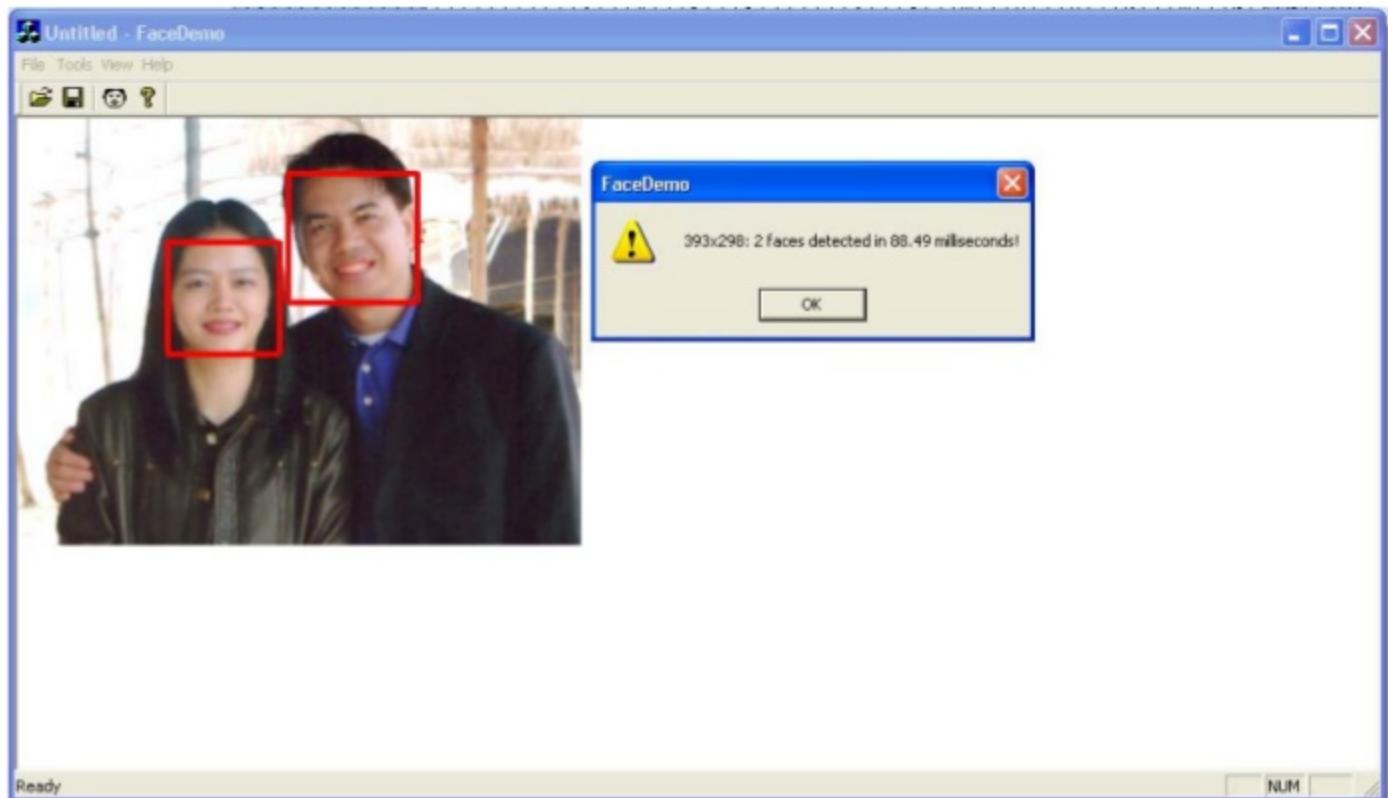


Hình 3.18: 0 face được phát hiện trong 150,40s

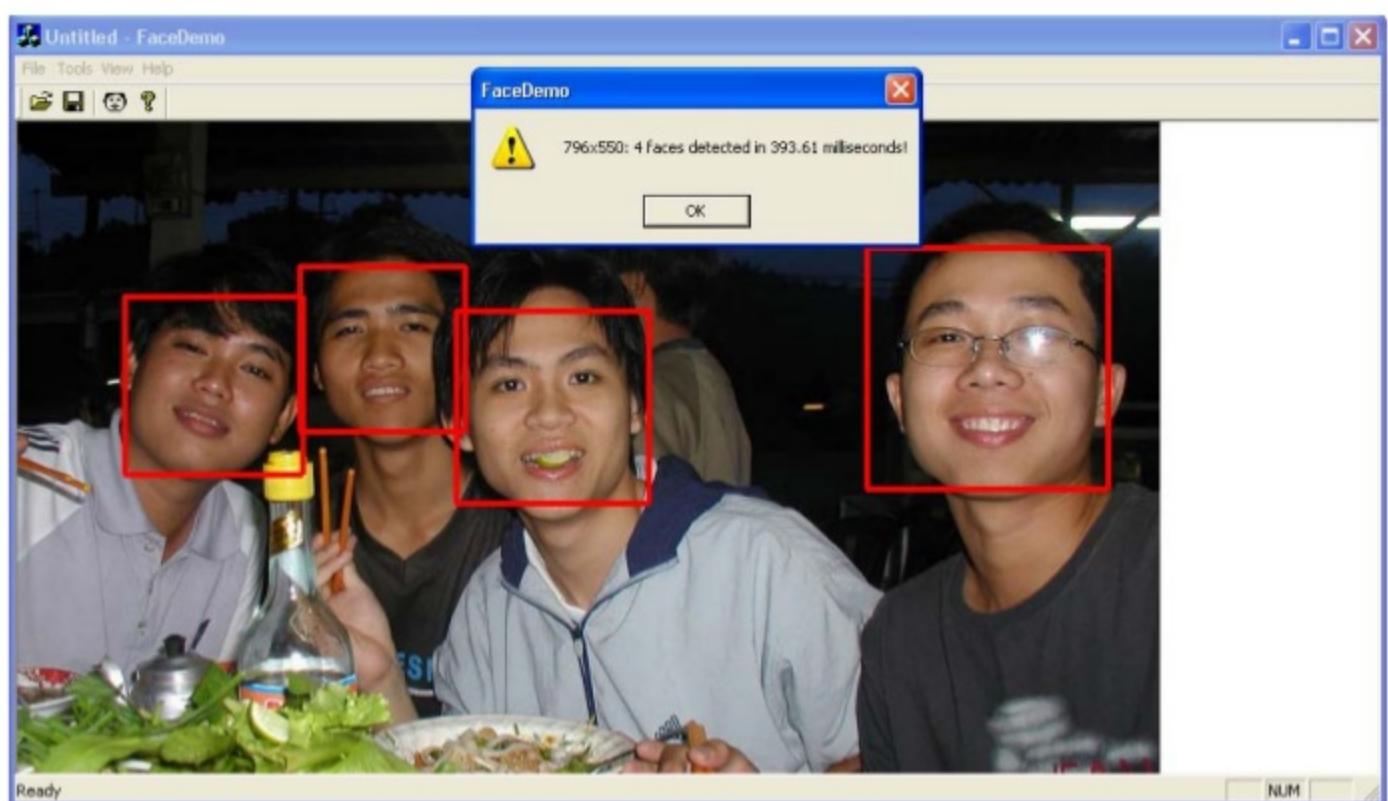


Hình 3.19: 0 face được phát hiện trong 159,99s

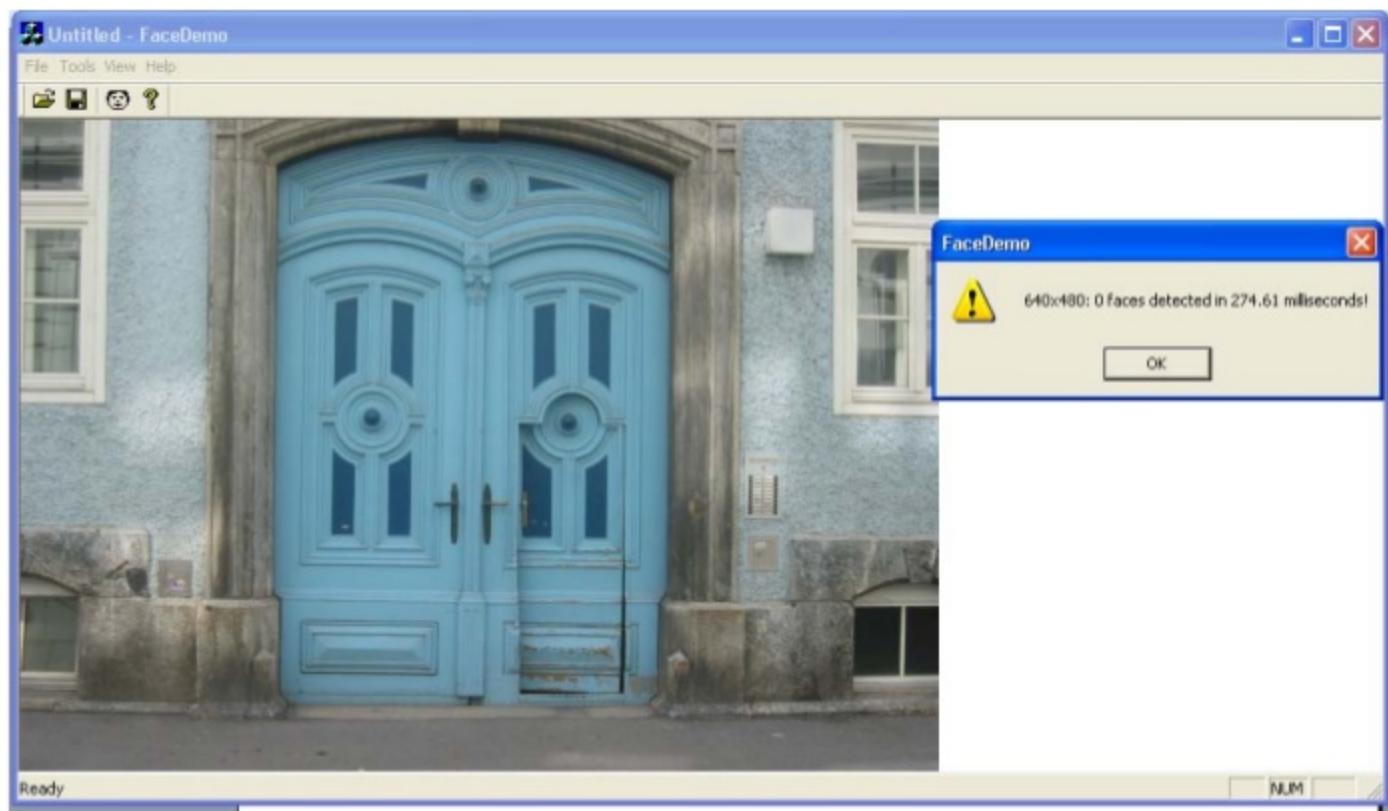
Minh họa kết quả kiểm thử trên một số ảnh thử nghiệm



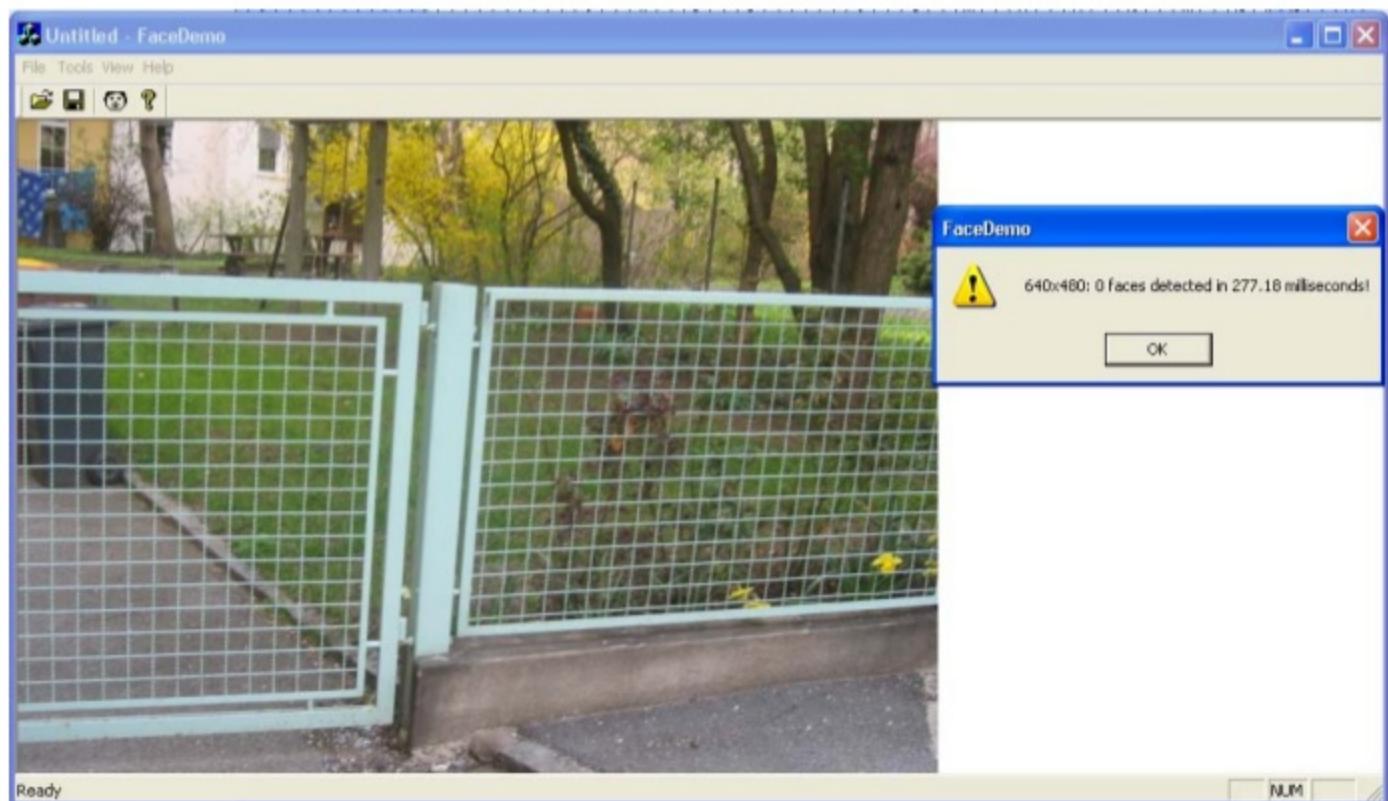
Hình 3.20: 2 face được phát hiện trong 88,49s



Hình 3.21: 4 face được phát hiện trong 393,61s



Hình 3.22: 0 face được phát hiện trong 274,61s



Hình 3.23: 0 face được phát hiện trong 277,18s

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Kết luận

Trên cơ sở tìm hiểu và nghiên cứu thuật toán AdaBoost và kỹ thuật nhận dạng đối tượng cụ thể là nhận dạng khuôn mặt, luận văn đã đạt được những kết quả như sau:

- Giới thiệu khái quát về bài toán nhận dạng đối tượng, mô hình nhận dạng đã và đang được sử dụng trên thế giới, khái quát về lịch sử phát triển của thuật toán AdaBoost và những công trình nghiên cứu uy tín về vấn đề này.

- Trình bày thuật toán AdaBoost, tìm hiểu về mô hình cascade bộ phân loại áp dụng thuật toán cho bài toán nhận dạng đối tượng (cụ thể là nhận dạng khuôn mặt) đó là xây dựng mô hình phân tầng với sử dụng đặc trưng Haar-like.

- Tiến hành thực nghiệm được việc huấn luyện với bộ dữ liệu khuôn mặt Feret, đánh giá kết quả thu được với hai bộ dữ liệu là CBCL và CMU, kết quả thu được tuy còn rất hạn chế do thời gian tiến hành thực nghiệm không nhiều tuy nhiên qua đó đã giới thiệu được quy trình huấn luyện và phương pháp huấn luyện nhận dạng ảnh. Tuy thực nghiệm đã cho kết quả nhưng đây là kết quả chưa tốt, do bởi số lượng mẫu thử ít và thời gian để thực hiện không được nhiều.

2. Hướng phát triển

- Qua kết quả thực nghiệm thu được đã cho thấy việc nhận dạng đối tượng đòi hỏi một quá trình lâu dài vì thế trước mắt tiếp tục tiến hành huấn luyện với những bộ dữ liệu khác với số lượng ảnh lớn hơn để nhận dạng đạt kết quả tốt hơn.

- Nhận dạng đối tượng là một đề tài rất thú vị và có nhiều ứng dụng trong khoa học kỹ thuật và đời sống. Trong luận văn này, chỉ nghiên cứu thuật toán AdaBoost trong việc nhận dạng và kết quả còn hạn chế nên việc cải thiện các thuật toán cũng như xây dựng hoàn thiện một hệ thống tìm kiếm và nhận dạng với độ chính xác cao để có thể ứng dụng trong đời sống xã hội cũng là một hướng nghiên cứu thú vị và phức tạp đòi hỏi nhiều công sức lao động và nghiên cứu. Đề

hướng tới mục tiêu này, cần nghiên cứu nhiều mô hình nhận dạng khác đang rất được các nhà nghiên cứu quan tâm như sử dụng mô hình Markov nhiều chiều, thực hiện lấy mẫu ảnh với độ chính xác cao hơn.

- Một hướng phát triển khác là áp dụng thuật toán AdaBoost nhưng với các đặc trưng khác như: đặc trưng local, garbor,... Tuy nhiên, một vấn đề cần phải thừa nhận rằng rất khó có thể tồn tại một giải pháp tổng thể cho tất cả các bài toán đặc biệt khi áp dụng vào thực tiễn cuộc sống. Do đó cần phải có những hạn chế cần thiết cho phù hợp với miền ứng dụng ví dụ: có thể chỉ tìm kiếm một khuôn mặt duy nhất hoặc yêu cầu hạn chế vùng tìm kiếm khi xây dựng các ứng dụng dựa trên mặt người...

TÀI LIỆU THAM KHẢO

Tiếng Anh

1. A. Lanitis., C.J. Taylor and T.F. Cootes (1995), *A Unified Approach to Coding and Interpreting Face Images*, Int'l Conf. Computer Vision, pp. 368-373.
2. Agarwal S and Roth D (2002), *Learning a sparse representation for object detection*, In ECCV-2002.
3. Carbonetto, S.T., M.M. Gruver., D.C. Turner (1982), *Nerve fiber growth on defined hydrogel substrates*, Science 21,pp. 897-899
4. Eng-Jon Ong and Bowden R (2004), *A Boosted Bô phân loại Tree for Hand Shape Detection*, In Proc. IEEE Intl. Conference on Automatic Face and Gesture Recognition.
5. G. Dorko and C. Schmid (2004), *Object class recognition using discriminative local features*, IEEE Transactions on Pattern Analysis and Machine Intelligence.
6. I. Craw and P. Cameron (1992), *Face Recognition by Computer*, British Machine Vision Conf., pp. 498–507.
7. Johan Thureson and Stefan Carlsson (2004), *Appearance Based Qualitative Image Description for Object Class Recognition*, ECCV (2) 2004, pp. 518-529.
8. K.C.Yow, R.Cipolla (1997), *Feature-based Human Face Detection*, Image and Vision Computing, 15(9), pp. 713-735.
9. K. Sung and T. Poggio (1998), *Example-Based Learning for View-Based Human Face Detection*, Journal of Artificial Intelligence Research, v.9 n.1, pp.167-217.

10. M. Turk and A. Pentland (1991), *Face recognition using eigenfaces*, IEEE Conference on Computer Vision and Pattern Recognition, pp. 586–591.
11. Ming-Hsuan Yang., David J. Kriegman., Narendra Ahuja (2002), *Detecting Faces in Images: A Survey*, IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI), vol 1, pp. 34-47.
12. E.Osuna., R.Freund and F.Girosi (1997), *An improved huân luyện algorithm for support vector machines*, IEEE CVPR.
13. Paul Viola and Michael J. Jones (2001), *Rapid Object Detection using a Boosted Cascade of Simple feature* , IEEE CVPR.
14. P. Viola and M. J. Jones (2004), *Robust real-time face detection*, International Journal of Computer Vision, 57(2), pp.137-154.
15. Robert E. Schapire and Yoram Singer (1999), *Improved boosting algorithms using confidence-rated predictions*, Machine Learning, 37(3), pp.297–336.
16. Rajagopal and A. Kahn (1998), *Molecular Level Alignment at Organic Semiconductor-metal Interfaces*, Appl. Phys. Lett, pp.73- 662.
17. Rowley H., Baluja S., and Kanade T (1998), *Neural network-based face detection*, IEEE Patt, Anal, Mach Intell, pp.22–38.
18. R Bowden and M Sarhadi (2000), *Building temporal models for gesture recognition*, In Proc, BMVC volume 1, pages 32–41.
19. Schneiderman H and Kanade T (2000), *A statistical method for 3D object detection applied to faces and car*, In International Conference on Computer Vision.

20. Stan Z. Li, Zhen Qiu Zhang (2004), *FloatBoost Learning and Statistical Face Detection*, IEEE Transaction on Pattern Analysis and Machine Intelligence PAMI), Vol. 26, pp.1-12.
21. Stan Z. Li., Juwei Lu (2004), *Face Detection, Alignment and Recognition*, Book Chapter 9, pp. 385-455.
22. T.K. Leung., M.C. Burl., and P. Perona (1995), *Finding faces in cluttered scenes using random labelled graph matching*, In Int. Conf. on Computer Vision.
23. Yoav Freund and Robert E. Schapire (1995), *A decision-theoretic generalization of on-line learning and an application to boosting*, In Computational Learning Theory: Eurocolt '95, pp.23–37.
24. Yoav Freund and Robert E. Schapire (1996), *Experiments with a new boosting algorithm*, In Machine Learning:Proceedings of the Thirteenth International Conference, pp.148–156.
25. Yoav Freund and R. Schapire (1997), *A decision-theoretic generalization of on-line learning and an application to boosting*, Journal of Computer and System Sciences, 55(1), pp.119–139.
26. William T. Freeman., Michal Roth (1995), *Orientation Histograms for Hand Gesture Recognition*, In Proc, IEEE Intl, Wkshp, on Automatic Face and Gesture Recognition, Zurich.