

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG**

-----o0o-----



**ISO 9001 : 2008**

# **ĐỒ ÁN TỐT NGHIỆP**

## **NGÀNH CÔNG NGHỆ THÔNG TIN**

**HẢI PHÒNG 2017**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**  
**TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG**

-----o0o-----

**ỨNG DỤNG PHÉP TOÁN HÌNH THÁI CHO**  
**BÀI TOÁN PHÁT HIỆN KHUÔN MẶT**  
**DỰA VÀO MÀU DA**

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY**

**Ngành: Công nghệ Thông tin**

Sinh viên thực hiện: Trịnh Trung Thành

Mã số sinh viên: 1312101015

Cán bộ hướng dẫn: Ts. Ngô Trường Giang

**HẢI PHÒNG – 2017**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**  
**TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG**

**CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  
*Độc lập - Tự do - Hạnh phúc*  
-----000-----

## **NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP**

Sinh viên: Trịnh Trung Thành

Mã sinh viên: 1312101015

Lớp: CT1701

Ngành: Công nghệ Thông tin

Tên đề tài: Ứng dụng phép toán hình thái cho bài toán phát hiện khuôn mặt dựa vào màu da

## NHIỆM VỤ ĐỀ TÀI

1. Nội dung và các yêu cầu cần giải quyết trong nhiệm vụ đề tài tốt nghiệp

a. Nội dung

b. Các yêu cầu cần giải quyết

2. Các số liệu cần thiết để thiết kế, tính toán

3. Địa điểm thực tập

## CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

**Người hướng dẫn thứ nhất:**

Họ và tên: Ngô Trường Giang

Học hàm, học vị: Tiến Sĩ

Cơ quan công tác: Trường Đại Học Dân Lập Hải Phòng

Nội dung hướng dẫn:

.....  
.....  
.....  
.....

**Người hướng dẫn thứ hai:**

Họ và tên:

Học hàm, học vị:

Cơ quan công tác:

Nội dung hướng dẫn:

.....  
.....  
.....  
.....

Đề tài tốt nghiệp được giao      ngày      tháng      năm 2017

Yêu cầu phải hoàn thành trước      ngày      tháng      năm 2017

Đã nhận nhiệm vụ: Đ.T.T.N

Đã nhận nhiệm vụ: Đ.T.T.N

**Sinh viên**

**Cán bộ hướng dẫn Đ.T.T.N**

**Ts. Ngô Trường Giang**

*Hải Phòng, ngày ..... tháng ..... năm 2017*

**HIỆU TRƯỞNG**

**GS.TS.NGUYỄN Trần Hữu Nghị**

**PHÂN NHẬN XÉT TÓM TẮT CỦA CÁN BỘ HƯỚNG DẪN**

**1. Tinh thần thái độ của sinh viên trong quá trình làm đề tài tốt nghiệp:**

.....

.....

.....

.....

.....

.....

**2. Đánh giá chất lượng của đề tài tốt nghiệp (so với nội dung yêu cầu đã đề ra trong nhiệm vụ đề tài tốt nghiệp)**

.....

.....

.....

.....

.....

.....

.....

**3. Cho điểm của cán bộ hướng dẫn:**

**(Điểm ghi bằng số và chữ)**

.....

.....

*Ngày ..... tháng ..... năm 2017*

**Cán bộ hướng dẫn chính**

*(Ký, ghi rõ họ tên)*

**PHÂN NHẬN XÉT ĐÁNH GIÁ CỦA CÁN BỘ CHẤM PHẢN BIỆN ĐỀ  
TÀI TỐT NGHIỆP**

**1. Đánh giá chất lượng đề tài tốt nghiệp (về các mặt như cơ sở lý luận, thuyết minh chương trình, giá trị thực tế, ...)**

.....  
.....  
.....  
.....  
.....  
.....

**2. Cho điểm của cán bộ phản biện  
(Điểm ghi bằng số và chữ)**

.....  
.....

*Ngày ..... tháng ..... năm 2017*

**Cán bộ chấm phản biện**

*(Ký, ghi rõ họ tên)*

## LỜI CẢM ƠN

*Trước hết, em xin bày tỏ lòng biết ơn sâu sắc tới giáo viên hướng dẫn là Thầy giáo, TS Ngô Trường Giang, thầy đã tận tình hướng dẫn, chỉ bảo trong suốt quá trình tìm hiểu nghiên cứu và hoàn thành đề án từ lý thuyết cho tới khi hoàn thiện chương trình.*

*Em xin chân thành cảm ơn đến các quý thầy, cô trong khoa Công nghệ Thông tin - Trường Đại học Dân lập Hải Phòng, chân thành cảm ơn vì thầy, cô đã tham gia giảng dạy và truyền đạt những kiến thức quý báu trong suốt thời gian em học tập tại trường.*

*Cuối cùng, em xin chân thành cảm ơn đến các thầy, cô và bạn bè đã tận tình giúp đỡ em những gì còn thiếu sót trong quá trình làm báo cáo và hoàn thành đề án.*

*Vì thời gian có hạn, trình độ hiểu biết và nhận thức còn chưa cao cho nên trong đề án không thể tránh khỏi những thiếu sót, em rất mong nhận được sự đóng góp ý kiến của các thầy cô và bạn bè để em có thể hoàn thiện đề án này tốt hơn.*

*Em xin chân thành cảm ơn!*

*Hải Phòng, ngày 27 tháng 12 năm 2017*

*Sinh viên thực hiện*

*Trịnh Trung Thành*



## MỤC LỤC

LỜI CẢM ƠN .....	1
MỤC LỤC.....	9
DANH MỤC HÌNH VẼ VÀ BẢNG BIỂU .....	11
MỞ ĐẦU.....	12
<b>CHƯƠNG 1: TỔNG QUAN VỀ PHÁT HIỆN KHUÔN MẶT.....</b>	<b>13</b>
1.1 Tổng quan về xử lý ảnh .....	13
1.1.1 Một số khái niệm.....	13
1.1.2 Các không gian màu.....	14
1.1.3 Phân đoạn ảnh .....	18
1.2 Phép toán hình thái trong xử lý ảnh.....	19
1.2.1 Phân tử cấu trúc.....	19
1.2.2 Phép giãn nở ( Dilation ).....	20
1.2.3 Phép co ( Erosion ).....	20
1.2.4 Phép mở ( Open ) .....	20
1.2.5 Phép đóng ( Close ).....	21
1.2.6 Một số ứng dụng của phép toán hình thái.....	21
1.3 Phát hiện khuôn mặt trong ảnh .....	26
1.3.1 Giới thiệu về phát hiện khuôn mặt.....	26
1.3.2 Các ứng dụng của phát hiện khuôn mặt.....	26
1.3.3 Một số hướng tiếp cận trong bài toán phát hiện khuôn mặt .....	27
<b>CHƯƠNG 2: MỘT SỐ PHƯƠNG PHÁP PHÁT HIỆN KHUÔN MẶT .....</b>	<b>29</b>
2.1 Phương pháp dựa trên màu da và thông tin biên. ....	29
2.1.1 Xác định các vùng da trong ảnh.....	29
2.1.2 Xác thực khuôn mặt .....	31
2.2 Phương pháp dựa trên nhiều chứng cứ .....	32
2.2.1 Xác định các vùng da trong ảnh.....	32
2.2.2 Xác thực khuôn mặt .....	33
2.3 Phương pháp dựa trên phân tích wavelet.....	35
2.3.1 Xác định các vùng da trong ảnh.....	36
2.3.2 Xác thực khuôn mặt .....	39

2.4	Phương pháp phát hiện khuôn mặt dựa vào đặc trưng Haarlike .....	41
2.4.1	Đặc trưng Haar-like.....	41
2.4.2	Xây dựng bộ phân lớp sử dụng Adaboost.....	44
2.4.3	Xác thực khuôn mặt .....	45
<b>CHƯƠNG 3: ỨNG DỤNG PHÉP TOÁN HÌNH THÁI TRONG BÀI TOÁN</b>		
<b>PHÁT HIỆN KHUÔN MẶT .....</b>		<b>46</b>
3.1	Phát biểu bài toán.....	46
3.2	Xây dựng bộ phân lớp.....	47
3.3	Xác định các vùng da trong ảnh.....	48
3.3.1	Phân đoạn ảnh dựa vào màu da.....	48
3.3.2	Nâng cấp ảnh bằng phép toán hình thái học .....	49
3.4	Xác thực khuôn mặt .....	52
3.5	Cài đặt chương trình.....	55
3.5.1	Ngôn ngữ lập trình MATLAB .....	55
3.5.2	Giao diện chương trình .....	55
3.5.3	Kết quả xác định các vùng màu da .....	57
3.5.4	Kết quả xác thực khuôn mặt .....	62
3.5.5	Nhận xét sau khi thực nghiệm bằng chương trình .....	65
<b>KẾT LUẬN .....</b>		<b>66</b>
<b>TÀI LIỆU THAM KHẢO .....</b>		<b>67</b>

## DANH MỤC HÌNH VẼ VÀ BẢNG BIỂU

- Hình 1.1: Sơ đồ tổng quát hệ thống xử lý ảnh  
Hình 1.2: Mô hình không gian màu RGB  
Hình 1.3: Mô hình không gian màu YcbCr  
Hình 1.4: Mô hình biểu diễn 2 thành phần màu H,S  
Hình 1.5: Mô hình không gian màu HSV  
Hình 1.6: Một số phần tử cấu trúc của ảnh nhị phân  
Hình 1.7: Ví dụ về trích lọc biên đối tượng trong ảnh  
Hình 1.8: Ví dụ về tô đầy vùng đối tượng trong ảnh  
Hình 1.9: Ví dụ về làm mỏng đối tượng trong ảnh  
Hình 1.10: Ví dụ về làm dày đối tượng trong ảnh  
Hình 1.11: Ví dụ về tìm xương đối tượng trong ảnh  
Hình 1.12: Ví dụ về tìm xương đối tượng trong ảnh  
Hình 2.1: Ví dụ về ảnh mẫu trong phương pháp của K. Sandeep  
Hình 2.2: Mô hình biểu diễn của RPROP  
Hình 2.3: Ví dụ về các tư thế đầu khác nhau trong Gabor  
Hình 2.4: Sự phân bố màu da trong không gian màu YcbCr và HSV  
Hình 2.5: Kết quả thực hiện sau thuật toán phát hiện vùng da đã đưa ra  
Hình 2.6: Minh họa outer và inner  
Hình 2.7: Đặc trưng Haarlike trên khuôn mặt người  
Hình 3.1: Mô hình biểu diễn các bước thực hiện  
Hình 3.2: Kết quả sau khi phát hiện màu da  
Hình 3.3: Ảnh sau khi phân đoạn xuất hiện nhiều nhiễu và các lỗ  
Hình 3.4 Thực hiện phép toán hình thái trên ảnh nhị phân  
Hình 3.5: Kết quả sau khi nâng cấp ảnh trên ảnh xám  
Hình 3.6: Biến đổi sang ảnh nhị phân  
Hình 3.7: Kết quả là các ứng viên được đóng khung  
Hình 3.8: Vị trí các khuôn mặt trên mỗi ứng viên  
Hình 3.9: Các vị trí khuôn mặt trong ảnh đầu vào  
Hình 3.10: Giao diện chính của chương trình  
Hình 3.11: Kết quả của bức ảnh có 1 khuôn mặt người  
Hình 3.12: Kết quả của ảnh có nhiều mặt người  
Hình 3.13: Một số bức ảnh có nền đơn giản  
Hình 3.14 Kết quả trên ảnh có nhiều vùng giống với nền  
Hình 3.15: Kết quả với các tham số khác nhau xác định màu da  
Hình 3.16: Kết quả nâng cấp ảnh không sử dụng phép toán hình thái  
Hình 3.17: Sự khác biệt khi sử dụng phép toán hình thái trên ảnh nhị phân và ảnh xám  
Hình 3.18: Kết quả chương trình với các ảnh có nền đơn giản  
Hình 3.19: Kết quả trong các ảnh có nhiều khuôn mặt  
Hình 3.20: Kết quả thực nghiệm trên một số hình có khuôn mặt không thẳng hoặc bị che khuất

## MỞ ĐẦU

Xử lý ảnh là một lĩnh vực mang tính khoa học và công nghệ. Là một trong những chuyên ngành quan trọng của công nghệ thông tin hiện nay được áp dụng trong những lĩnh vực khác nhau như y học, vật lý, toán học, tìm kiếm, bảo mật và rất nhiều lĩnh vực khoa học khác...

Phát hiện khuôn mặt là một phần trong lĩnh vực xử lý ảnh, là một vấn đề cơ bản trong ngành học thị giác máy. Đây là một trong những giai đoạn của hệ thống nhận dạng mặt người cùng với nhiều ứng dụng rộng rãi và phổ biến khác như chỉ số hóa nội dung trong ảnh, hệ thống giám sát, hội thảo truyền hình, phát hiện khuôn mặt đã và đang dành được sự quan tâm nghiên cứu của nhiều người trong suốt hai thập kỷ qua.

Để loại bỏ một số yếu tố ảnh hưởng tới độ chính xác trong phát hiện khuôn mặt thì cần thực hiện một số kỹ thuật tiền xử lý. Đồ án này sẽ trình bày một phương pháp phát hiện khuôn mặt, trong đó phép toán hình thái học được sử dụng trong giai đoạn tiền xử lý để xác định vùng màu da của ảnh. Đồ án được trình bày theo các nội dung như sau:

- Chương 1: Tổng quan về phát hiện khuôn mặt.
- Chương 2: Một số phương pháp phát hiện khuôn mặt
- Chương 3: Ứng dụng phép toán hình thái trong bài toán phát hiện khuôn mặt.

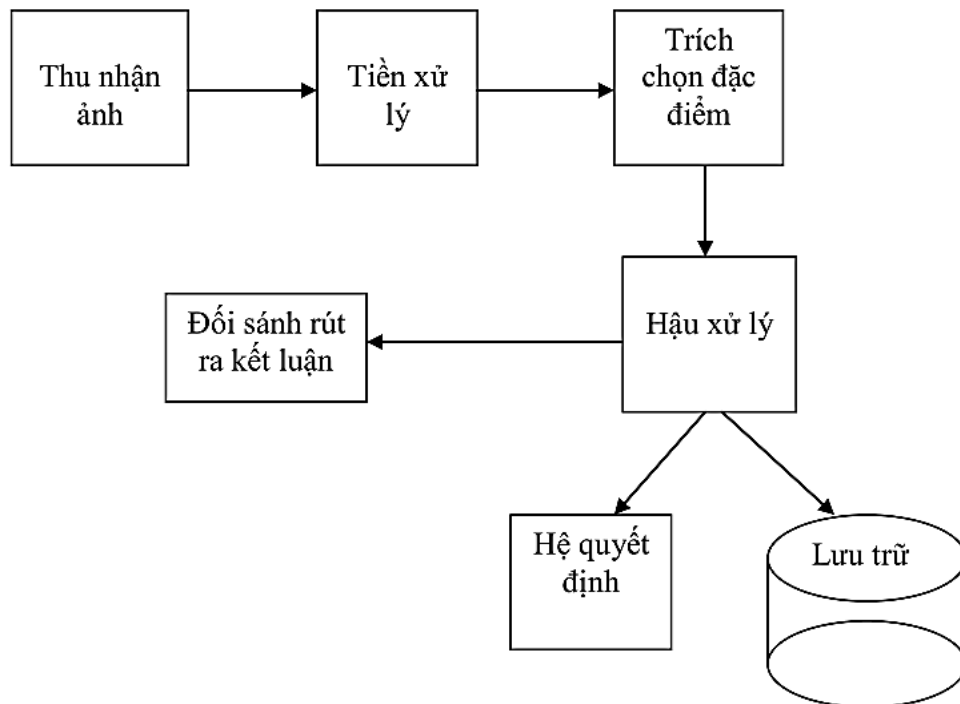
## CHƯƠNG 1: TỔNG QUAN VỀ PHÁT HIỆN KHUÔN MẶT

### 1.1 Tổng quan về xử lý ảnh

#### 1.1.1 Một số khái niệm

Ảnh là tập hợp của các điểm ảnh. Góc của ảnh là ảnh liên tục về không gian và độ sáng. Để xử lý bằng máy tính, ảnh cần phải được số hoá. Số hoá ảnh là sự biến đổi gần đúng một ảnh liên tục thành một tập điểm phù hợp với ảnh thật về vị trí (không gian) và độ sáng (mức xám). Khoảng cách giữa các điểm ảnh đó được thiết lập sao cho mắt người không phân biệt được ranh giới giữa chúng. Mỗi một điểm như vậy gọi là điểm ảnh (PEL:Picture Element) hay gọi tắt là Pixel. Trong khuôn khổ ảnh hai chiều, mỗi điểm ảnh (Pixel) ứng với cặp tọa độ  $(x, y)$ . Điểm ảnh (Pixel) là một phần tử của ảnh số tại tọa độ  $(x, y)$  với độ xám hoặc màu nhất định.

Xử lý ảnh là một loạt các thao tác và phân tích ảnh bằng máy tính nhằm cải thiện chất lượng ảnh cho tốt hơn và xử lý dữ liệu tự động trên máy. Quá trình này được xem như là thao tác ảnh đầu vào nhằm cho ra kết quả mong muốn. Kết quả đầu ra của quá trình sẽ là một ảnh tốt hơn hoặc một kết luận.



Hình 1.1: Sơ đồ tổng quát hệ thống xử lý ảnh

Ảnh đen trắng chỉ bao gồm 2 màu: màu đen và màu trắng. Người ta phân mức đen trắng đó thành L mức. Nếu L bằng 2, nghĩa là chỉ có 2 mức: mức 0 và mức 1 và còn gọi là ảnh nhị phân. Mức 1 ứng với màu sáng, còn mức 0 ứng với màu tối. Nếu L lớn hơn 2 ta có ảnh đa cấp xám. Việc xác định số mức là phụ thuộc vào tiêu chí lượng hóa. L thường chọn có 32, 64, 128 và 256 mức. Ảnh 256 mức là ảnh có chất lượng cao và thường được sử dụng.

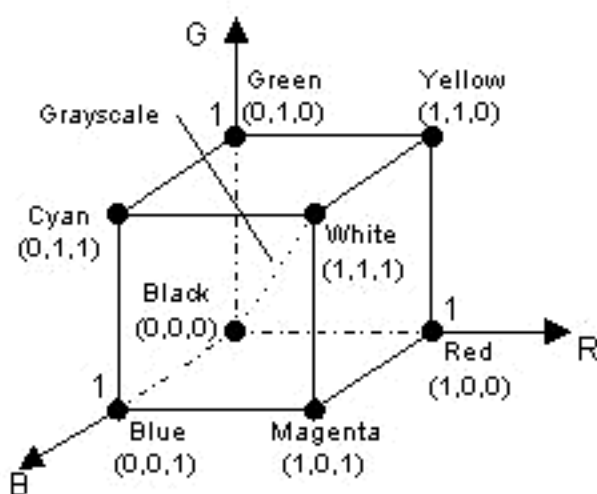
Với ảnh nhị phân, mỗi pixel mã hóa trên 1 bit; còn với ảnh 256 mức, mỗi pixel mã hóa trên 8 bit. Ví dụ với ảnh 256 mức xám, kích thước 512x512 cần không gian lưu trữ là 512x512 bytes hay 245 Kbytes

Ảnh màu là ảnh được tổ hợp từ 3 màu cơ bản: đỏ(R), lục(G), lam(B). Với ảnh màu, người ta lưu trữ thành từng màu riêng biệt, mỗi màu được lưu trữ như một ảnh đa cấp xám nên không gian nhớ dành cho một ảnh màu lớn gấp 3 lần một ảnh đa cấp xám cùng kích cỡ.

## 1.1.2 Các không gian màu

### 1.1.2.1 Không gian màu RGB

Không gian màu RGB mô tả màu sắc bằng 3 thành phần chính là Đỏ(R), Xanh lục(G) và Xanh lam(B). Không gian màu này có thể được biểu diễn như một khối lập phương 3 chiều với màu đỏ là trục x, màu xanh lục là trục y, và màu xanh lam là trục z. Hình 1.2 mô tả không gian màu RGB.



Hình 1.2: Mô hình không gian màu RGB

Giả sử một ảnh màu RGB được mã hóa bằng 24 bit với 8 bit cho một kênh màu, khi đó mỗi kênh màu này sẽ nhận giá trị trong khoảng từ 0 đến 255. Với mỗi giá trị khác nhau của mỗi kênh màu khi kết hợp với nhau sẽ được một màu khác nhau, vậy tổng cộng có  $255 \times 255 \times 255 = 1,66$  triệu màu sắc cho biểu diễn ảnh này.

### 1.1.2.2 Không gian màu YCbCr

Không gian màu YCbCr được sử dụng trong các hệ thống ảnh số, video hay trong các chuẩn nén JPEG và MPEG. Vì không gian màu RGB chỉ thuận tiện cho hiển thị lên màn hình nhưng lại không thuận tiện trong lưu trữ và chuyển đổi nên người ta đã đề xuất ra không gian màu này để thuận lợi cho việc nén và chuyển đổi.

Không gian màu YCbCr mô tả màu sắc bằng 3 thông số là Y – biểu diễn thành phần độ sáng, Cb – biểu diễn sắc tố xanh lục, Cr biểu diễn sắc tố đỏ. YCbCr là dạng biểu diễn số của hệ màu này (tức là các thành phần Y, Cr, Cb nhận các giá trị số nguyên), nó còn có dạng analog nữa, kí hiệu là YPrPb.

Công thức để chuyển đổi từ không gian màu RGB sang YCbCr phát biểu dưới dạng ma trận:

$$\begin{array}{r} Y \\ CB \\ CR \end{array} = \begin{array}{r} 16 \\ 128 \\ 128 \end{array} + \begin{array}{ccc} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{array} \begin{array}{l} R \\ * G \\ B \end{array} \quad [1.1]$$

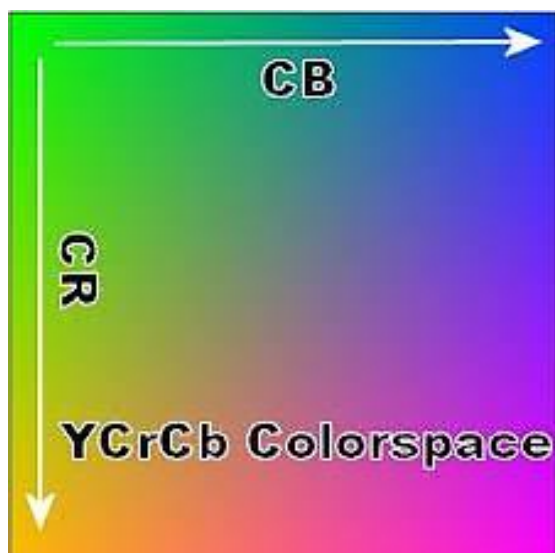
- Trong đó:

$$R/G/B = [0...255]$$

$$Y = [16...235]$$

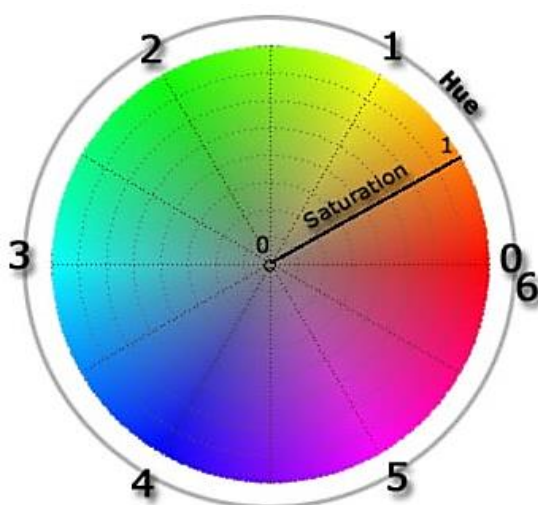
$$Cb/Cr = [16...240]$$

Đây là không gian màu gồm có 3 thành phần H, S, V ( Hue, Saturation, Value). Trong không gian màu này, các màu đều được biểu diễn dựa trên 3 thành phần H, S, V này.



**Hình 1.3: Mô hình không gian màu YcbCr**

H biểu diễn màu sắc vốn có của màu như: đỏ, xanh da trời, da cam,... Nó có giá trị từ 0 đến 360°. Hình sau minh họa giá trị của H,S và màu tương ứng:



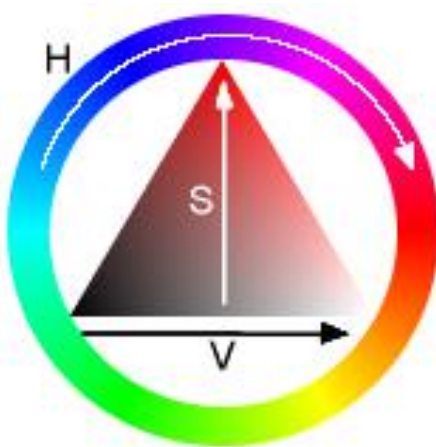
**Hình 1.4: Mô hình biểu diễn 2 thành phần màu H,S**

Ở trong hình, các giá trị của H đã được quy về dải  $[0,6]$  tương ứng với chia các góc cho 60 độ.

S biểu diễn độ bão hòa. Có thể hiểu nó giống như khái niệm nồng độ trong hóa học, với dung môi là màu trắng và chất tan là màu. S có giá trị nằm trong đoạn  $[0,1]$ . Khi giá trị của S lớn, lượng màu hòa trong màu trắng nhiều lên, màu sẽ đậm hơn.  $S=1$ , màu đậm nhất, chiếm hoàn toàn màu trắng. Và ngược lại, khi giá trị S nhỏ, lượng màu hòa trong màu trắng ít đi, màu sẽ nhạt



hơn.  $S=0$  tương ứng với không có một chút màu nào pha trong màu trắng, kết quả sẽ không có màu (cũng không khẳng định ngay kết quả là màu trắng, vì nó còn phụ thuộc vào thành phần  $V$  nữa, nhưng có thể khẳng định nó là màu xám). Ví dụ: khi  $H=0$ , ta có màu đỏ. Nhưng không phải cứ  $H=0$  là ta có được 1 màu đỏ đậm đà, nó còn phụ thuộc vào  $S$  (tức là phụ thuộc vào lượng màu đỏ hòa trong màu trắng). Khi  $S$  nhỏ, ta có màu đỏ nhạt. Khi  $S$  lớn ta có màu đỏ đậm hơn.  $S=1$  màu đỏ đậm nhất. Khi  $S=0$ , giá trị của  $H$  bằng bao nhiêu là vô nghĩa, kết quả cũng chỉ là 1 màu trắng (giả sử  $V=\max$ ).



**Hình 1.5: Mô hình không gian màu HSV**

Thành phần  $V$  biểu thị thành phần độ sáng.  $V$  có giá trị nằm trong đoạn  $[0,1]$ . Với 1 giá trị  $H$  xác định, ta có 1 màu xác định. Với giá trị  $S$ , ta có được độ đậm nhạt cho màu đó. Nhưng ta sẽ không thể cảm nhận đúng màu đó với độ đậm nhạt đó, nếu độ sáng không chuẩn. Khi  $V=1$  ta mới cảm nhận được đúng bản chất vốn có của màu. Khi  $V$  tăng tương ứng với độ sáng tăng dần.  $V=0$ : không có ánh sáng, tất cả chỉ là 1 màu đen với mọi  $H$  và  $S$ . Như ở trên, ta đã nói, khi  $S=0$  thì màu kết quả không phụ thuộc vào  $H$ , nhưng chưa thể nói ngay ta có màu trắng vì còn phụ thuộc vào  $V$ . Nếu  $V=0$  ta có màu đen,  $V=1$  ta có màu trắng, còn nếu  $0 < V < 1$  thì ta có màu xám.

Trong xử lý ảnh ta thường có nhu cầu chuyển đổi từ không gian màu RGB sang không gian màu HSV và ngược lại, ta có công thức như sau:

$$H = \begin{cases} \text{undefined}, & \text{if } \max = \min \\ 60^\circ \times \frac{g-b}{\max - \min} + 0^\circ, & \text{if } \max = r \text{ and } g \geq b \\ 60^\circ \times \frac{g-b}{\max - \min} + 360^\circ, & \text{if } \max = r \text{ and } g < b \\ 60^\circ \times \frac{b-r}{\max - \min} + 120^\circ, & \text{if } \max = g \\ 60^\circ \times \frac{r-g}{\max - \min} + 240^\circ, & \text{if } \max = b \end{cases} \quad [1.2]$$

$$S = \begin{cases} 0, & \text{if } \max = 0 \\ \frac{\max - \min}{\max} = 1 - \frac{\min}{\max}, & \text{otherwise} \end{cases}$$

$V = \max$

Trong đó:

$$\max = \max(r, g, b); \min = \min(r, g, b).$$

Các thành phần màu r, g, b được chuẩn hóa trong đoạn [0,1].

### 1.1.3 Phân đoạn ảnh

Phân đoạn ảnh là một quá trình chia một bức ảnh số thành nhiều phần khác nhau (tập hợp các điểm ảnh, hay có thể gọi là superpixels). Mục tiêu của phân đoạn ảnh là để đơn giản hóa và hoặc thay đổi biểu diễn của một tấm ảnh vào điều gì đó có ý nghĩa hơn và dễ dàng để phân tích. Phân đoạn ảnh thường được sử dụng để xác định vị trí các đối tượng, đường biên (đường thẳng, cong.vv), hay nói cách khác phân đoạn ảnh là một quá trình gán nhãn (assigning a label) cho mỗi điểm ảnh trong một bức ảnh, các điểm ảnh trong cùng một nhãn sẽ có những đặc tính giống nhau về màu sắc, cường độ hoặc kết cấu của ảnh.

Kết quả của việc phân đoạn ảnh là tập hợp các phân đoạn (segments) bao gồm có thể là toàn bộ bức ảnh hoặc tập hợp các đường biên chiết xuất từ hình ảnh. Các điểm ảnh trong cùng một vùng có đặc tính tương tự nhau về màu sắc, cường độ hoặc kết cấu. Các vùng lân cận thì khác nhau đáng kể về các đặc trưng trên.

## 1.2 Phép toán hình thái trong xử lý ảnh

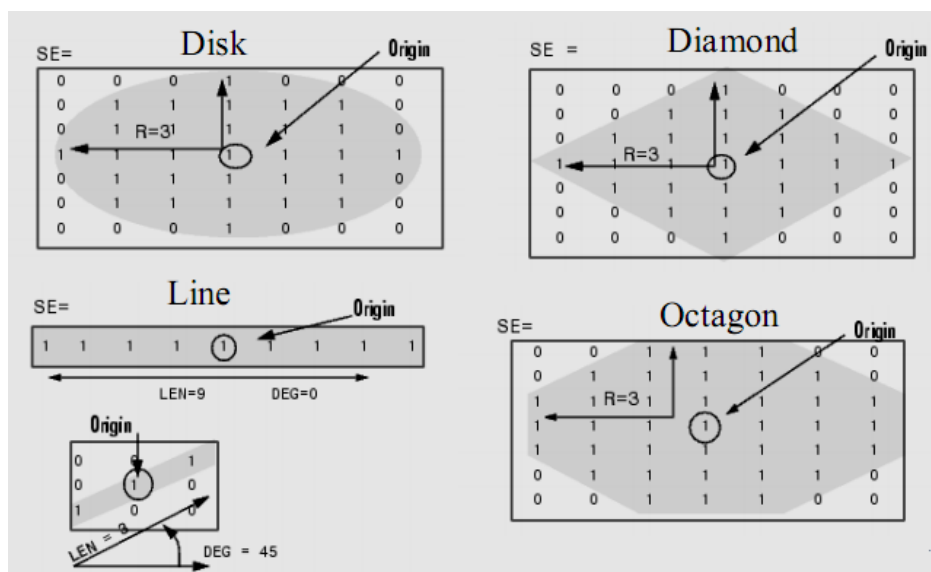
Hình thái học là lý thuyết và kỹ thuật để phân tích và xử lý cấu trúc, hình học dựa trên lý thuyết tập hợp, lý thuyết lưới, cấu trúc liên kết và chức năng ngẫu nhiên. Phép toán hình thái học được ứng dụng nhiều trong xử lý ảnh, đặc biệt là trong giai đoạn tiền xử lý và phân tích ảnh [2,7].

### 1.2.1 Phần tử cấu trúc

*Đối với ảnh nhị phân*, phần tử cấu trúc là một ảnh có kích thước nhỏ gồm có hai giá trị 0 và 1, các giá trị bằng 0 được bỏ qua trong quá trình tính toán, gọi  $B(i, j)$  là phần tử cấu trúc của ảnh nhị phân và được thể hiện như sau:

$$B(i, j) \in \{0; 1\}$$

Phần tử cấu trúc được sử dụng trên ảnh nhị phân thường có dạng đường theo chiều ngang, chiều dọc, đường elip hoặc hình vuông...



**Hình 1.6. Một số phần tử cấu trúc của ảnh nhị phân**

*Đối với ảnh xám*, phần tử cấu trúc là không phẳng, tức là các phần tử cấu trúc sử dụng các giá trị 0 và 1 để xác định phạm vi của phần tử cấu trúc trong mặt phẳng x và mặt phẳng y và thêm giá trị độ cao để xác định chiều thứ ba. Cấu trúc phần tử không phẳng gồm có hai phần:

- Phần thứ nhất : Một mảng hai chiều gồm có các giá trị 0 và 1, trong đó giá trị bằng 1 xác định hàng xóm của phần tử cấu trúc.
- Phần thứ hai: Một mảng hai chiều có kích thước bằng với kích thước của mảng hai chiều ở phần thứ nhất nhưng chứa các giá trị thực của phần tử cấu trúc.

### 1.2.2 Phép giãn nở ( Dilation )

Gọi A là ảnh gốc, B là một phần tử cấu trúc. **Phép giãn nhị phân** của ảnh A với phần tử cấu trúc B được kí hiệu  $A \oplus B$  có thể biểu diễn dưới dạng phép toán tổ hợp như sau:

$$A \oplus B = \{c \mid c = a + b, a \in A, b \in B\} \quad [1.3]$$

**Phép toán giãn nở của ảnh xám** I với phần tử cấu trúc không phẳng H tại vị trí (x,y) của ảnh I được xác định như sau :

$$(I \oplus H)(x,y) = \max(I(x+i, y+i) + H(i, j) \mid (i, j) \in D_H) \quad [1.4]$$

Trong đó,  $D_H$  là không gian ảnh của phần tử cấu trúc không phẳng H.

### 1.2.3 Phép co ( Erosion )

**Phép co ảnh nhị phân** của tập hợp A bởi phần tử cấu trúc B được kí hiệu  $A \ominus B$  và được biểu diễn dưới dạng phép toán tổ hợp như sau:

$$A \ominus B = \{c \mid (B)_c \subseteq A\} \quad [1.5]$$

**Phép co ảnh của ảnh xám** I với phần tử cấu trúc không phẳng H tại vị trí ( x, y ) của ảnh I được xác định như sau:

$$(I \ominus H)(x, y) = \min(I(x+i, y+j) - H(i, j) \mid (i, j) \in D_H) \quad [1.6]$$

Trong đó,  $D_H$  là không gian ảnh của phần tử cấu trúc không phẳng H.

### 1.2.4 Phép mở ( Open )

Gọi A là ảnh gốc và B là phần tử cấu trúc, phép mở ảnh được xác định bởi công thức:

$$\text{Open}(A) = (A \ominus B) \oplus B \quad [1.7]$$

### 1.2.5 Phép đóng ( Close )

Gọi A là ảnh gốc và B là phần tử cấu trúc, phép đóng ảnh được xác định bởi công thức:

$$\text{Close}(A) = (A \oplus B) \ominus B \quad [1.8]$$

### 1.2.6 Một số ứng dụng của phép toán hình thái

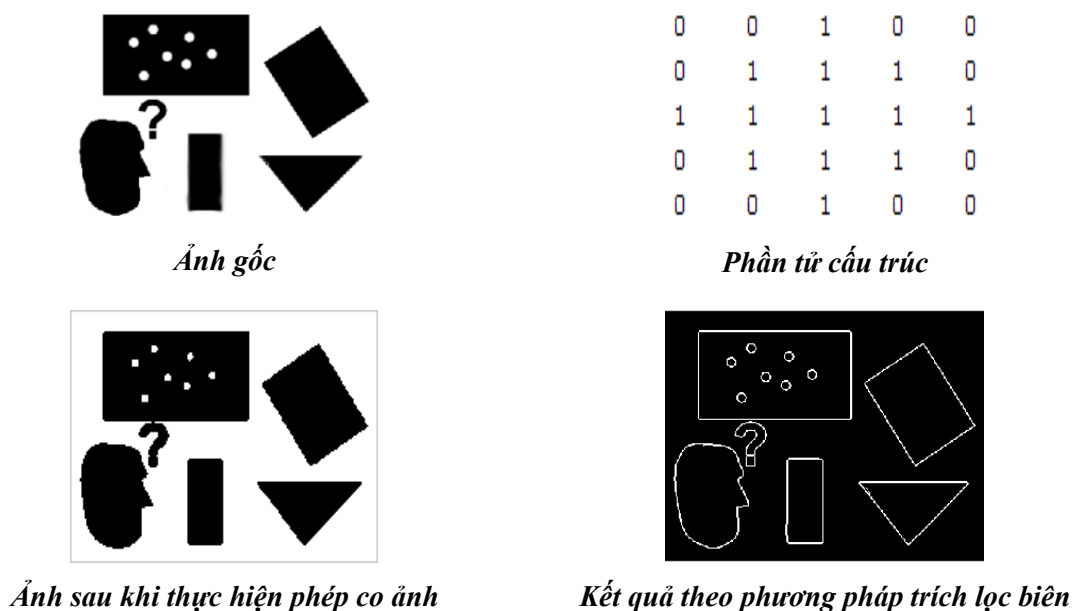
#### 1.2.6.1 Trích lọc biên ảnh ( Boundary extractoin )

Để trích lọc biên của ảnh nhị phân A, ta thực hiện phép co ảnh với phần tử cấu trúc B, sau đó thực hiện khử nền của ảnh A bằng cách lấy ảnh gốc A trừ cho ảnh đã thực hiện co ở bước 1.

Như vậy ta có thể trích lọc biên của ảnh A, ký hiệu là  $A_b$  với một phần tử cấu trúc B bằng công thức như sau:

$$A_b = A - (A \ominus B) \quad [1.9]$$

Ví dụ về trích lọc biên đối tượng trong ảnh bằng morphology :



**Hình 1.7: Ví dụ về trích lọc biên đối tượng trong ảnh**

### 1.2.6.2 Tô đầy vùng ( Region fill )

Ảnh nhị phân thường là kết quả của các phép thực hiện phân ngưỡng hoặc phân đoạn ảnh xám hoặc ảnh màu đầu vào. Những phép biến đổi này rất hiếm khi “hoàn hảo” do những nhân tố bên ngoài mà trong quá trình lấy mẫu ảnh chúng ta không kiểm soát được như cường độ sáng hay độ chói xuất hiện trong ảnh và nó có thể để lại những “lỗ hổng” sau khi thực hiện lấy ngưỡng hoặc phân đoạn ảnh. Sử dụng các phép xử lý hình thái học để lấp đầy các lỗ hổng thực sự rất hiệu quả.

Cho một ảnh nhị phân  $A$  gồm có: các điểm ảnh là biên của đối tượng được gán nhãn bằng 1 và các điểm ảnh không phải là biên được gán nhãn bằng 0. Đặt  $B$  là cấu trúc phần tử và  $x_0$  là một điểm ảnh bất kì nằm trong lỗ hổng được bao bọc bởi biên của đối tượng (điểm xuất phát).

Việc làm đầy đối tượng được thực hiện bằng cách lặp đi lặp lại biểu thức sau đây :

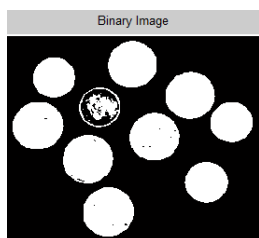
$$x_0 = 1;$$

Do

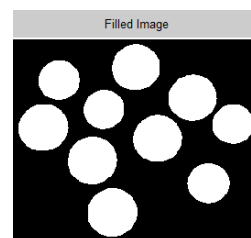
$$x_k = (x_{k-1} \oplus C) \cap B, \text{ với } k = 1, 2, 3, \dots$$

Until  $x_k = x_{k-1}$

Kết quả vùng đối tượng được lấp đầy cuối cùng chúng ta có được là  $H = A \cup x_k$ .



Ảnh nhị phân với các lỗ hổng



Ảnh sau khi được lấp đầy

**Hình 1.8: Ví dụ về tô đầy vùng đối tượng trong ảnh**

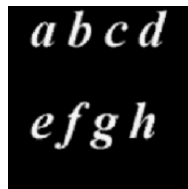
### 1.2.6.3 Làm mỏng đối tượng trong ảnh ( Thinning )

Để làm mỏng đối tượng trong ảnh A với phần tử cấu trúc B được xác định như sau:

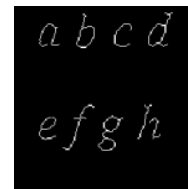
$$\text{Thin}(A, B) = A \cap \overline{(A \otimes B)} = A \cap \overline{((A \ominus B1) \cap (A \ominus B2))} \quad [1.10]$$

Trong đó:

- B = (B1, B2)
- B2 chính là phần tử B1 được thay đổi vị trí các giá trị 1( số vị trí các giá trị bằng 1 là như nhau ).



*Ảnh gốc*



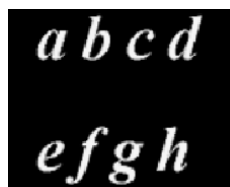
*Ảnh đã làm mỏng*

**Hình 1.9: Ví dụ về làm mỏng đối tượng trong ảnh**

### 1.2.6.4 Làm dày đối tượng trong ảnh ( Thickening )

Làm dày đối tượng tương tự như phép giãn ảnh, nhưng nó không sát nhập các đối tượng không kết nối với nhau và nó được sử dụng để làm to các đối tượng bị lốm và có thể biểu diễn qua công thức sau:

$$\text{Thicken}(A, B) = A \cup (A \otimes B) \quad [1.11]$$



*Ảnh gốc*



*Ảnh đã làm dày*

**Hình 1.10: Ví dụ về làm dày đối tượng trong ảnh**

### 1.2.6.5 Tìm xương đối tượng trong ảnh (Skeletons)

Thuật toán tìm xương của đối tượng tương tự như phép làm mỏng đối tượng nhưng nó cho chúng ta biết nhiều thông tin của một đối tượng, nó nhấn

manh một số đặc tính của hình ảnh: độ cong, đường viền tương ứng với tính chất hình học của bộ xương.

Nếu A là ảnh nhị phân và B là phần tử cấu trúc (thường có kích thước 3x3). Phép tìm xương của đối tượng trong ảnh A, ký hiệu là S(A) được xác định như sau:

$$S(A) = \bigcup_{k=0}^k S_k(A) \quad [1.12]$$

trong đó :

$$S_k(A) = \bigcup_{k=0}^k \{(A \ominus kB) - [(A \ominus kB) * B]\}$$

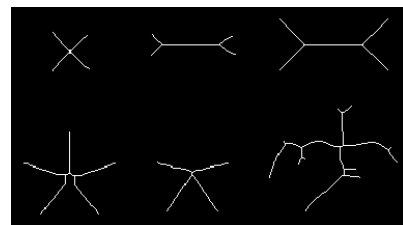
Với k là số lần áp dụng phép làm mỏng đối tượng A và K lần làm mỏng cuối cùng trước khi A suy biến thành tập rỗng.

$$K = \max\{k | (A \ominus kB) \neq \emptyset\} \quad [1.13]$$

Hình ảnh của đối tượng sau phép tìm xương



*Ảnh gốc*



*Ảnh sau phép tìm xương*

**Hình 1.11: Ví dụ về tìm xương đối tượng trong ảnh**

### 1.2.6.6 Cắt tỉa đối tượng trong ảnh ( pruning )

Xương của một mẫu đối tượng sau khi thực hiện làm mỏng thường xuất hiện những nhánh nhiều ngón trong ảnh, vì vậy xương ảnh cần phải được “làm sạch” trước khi đưa vào khâu xử lý tiếp theo trong mô hình xử lý ảnh tổng quát. Quá trình làm sạch này gọi là cắt tỉa ảnh. Các bước cắt tỉa ảnh được thực hiện qua các bước sau:



- B1: Thực hiện làm mỏng đối tượng:

$$X_1 = A \otimes \{B\}$$

- B2 : Khôi phục lại ảnh gốc sau khi đã loại bỏ ảnh cuối:

$$X_2 = \bigcup_{k=1}^8 (X_1 \circledast B^k)$$

- B3: Thực hiện giãn các điểm cuối:

$$X_3 = (X_2 \oplus H) \cap A$$

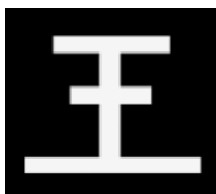
- B4 : Kết quả của cắt tỉa ảnh A được thực hiện thông qua phép hợp của  $X_1$  với  $X_3$ :

$$X_4 = X_1 \cup X_3$$

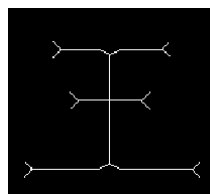
Trong đó,  $\{B\} = \{B_1, B_2, \dots, B_8\}$  là các phần tử cấu trúc có hình dạng là ma trận 3x3 gồm 0,1,x với 0,1,x có vị trí thay đổi và giá trị x tùy chọn. Với:

$$B^k = \begin{matrix} x & 0 & 0 \\ 1 & 1 & 0 \\ x & 0 & 0 \end{matrix}$$

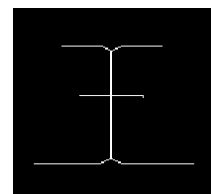
Và H là phần tử cấu trúc có kích thước 3x3 với giá trị bằng 1.



*Ảnh gốc*



*Ảnh sau khi tìm xương*



*Ảnh sau khi tìm xương rồi cắt tỉa*

**Hình 1.12: Ví dụ về tìm xương đối tượng trong ảnh**

**Kết luận:**

Trong lĩnh vực xử lý và phân tích ảnh, một trong những công việc quan trọng là trích lọc được những đặc trưng của đối tượng, mô tả hình dáng và nhận dạng mẫu. Một trong những nhiệm vụ thường đề cập đến khái niệm hình học của đối tượng, ví dụ như kích thước, hình dáng và hướng của đối tượng

trong ảnh. Xử lý hình thái học được hình thành từ lý thuyết tập hợp, hình học và hình học topo, ... để phân tích các cấu trúc hình học trong ảnh.

Một trong những vấn đề quan trọng trong các thuật toán xử lý hình thái học là tìm và sử dụng cấu trúc phần tử phù hợp để có được kết quả tốt nhất. Hầu hết các thuật toán xử lý hình thái học đều dựa trên những thuật toán cơ bản như phép co ảnh, giãn ảnh, đóng ảnh và mở ảnh để ứng dụng thành các phép toán cắt tỉa, lọc biên tìm xương sao cho phù hợp.

### **1.3 Phát hiện khuôn mặt trong ảnh**

#### **1.3.1 Giới thiệu về phát hiện khuôn mặt**

Các nghiên cứu về phát hiện khuôn mặt tuy mới chỉ xuất hiện cách đây không lâu, chỉ khoảng vài thập niên, nhưng đã có rất nhiều các nghiên cứu về nó. Các nghiên cứu đi từ bài toán đơn giản, từ phát hiện một khuôn mặt trong một bức ảnh đơn giản đến những bài toán phức tạp có nhiều khuôn mặt với nhiều tư thế thay đổi. Không những vậy mà còn mở rộng cả phạm vi môi trường xung quanh từ khá đơn giản (trong phòng thí nghiệm) cho đến môi trường xung quanh rất phức tạp (như trong tự nhiên) nhằm đáp ứng những nhu cầu của thực tế khác nhau.

Xác định khuôn mặt người ( Face Detection ) là một kỹ thuật máy tính để xác định các vị trí là khuôn mặt người trong một ảnh bất kỳ. Kỹ thuật này xác định vùng của khuôn mặt và đánh dấu, bỏ qua các thứ khác, như cơ thể, cây cối, đồ đạc...

#### **1.3.2 Các ứng dụng của phát hiện khuôn mặt**

Ứng dụng của phát hiện khuôn mặt (Face detection) là bước đầu trong bài toán nhận diện (Face recognition), trước khi đưa vào nhận diện một người trong ảnh, cần đưa ra được đâu là khuôn mặt người và có bao nhiêu khuôn mặt trong ảnh đó.

Tìm kiếm, phân loại, đo đếm để tổ chức dữ liệu có liên quan đến con người thông qua khuôn mặt trên nhiều cơ sở dữ liệu lớn.

Ứng dụng trong các hệ thống quan sát, theo dõi và bảo vệ. Các thiết bị an ninh sẽ xác định được đâu là con người và tập trung kiểm tra hành vi của người đó, ví dụ các khu vực cấm con người...

Các ứng dụng liên quan đến tương tác giữa con người và máy móc cần đến nhận biết con người thông qua khuôn mặt.

Ứng dụng trong xác thực danh tính, mã hóa và bảo mật trên các thiết bị như di động, máy chấm công, camera...

### **1.3.3 Một số hướng tiếp cận trong bài toán phát hiện khuôn mặt**

Hướng tiếp cận dựa trên mô hình (top-down model-based approach): trong hướng tiếp cận này, người ta sử dụng các mô hình mặt người khác nhau có tỉ lệ khác nhau từ thô nhất đến tốt nhất. Đầu tiên, ảnh sẽ được quét bởi mô hình có tỉ lệ thô nhất, sau đó ảnh được quét với mô hình có tỉ lệ tốt hơn và cuối cùng được quét với mô hình có tỉ lệ tốt nhất. Ứng với mỗi tỉ lệ chỉ có 1 mô hình khuôn mặt mà 1 mô hình khuôn mặt ứng với 1 góc nhìn khuôn mặt cụ thể.

Hướng tiếp cận dựa trên đặc trưng (bottom-up feature-based approach): trong hướng tiếp cận này, người ta sẽ tiến hành tìm kiếm trong ảnh các đặc trưng của khuôn mặt (như mắt, mũi, miệng,...) sau đó gom nhóm chúng lại với nhau (dựa trên mối quan hệ hình học giữa chúng) tạo thành các ứng cử viên cho khuôn mặt.

Hướng tiếp cận dựa trên bề mặt (texture-based approach): trong hướng tiếp cận này, các khuôn mặt được phát hiện dựa trên sự phân bố không gian các mức xám của các điểm trong ma trận ảnh con từ các ảnh lớn.

Hướng tiếp cận dựa trên mạng nơron (neural network approach): trong giải pháp này, người ta tiến hành huấn luyện trên 1 tập ảnh khuôn mặt và không phải khuôn mặt. Sau khi huấn luyện xong, sẽ được dùng vào xác thực khuôn mặt. Ảnh sẽ được quét bằng 1 cửa sổ, tại mỗi vị trí cửa sổ, nội dung

ảnh được lấy ra (có thể phải thay đổi tỉ lệ (co, giãn) rồi xác thực khuôn mặt qua bộ lọc. Nhìn chung, hướng tiếp cận này cho kết quả tốt, tuy nhiên tốn thời gian huấn luyện và sưu tầm mẫu.

Hướng tiếp cận dựa trên màu sắc (color-based approach): trong giải pháp này, đầu tiên người ta xác định các điểm trong ảnh có màu giống màu da mặt. Sau đó, người ta khoanh vùng các điểm đó lại, các vùng này có thể là khuôn mặt hoặc không. Để xác định có phải là mặt hay không, có thể dựa vào tỉ lệ kích thước của vùng có tương tự tỉ lệ khuôn mặt, hoặc dựa vào tỉ lệ số điểm màu da trong hình chữ nhật bao vùng đó,...

Hướng tiếp cận dựa trên chuyển động (motion-based approach): Từ các đối tượng chuyển động trong ảnh, dựa trên một số tiêu chí, người ta sẽ xác định được vùng mặt. Ví dụ như một cái lắc đầu, hay nháy mắt sẽ là cơ sở để phát hiện khuôn mặt.

Trong thực tế, ngoài những phương pháp đơn thuần đi theo một hướng tiếp cận, thì cũng có những phương pháp kết hợp nhiều hướng tiếp cận để cho ra kết quả chính xác hơn, tuy nhiên cũng phải tiêu tốn về thời gian.

## CHƯƠNG 2: MỘT SỐ PHƯƠNG PHÁP PHÁT HIỆN KHUÔN MẶT

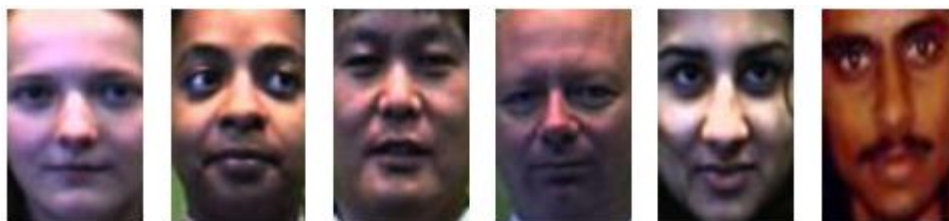
### 2.1 Phương pháp dựa trên màu da và thông tin biên.

Phương pháp này được giới thiệu bởi K. Sandeep và cộng sự [9] bằng cách sử dụng một biểu đồ màu trong mô hình màu HSV để xác định vùng có thể là vùng da, sau đó dựa vào các tỉ lệ như tỉ lệ điểm màu da, tỉ lệ chiều cao/rộng... để quyết định có phải vùng da mặt hay không. Nội dung chính của phương pháp được trình bày như sau:

#### 2.1.1 Xác định các vùng da trong ảnh

##### 2.1.1.1 Phân đoạn dựa vào màu da

Để xác định 1 điểm có phải là màu da nhóm tác giả đã tính toán một biểu đồ màu trong không gian màu HSV. Các tác giả đã thực hiện quá trình học màu da dựa trên các ảnh mẫu về khuôn mặt kiểu như sau:



**Hình 2.1.** Ví dụ về ảnh mẫu trong phương pháp của K. Sandeep

Với mỗi điểm trong ảnh, lấy về các giá trị H (hue) và S (saturation) trong mô hình màu HSV. Do màu da của con người trong ảnh còn phụ thuộc vào yếu tố ánh sáng, ảnh tối thì màu da sẽ hơi sẫm, ảnh sáng thì màu da sẽ trắng hơn, với cùng 1 người nhưng trong điều kiện ánh sáng khác nhau màu da cũng sẽ bị thay đổi. Vậy nên cần phải loại bỏ tác động của yếu tố ánh sáng này để lấy về màu da đúng của mỗi người và để hạn chế bớt sự đa dạng trong màu da. Thành phần V trong mô hình màu HSV chính là thành phần biểu thị độ sáng trong ảnh vì thế nó bị loại bỏ. Sau quá trình tính toán, Biểu đồ màu thu được sẽ được chuẩn hóa để có được các giá trị trong khoảng (0,1).

Để xác định 1 điểm ảnh có phải là màu da hay không, ban đầu phải lấy về thành phần H,S của điểm ảnh đó. Sau đó, dựa vào biểu đồ màu thu được xác định xem cột ứng với giá trị H,S này có giá trị bằng bao nhiêu, nếu nhỏ

hơn 1 ngưỡng cho trước thì đó không phải là màu da, ngược lại thì là màu da. Các tác giả thấy rằng nếu chỉ dùng thông tin này thì kết quả không tốt bằng việc dựa thêm thông tin về biên nữa để xác định 1 điểm có phải là màu da hay không. Do vậy, để xác định 1 điểm trong ảnh có phải là màu da hay không được dựa trên 2 tiêu chí:

- Biểu đồ màu (H,S) phải có giá trị lớn hơn 1 giá trị ngưỡng - skinthreshold.
- Điểm đó không nằm trên biên. Để xác định nó có nằm trên biên không, cần sử dụng ảnh biên. Trong ảnh biên, 1 điểm được coi là nằm trên biên nếu giá trị của nó lớn hơn 1 giá trị ngưỡng – edgethreshold.

Bằng thực nghiệm, các tác giả chọn ra các giá trị cụ thể cho các giá trị ngưỡng kia như sau:

$$+ \text{Skinthreshold} = 0,1.$$

$$+ \text{Edgethreshold} = 125.$$

Sau khi được phân vùng dựa vào màu da, ảnh thu được có thể được biểu diễn dưới dạng ảnh nhị phân, các điểm màu da có giá trị 1, các điểm không phải có giá trị 0.

#### 2.1.1.2 Xác định vị trí các vùng da trong ảnh:

Để xác định vùng trong ảnh cần phải dựa trên 8 láng giềng của điểm ảnh. Hai điểm được coi là thuộc cùng 1 vùng nếu điểm này là 1 trong 8 láng giềng của điểm kia. Để xác định kích thước của vùng, thì các tác giả làm như sau:

- Đầu tiên là xác định điểm trung tâm của vùng. Nó được xác định bằng cách tính trung bình cộng tọa độ của các điểm trong vùng.
- Sau khi có được tọa độ của điểm này, chiều cao của vùng được xác định như sau:

- + Lấy y của các điểm trong vùng trừ đi y của điểm trung tâm.
  - + Lấy trung bình cộng của các giá trị âm, và lấy trung bình cộng của các giá trị dương.
  - + Cộng trị tuyệt đối 2 giá trị trung bình đó lại, được bao nhiêu nhân 2 ra chiều cao.
- Với chiều rộng của vùng làm tương tự.

Với cách làm này, khung hình chữ nhật bao quanh vùng sẽ không bao hết các điểm nằm trong vùng, và quá trình duyệt đòi hỏi phải lưu lại tất cả tọa độ các điểm trong vùng để phục vụ cho tính chiều rộng, chiều cao sau này. Do vậy, các tác giả đã đề xuất cách tính có thể giải quyết được vấn đề trên là lấy x max, x min, y max, y min của tất cả các điểm trong vùng. Tuy nhiên có thể làm vậy lại ảnh hưởng đến chuẩn đoán vùng mặt sau này.

### 2.1.2 Xác thực khuôn mặt

Vì vùng da tìm thấy có thể là da tay, da chân, ... nên để xác định vùng đó có phải là khuôn mặt hay không, dựa trên các tiêu chí:

- Cao/rộng hoặc rộng/cao phải nằm trong khoảng: tỉ lệ vàng  $\pm \delta$ . Rộng/ cao là cho trường hợp mặt nằm ngang. Delta được đưa vào vì tỉ lệ không thể nào đúng bằng tỉ lệ vàng được; hơn nữa, những khuôn mặt khác nhau cũng lệch nhau đôi chút, và cuối cùng là vì những trường hợp mặt nghiêng. Tỉ lệ vàng =  $\frac{1+\sqrt{5}}{2}$ .
- Tỉ lệ số điểm màu da trong khung hình chữ nhật của vùng đó phải lớn hơn 1 giá trị ngưỡng – PercentageThreshold.
- Bằng thực nghiệm, nhóm tác giả đã đưa ra giá trị cho các tham số là: Delta = 0,65 và PercentageThreshold = 55.

## 2.2 Phương pháp dựa trên nhiều chứng cứ

Trong phương pháp dựa trên màu da và thông tin biên, các tác giả chỉ dựa trên màu da để xác định vùng mặt và dựa vào tỉ lệ kích thước các chiều để xác định xem đó có phải là mặt hay không. Với phương pháp như vậy thì có thể phát hiện được các khuôn mặt với kích thước khác nhau, không cần giả định trước kích thước khuôn mặt và cho tốc độ phát hiện khá nhanh. Tuy nhiên, việc xác định nhầm là rất cao, ví dụ như khi xác định được 1 vùng màu da, để xác định xem nó có phải là khuôn mặt hay không thì dựa vào tỉ lệ kích thước hai chiều của vùng, và tỉ lệ số điểm màu da trong vùng. Như vậy chỉ cần 1 vùng có màu giống màu da và có kích thước thích hợp là sẽ bị coi là khuôn mặt. Vì vậy, phương pháp này hay bắt nhầm vùng tay.

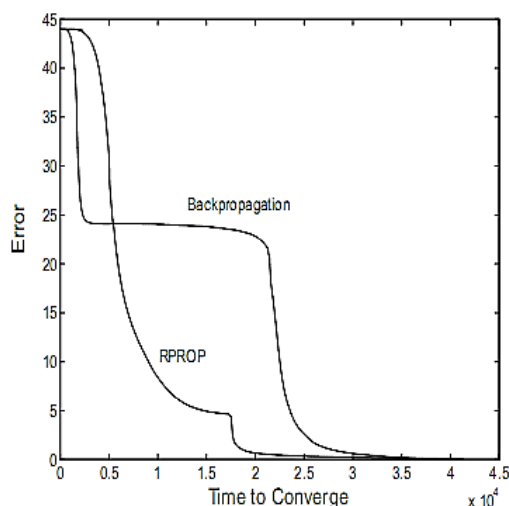
Trong phương pháp được giới thiệu bởi Manoj Seshadrinathan và Jezekiel Ben – Arie[5] các tác giả sẽ đưa ra quyết định đó có phải là mặt hay không dựa trên nhiều chứng cứ. Trước tiên sử dụng công thức màu da trong hệ màu YCbCr xác định vị trí có thể là vùng da, sau đó dựa hình dạng và kết quả lọc Gabor để xác thực kết quả.

### 2.2.1 Xác định các vùng da trong ảnh

Để xác định 1 điểm có phải là da mặt hay không, các tác giả dựa vào 2 thành phần Cr,Cb của hệ màu YCbCr và không sử dụng thành phần Y của hệ màu này vì nó là thành phần biểu thị độ sáng, không phải là cái vốn có của màu sắc da mặt, tức là nó thay đổi theo các điều kiện ánh sáng khác nhau nên việc loại bỏ nó sẽ giúp xác định màu da mặt một cách đơn giản hơn.

Để dựa vào Cr, Cb phát hiện 1 điểm có màu da mặt hay không, các tác giả thực hiện quá trình học màu da mặt bằng phương pháp mạng Noron có tên là: RPROP (Resilient Propagation). Đây là 1 thuật toán học mạng nơron cải tiến của thuật toán Backpropagation.





**Hình 2.2: Mô hình biểu diễn của RPROP**

## 2.2.2 Xác thực khuôn mặt

### 2.2.2.1 Xác thực dựa vào hình dạng

Sau khi phân đoạn ảnh trên màu da, các vùng có thể là da mặt được xác định. Tuy nhiên, không phải vùng nào cũng là khuôn mặt. Để loại bỏ những vùng không phải là khuôn mặt, các tác giả dựa trên các tiêu chuẩn về hình dạng.

- Tiêu chuẩn đầu tiên được áp dụng là tiêu chuẩn tỉ lệ mặt 1/3. Tức là chiều rộng trên chiều cao khuôn mặt phải  $\geq 1/3$ .
- Tiêu chuẩn thứ 2 được sử dụng là tiêu chuẩn vùng elip. Mỗi khuôn mặt có hình gần giống với hình elip nên sẽ coi các vùng phát hiện được đó như là các hình elip. Cần xác định các độ dài các trục của nó: trục chính và trục phụ. Để xác định điều này, các tác giả cần sử dụng một số phép biến đổi. Sau khi, xác định được độ dài các trục, các tác giả đưa ra công thức đánh giá cho vùng đó như sau:

$$S_{ei(x,y)} = \frac{4 * N_i}{\pi * l_1 * l_2} \quad [2.1]$$

Trong đó:  $l_1, l_2$  là chiều dài các trục,  $N_i$  là số điểm trong vùng  $i$  đang xem xét.

Ngoài ra, tác giả còn xác định cả tỉ lệ tóc trong vùng elip này. Việc xác định màu tóc cũng được thực hiện bằng học mạng nơron RPROP.

### 2.2.2.2 Xác thực khuôn mặt sử dụng bộ lọc Gabor

Các tác giả còn sử dụng bộ lọc Gabor để nhận ra các vùng mặt. Các sóng 2 chiều Gabor được sử dụng để nhận ra khuôn mặt. Nó cung cấp các tính chất đáp ứng tần số của ảnh. Biểu thức của hàm Gabor 2 chiều như sau:

$$\phi(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{1}{2}\left[\left(\frac{x-x_0}{\sigma_x}\right)^2 + \left(\frac{y-y_0}{\sigma_y}\right)^2\right]} e^{-j[u_0(x-x_0) + v_0(y-y_0)]} \quad [2.2]$$

Trong đó:  $(x, y)$  là các điểm trong ảnh,  $\sigma_x, \sigma_y$  là kí hiệu của các tỉ lệ Gaussian dọc theo các trục tương ứng,  $(x_0, y_0)$  là tọa độ tâm của hàm trong miền không gian,  $u_0, v_0$  là các tần số góc.

Phân tách các thành phần thực và ảo trong hàm trên theo công thức:  $e^{j\varphi} = \cos\varphi + j\sin\varphi$ , được các hàm dạng sin và cosin với 2 tham số độc lập là: tỉ lệ  $\sigma$  và góc quay  $\theta$ :

$$\begin{aligned} \phi_1(x, y) &= \sin((x \cos \theta + y \sin \theta) \omega) e^{-\frac{1}{2}\left(\frac{x^2 + y^2}{\sigma^2}\right)} \\ \phi_2(x, y) &= \cos((x \cos \theta + y \sin \theta) \omega) e^{-\frac{1}{2}\left(\frac{x^2 + y^2}{\sigma^2}\right)} \end{aligned} \quad [2.3]$$

Trong thử nghiệm, các tác giả đã sử dụng 64 hàm Gabor với 4 hướng và 8 tỉ lệ (32 hàm cos và 32 hàm sin). Các tác giả đã sử dụng tới 5 mẫu học để nhận về các dấu hiệu của hàm Gabor cho các tư thế đầu khác nhau:



**Hình 2.3: Ví dụ về các tư thế đầu khác nhau trong Gabor**

Áp dụng các bộ lọc Gabor lên ảnh vào, thu được 1 tập dấu hiệu. Sau đó, các tác giả sẽ tính toán sự tương đồng của tập dấu hiệu thu được đó với 5 mô hình dấu hiệu thu được trong quá trình học ở trên, lấy về giá trị lớn nhất và gán cho điểm ảnh đang tính đó. Như vậy, mỗi điểm trong ảnh vào sẽ có một điểm số  $S_g(x,y)$  thể hiện sự giống của nó với khuôn mặt mô hình.

Sau phân tích ảnh dựa trên một loạt các tham số như trên, tác giả tiến hành tổng hợp các kết quả lại bằng 1 công thức, sau đó đưa ra 1 giá trị ngưỡng để quyết định xem đó có phải là mặt hay không.

Quá trình thực hiện được cho trong sơ đồ sau:

### **2.3 Phương pháp dựa trên phân tích wavelet**

Với việc sử dụng phương pháp phát hiện khuôn mặt dựa trên nhiều chứng cứ như trên có tính chính xác cao, tuy nhiên lại chỉ trên các ảnh có khuôn mặt có kích thước nhất định hoặc phải giả định kích thước khuôn mặt từ trước.

Phương pháp của C. Garcia, G. Zikos, G. Tziritas[8] đưa ra là phương pháp phát hiện tự động và nhanh các mặt người trong ảnh màu dựa trên phân tích wavelet. Phương pháp này không giả định trước về số lượng, vị trí, hướng và kích thước các khuôn mặt trong ảnh. Nó có thể phát hiện tốt các khuôn mặt trên các nền ảnh phức tạp, cũng như trong những điều kiện ánh sáng khác nhau. Sơ lược về các bước của phương pháp như sau:

Với mỗi ảnh đưa vào, đầu tiên, các tác giả sử dụng các giá trị sắc tố da mặt để đưa ra các vùng trong ảnh có khả năng là vùng mặt, gọi là các ứng cử viên vùng mặt.

Sau đó, các công việc phân tích hình dạng và phân tách các wavelet được thực hiện trên mỗi ứng cử viên vùng mặt. Mỗi vùng ứng cử viên này sẽ được biểu diễn như là 1 tập con của các ảnh được lọc chứa các hệ số sóng. Các hệ số này mô tả bề mặt của mặt và một tập các dữ liệu thống kê đơn giản

được đưa ra trong trình tự tới hình thức cô đọng và các vector đặc trưng có ý nghĩa.

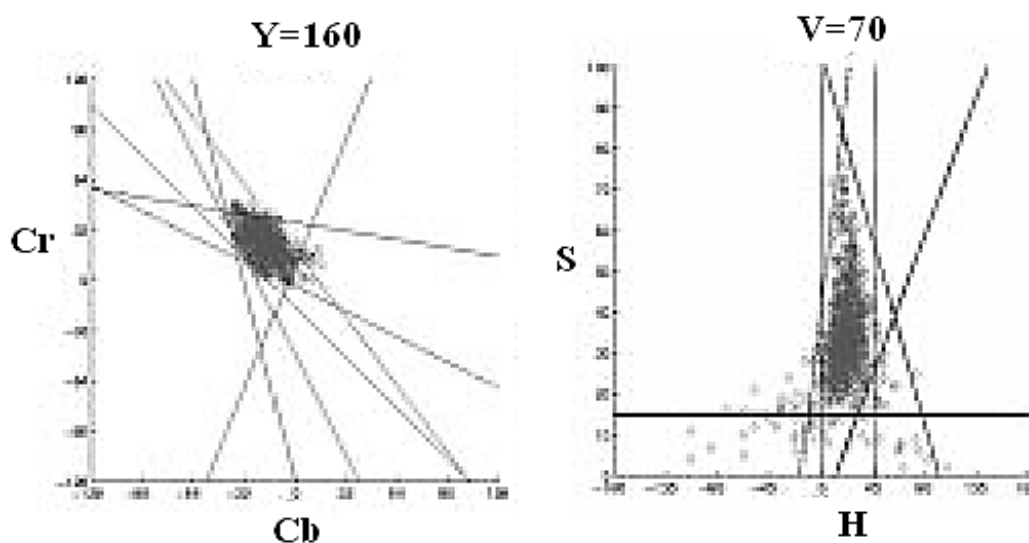
Tiếp theo, một độ đo xác suất tin cậy và hiệu quả, được suy ra từ khoảng cách nhất định, sử dụng để phân xác định xem các vùng ứng cử viên đó, vùng nào là khuôn mặt, vùng nào không, dựa trên các vector đặc trưng thu được từ trên.

### 2.3.1 Xác định các vùng da trong ảnh

#### 2.3.1.1 Phân đoạn ảnh dựa vào màu da

Để xác định các vùng này, các tác giả tiến hành nghiên cứu màu da mặt trong các không gian màu YCbCr và HSV. Khác với không gian màu RGB, biểu diễn màu sắc như là tổ hợp của 3 màu cơ bản đỏ (Red), xanh lá cây (Green), xanh da trời (Blue); các không gian màu YCbCr và HSV biểu diễn màu sắc khác với quan điểm nhìn màu sắc của con người.

Qua thực nghiệm (lấy mẫu các màu da mặt của nhiều chủng tộc người khác nhau với những điều kiện ánh sáng khác nhau), các tác giả thấy rằng, các màu biểu diễn màu da con người chỉ chiếm 1 không gian rất nhỏ trong không gian màu YCbCr hoặc HSV. Minh họa qua hình sau:



Hình 2.4: Sự phân bố màu da trong không gian màu YcbCr và HSV

Trong không gian màu HSV, không gian màu da mặt có vẻ kém cô đọng hơn so với YCbCr. Từ kết quả thực nghiệm, các tác giả đã tính ra công thức cho việc xác định một màu có phải là màu da mặt hay không, như sau:

- Trong không gian màu YCbCr:

$$\left\{ \begin{array}{l} \text{if } (Y \leq 128) \theta_1 = -2 + \frac{256 - Y}{16}; \theta_2 = 20 - \frac{256 - Y}{16}; \theta_3 = 6; \theta_4 = -8 \\ \text{if } (Y > 128) \theta_1 = 6; \theta_2 = 12; \theta_3 = 2 + \frac{Y}{32}; \theta_4 = 16 + \frac{Y}{16} \end{array} \right.$$

$$\left\{ \begin{array}{l} Cr \geq -2 + (Cb + 24); Cr \geq -(Cb + 17); Cr \geq -4(Cb + 32); \\ Cr \geq 2.5(Cb + \theta_1); Cr \geq \theta_3; Cr \geq -0.5(Cb - \theta_4) \\ Cr \leq -\frac{Cb - 220}{6}; Cr \leq -1.34(Cb - \theta_2) \end{array} \right. \quad [2.4]$$

- Trong không gian màu HSV:

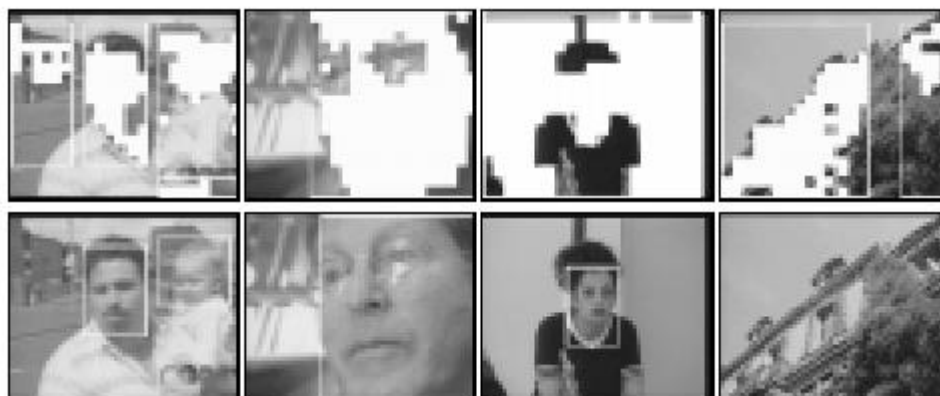
$$\left\{ \begin{array}{l} H \geq 0; S \geq 15; S \geq 0.75H + 0.3V - 30 \\ S \leq -H - 0.1V + 110; H \leq -0.4V + 75; S \leq 0.08(100 - V)H + 0.6V \end{array} \right.$$

Để xác định màu da, đầu tiên phải duyệt qua ảnh để xác định những điểm nào là màu da, điểm nào không phải màu da, dựa vào 1 trong 2 công thức trên. Chú ý ở đây, các tác giả không thao tác trên từng điểm ảnh (pixel) mà thao tác trên cả 1 khối điểm vuông kích thước 16x16 pixel gọi là macro-block. Khi đó màu được so sánh là màu trung bình của cả khối.

Sau khi xác định được các block nào là màu da, block nào không, các tác giả đưa ảnh về ảnh nhị phân với các block màu da=1 và block không phải màu da=0. Sau đó, để xác định các vùng màu da (vùng 1) và đảm bảo tốc độ thực hiện thuật toán, các tác giả đã sử dụng 1 phép chiếu đơn giản lên 2 trục ngang dọc dựa vào các băng 0, 1 (những giá trị 0 (1) liên tiếp được gọi là băng 0 (băng 1)) để xác định các hình chữ nhật bao lấy các vùng màu da.

Hình minh họa trong hình 2.6. Dòng trên là kết quả xác định vùng màu da thực hiện theo thuật toán đưa ra, còn bên dưới là kết quả cuối cùng của

phương pháp này (tức là còn phải trải qua vài khâu nữa mới đưa ra được kết quả này).



**Hình 2.5:** Kết quả thực hiện sau thuật toán phát hiện vùng da đã đưa ra

### 2.3.1.2 Xác định vị trí vùng da mặt

Ở trên đã xác định được các vùng màu da được đóng khung trong các hình chữ nhật, ở bước này sẽ phải tìm khuôn mặt trong các hình chữ nhật đó - gọi là ứng cử viên vùng mặt.

Vì không biết trước kích thước của các khuôn mặt trong ảnh nên các tác giả bắt đầu tìm những vùng ứng cử viên lớn nhất, sau đó giảm dần kích thước. Các tác giả sử dụng một cửa sổ quét, kích thước của cửa sổ đó được xác định dựa theo hai tiêu chí là: giới hạn kích thước của cửa sổ và tỉ lệ khuôn mặt.

- Về giới hạn kích thước của cửa sổ, kích thước nhỏ nhất là: 5 x 3 macro-block tương đương với 80 x 48 pixel, còn kích thước lớn nhất là cả khung hình chữ nhật đó.
- Về tỉ lệ khuôn mặt, tức là tỉ lệ giữa chiều cao và chiều rộng, các tác giả đưa ra khoảng giá trị là: [1;1,8]. Khoảng giá trị đưa ra rộng là để cho các hướng và các tư thế khác nhau của khuôn mặt.

Với mỗi vị trí cửa sổ quét, để xác định xem vùng đó có thể là ứng cử viên không, các tác giả sử dụng tiêu chí là độ thuần nhất, tức là tỉ lệ các điểm 1 ở trong ô đó phải lớn hơn 1 ngưỡng nào đó. Các tác giả chia ô đó ra thành 2

miền là: miền ngoài (outer) và miền trong (inner). Miền ngoài là đường bao của ô đó (có kích thước khoảng 15% mỗi chiều), còn miền trong là vùng tâm còn lại. Hình minh họa như sau:



**Hình 2.6: Minh họa outer và inner**

Khi đó, tiêu chuẩn về sự đồng nhất đặt ra cho mỗi vùng như sau: trong vùng outer, tỉ lệ 1 phải lớn hơn 65%. Còn trong vùng inner tỉ lệ 1 phải lớn hơn 1 giá trị ngưỡng  $p$  được xác định như sau:

$$\begin{cases} p = 1 - 0.3 \frac{(w-2*\alpha)(h-2*\alpha)}{wh} & ; \text{if } w > 2 * \alpha \\ p = 0.80 & \text{otherwise} \end{cases} \quad [2.5]$$

trong đó  $\alpha = 2$ , còn  $w, h$  tương ứng là chiều rộng, cao của vùng inner đó với đơn vị tính là macro-block chứ không phải pixel.

### 2.3.2 Xác thực khuôn mặt

Cuối cùng, cần xác định xem vùng ứng cử viên tìm được ở trên có phải là mặt hay không là mặt dựa vào phân tích wavelet. Đặc tính chính của wavelet là có khả năng cung cấp một sự phân tích ảnh nhiều độ phân giải trong hình thức các ma trận hệ số với sự phân rã ảnh trong miền không gian và tần số cùng một lúc. Trong trường hợp 2D (ảnh số), biến đổi wavelet thường được thực hiện bằng việc áp dụng các bộ lọc riêng rẽ. Cụ thể, một bộ lọc thông thấp và 1 bộ lọc dải thông được áp dụng. Kết quả của việc áp dụng bộ lọc thông thấp cho ra 1 ảnh gọi là ảnh xấp xỉ. Kết quả của việc áp dụng bộ lọc dải thông cho ra nhiều ảnh, gọi là các ảnh chi tiết.

Như vậy, kết quả của việc áp dụng các bộ lọc cho ra 1 ảnh xấp xỉ và nhiều ảnh chi tiết. Sau đó, ảnh xấp xỉ đó lại có thể phân rã tiếp thành ảnh xấp

xỉ và các ảnh chi tiết mức 2,... Trong phương pháp đưa ra của các tác giả, cả ảnh xấp xỉ và ảnh chi tiết đều có thể phân rã tiếp, kết quả thu được 1 cây phân rã. Qua nhiều thí nghiệm, các tác giả đã chọn ra các bộ lọc được sử dụng là các bộ lọc cầu liên hợp.

Các tác giả đã xây dựng 1 cơ sở dữ liệu gồm 50 vùng mặt được lấy ra thủ công (tức là dùng Paint cắt vùng mặt trong ảnh ra), với các kích thước, hướng, ánh sáng và vị trí khác nhau. Các mẫu vùng mặt này được phân làm 2 nhóm:

- Nhóm trung bình: gồm các mẫu có chiều cao  $<128$  pixel.
- Nhóm lớn: gồm các mẫu có chiều cao  $>128$  pixel.

Qua thử nghiệm, các tác giả quyết định độ sâu của cây phân rã với các ảnh cỡ lớn là 3 và với các ảnh cỡ trung bình là 2. Như vậy, một ảnh sau khi được phân rã sẽ được biểu diễn bởi một tập gồm  $n$  ma trận hệ số sóng thuộc về mức sâu nhất của quá trình phân rã. Tức là với ảnh trung bình sẽ có  $n=16$ , 1 ảnh xấp xỉ và 15 ảnh chi tiết; với ảnh lớn sẽ có  $n=64$ , 1 ảnh xấp xỉ và 63 ảnh chi tiết.

Để lấy ra được các vector đặc trưng của vùng mặt, các tác giả chia vùng mặt ra thành 4 phần bằng nhau, và phân tích bó sóng trên mỗi phần đó để lấy về thông tin bề mặt khuôn mặt liên quan đến các phần khác nhau của khuôn mặt như mắt, mũi, miệng. Trên mỗi phần đó, sau khi phân tích, các tác giả thu được các biến, kí hiệu là  $\delta_i^2$ , như vậy là có 4 biến. Với ảnh xấp xỉ sẽ lấy về 4 biến (bằng cách phân chia ảnh đó thành 4 phần), và với mỗi ảnh chi tiết sẽ chỉ lấy về 1 biến (không phân chia). Kết quả thu được 1 vector đặc trưng có  $m + 4$  biến gồm 4 biến lấy từ ảnh xấp xỉ và  $m$  biến lấy từ  $m$  ảnh chi tiết ( $m = 15$  với ảnh cỡ trung bình, và  $= 63$  với ảnh cỡ lớn). Tóm lại, với ảnh cỡ lớn ta sẽ thu được 1 vector gồm 67 biến và với ảnh cỡ trung bình ta thu được 1 vector gồm 19 biến.



Với mỗi ảnh trong cơ sở dữ liệu, các tác giả cũng tiến hành lấy ra các vector đặc trưng. Tuy nhiên, nếu lấy vector đặc trưng thu được từ vùng ảnh ứng cử viên lấy được ở pha trước mà đem so sánh với từng vector đặc trưng của các ảnh trong cơ sở dữ liệu thì rõ ràng là khả năng khớp hoàn toàn là không thể. Vì vậy, các tác giả tiến hành lấy trung bình các vector đặc trưng của các ảnh trong cơ sở dữ liệu để cho ra các vector đặc trưng mẫu. Sau đó, tính khoảng cách của vector đặc trưng của vùng ảnh ứng cử viên so với vector đặc trưng mẫu để quyết định xem vùng ứng cử viên đó có phải là mặt không. Vì ảnh có 2 loại, nên cũng có 2 vector đặc trưng mẫu, 1 cho loại trung bình, 1 cho loại lớn. Công thức xác định khoảng cách như sau:

$$D(V_k, V_l) = \frac{1}{2} \sum_{i=0}^{m+3} \ln\left(\frac{\frac{1}{2}(\alpha_{ik}^2 + \alpha_{il}^2)}{\sqrt{(\alpha_{ik}^2 \alpha_{il}^2)}}\right) + \frac{1}{2} \sum_{C \in \{Cb, Cr\}} \ln\left\{\left(\frac{\frac{1}{2}(\alpha^2 + Vc)}{\sqrt{(\alpha_{ik}^2 Vc)}}\right)\right\} \quad [2.6]$$

Nếu khoảng cách trên lớn hơn 1 giá trị ngưỡng (được xác định trước) thì vùng ứng cử viên đó không phải là mặt, nếu nhỏ hơn thì đó là khuôn mặt. Qua thực nghiệm, các tác giả tìm được giá trị ngưỡng phù hợp là 7,0.

## 2.4 Phương pháp phát hiện khuôn mặt dựa vào đặc trưng Haarlike

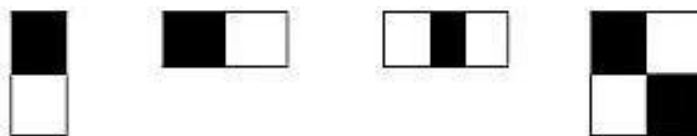
Ở các phương pháp trước, các tác giả thường thực hiện theo hai bước:

- Xác định các vùng da trong ảnh.
- Xác thực khuôn mặt.

Còn theo phương pháp này của Viola và Jones[4],[6] các tác giả sẽ quét tất cả các vị trí trong bức ảnh để xác thực khuôn mặt với các đặc trưng Haarlike.

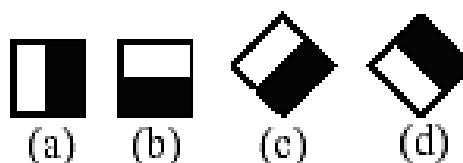
### 2.4.1 Đặc trưng Haar-like

Đặc trưng Haar – Like gồm có 4 đặc trưng cơ bản với mỗi đặc trưng là sự kết hợp của hai hay ba hình chữ nhật "trắng" và "đen" như trong hình sau:

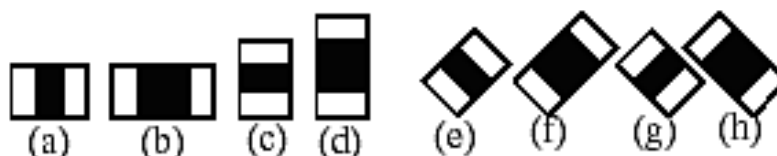


Để sử dụng các đặt trưng này vào việc xác định khuôn mặt người, 4 đặc trưng Haar-like cơ bản được mở rộng ra, và được chia làm 3 tập đặc trưng như sau:

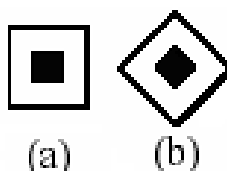
1. Đặc trưng cạnh (edge features):



2. Đặc trưng đường (line features):



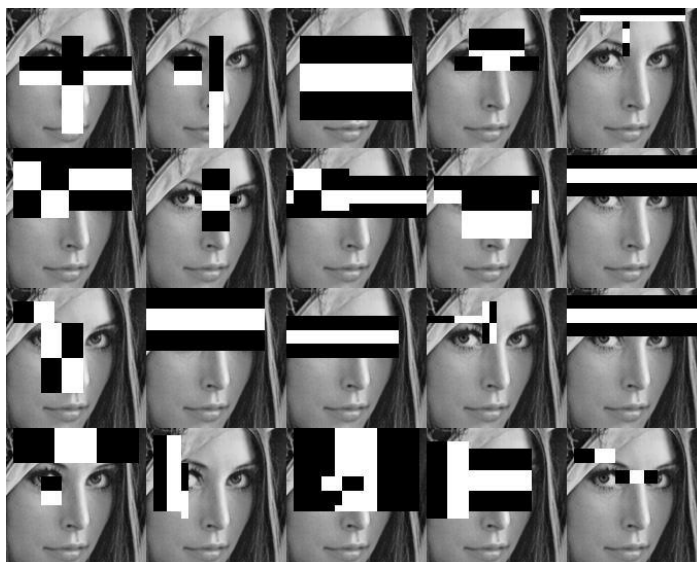
3. Đặc trưng xung quanh tâm (center-surround features):



Dùng các đặc trưng trên, có thể tính được giá trị của đặc trưng Haar-like là sự chênh lệch giữa tổng của các pixel của các vùng đen và các vùng trắng như trong công thức sau:

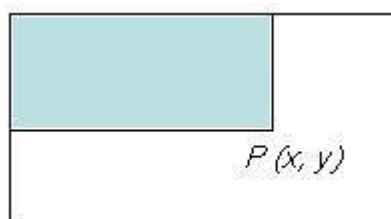
$$f(x) = \text{Tổng vùng đen (mức xám của pixel)} - \text{Tổng vùng trắng (mức xám của pixel)}$$

Sử dụng giá trị này, so sánh với các giá trị của các giá trị pixel thô, các đặc trưng Haar-like có thể tăng/giảm sự thay đổi in-class/out-of-class (bên trong hay bên ngoài lớp khuôn mặt người), do đó sẽ làm cho bộ phân lớp dễ hơn.



**Hình 2.7: Đặc trưng Haarlike trên khuôn mặt người**

Như vậy có thể thấy rằng, để tính các giá trị của đặc trưng Haar-like phải tính tổng của các vùng pixel trên ảnh. Nhưng để tính toán các giá trị của các đặc trưng Haar-like cho tất cả các vị trí trên ảnh đòi hỏi chi phí tính toán khá lớn, không đáp ứng được cho các ứng dụng đòi hỏi tính thời gian thực. Do đó, Viola và Jones đưa ra một khái niệm gọi là Integral Image, là một mảng 2 chiều với kích thước bằng với kích của ảnh cần tính các đặc trưng Haar-like, với mỗi phần tử của mảng này được tính bằng cách tính tổng của điểm ảnh phía trên (dòng - 1) và bên trái (cột - 1) của nó. Bắt đầu từ vị trí trên, bên trái đến vị trí dưới, phải của ảnh, việc tính toán này đơn thuần chỉ dựa trên phép cộng số nguyên đơn giản, do đó tốc độ thực hiện rất nhanh.



Cách tính Integral Image của ảnh:

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

Sau khi biểu diễn ảnh bằng các đặc trưng Haar-like sẽ được đưa vào một tập học được học bằng thuật toán học máy Adaboost sử dụng bộ phân lớp Cascades để xác thực khuôn mặt.

#### **2.4.2 Xây dựng bộ phân lớp sử dụng Adaboost**

AdaBoost là một thuật toán Boosting dùng để xây dựng bộ phân lớp (Classifiers). Nó thường được kết hợp với các mô hình Cascades of Classifiers để tăng tốc độ phát hiện khuôn mặt trong ảnh. Tư tưởng của thuật toán Adaboost là kết hợp các bộ phân lớp yếu ( weak classifiers) thành một bộ phân lớp mạnh ( strong classifiers ). Trong quá trình xây dựng, bộ phân lớp yếu tiếp theo sẽ được xây dựng dựa trên bộ phân lớp yếu trước, cuối cùng các bộ phân lớp yếu sẽ được kết hợp để trở thành bộ phân lớp mạnh.

Boosting là thuật toán học quần thể bằng xây dựng nhiều thuật toán học cùng một lúc ( ví dụ như cây quyết định ) và kết hợp chúng lại. Mục đích là để một cụm nhóm các week learner sau đó kết hợp chúng lại tạo ra một strong learner duy nhất.

Mô hình Cascade of Classifiers được xây dựng nhằm rút ngắn thời gian xử lý, giảm thiểu nhận dạng lầm (false alarm) cho bộ phân lớp. Cascade trees gồm nhiều tầng (stage hay còn gọi là layer), mỗi tầng là một mô hình học với bộ phân lớp yếu là các cây quyết định. Một mẫu để được phân lớp là đối tượng thì nó cần phải đi qua hết tất cả các tầng. Các tầng sau được huấn luyện bằng những mẫu âm negative (không phải mặt người) mà tầng trước nó nhận dạng sai, tức là nó sẽ tập trung học từ các mẫu background khó hơn, do đó sự kết hợp các tầng học này lại sẽ giúp bộ phân lớp giảm thiểu nhận dạng lầm. Với cấu trúc này, những mẫu âm negative dễ nhận dạng sẽ bị loại ngay từ những tầng đầu tiên, giúp đáp ứng tốt nhất thời gian xử lý và vẫn duy trì được hiệu quả phát hiện khuôn mặt.

Các ảnh khuôn mặt mẫu sau khi được chuyển sang ảnh xám sẽ được trích chọn các đặc trưng Haar-like và đưa vào học. Việc sử dụng đặc trưng

Haar-like cho phương pháp học Adaboost với phân lớp bằng Cascade sẽ cho kết quả tốt hơn với thời gian học ngắn hơn so với sử dụng các đặc trưng khác.

### **2.4.3 Xác thực khuôn mặt**

Viola và Jones sử dụng các cửa sổ quét, các cửa sổ sẽ quét lần lượt hết bức ảnh sau đó thay đổi kích thước và quét lại cho đến khi kích thước cửa sổ là nhỏ nhất so với khuôn mẫu. Với mỗi cửa sổ quét được sẽ được biểu diễn bằng các đặc trưng Haar-like rồi đưa qua bộ phân lớp đã được xây dựng.

## CHƯƠNG 3: ỨNG DỤNG PHÉP TOÁN HÌNH THÁI TRONG BÀI TOÁN PHÁT HIỆN KHUÔN MẶT

### 3.1 Phát biểu bài toán

Từ các phương pháp được đưa ra ở phần trước, bài toán phát hiện khuôn mặt được thực hiện theo hai bước chính:

- Xác định vị trí các vùng có thể là khuôn mặt trong ảnh.
- Xác thực khuôn mặt.

Ở bước xác định vị trí các vùng có thể là khuôn mặt trong ảnh, một số phương pháp đã sử dụng kỹ thuật tiền xử lý để thu hẹp không gian tìm kiếm các vùng có thể là khuôn mặt trên ảnh. Kỹ thuật này chỉ xét tới các vùng là màu da dựa trên thuật toán phân đoạn dựa trên màu da. Sau khi phân đoạn ảnh dựa trên màu da, ảnh thu được thường xuất hiện các phần nhiễu gây ảnh hưởng tới độ chính xác khi xác thực khuôn. Để giải quyết vấn đề này, một số kỹ thuật lọc nhiễu, lấp lỗ hổng dựa trên phép toán hình thái sẽ được áp dụng.

Ở bước xác thực khuôn mặt, các phương pháp sử dụng các tập ảnh mẫu (ảnh khuôn mặt và ảnh không phải khuôn mặt) và áp dụng kỹ thuật học máy để xây dựng bộ phân lớp. Bộ phân lớp này sau đó dùng để xác thực các vùng ảnh có phải khuôn mặt hay không.

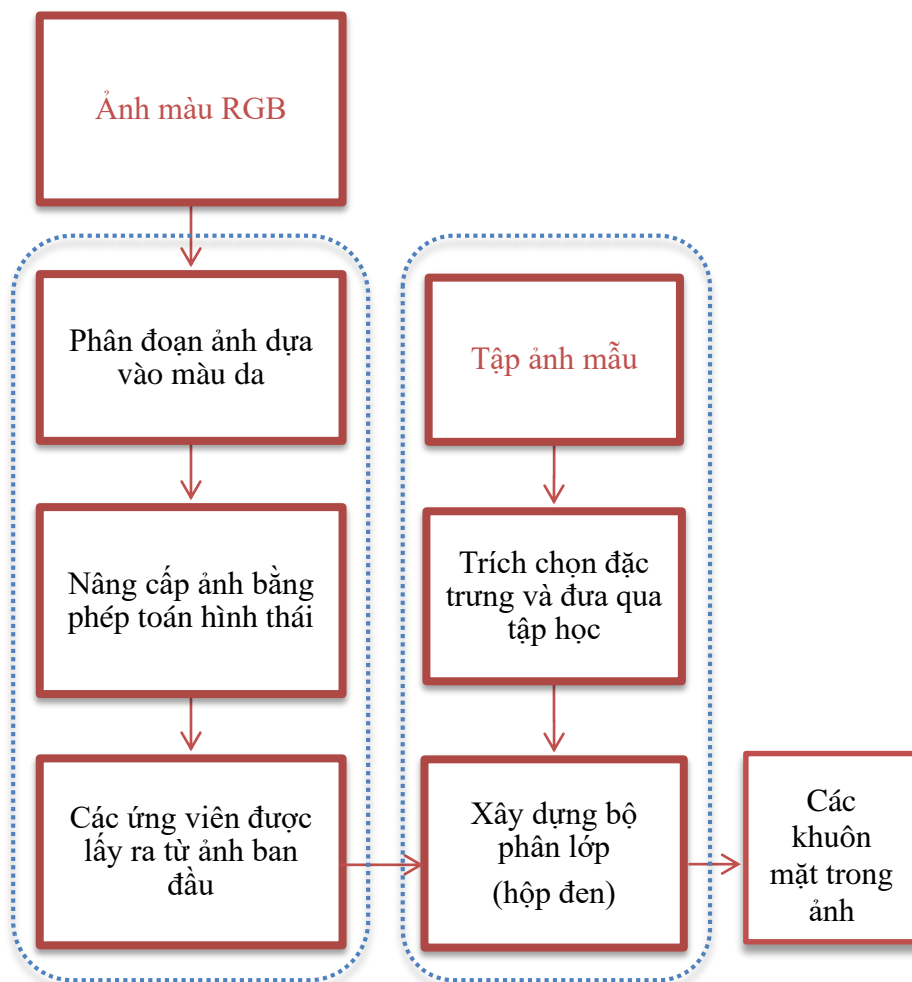
Bài toán phát hiện khuôn mặt được tiến hành theo trình tự sau :

1. Xây dựng bộ phân lớp
  - Trích chọn đặc trưng Haar-like của tập ảnh mẫu
  - Huấn luyện bộ phân lớp sử dụng phương pháp học Adaboost
2. Xác định các vùng có thể là khuôn mặt
  - Phân đoạn ảnh dựa trên màu da
  - Nâng cấp ảnh với phép toán hình thái
  - Xác định các vùng có thể là khuôn mặt

### 3. Xác thực khuôn mặt

- Trích chọn đặc trưng các vùng ảnh cần xác thực
- Xác thực thông qua bộ phân lớp đã được huấn luyện

Sơ đồ khối của bài toán phát hiện khuôn mặt được mô tả như hình 3.1:



**Hình 3.1: Mô hình biểu diễn các bước thực hiện**

### 3.2 Xây dựng bộ phân lớp

Đầu tiên, một tập mẫu là các ảnh khuôn mặt người và ảnh không phải khuôn mặt người có kích thước 24x24 được biến đổi sang ảnh đa mức xám. Với mỗi ảnh mẫu đó, chương trình sẽ trích chọn các đặc trưng Haar-like theo phương pháp đã đề cập đến ở phần 2.4.1. Khi đó, ảnh sẽ được biểu diễn bằng một ma trận có số cột bằng tổng số đặc trưng trên ảnh đó và số hàng là các chỉ số bao gồm:

$$F = [if; xf; yf; wf; hf; ir]$$

Trong đó:

- $if$ : chỉ số của đặc trưng Haar.
- $xf$ ;  $yf$ : vị trí trên cùng bên trái của đặc trưng Haar.
- $wf$ ;  $hf$ : chiều rộng và chiều cao của đặc trưng Haar.
- $ir$ : chỉ số của hình chữ nhật đầu tiên theo hàm tính đặc trưng.

Với cách biểu diễn như vậy thì một tập ảnh mẫu sẽ cho một tập các ma trận. Với tập các ma trận biểu diễn các đặc trưng Haar-like đó, chương trình sẽ sử dụng thuật toán học để xây dựng một bộ phân lớp theo phương pháp học AdaBoost. Cách xây dựng bộ phân lớp sử dụng Adaboost dựa theo phương pháp đã trình bày phần 2.4.2.

### 3.3 Xác định các vùng da trong ảnh

#### 3.3.1 Phân đoạn ảnh dựa vào màu da

Để xác định được vùng ảnh nào là vùng màu da, vùng ảnh nào là vùng không là màu da (vùng nền) cần mô hình màu da trong các không gian màu sắc mà yêu cầu là chọn ra các mô hình màu da thích hợp để ít bị nhầm nhất. Đồ án sử dụng ba mô hình màu da của các không gian màu RGB, YCbCr và HSV dựa theo công thức màu da của Abdellatif Hajraoui và Mohamed Sabri[1] đưa ra để phát hiện ra vùng màu da trong ảnh.

- Công thức màu da trong không gian màu RGB:

$$\begin{cases} R > 95 \text{ and } G > 40 \text{ and } B > 20 \\ \text{Max}\{R, G, B\} - \text{Min}\{R, G, B\} > 15 \\ |R - G| > 15 \\ R > G \text{ and } R > B \end{cases} \quad [3.1]$$

- Công thức màu da trong không gian màu HSV:

$$\begin{cases} 0 \leq H \leq 50 \\ 0.2 \leq S \leq 0.68 \text{ and } 0.35 \leq V \leq 1 \end{cases} \quad [3.2]$$

- Công thức màu da trong không gian màu YCbCr:

$$97.5 \leq Cb \leq 142.5 \text{ and } 134 \leq Cr \leq 176 \quad [3.3]$$



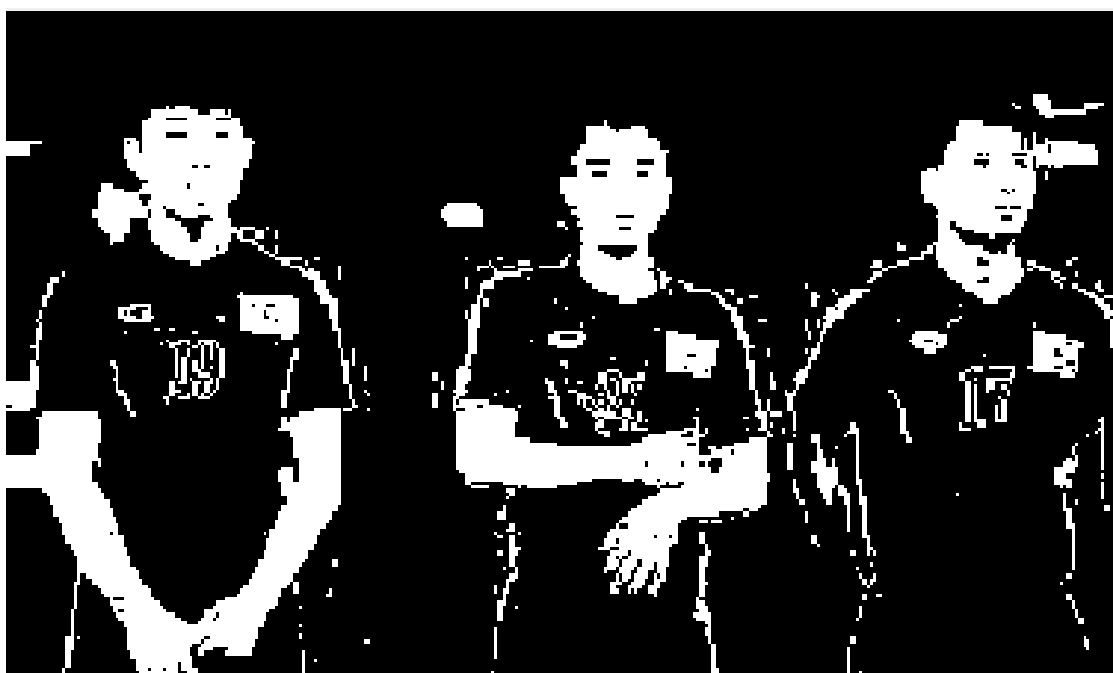
Kết quả phân đoạn màu da được biểu diễn như hình 3.2:



**Hình 3.2:** Kết quả sau khi phát hiện màu da

### 3.3.2 Nâng cấp ảnh bằng phép toán hình thái học

Qua phân đoạn ảnh dựa vào màu da sẽ loại được các vùng không là màu da nhưng hình ảnh sẽ có rất nhiều phân gậy nhiễu và các lỗ, minh họa như hình 3.3.



**Hình 3.3:** Ảnh sau khi phân đoạn xuất hiện nhiễu nhiều và các lỗ

Các phần gây nhiễu và các lỗ đố ít nhiễu sẽ gây ảnh hưởng tới tiến độ khi thực hiện xác thực khuôn mặt sau này. Do vậy, để loại bỏ các phần nhiễu này, đồ án áp dụng một bước tiền xử lý dựa trên phép toán hình thái. Ý tưởng cơ bản của phép toán là sử dụng phần tử cấu trúc và áp dụng các phép toán hình thái để loại bỏ các điểm gây nhiễu này.

Tuy nhiên, nếu thực hiện phép toán hình thái trên ảnh nhị phân, kết quả sẽ không được tốt, ví dụ minh họa như hình 3.4:



**Hình 3.4 Thực hiện phép toán hình thái trên ảnh nhị phân**

Do vậy, đồ án lựa chọn thực hiện phép toán hình thái trên ảnh đa mức xám. Để áp dụng phép toán hình thái trên ảnh xám, đồ án đã thực hiện các bước sau:

Đầu tiên đồ án chuyển đổi ảnh ban đầu sang ảnh đa cấp xám rồi sử dụng phép mở (Open) để loại bỏ các vật thể rất nhỏ khỏi hình ảnh sao cho vẫn giữ được hình dạng và kích thước của các vật thể lớn hơn trong hình ảnh. Sử dụng một phần tử cấu trúc dạng đĩa (DISK) có bán kính 1 để làm điều đó.

Sau khi sử dụng phép mở để thực hiện lọc các phần tử nhỏ gây nhiễu ra ngoài có thể sinh ra các lỗ nhỏ, cần làm đầy các lỗ để các khuôn mặt trở thành

một vùng liên tục. Để làm được điều đó, đồ án sử dụng phép tô đầy vùng ( region fill ). Kết quả áp dụng tiền xử lý bằng phép toán hình thái trên ảnh xám được minh họa trong hình 3.5.



**Hình 3.5: Kết quả sau khi nâng cấp ảnh trên ảnh xám**

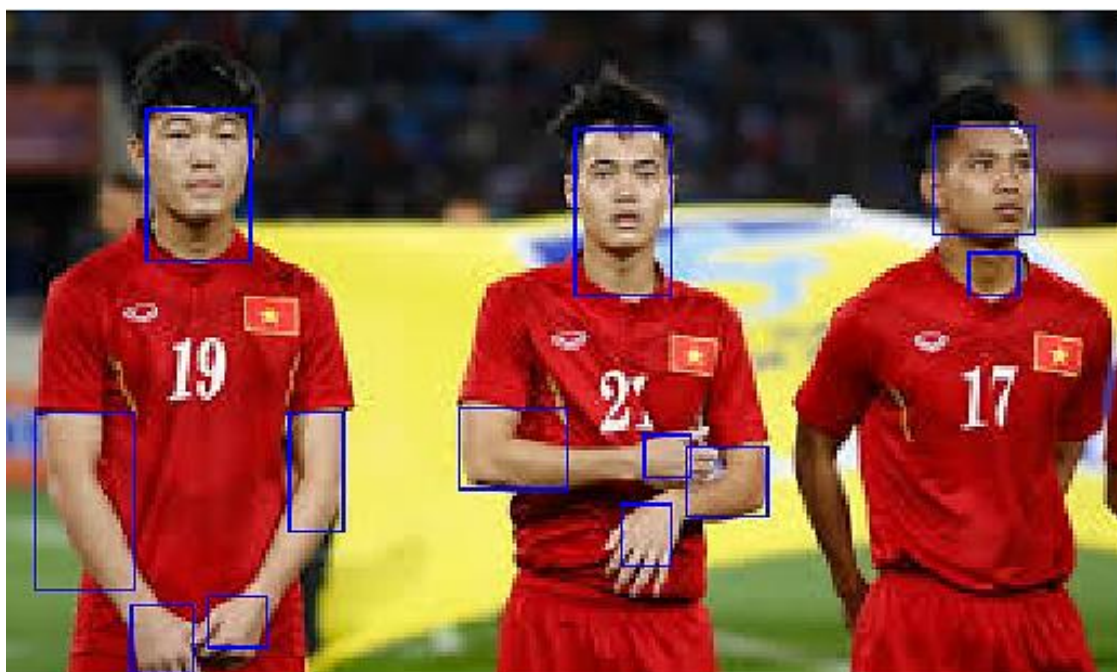


**Hình 3.6: Biến đổi sang ảnh nhị phân**

Để thực hiện thuật toán khoanh vùng các khu vực này lại, cần chuyển đổi sang một ảnh nhị phân để sử dụng trong thuật toán đóng khung các đối

tượng ( Bounding Box ). Kết quả phân ngưỡng trên ảnh xám sau khi thực hiện nâng cấp với phép toán hình thái học được chỉ ra trong hình 3.6. Có thể thấy chất lượng phân vùng chính xác hơn so với thực hiện phép toán hình thái học trên ảnh nhị phân ( như chỉ ra trong hình 3.4 ).

Ranh giới ( Bounding Box ) của các đối tượng có thể là vùng màu da được gọi là các ứng viên được đóng khung lại. Việc tạo các khung trên ảnh nhị phân sẽ có được vị trí và kích thước của các hộp đó, sau đó chương trình sẽ đối chiếu lên ảnh đầu vào để có vị trí của các hộp ranh giới trên đó. Kết quả được minh họa như hình 3.7:



**Hình 3.7: Kết quả là các ứng viên được đóng khung**

Quan sát trong hình trên sẽ thấy, các vùng như cánh tay cũng được đóng khung trong hộp ranh giới BoundingBox ( gọi là các ứng viên ) vì có màu giống với màu da mặt. Đó đó ta cần đưa qua bộ phân lớp đã được xây dựng ở bước đầu để có thể xác thực được đâu là khuôn mặt trong các ứng viên đó.

### 3.4 Xác thực khuôn mặt

Sau khi xây dựng bộ phân lớp và có được các ứng viên là các vùng màu da, chương trình sẽ đưa từng ứng viên qua bộ phân lớp. Một cửa sổ quét sẽ

quét hết các vị trí trên các ảnh ứng viên để có được kết quả là khuôn mặt trong các ứng viên đó. Chi tiết các bước như sau.

Khi một ứng viên được đưa vào bộ phân lớp, chương trình sẽ chuyển ảnh ứng viên sang ảnh đa mức xám rồi chuẩn hóa kích thước bằng cách chia chiều cao, chiều rộng cho 384, sau đó mới tính ra ảnh tích phân của ứng viên đó. Có được ảnh tích phân rồi, hệ thống sẽ tính toán ra các tỉ lệ cho giai đoạn tính toán sau này:

$$\text{ScaleWidth} = \text{IntegralImage.width}/\text{Haarcascade.width}$$

$$\text{ScaleHeight} = \text{IntegralImage.height}/\text{Haarcascade.height}$$

$$\text{StartScale} = \text{ScaleWidth} \text{ nếu } \text{ScaleWidth} < \text{Scale.Height} \text{ và ngược lại}$$

Trong đó:

- $\text{Haarcascade.width/height}$ : Chiều rộng/cao của đặc trưng Haar.
- $\text{Integral.Width/Height}$ : Chiều rộng/cao của ảnh tích phân.
- $\text{StartScale}$  tìm được sẽ là tỉ kích thước ban đầu của cửa sổ quét.

Có tỉ lệ  $\text{StartScale}$  sẽ tính được kích thước, vị trí ban đầu của cửa sổ quét. Tiếp theo chương trình sẽ liên tục thực hiện thay đổi vị trí, kích thước cửa sổ theo một tỉ lệ ( $\text{Scale}$ ) rồi quét lần lượt hết bức ảnh và lặp lại.

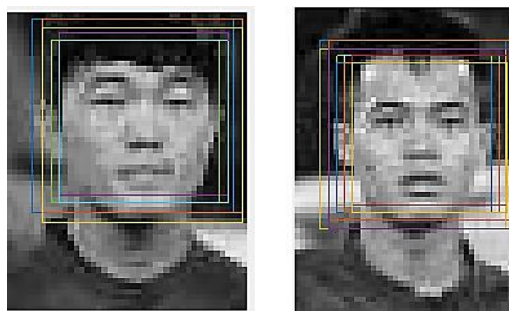
$$\text{Scale} = \text{StartScale} * \text{Option.ScaleUpdate}^{(i-1)}$$

Trong đó:

- $\text{Option.ScaleUpdate}$  là chỉ số nhất định, ở đây được gán bằng 1/1.2.
- $i$  là số lần lặp được tính  $i=1$  đến  $\text{init}$ .
- $\text{init} = \text{Log}(1/\text{StartScale})/\text{Log}(\text{Option.ScaleUpdate})$ .

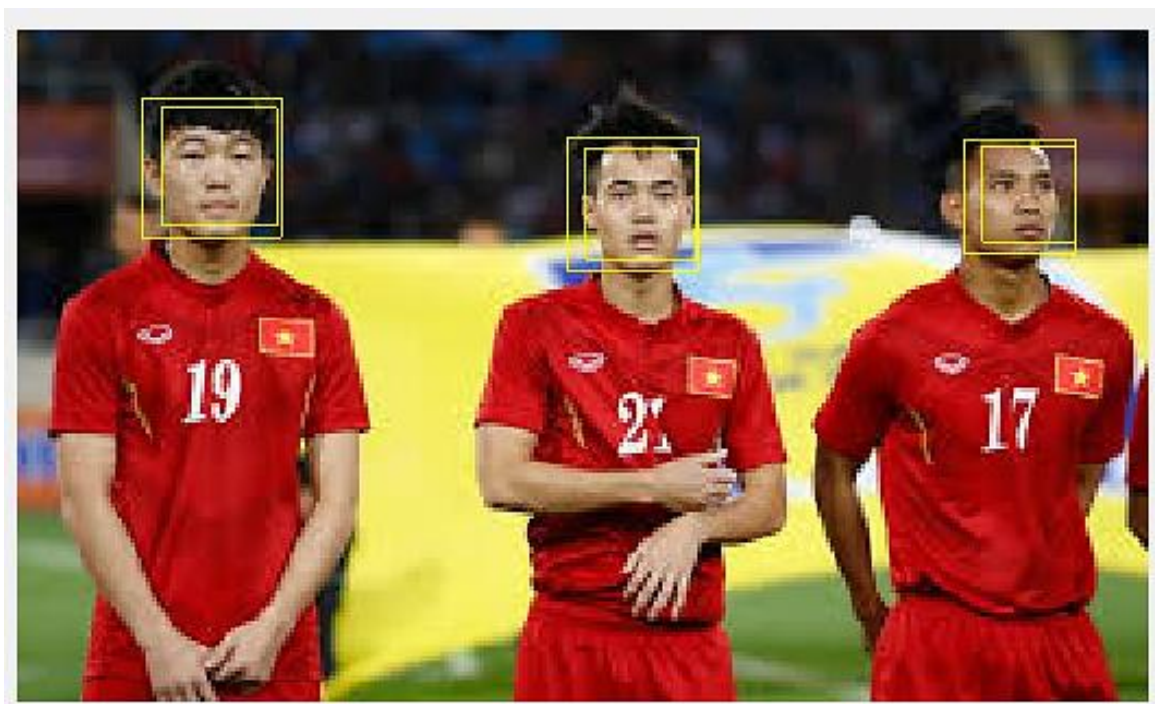
Sau mỗi lần thay đổi kích thước và vị trí như vậy, cửa sổ đó sẽ được trích chọn các đặc trưng Haar-like rồi đưa qua bộ phân lớp để xác thực khuôn mặt. Nếu đó là khuôn mặt, chương trình sẽ lưu lại vị trí và kích thước của cửa

số đó gồm: tọa độ (x,y) góc trên cùng bên trái và kích thước gồm chiều rộng, chiều cao (w,h) vào một mảng Array = [x,y,w,h];



**Hình 3.8: Vị trí các khuôn mặt trên mỗi ứng viên**

Cuối cùng, khi có được danh sách các ứng viên với các vị trí và kích thước, ta cần trả lại vị trí của các ứng viên đó vào trong ảnh gốc ban đầu. Thực hiện cộng tọa độ trên cùng bên trái của ứng viên đó khi được lấy ra ở bước xác định vùng da với vị trí của khuôn mặt trong mảng (Array) ứng với ứng viên đó. Kết quả sẽ cho vị trí của các khuôn mặt trong ảnh đầu vào.



**Hình 3.9: Các vị trí khuôn mặt trong ảnh đầu vào**



### 3.5 Cài đặt chương trình

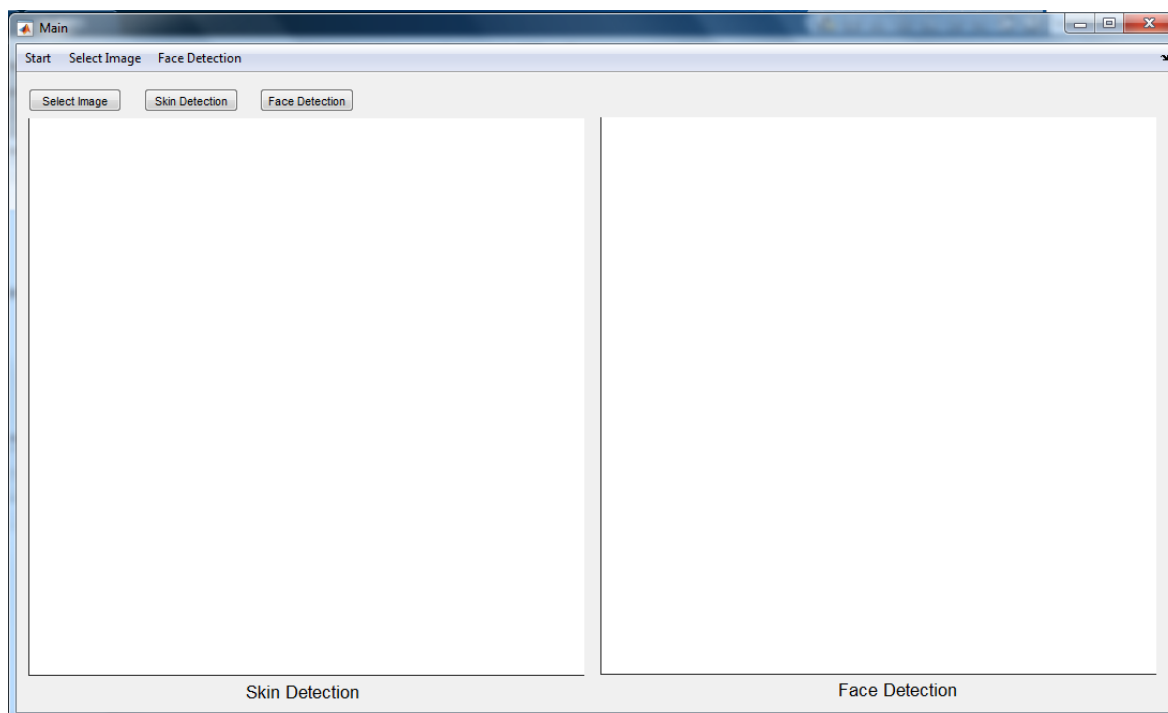
#### 3.5.1 Ngôn ngữ lập trình MATLAB

MATLAB là một môi trường tính toán số học và lập trình được thiết kế bởi công ty MathWorks. MATLAB cho phép tính toán số với ma trận, vẽ đồ thị hàm số hay biểu đồ thông tin, thực hiện thuật toán, tạo các giao diện người dùng và liên kết với những chương trình máy tính viết trên nhiều ngôn ngữ lập trình khác. MATLAB giúp đơn giản hóa việc giải quyết các bài toán tính toán kỹ thuật so với các ngôn ngữ lập trình truyền thống như C, C++, và Fortran.

MATLAB được sử dụng trong nhiều lĩnh vực, bao gồm xử lý tín hiệu và ảnh, truyền thông, thiết kế điều khiển tự động, đo lường kiểm tra, phân tích mô hình tài chính, hay tính toán sinh học. Với hàng triệu kỹ sư và nhà khoa học làm việc trong môi trường công nghiệp cũng như ở môi trường hàn lâm, MATLAB là ngôn ngữ của tính toán và nghiên cứu khoa học.

#### 3.5.2 Giao diện chương trình

- Giao diện chính của chương trình:



**Hình 3.10: Giao diện chính của chương trình**

Giao diện chương trình bao gồm:

**Button Select Image:** Mở file truy vấn để chọn ảnh.

- Ảnh được chọn ở đây là ảnh màu có đuôi \*.jpg

**Button Skin Detection:** Thực hiện phân đoạn dựa vào màu da

- Sử dụng công thức màu da để phân đoạn và áp dụng phép toán hình thái để tiền xử lý nâng cao chất lượng ảnh.

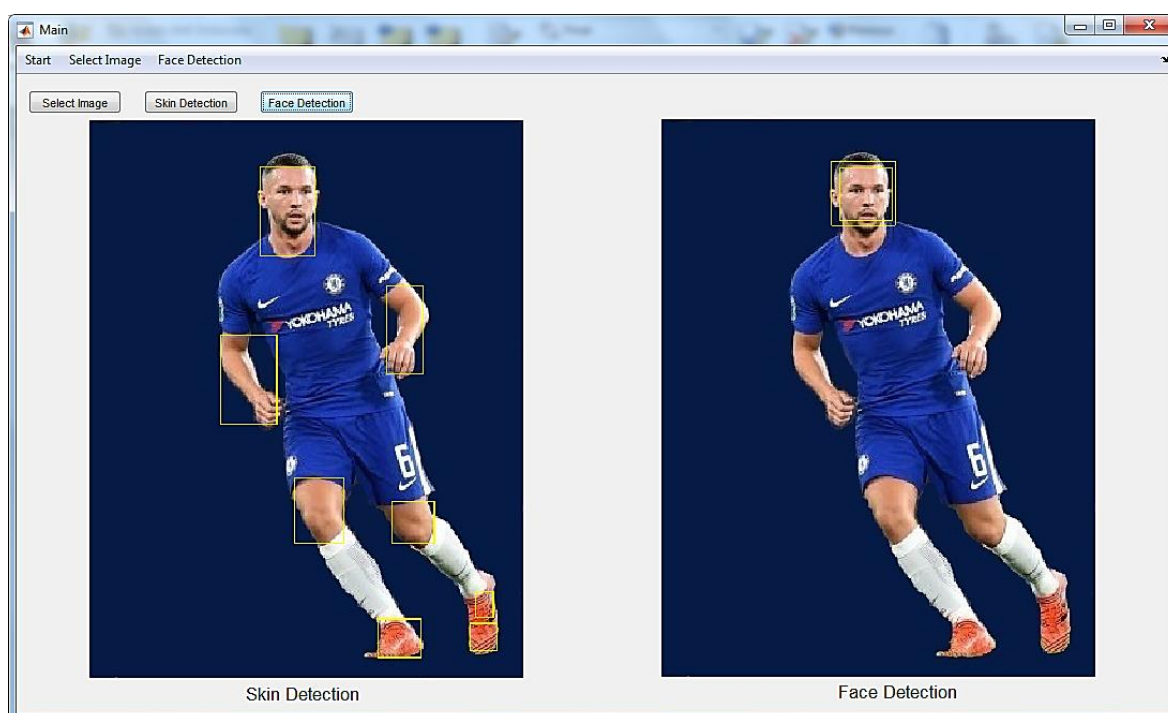
**Button Face Detection:** Thực hiện xác thực khuôn mặt

- Xác thực khuôn mặt trong các vùng màu da.

**Axes1:** Hiển thị kết quả của bước xác định các vùng da

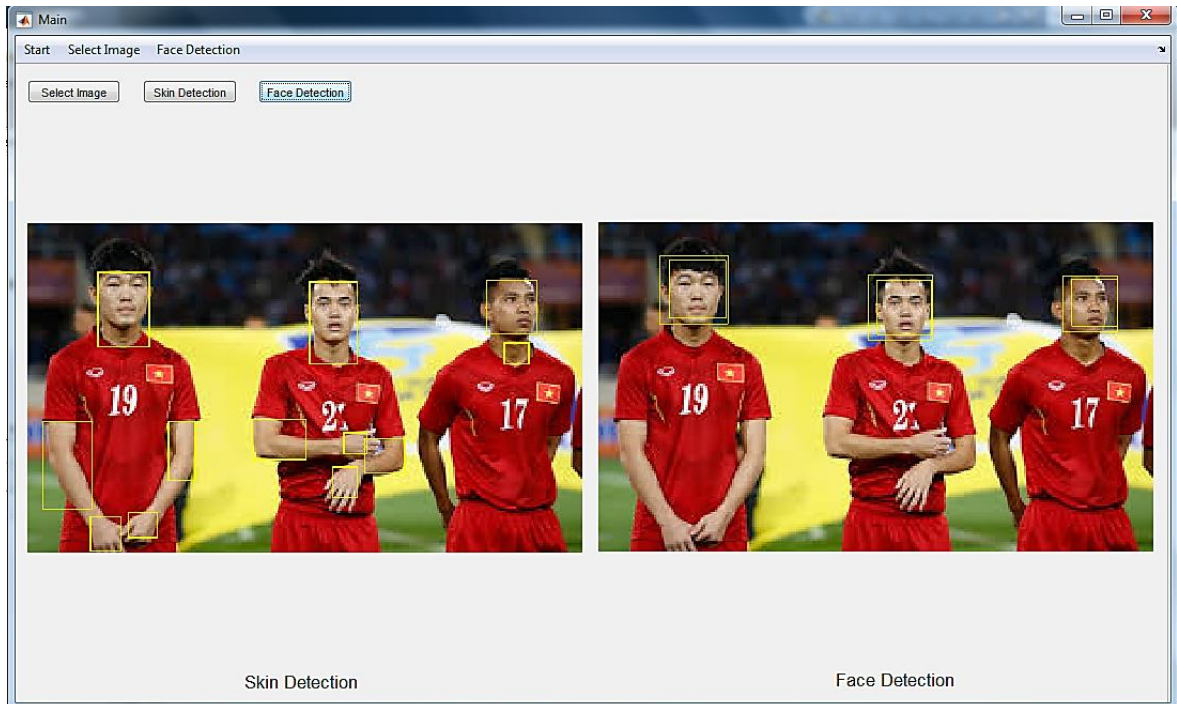
**Axes2:** Hiển thị kết quả bước xác thực khuôn mặt

- Kết quả khi thực hiện các chức năng của chương trình:



**Hình 3.11: Kết quả của bức ảnh có 1 khuôn mặt người**



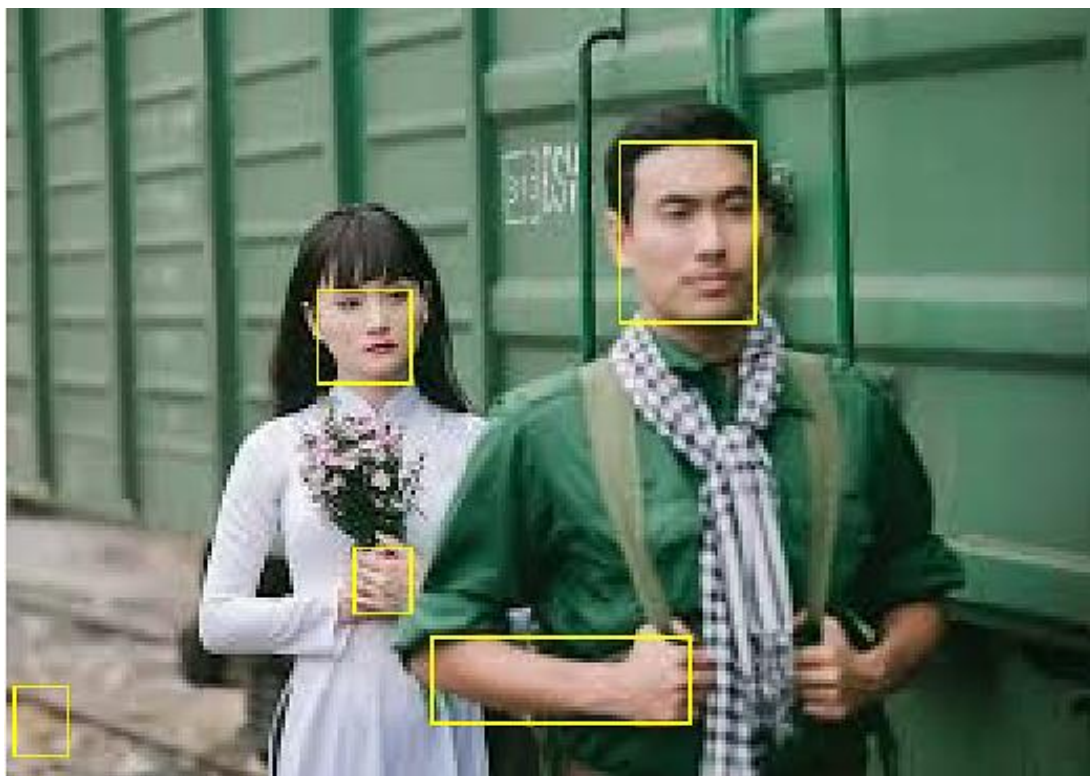


**Hình 3.12: Kết quả của ảnh có nhiều mặt người**

### 3.5.3 Kết quả xác định các vùng màu da

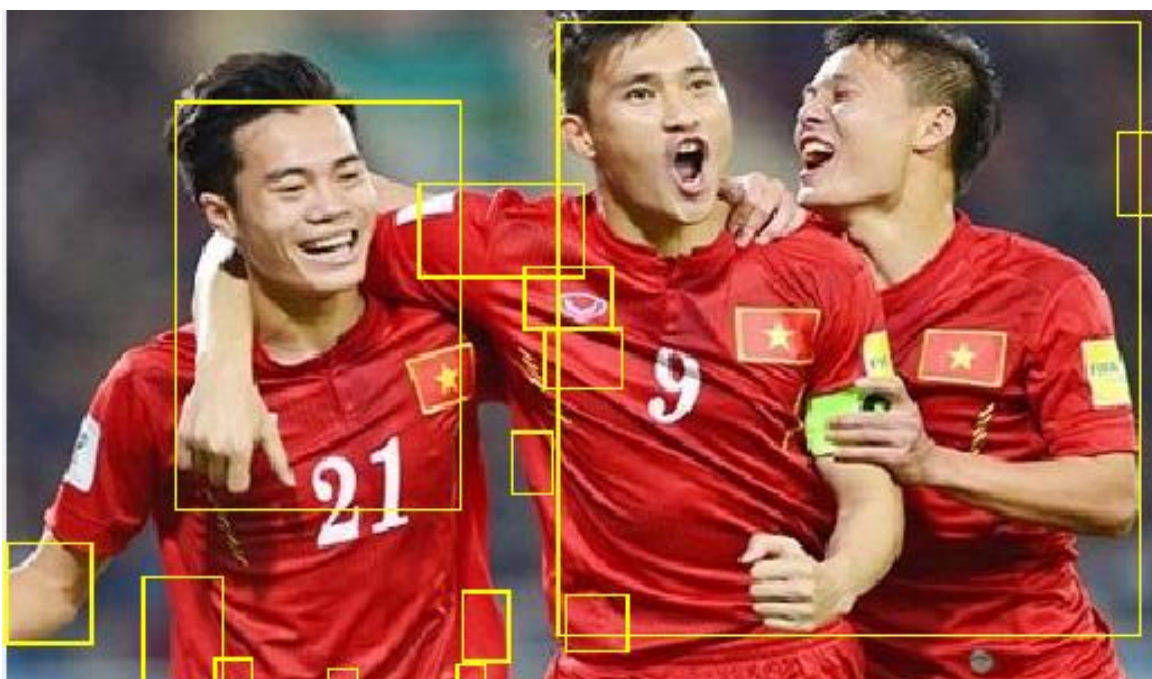
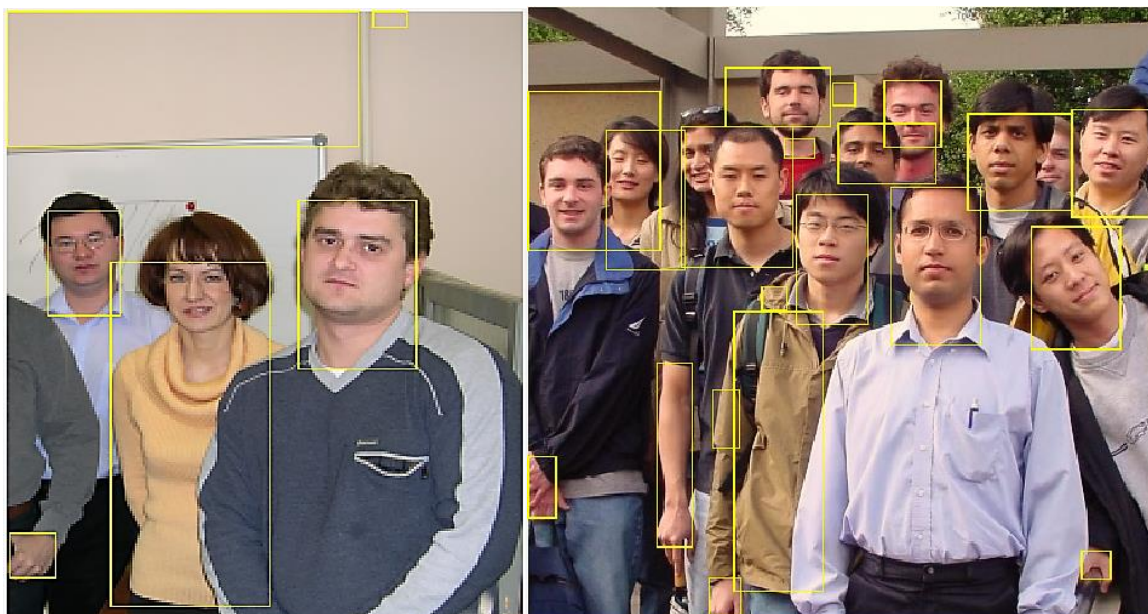
Một số kết quả của chương trình khi xác định các vùng da trong ảnh:

- Phát hiện vùng da trên một bức ảnh có nền đơn giản:



**Hình 3.13: Một số bức ảnh có nền đơn giản**

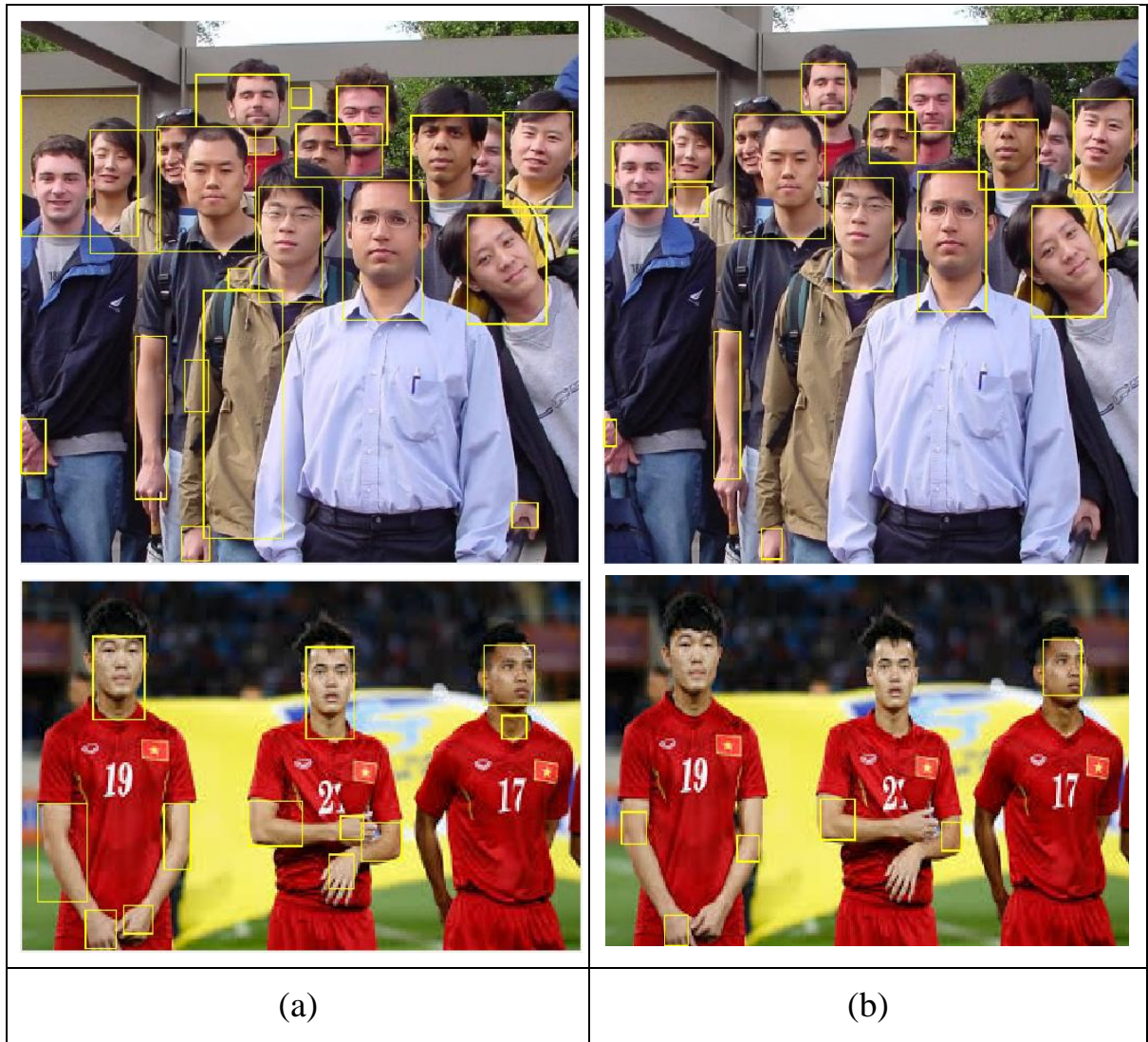
- Ví dụ về một bức ảnh có nhiều khu vực bị nhầm do có nhiều vùng có sắc tố giống với màu da mặt:



**Hình 3.14** Kết quả trên ảnh có nhiều vùng giống với nền

Các kết quả như trong các hình minh họa trên có thể thấy, các ảnh có vùng nền giống với sắc tố da nên dễ bị nhầm thành vùng da mặt. Tuy nhiên, ta có thể thay đổi một số giá trị các chỉ số trong công thức màu da để có được kết quả tốt hơn. Ví dụ minh họa như hình 3.15.

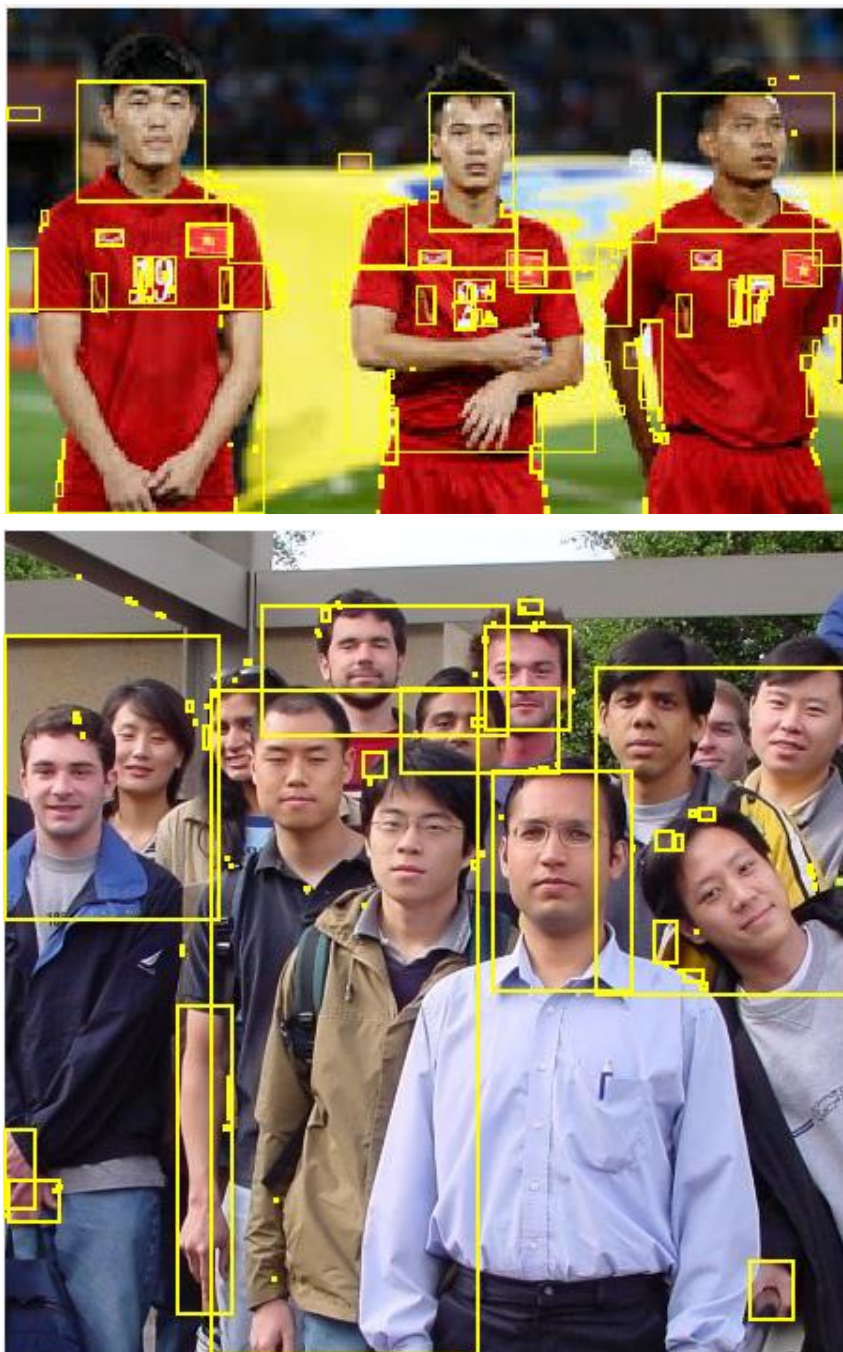




**Hình 3.15:** Kết quả với các tham số khác nhau xác định màu da

$$\begin{aligned}
 (a) \quad & \left\{ \begin{array}{l} R > 95 \text{ and } G > 40 \text{ and } B > 20 \\ \text{Max}\{R, G, B\} - \text{Min}\{R, G, B\} > 15 \\ |R - G| > 15 \\ R > G \text{ and } R > B \end{array} \right. \\
 (b) \quad & \left\{ \begin{array}{l} R > 120 \text{ and } G > 50 \text{ and } B > 30 \\ \text{Max}\{R, G, B\} - \text{Min}\{R, G, B\} > 35 \\ |R - G| > 25 \\ R > G \text{ and } R > B \end{array} \right.
 \end{aligned}$$

Một số ví dụ minh họa kết quả chương trình khi không áp dụng phép toán hình thái để tiền xử lý như hình 3.16.

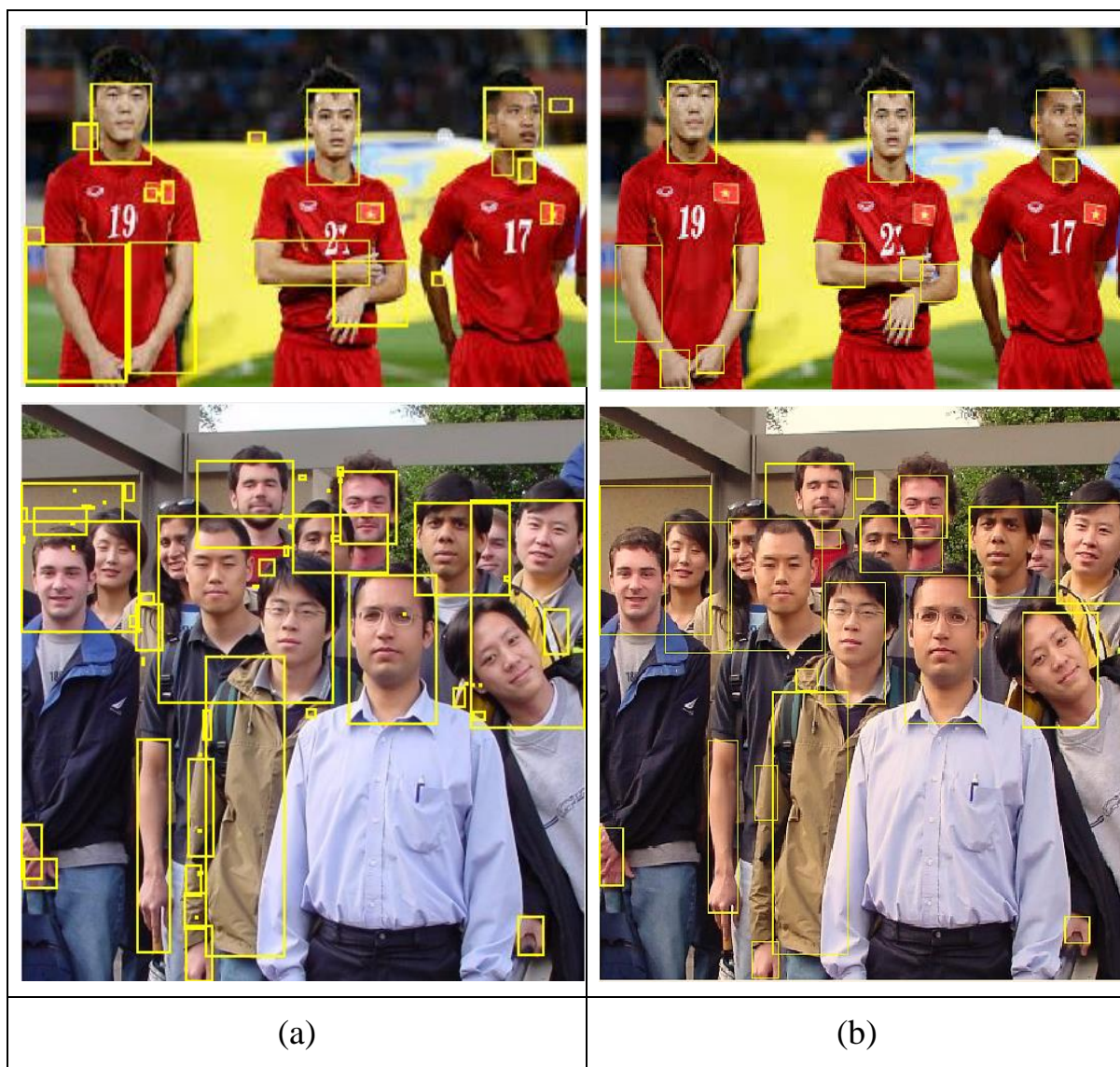


**Hình 3.16: Kết quả nâng cấp ảnh không sử dụng phép toán hình thái**

Từ thực nghiệm có thể thấy rằng, nếu không áp dụng phép toán hình thái để tiền xử lý, các vùng gây nhiễu và các lỗ sẽ gây ra rất nhiều nhầm lẫn về kích thước, số lượng các vùng màu da.

Kết quả chương trình thực hiện phát hiện vùng da sử dụng phép toán hình thái trên ảnh nhị phân và trên ảnh xám:





**Hình 3.17: Sự khác biệt khi sử dụng phép toán hình thái trên ảnh nhị phân và ảnh xám**

(a): Áp dụng phép toán hình thái trên ảnh nhị phân

(b): Áp dụng phép toán hình thái trên ảnh đa cấp xám

**Nhận xét:**

- Việc phát hiện vùng da là khá tốt vì đã không bỏ sót phần màu da nào, tuy nhiên vẫn có những nhầm lẫn vì các vùng bị nhầm có màu khá giống với màu da.

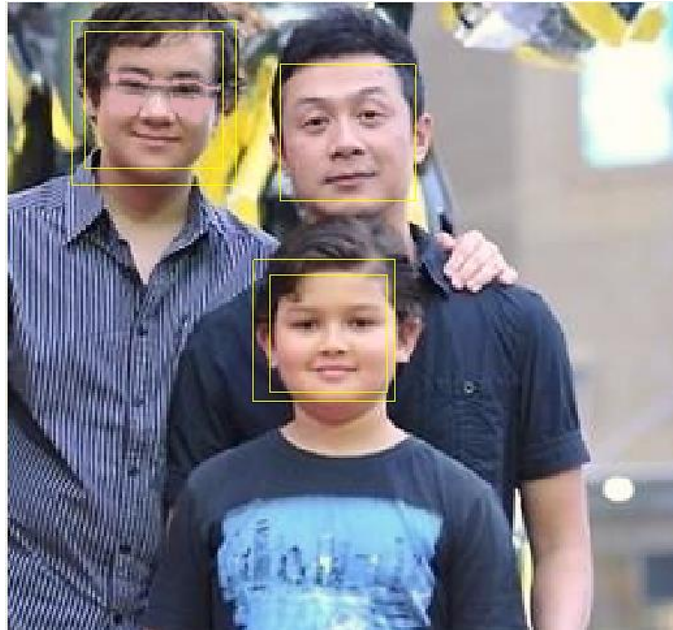
- Công thức màu da và phần tử cấu trúc của phép toán hình thái trong đồ án mang tính tương đối, tức là công thức và kích thước phần tử có thể tốt trong một số bức ảnh chứ không phải tất cả các bức ảnh.
- Sử dụng phép toán hình thái trên ảnh xám để tiền xử lý sẽ cho các đối tượng chính xác và rõ ràng, thuận tiện cho các bước xác thực khuôn mặt.

### 3.5.4 Kết quả xác thực khuôn mặt

Kết quả xác thực khuôn mặt sau khi xác định vùng màu da.

- Thực nghiệm trên những bức ảnh có khuôn mặt thẳng, không bị che khuất:





**Hình 3.18: Kết quả chương trình với các ảnh có nền đơn giản**



**Hình 3.19: Kết quả trong các ảnh có nhiều khuôn mặt**



- Thực nghiệm trên các hình có khuôn mặt không thẳng hoặc bị che khuất:







**Hình 3.20: Kết quả thực nghiệm trên một số hình có khuôn mặt không thẳng hoặc bị che khuất**

**Nhận xét:**

- Tốc độ xử lý nhanh hay chậm phụ thuộc vào số vùng da được phát hiện trong ảnh nhiều hay ít nên có thể bị chậm khi số ảnh có ứng viên lớn.
- Tỷ lệ dò tìm thành công cao: trong số các hình sử dụng làm ví dụ, số khuôn mặt bị bỏ sót là 6 trên tổng số 50 khuôn mặt trong ảnh có các nền và đối tượng khác nhau.

**3.5.5 Nhận xét sau khi thực nghiệm bằng chương trình**

Chương trình có tốc độ không cao khi có nhiều các ứng viên nhưng bù lại sẽ có những điểm tích cực sau:

- Không cần giả định trước về kích thước các khuôn mặt và có thể sử dụng cho các khuôn mặt có kích thước khác nhau.
- Không phụ thuộc vào độ sáng của ảnh.
- Không phụ thuộc vào số lượng các khuôn mặt
- Chương trình sử dụng được trên các ảnh có kích thước lớn.
- Cho kết quả có tính chính xác cao

## KẾT LUẬN

Sau một thời gian tìm hiểu và nghiên cứu đề tài này, em đã đạt được một số kết quả sau:

- Tìm hiểu được các hướng tiếp cận trong phát hiện khuôn mặt.
- Tìm hiểu về phép hình thái học (morphology) và các ứng dụng của nó để nâng cấp ảnh.
- Tìm hiểu một số phương pháp phát hiện khuôn mặt.
- Áp dụng các kỹ thuật xử lý ảnh để xây dựng một chương trình thử nghiệm phát hiện khuôn mặt dựa vào màu da.

Tuy nhiên, vẫn còn tồn tại một số vấn đề:

- Chương trình chạy chậm do công thức màu da và phần tử cấu trúc trong phép toán hình thái chưa được tối ưu hóa.
- Mới chỉ phát hiện được các khuôn mặt có tư thế thẳng.
- Chương trình được cài đặt bằng MATLAB nên phần giao diện của chương trình không được đẹp và sơ sài. Tốc độ xử lý không nhanh do khả năng phân cứng còn hạn chế.

Hướng phát triển tiếp theo:

- Thực nghiệm nhiều hơn để tìm ra các tham số tối ưu phát hiện màu da.
- Thử nghiệm để xây dựng bộ phân lớp cho kết quả phân lớp tốt hơn.
- Tìm hiểu, vận dụng các kỹ thuật để tăng tốc độ xử lý của chương trình.

## TÀI LIỆU THAM KHẢO

- [1] Abdellatif Hajraoui, Mohamed Sabri, “*Face Detection Algorithm based on Skin Detection, Watershed Method and Gabor Filters*”, May 2014.
- [2] <http://tailieu.vn/doc/mot-so-tinh-chat-cua-phep-toan-hinh-thai-va-ung-dung-trong-phat-hien-bien--1556027.html>
- [3] Iyas Saeed, Bassel Shanwar, Nizar Zarka, “*Face detection based on skin color information, morphology and neuralnetwork*”, June 2016.
- [4] <https://www.mathworks.com/matlabcentral/fileexchange/29437-viola-jones-object-detection.html>
- [5] Manoj Seshadrinathan, Jezekiel Ben – Arie, “*Face Detection by Integration of Evidence*”, September 2005
- [6] Dmytro Peleshko, Kateryna Soroka, “*Research of usage of Haar-like features and Adaboost algorithm in Viola-Jones method of object detection*”, December 2013.
- [7] Hồ Đức Lĩnh, “*Xử lý hình thái học trên ảnh và ứng dụng*”, Khoa công nghệ thông tin - Đại học Đông Á.
- [8] C. Garcia, G. Zikos, G. Tziritas, “*Face Detection in color images using Wavelet packet analysis*”, Institute of Computer Science Foundation for Research and Technology-Hellas.
- [9] K. Sandeep và A.N. Rajagopalan, “*Human Face Detection in Clutterd Using Skin Color And Edge Information*”, Electrical Engineering, Citeseer, 2002.