TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN  
& TRUYỀN THÔNG VIỆT- HÀN

**KHOA KỸ THUẬT MÁY TÍNH VÀ ĐIỆN TỬ**

**🙟🕮🙝**

****

**BÁO CÁO**

**ĐỒ ÁN CHUYÊN NGÀNH 1**

**ĐỀ TÀI:**

**PHÁT HIỆN ĐÁM CHÁY BẰNG XỬ LÝ ẢNH**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sinh viên thực hiện** | **:** | **Trần Văn Thanh** |  | **20CE044** |  |
|  |  | **Trần Huỳnh Anh Nhật** |  | **20CE040** |  |
| **Lớp** | **:** | **20CE** |  |  |  |
| **Giảng viên hướng dẫn** | **:** | **ThS. Trần Thu Thủy** |  |  |  |

***Đà Nẵng, tháng 05 năm 2023***

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN &

TRUYỀN THÔNG VIỆT - HÀN

**KHOA KỸ THUẬT MÁY TÍNH VÀ ĐIỆN TỬ**



**BÁO CÁO**

**ĐỒ ÁN CHUYÊN NGÀNH 1**

**ĐỀ TÀI:**

**PHÁT HIỆN ĐÁM CHÁY BẰNG XỬ LÝ ẢNH**

| **Sinh viên thực hiện** | **:** | **Trần Văn Thanh** | **20CE044** |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | **Trần Huỳnh Anh Nhật** | **20CE040** |  |  |
| **Lớp** | **:** | **20CE** |  |  |  |
| **Giảng viên hướng dẫn** | **:** | **ThS. Trần Thu Thủy** |  |  |  |

**Đà Nẵng, tháng 05 năm 2023**

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

**LỜI CẢM ƠN**

Chúng em xin trân trọng cảm ơn chân thành đến ban giám hiệu nhà trường, cùng các thầy cô đã tận tình chỉ bảo giảng dạy, cung cấp kiến thức nền tảng tạo điều kiện tốt nhất để chúng em học tập trau dồi.

Đặc biệt em xin chân thành cảm ơn ThS. Trần Thu Thủy đã nhiệt tình hướng dẫn chúng em trong suốt quá trình xây dựng sản phẩm.

Trong quá trình thực hiện đề tài, chúng em đã rất cố gắng nỗ lực tuy nhiên không tránh khỏi sai sót. Chúng em mong nhận được sự góp ý của thầy cô giáo và từ phía hội đồng nhà trường để có thể thực hiện hoàn chỉnh hơn ở những chủ đề.

*Chúng em xin chân thành cảm ơn!*

*Sinh viên,*

Trần Văn Thanh

Trần Huỳnh Anh Nhật

**MỤC LỤC**

[**LỜI CẢM ƠN iii**](#_heading=h.30j0zll)

[**MỤC LỤC iv**](#_heading=h.2et92p0)

[**DANH MỤC HÌNH VẼ vi**](#_heading=h.tyjcwt)

[**MỞ ĐẦU 1**](#_heading=h.3dy6vkm)

[1. Giới thiệu 1](#_heading=h.1t3h5sf)

[2. Mục tiêu của đề tài 1](#_heading=h.2s8eyo1)

[3. Nội dung và kế hoạch thực hiện 1](#_heading=h.17dp8vu)

[4. Bố cục báo cáo 2](#_heading=h.1ksv4uv)

[**Chương 1. TỔNG QUAN VỀ ĐỀ TÀI 3**](#_heading=h.z337ya)

[1.1. Đặt vấn đề 3](#_heading=h.3j2qqm3)

[1.2. Các phương pháp cảnh báo cháy phổ biến 3](#_heading=h.4i7ojhp)

[1.2.1. Cảm biến nhiệt độ 3](#_heading=h.2xcytpi)

[1.2.2. Cảm biến khói 4](#_heading=h.3whwml4)

[1.3. Tổng quan về thị giác máy tính 5](#_heading=h.qsh70q)

[**Chương 2. CƠ SỞ LÝ THUYẾT 7**](#_heading=h.49x2ik5)

[2.1. Các bước cơ bản trong xử lý ảnh 7](#_heading=h.2p2csry)

[2.1.1. Phần thu nhận ảnh 7](#_heading=h.3o7alnk)

[2.1.2. Tiền xử lý 8](#_heading=h.23ckvvd)

[2.1.3. Phát hiện 8](#_heading=h.ihv636)

[2.1.4. Nhận diện 8](#_heading=h.32hioqz)

[2.2. Một số kỹ thuật cơ bản trong xử lý ảnh 8](#_heading=h.1hmsyys)

[2.2.1. Chuyển đổi hệ màu 8](#_heading=h.41mghml)

[2.2.2. Cân bằng biểu đồ tần số 10](#_heading=h.2grqrue)

[2.2.3. Phân đoạn ảnh dùng ngưỡng 11](#_heading=h.19c6y18)

[2.3. Nhận diện đối tượng sử dụng thuật toán Harr cascade 12](#_heading=h.3tbugp1)

[2.3.1. Nhận diện đối tượng 12](#_heading=h.28h4qwu)

[2.3.2. Adaboost với đặc trưng Harr-like 13](#_heading=h.nmf14n)

[2.3.3. Cascade of Classifiers: 16](#_heading=h.2lwamvv)

[2.3.4. Cascade of boosting classifiers: 17](#_heading=h.3l18frh)

[**Chương 3. PHÂN TÍCH VÀ XÂY DỰNG HỆ THỐNG 19**](#_heading=h.4k668n3)

[3.1. Tổng quan về openCV 19](#_heading=h.2zbgiuw)

[3.2. Phân tích 19](#_heading=h.3ygebqi)

[3.2.1. Chức năng của hệ thống 19](#_heading=h.2dlolyb)

[3.2.2. Cách thức hoạt động của hệ thống 20](#_heading=h.3cqmetx)

[3.3. Xây dựng hệ thống 21](#_heading=h.4bvk7pj)

[3.3.1. Xây dựng tập tin để huấn luyện 21](#_heading=h.2r0uhxc)

[3.3.2. Kết quả sau khi chạy chương trình 22](#_heading=h.3q5sasy)

[**KẾT LUẬN VÀ HƯỚNG NGHIÊN CỨU 24**](#_heading=h.43ky6rz)

[**1.**](#_heading=h.2iq8gzs) **Kết quả** 24

[**2.**](#_heading=h.xvir7l) **Hướng nghiên cứu** 24

[**TÀI LIỆU THAM KHẢO 25**](#_heading=h.3hv69ve)

# DANH MỤC HÌNH VẼ

[Hình 1.1 Cảm biến nhiệt độ PT100 4](#_heading=h.1ci93xb)

[Hình 1.2 Cảm biến khói 4](#_heading=h.2bn6wsx)

[Hình 1.3 Ba mức xử lý của một hệ thống thị giác máy tính 6](#_heading=h.3as4poj)

[Hình 2.1 Các bước cơ bản trong xử lý ảnh 7](#_heading=h.147n2zr)

[Hình 2.2 Biểu đồ tần số của một ảnh có kích thước 5x5 pixels 10](#_heading=h.3fwokq0)

[Hình 2.3 Nguyên tắc cân bằng biểu đồ tần số 11](#_heading=h.1v1yuxt)

[Hình 2.4 Quá trình cân bằng biểu đồ tần số 11](#_heading=h.4f1mdlm)

[Hình 2.5 Ảnh biểu đồ tần số được làm mượt 11](#_heading=h.2u6wntf)

[Hình 2.6 Đặc trưng Harr-like cơ bản 13](#_heading=h.37m2jsg)

[Hình 2.7 Đặc trưng Harr-like mở rộng 13](#_heading=h.1mrcu09)

[Hình 2.8 Tìm tổng giá trị điểm ảnh ABCD bằng cách sử dụng ảnh tích lũy 14](#_heading=h.46r0co2)

[Hình 2.9 Cascade of classifiers 17](#_heading=h.111kx3o)

[Hình 2.10 Cascade of boosting classifiers 18](#_heading=h.206ipza)

[Hình 3.1 Cấu trúc cơ bản của OpenCV 19](#_heading=h.1egqt2p)

[Hình 3.2 Sơ đồ khối hệ thống 20](#_heading=h.sqyw64)

[Hình 3.3 Khái quát quá trình thực hiện hệ thống 21](#_heading=h.1rvwp1q)

[Hình 3.4 Dữ liệu dùng để huấn luyện 22](#_heading=h.1664s55)

[Hình 3.5 Lửa được nhận diện 22](#_heading=h.25b2l0r)

[Hình 3.6 Lửa được tách ra khi sử dụng phân vùng ảnh 22](#_heading=h.kgcv8k)

[Hình 3.7 Cảnh báo có đám cháy 23](#_heading=h.34g0dwd)

# MỞ ĐẦU

1. Giới thiệu

Hỏa hoạn luôn mang đến những hậu quả hết sức nghiêm trọng. Tình trạng hỏa hoạn ở Việt Nam và trên toàn thế giới hiện nay diễn ra rất phức tạp. Các vụ cháy nổ ở các chung cư, nhà máy xí nghiệp, kho bãi, các chung cư… diễn ra ngày càng nhiều gây thiệt hại nghiêm trọng về tài sản cũng như về tính mạng con người. Vì vậy việc nghiên cứu chế tạo một hệ thống giám sát và tự động cảnh báo hỏa hoạn là hoàn toàn cần thiết trong đời sống.

Với sự phát triển nhanh chóng của công nghệ, nhiều hệ thống đã ra đời để đưa ra cảnh báo sớm cho hỏa hoạn. Có thể kể đến các biết bị cảnh báo cháy sử dụng cảm biến nhiệt độ, cảm biến khói, phát hiện các vụ cháy qua video, hình ảnh từ camera quan trắc. Tuy nhiên phương pháp phân tích video, hình ảnh mang lại nhiều ưu điểm và chính xác hơn.

**2. Mục tiêu của đề tài**

- Xây dựng thuật toán nhận diện đám cháy bằng cách phát hiện lửa nhờ công nghệ Harr Cascade kết hợp với nhận diện lửa nhờ vào màu sắc.

- Phát hiện đám cháy bằng cách nhận diện lửa thông qua video.

- Đưa ra cảnh báo cho con người khi có hỏa hoạn.

**3. Nội dung và kế hoạch thực hiện**

| **Nội dung** | **Kế hoạch thực hiện** |
| --- | --- |
| Lên ý tưởng và tìm nguồn tham khảo | Thành viên trong nhóm sẽ tự nghĩ ý tưởng trong một tuần và tìm ý tưởng từ nhiều nguồn sau đó cùng nhau thống nhất đề tài, làm đề cương chi tiết. |
| Nghiên cứu lý thuyết | Đọc tài liệu, tìm hiểu về thị giác máy tính và nhận diện đối tượng. |
| Phân tích và thiết kế hệ thống điều khiển thiết bị chiếu sáng | - Xây dựng sơ đồ khối, lưu đồ thuật toán.  - Thiết kế chương trình cho  - Lắp ráp hệ thống. |
| Chạy thử sản phẩm | Kiểm tra sản phẩm sau khi hoàn thành. |
| Chuẩn bị bài thuyết trình và báo cáo chi tiết bảo vệ đồ án |  |

4. Bố cục báo cáo

Sau phần *Mở đầu*, báo cáo được trình bày trong ba chương, cụ thể như sau:

Chương 1:*Tổng quan về đề tài.*Đặt ra vấn đề, giới thiệu về các phương pháp để cảnh báo cháy, hỏa hoạn phổ biến hiện nay

Chương 2:*Cơ sở lý thuyết.*Trình bày tổng quan về thị giác máy tính. Đưa ra một số kỹ thuật cơ bản và thường sử dụng trong xử lý ảnh. Giới thiệu về thuật toán chính sử dụng trong đề tài đó là **Harr Cascasde**

Chương 3: *Phân tích và xây hệ thống đèn chiếu sáng*.Phân tích và thiết kế hệ thống điều khiển thiết bị chiếu sáng áp dụng công nghệ Bluetooth mesh

*Kết luận:*Kết luận lại những kiến thức nhận được trong quá trình nghiên cứu. Đưa ra những vấn đề chưa thể hoàn thiện và định hướng phát triển trong tương lai.

# 

# TỔNG QUAN VỀ ĐỀ TÀI

## Đặt vấn đề

Hiện nay trên thế giới cũng như Việt Nam, mỗi năm có hàng nghìn vụ cháy, gây thiệt hại lớn về người và kinh tế. Vì vậy, việc tìm kiếm và phát triển những phương pháp phát hiện sớm các khu vực sắp cháy, cháy nhỏ một cách chính xác, kịp thời thực sự cấp thiết. Hiện tại đã có nhiều nghiên cứu đề xuất những biện pháp phát hiện và cảnh báo cháy như dung các đầu báo cháy nhiệt, đầu báo cháy khói và đầu báo cháy lửa. Hệ thống phát hiện và cảnh báo cháy dựa trên đầu báo cháy truyền thống đã phát huy hiệu quả trong nhiều tình huống, cảnh báo cháy chính xác khi nhiệt độ, khói lan tỏa tới đầu báo cháy và đạt ngưỡng hoạt động, khi đó thường đám cháy đã phát triển lớn. Vì vậy, hệ thống chỉ hiệu quả trong không gian nhỏ và kín (như trong tòa nhà), phát hiện cháy khi đám cháy bùng phát không nhanh, còn với vùng giám sát có không gian mở như các hành lang, phòng không kín, ảnh hưởng gió thì hệ thống hoạt động kém hiệu quả. Việc phát hiện và cảnh báo cháy sử dụng đầu dò cũng còn nhiều hạn chế do phụ thuộc hoàn toàn vào thời gian lan truyền nhiệt và khói kể từ khi bắt đầu xảy ra hiện tượng cháy cho đến khi khói hoặc nhiệt độ lan tỏa tới đầu dò.

Những năm gần đây, một hướng mở ra trong nghiên cứu cảnh báo cháy là sử dụng kỹ thuật xử lý ảnh nhận dạng thông qua hệ thống quan sát camera. Những nghiên cứu ứng dụng kỹ thuật xử lý ảnh, video vào các hệ thống camera cảnh giới với mục đích phát hiện và cảnh báo cháy phát triển mạnh và đã có những kết quả nhất định. Phần lớn các giải pháp được đề xuất cho bài toán phát hiện ngọn lửa sử dụng kỹ thuật xử lý ảnh, video hiện đều dựa trên những tính chất có thể quan sát được của ngọn lửa như màu sắc, sự thay đổi về vị trí các điểm ảnh của ngọn lửa theo thời gian.

Trong bài báo cáo này, nhóm chúng em trình bày về hệ thống nhận diện đám cháy bằng kỹ thuật xử lý ảnh. Chương trình sử dụng kết hợp thuật toán Harr Cascade với kỹ thuật nhận diện ngọn lửa bởi màu sắc.Tổng quan về các phương pháp báo cháy.

## Các phương pháp cảnh báo cháy phổ biến

### Cảm biến nhiệt độ

Cảm biến nhiệt độ thường được sử dụng nhiều trong các môi trường có chất dễ gây cháy nổ. Nó dùng để đo nhiệt độ của môi trường, tùy vào nhiệt độ môi trường mà sử dụng các loại cảm biến nhiệt độ khác nhau. Cụ thể ta thường sử thấy các loại cảm biến nhiệt độ khác nhau. Cụ thể ta thường thấy các loại cảm biến đo nhiệt độ chống cháy nổ ở các khu vực gas, chứa dầu, xăng cũng như các kho bãi, siêu thị, hầm xe và các tòa cao ốc. Các loại cảm biến nhiệt độ thường được dung là: PT100, Can nhiệt loại T,…



*Hình 1.1 Cảm biến nhiệt độ PT100*

### Cảm biến khói

Cảm biến khói là thiết bị dò khói, đầu báo khói là nơi phát ra tín hiệu để đưa đến trung tâm báo cháy. Một số cảm biến khói cũng phát ra âm thanh báo động khi dò được khói. Hiện nay có 2 loại thiết bị cảm biến khói thông dụng là: quang điện và ion hóa. Đầu báo khói ion hóa: đầu báo khói ion sử dụng đồng vị phóng xạ là nguồn phát hạt alpha – α mang tên Americium 241 để có thể tạo ra ion hóa trong không khí. Đầu báo khói ion hóa có độ nhạy cảm rất cao với những đám cháy rực. Khi có một số phần tử khói chui vào buồng ion hóa lúc này bên trong sẽ có mạch phát hiện sự suy giảm dòng điện giữa hai cực và phát đi tín hiệu báo động. Giá thành của đầu báo khói ion hóa rẻ hơn với đầu báo khói quang điện, tuy nhiên dễ xảy ra những tiếng báo động giả và chỉ thích hợp với những đám cháy có hạt khói không nhìn thấy được.



*Hình 1.2 Cảm biến khói*

## Tổng quan về thị giác máy tính

Thị giác máy tính là một ngành khoa học, kỹ thuật đang phát triển mạnh mẽ trên thế giới. Ngành học này phát triển dựa trên bối cảnh của việc máy tính mô phỏng các khả năng thị giác của con người thông qua sự nhận dạng và hiểu biết của các ảnh số và đưa ra các quyết định dựa trên thông tin thu nhận được. Thị giác máy tính có thể được xem như một hướng nghiên cứu của trí tuệ nhân tạo hay khoa học máy tính nói chung và các lĩnh vực liên quan đến vật lý, hình học và thống kê để hỗ trợ giải quyết các vấn đề của một hệ thống nhân tạo. Các ứng dụng của thị giác máy tính rất đa dạng và có nhu cầu không ngừng tăng lên trong nhiều lĩnh vực như y khoa, quốc phòng, an ninh, giáo dục, nông nghiệp, công nghiệp, giao thông, khoa học không gian, giải trí, thực tế tăng cường, sinh trắc học, khoa học nhận dạng,… Xử lý ảnh nhận dạng đám cháy cũng là một ví dụ điển hình trong việc sử dụng thị giác máy tính để cảnh báo hỏa hoạn và giảm thiểu tối đa thiệt hại tài sản cũng như tính mạng con người.

Mục tiêu của thị giác máy tính là thu nhận lại các thông tin hữu dụng về một quang cảnh từ các hình ảnh của quang cảnh đó. Nó phỏng đoán các cấu trúc hay thuộc tính của thế giới ba chiều từ một hoặc nhiều hình ảnh hai chiều, và nhận biết các vật thể thông qua các thuộc tính này. Sự đa dạng của các bề mặt vật thể với các thuộc tính chất liệu và hình học khác nhau, và các điều kiện ánh sáng khác nhau có thể dẫn đến các hình ảnh giống nhau. Để thu nhận lại thông tin về một quang cảnh, nhiều loại kiến thức khác nhau được yêu cầu. Một vài ví dụ về các loại kiến thức này là kiến thức về các vật thể trong quang cảnh, mối quan hệ giữa các vật thể, kiến thức về sự tạo thành ảnh, kiến thức về điều kiện chiếu sáng…

Hệ thống thị giác máy tính có thể được cảm nhận với ba mức: xử lý thị giác ở cấp thấp, xử lý thị giác ở cấp trung và xử lý thị giác ở cấp cao. Xử lý thị giác ở cấp thấp có thể bao gồm thu nhận ảnh, giảm nhiễu ảnh và nâng cao chất lượng ảnh. Xử lý thị giác ở cấp trung thực hiện phân đoạn ảnh, và rút trích đặc trưng. Xử lý thị giác ở cấp cao thực hiện các tác vụ nhận dạng và thông dịch ảnh. Nó so sánh các đặc trưng ảnh dựa vào một mô hình hoặc dung tri thức lĩnh vực theo một cách thích hợp. Xử lý thị giác ở cấp thấp giải quyết tiến trình ở võng mạc, còn xử lý thị giác ở cấp cao giải quyết tiến trình ở não với việc sử dụng tri thức dựa trên cảm nhận.



*Hình 1.3 Ba mức xử lý của một hệ thống thị giác máy tính*

# CƠ SỞ LÝ THUYẾT

## Các bước cơ bản trong xử lý ảnh

Các phương pháp xử lý ảnh bắt đầu từ các ứng dụng chính: nâng cao chất lượng ảnh và phân tích ảnh. Ứng dụng đầu tiên được biết đến là nâng cao chất lượng ảnh báo được truyền qua cáp từ Luân đôn đến New York từ những năm 1920. Vấn đề nâng cao chất lượng ảnh có liên quan tới phân bố mức sáng và độ phân giải của ảnh. Việc nâng cao chất lượng ảnh được phát triển vào khoảng những năm 1955. Điều này có thể giải thích được vì sau thế chiến thứ hai, máy tính phát triển nhanh tạo điều kiện cho quá trình xử lý ảnh số thuận lợi. Năm 1964, máy tính đã có khả năng xử lý và nâng cao chất lượng ảnh từ mặt trăng và vệ tinh Ranger 7 của Mỹ bao gồm: làm nổi đường biên, lưu ảnh. Từ năm 1964 đến nay, các phương tiện xử lý, nâng cao chất lượng, nhận dạng ảnh phát triển không ngừng. Các phương pháp tri thức nhân tạo như mạng nơ ron nhân tạo, các thuật toán xử lý hiện đại và cải tiến, các công cụ nén ảnh ngày càng được áp dụng rộng rãi và thu nhiều kết quả khả quan.

Để dễ tưởng tượng, xét các bước cần thiết trong xử lý ảnh. Đầu tiên, ảnh tự nhiên từ thế giới ngoài được thu nhận qua các thiết bị thu (như Camera, máy chụp ảnh). Trước đây, ảnh thu qua Camera là các ảnh tương tự (loại Camera ống kiểu CCIR). Gần đây, với sự phát triển của công nghệ, ảnh màu hoặc đen trắng được lấy ra từ Camera, sau đó nó được chuyển trực tiếp thành ảnh số tạo thuận lợi cho xử lý tiếp theo. (Máy ảnh số hiện nay là một thí dụ gần gũi). Mặt khác, ảnh cũng có thể tiếp nhận từ vệ tinh; có thể quét từ ảnh chụp bằng máy quét ảnh. Hình dưới đây mô tả các bước cơ bản trong xử lý ảnh.



*Hình 2.1 Các bước cơ bản trong xử lý ảnh*

### Phần thu nhận ảnh

Ảnh có thể nhận qua camera màu hoặc đen trắng. Thường ảnh nhận qua camera là ảnh tương tự (loại camera ống chuẩn CCIR với tần số 1/25, mỗi ảnh 25 dòng), cũng có loại camera đã số hóa (như loại CCD – Change Coupled Device) là loại photodiot tạo cường độ sáng tại mỗi điểm ảnh.

Camera thường dung là loại quét dòng; ảnh tạo ra có dạng hai chiều. Chất lượng một ảnh thu nhận được phụ thuộc vào thiết bị thu, vào môi trường (ánh sáng, phong cảnh).

### Tiền xử lý

Sau bộ thu nhận, ảnh có thể nhiễu độ tương phản thấp nên cần đưa vào bộ tiền xử lý để nâng cao chất lượng. Chức năng chính của bộ tiền xử lý là lọc nhiễu, nâng độ tương phản để làm ảnh rõ nét, nét hơn, biến đổi ảnh thành các dạng khác (ảnh màu sang ảnh xám, ảnh xám sang nhị phân, RGB -> HSV,…).

### Phát hiện

Phát hiện vật thể dùng để tìm vật thể trong ảnh. Sau bước này sẽ biết được có vật thể cần tìm không, nếu có thì ở vị trí nào và số lượng bao nhiêu. Tùy theo loại đối tượng cần tìm mà có các cách khác nhau, như Cascade là 1 cách. Các bài toán đơn giản như tìm màu sắc chỉ cần dùng kênh màu HSV là được.

### Nhận diện

Sau khi đã phát hiện được vật thể và muốn nhận diện vật thể loại nào thì cần bước này. Bước này phải sử dụng Machine Learning để phân lớp (classification) đối tượng. Ví dụ để nhận diện khuôn mặt cần crop khuôn mặt để Machine Learning nhận diện. Bước phân lớp tác dụng là tìm xem giống khuôn mặt đã học nào nhất. Kết quả trả về là khuôn mặt giống nhất kèm tỉ lệ giống.

## Một số kỹ thuật cơ bản trong xử lý ảnh

### Chuyển đổi hệ màu

Nguyên lý của việc cảm nhận màu sắc là do hiện tượng phản xạ ánh sáng. Nguồn sáng tới là ánh sáng trắng (bao gồm các tia sáng đơn sắc với những bước sóng khác nhau từ đỏ, cam, vàng, xanh lá, xanh dương, tím) chiếu vào vật thể cần quan sát. Tia sáng nào phản xạ lại mắt người quan sát thì người quan sát cảm nhận được vật thể có màu đó. Vì vậy màu sắc là vấn đề nhận thức theo cách hiểu chủ quan. Khi mô tả với cùng một màu sắc, mỗi người khác nhau sẽ diễn đạt theo nhiều ý khác nhau, và họ sẽ định nghĩa màu sắc đó theo từ ngữ riêng của mình. Dựa theo quan điểm kỹ thuật, một màu sắc có thể được mô tả bằng ba thuộc tính sau: Sắc màu, giá trị và sắc độ.

Mặc dù hệ màu RBG (Red, Green, Blue) tốt cho việc thu nhận hoặc hiển thị thông tin màu sắc nhưng nó không phải là hệ màu tốt để biểu diễn cho sự cảm nhận màu sắc. Hệ thống nhìn của con người có thể phán đoán tốt về sự phản xạ bề mặt liên quan dù có bước sóng độ chói khác nhau. Sự phản xạ này là những gì chúng ta muốn bàn đến về màu sắc bề mặt.

Đặc trưng quan trọng khác của cơ sở màu thể hiện bởi một khả năng cảm nhận về đen trắng, thông tin cường độ nhận được một cách hiệu quả từ các phép đo lường màu. Từ quan điểm tiến hóa, chúng ta có thể mong rằng sự cảm nhận màu trong loài vật có thể sẽ tương thích với các cơ chế cảm nhận không màu tồn tại từ trước.

Trong OpenCV có hàm cvCVtColor() để chuyển đổi ảnh nhập từ một không gian màu này đến một không gian màu khác. Trong trường hợp chuyển đổi từ hoặc đến không gian màu RGB, thứ tự của các kênh màu phải được chỉ rõ (RBG hay BGR).

Nguyên hàm của cvCVtColor() được mô tả sau đây:

void cvCvt

(

const CvArr\* src,

const CvArr\* dst,

int code

)

Trong đó, các tham số được diễn giải như sau:

+ src: ảnh nguồn với 8bit, 16bit.

+ dst: ảnh đích cùng loại dữ liệu như ảnh nguồn. Số kênh có thể khác nhau

+ code: thao tác chuyển đổi màu có thể được chỉ rõ bằng các hằng số CV\_src\_color\_space2dst\_color\_space, được liệt kê như sau:

* CV\_BGR2GRAY: chuyển đổi ảnh BGR sang ảnh GRAY
* CV\_BGR2YCrCb: chuyển đổi ảnh BGR sang ảnh YCrCb
* CV\_BGR2HLS: chuyển đổi ảnh BGR sang ảnh HLS
* CV\_BGR2HSV: chuyển đổi ảnh BGR sang ảnh HSV
* CV\_BGR2Lab: chuyển đổi ảnh BGR sang ảnh Lab
* CV\_BGR2Luv: chuyển đổi ảnh BGR sang ảnh Luv

### Cân bằng biểu đồ tần số

Cân bằng biểu đồ tần số (Histogram Equalization) là một số kỹ thuật điều chỉnh các cường độ của các điểm ảnh để nâng cao sự tương phản của ảnh. Đây là kỹ thuật được dùng rất rộng rãi trong xử lý ảnh. Kỹ thuật này cải thiện sự tương phản của ảnh mà có thể áp dụng cho toàn bộ ảnh hoặc chỉ một phần của ảnh.

Giả sử f là một ảnh đã được biểu diễn như một ma trận mr dòng, mc cột với các cường độ điểm ảnh nguyên từ 0 đến L-1. Trong đó L là số của các giá trị cường độ dương thường là 256. Và p là biểu đồ tần số đã được chuẩn hóa của f với một cột trên biểu đồ tương ứng cho mỗi cường độ điểm ảnh, như vậy:

pn = , n = 0, 1, …, L – 1

Ảnh đã được cân bằng với biểu đồ tần số g sẽ được định nghĩa bởi công thức:

gij = floor((L – 1))

Trong đó, hàm floor() làm tròn xuống đến số nguyên gần nhất. Điều này tương đương với việc biến đổi các cường độ điểm ảnh k, của f bởi hàm:

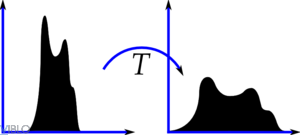
T(k) = floor((L – 1))

Hình sau đây minh họa biểu đồ tần số của một ảnh có kích thước 5x5 pixels với mức xám 8bit (256 giá trị, tức từ 0 đến 255).

| | 40 | 40 | 40 | 40 | 40 | | --- | --- | --- | --- | --- | | 50 | 50 | 50 | 60 | 60 | | 60 | 60 | 60 | 60 | 60 | | 60 | 70 | 70 | 70 | 80 | | 80 | 80 | 90 | 100 | 100 | |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

*Hình 2.2 Biểu đồ tần số của một ảnh có kích thước 5x5 pixels*

Hình sau minh họa nguyên tắc và quá trình cân bằng biểu đồ tần số:



*Hình 2.3 Nguyên tắc cân bằng biểu đồ tần số*

| | Độ tương phản | Giá trị tích lũy | Giá trị chuyển đổi | | --- | --- | --- | | 40 | 5 | 51 | | 50 | 8 | 81 | | 60 | 16 | 163 | | 70 | 19 | 193 | | 80 | 22 | 224 | | 90 | 23 | 234 | | 100 | 25 | 255 | | | 51 | 51 | 51 | 51 | 51 | | --- | --- | --- | --- | --- | | 81 | 81 | 81 | 163 | 163 | | 163 | 163 | 163 | 163 | 163 | | 163 | 193 | 193 | 193 | 224 | | 224 | 224 | 234 | 255 | 255 | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

*Hình 2.4 Quá trình cân bằng biểu đồ tần số*

Hình dưới trình bày ảnh gốc (hình bên trái) và ảnh biểu đồ tần số đã được làm mượt (hình bên phải).



*Hình 2.5 Ảnh biểu đồ tần số được làm mượt*

### Phân đoạn ảnh dùng ngưỡng

Lấy ngưỡng hay chọn ngưỡng là phương pháp biến đổi một ảnh mức xám thành một ảnh nhị phân sao cho các đối tượng quan tâm được tách biệt khỏi nền ảnh. Để việc chọn ngưỡng hiệu quả trong việc phân cách nền và đối tượng thì nền và đối tượng có đủ độ tương phản và chúng ta phải biết mức cường độ sáng hoặc là của nền hoặc là của đối tượng. Trong một số kỹ thuật lấy ngưỡng cố định, các đặc trưng cường độ này xác định giá trị của ngưỡng.

Giả sử rằng một ảnh nhị phân B[i,j] giống như một ảnh xám đã được lấy ngưỡng GT[i,j] bằng cách dùng một ngưỡng T cho ảnh xám gốc G[i,j]. Như vậy ta có công thức:

B[i,j] = GT[i,j]

Trong đó, đối tượng có màu tối hơn trên một nền có màu sáng hơn, và khi đó ta có thể biểu diễn bằng công thức:

GT[i,j] =

Nếu các giá trị cường độ của đối tượng nằm trong dãy [T1,T2] thì chúng ta có thể dùng công thức:

GT[i,j] =

Kỹ thuật lấy ngưỡng tổng quát trong đó các mức cường độ cho một đối tượng có thể gồm một vài khoảng rời nhau được biểu diễn như công thức:

GT[i,j] =

Trong đó, Z là một tập các giá trị cho các thành phần đối tượng.

## Nhận diện đối tượng sử dụng thuật toán Harr cascade

### Nhận diện đối tượng

Nhận dạng đối tượng trong thị giác máy tính là tác vụ tìm hoặc nhận biết các đối tượng trong một ảnh hoặc một chuỗi video. Con người nhận ra vô số đối tượng trong ảnh khá dễ dàng, mặc dù ảnh của các đối tượng có thể biến đổi theo các điểm quan sát khác nhau, với nhiều kích thước và tỉ lệ khác nhau, thậm chí chúng được tịnh tiến hoặc quay với các góc khác nhau. Các đối tượng này còn có thể được nhận biết ngay cả khi chúng bị che khuất một phần từ góc nhìn hay độ sáng của ảnh bị biến đổi theo môi trường xung quanh. Tuy nhiên đối với các hệ thống thị giác máy tính, tác vụ nhận dạng đối tượng vẫn là một thất thức lớn. Lý do mà việc nhận dạng đối tượng rất khó khăn là do hình dạng của vật thể sẽ thay đổi tùy theo góc của camera và vị trí của vật thể. Và nó cũng có thể là do giá trị độ sáng thay đổi tùy thuộc vào ánh sáng của môi trường xung quanh.

Tiến trình nhận dạng đối tượng nói chung được thực hiện theo ba bước. Trước hết là phát hiện đối tượng trong ảnh. Sau đó rút trích đặc trưng vùng đối tượng đã phát hiện. Cuối cùng là so sánh đặc trưng đã rút trích với mô hình có sẵn, tức là bước này được thực hiện thông qua tiến trình đối sánh với các mô hình đã có để xác định đối tượng hay vật thể.

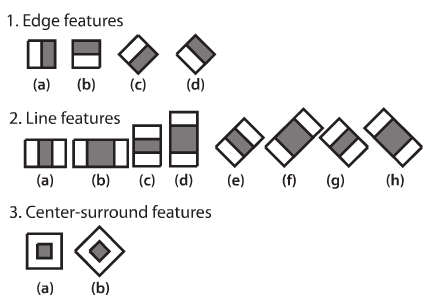
### Adaboost với đặc trưng Harr-like

Adaboots là phương pháp máy học (machine learning) giúp tăng cường khả năng học. Tức là kết hợp các bộ phân lớp yếu (weak classifier) để tạo thành các bộ phân lớp mạnh (strong classifier), một phân lớp sẽ nhận them trọng số hợp lý vào dữ liệu (data) để phân lớp khác có thể được phân lớp những dữ liệu đã bị phân lớp sai một cách chính xác. Đồng thời phương pháp Adaboots nhân trọng số phù hợp vào mỗi bộ phân lớp để tạo thành bộ phân lớp mạnh khi kết hợp các bộ phân lớp yếu. Nói tóm lại, trọng tâm của thuật toán Adaboots chính là quá trình học trọng số nhân vào trong dữ liệu.



*Hình 2.6 Đặc trưng Harr-like cơ bản*

Đặc trưng Harr-like do Viola và Jones đề xuất được sử dụng để xác định khuôn mặt người. Các đặc trưng cơ bản này được mở rộng thành các tập đặc trưng cạnh (edge features), đặc trưng đường (line features) và đặc trưng quanh tâm (center-surround features) đặc thể hiện ở hình bên dưới.



*Hình 2.7 Đặc trưng Harr-like mở rộng*

Phương pháp được thực hiện như sau. Trước hết là sắp đặt các mặt nạ được cấu tạo từ vùng màu đen và vùng màu trắng trong đối tượng. Giá trị tương ứng của các điểm ảnh thuộc vào vùng mặt nạ sẽ được tính bằng cách lấy tổng giá trị của các điểm ảnh thuộc vào vùng màu trắng trừ đi tổng các giá trị điểm ảnh thuộc vùng màu đen của mỗi mặt nạ. Giá trị đó là giá trị đặc trưng của mỗi mặt nạ và nếu giá trị này vượt quá một giá trị ngưỡng nhất định thì nó sẽ trở thành vùng khuôn mặt. Phân loại mỗi mặt nạ thuộc vào trong phân lớp yếu đẻ tạo nên phân lớp mạnh. Quá trình này được biểu thị như trong công thức sau:

F(x) = α1f1(x) + α2f2(x) + …

fi(x) =

Trong đó fi(x) chỉ ra phân lớp yếu của lần thứ i; gi(x) chỉ giá trị của mặt nạ lần thứ i. Tức là khi giá trị của mặt nạ lần thứ i vượt lên giá trị ngưỡng nhất định thì phân lớp yếu lần thứ I sẽ phán đoán vùng hiện hành là vùng khuôn mặt. Giá trị phân lớp mạnh F(x) tổng hợp các phân lớp yếu, nếu giá trị dương thì vùng hiện hành được phán đoán là vùng khuôn mặt.

Để tạo nên bộ phân lớp mạnh thì cần phải học giá trị ngưỡng θi của mỗi bộ phân lớp yếu và học trọng số αi để kết hợp các bộ phân lớp yếu. Có thể sử dụng ảnh tích lũy (integral image) giống như phía dưới đây để tính toán giá trị gi của mỗi mặt nạ trong quá trình tìm ra vùng khuôn mặt. Giá trị điểm ảnh của ảnh tích lũy (x,y) là tổng các giá trị của điểm ảnh gốc nằm ở bên trái và bên phải của tọa độ (x,y). Khi có ảnh tích lũy rồi thì tổng giá trị điểm ảnh nằm trong hình chữ nhật ABCD được tính như sau:

sum = A – B – C + D

| **D** | **B** |  |
| --- | --- | --- |
| **C** | **A** |  |
|  |  |  |

*Hình 2.8 Tìm tổng giá trị điểm ảnh ABCD bằng cách sử dụng ảnh tích lũy*

Quá trình tìm ra tổng các giá trị điểm ảnh nằm trong hình tứ giác sẽ trở nên đơn giản hơn khi sử dụng các ảnh tích lũy này. Do thuật toán Adaboots là thuật toán máy học nên nó cần có dữ liệu huấn luyện (training data) nhằm học giá trị θi αi. Dữ liệu học được biểu thị là (xi,yi) với xi chỉ ảnh học, và yi có giá trị là 1 hoặc -1, mang ý nghĩa ảnh học là ảnh đối tượng hoặc không phải là ảnh đối tượng.

Với mỗi lần “thúc đẩy”, chúng ta:

+ Đánh giá từng mặt nạ hình chữ nhật trên từng dữ liệu huấn luyện (tính g(x)).

+ Sắp xếp dữ liệu huấn luyện theo giá trị các mặt nạ.

+ Chọn ngưỡng tốt nhất với mỗi mặt nạ (nghĩa là chọn trường hợp ít lỗi nhất).

+ Chọn sự kết hợp mặt nạ/ngưỡng tốt nhất với tất cả mặt nạ ứng viên.

+ Tính trọng số và kết hợp các bộ phân lớp yếu để tạo thành bộ phân lớp mạnh F(x) ← F(x) + αf(x).

+ Cập nhật trọng số dữ liệu huấn luyện.

Thuật toán Adaboots thêm trọng số phù hợp vào trong dữ liệu huấn luyện trong quá trình học các bộ phân lớp mạnh và đóng góp vào việc hoàn thiện các điểm còn thiếu sót của bộ phân lớp yếu. Mỗi giai đoạn của việc học này sẽ được giải tích như sau:

Với m bộ dữ liệu huấn luyện (xi,yi), trong đó:

+ xi là ảnh dữ liệu

+ yi = {+1,-1}

Khởi tạo D1(i) = 1/m với mọi i

**for t=1 to T**

Phân lớp dữ liệu huấn luyện ft với giá trị lỗi tối thiểu (minimum error) là εt = PDt(ft(xi) ≠ yi)

Xác định ngưỡng tốt nhất θi

Tính trọng số αi =

Với mỗi dữ liệu huấn luyện i = 1 to m

Dt+1(i) =

**Output**

F(x) = sign

### Cascade of Classifiers:

Ta thấy quá trình huấn luyện, bộ phân loại phải duyệt qua tất cả các đặc trưng của các mẫu trong tập training. Việc này tốn rất nhiều thời gian. 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 phân loại 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.

Cascade of Classifiers được xây dựng chính là nhằm rút ngắn thời gian xử lý, giảm thiểu false alarm cho bộ phân loại. Cascade tree gồm nhiều stage (hay còn gọi là layer), mỗi stage của cây sẽ là một stage classifier. Một mẫu để được phân loại là đối tượng thì nó cần phải đi qua hết tất cả các stages của cây. Các stage classifiers ở stage sau được huấn luyện bằng những mẫu negative mà stage classifier trước nó nhận dạng sai, tức là nó sẽ tập trung học từ các mẫu background khó hơn, do đó sự kết hợp các stage classifiers này lại sẽ giúp bộ phân loại có false alarm thấp. Với cấu trúc này, những mẫu background dễ nhận diện sẽ bị loại ngay từ những stages 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.

Thuật toán Cascade training:

1. Gọi:

***F*:** là giá trị *false* alarm và ***d*** là độ chính xác của weak classifier ở mỗi stage

***Ftarget*:** Giá trị max false alarm.

***P, N*:** là số lượng mẫu positive và negative.

***Pi, Ni*:** là tập positive và negative cho bộ phân lớp ở tầng thứ ***i***.

***Fi, Di*:** Giá trị false alarm và độ chính xác của cascade trước khi đến tầng thứ ***i***.

1. Khởi tạo i=0; F0=1.0; D0 = 1.0
2. Lặp: while ***Fi* > *Ftarget***

* ***i = i+1;***
* Huấn luyện bộ phân loại ***hi*** từ tập ***Pi*** và ***Ni*** với detection rate ***d*** và max false alarm ***f***. Thêm ***hi*** vào cây phân lớp.

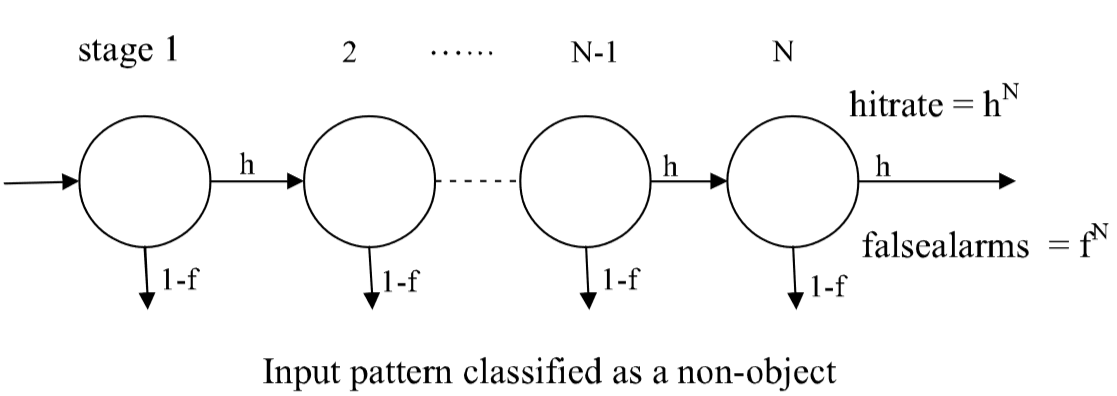
Dùng cây phân lớp hiện có để tính ***Fi***: Duyệt qua ***N*** mẫu negative cho đến khi nào tìm đủ ***n*** mẫu mà cây phân lớp hiện có phân loại sai.

* N := ∅
* Nếu Fi >Ftarget

N = { số mẫu sai ở stage hiện tại phân loại sai }.

P = { số mẫu positive mà stage hiện tại phân loại dúng }

Minh họa thuật toán Cascade training:

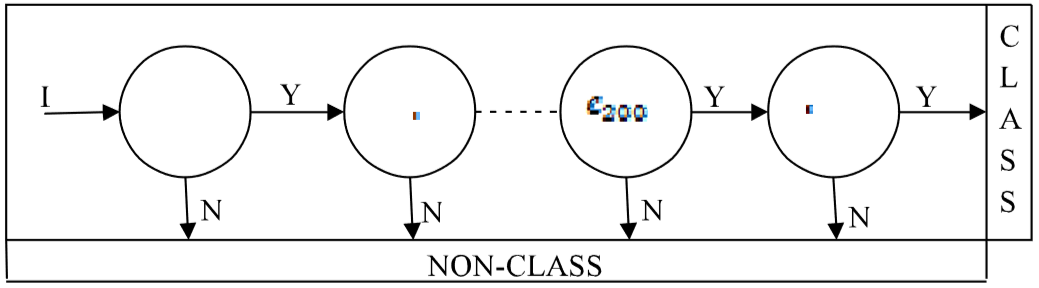


*Hình 2.9 Cascade of classifiers*

Hình trên minh họa sự huấn luyện của một cascade gồm **N** stages. Ở mỗi stage, weak classifier tương ứng sẽ được huấn luyện sao cho độ chính xác của nó là h và false alarm bằng f.

### Cascade of boosting classifiers:

Cascade of boosting classifiers là mô hình cascade of classifiers với mỗi classifier được xây dựng bằng Adaboost sử dụng haar-like.



*Hình 2.10 Cascade of boosting classifiers*

Viola và Jones đã sử dụng rất thành công cascade of boosting classifiers cho bài toán nhận dạng mặt người. Với tập huấn luyện gồm 4196 hình mặt người được đưa về ảnh grayscale kích thước 24x24 và 9500 hình background, hai ôngđã xây dựng cấu trúc cascade tree gồm 38 stage với tổng cộng 6060 đặc trưng haar-like. Thực nghiệm đã cho thấy classifier ở stage đầu tiên sử dụng 2 đặc trưng và loại được khoảng 50% mẫu background (không phải mặt người) và có độ chính xác là d = 100%. Classifier ở stage thứ 2 sử dụng 10 đặc trưng loại được 80% mẫu background với độ chính xác vẫn là 100%. Hệ thống này được so sánh với hệ thống của Rowley-Baluja-Kanade (sử dụng mạng neural), Schneiderman-Kanade (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.

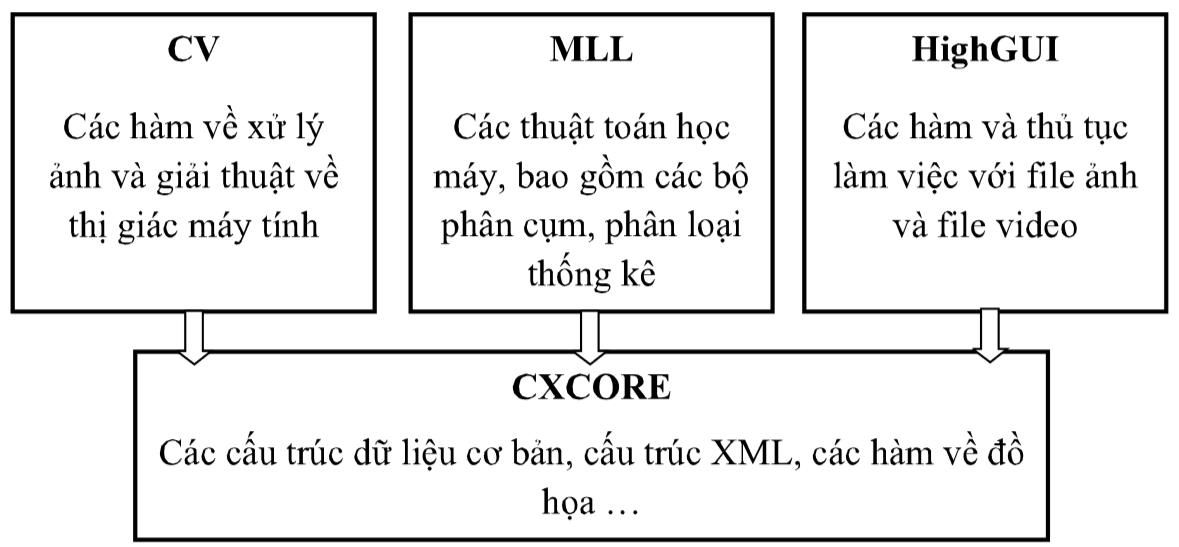
Lý do mà cấu trúc cascade đạt tốc độ nhận dạng nhanh chính là nhờ nó sớm loại bỏ được các mẫu background đơn giản (thường có số lượng lớn hơn nhiều so với các mẫu chứa object – các mẫu chưa khuôn mặt cần tiến hành nhận dạng). Bên cạnh đó, hệ thống của Viola và Jones cũng đạt được độ chính xác khá cao nhờ vào thuật toán cascade training, các bộ nhận dạng được huấn luyện bằng AdaBoost với đặc trưng Haar-like mô tả tốt thông tin đối tượng, cùng với cách Integral Image tính nhanh các đặc trưng, không làm giảm tốc độ nhận dạng của hệ thống. Như vậy, mô hình Cascade of Boosted Classifiers thật sự là một cách tiếp cận tốt cả về tốc độ lẫn khả năng nhận dạng, rất phù hợp với bài nhận dạng mặt người.

# PHÂN TÍCH VÀ XÂY DỰNG HỆ THỐNG

## Tổng quan về openCV

OpenCV là thư viện mã nguồn mở của intel về thị giác máy tính. Nó cung cấp một bộ mã nguồn bao gồm hàng trăm hàm, lớp dựa trên các thuật toán về xử lý ảnh cũng như Computer vision dùng ngôn ngữ Python/C++. Open CV thể hiện sự đa dạng của trí tuệ nhân tạo. Được ứng dụng nhiều trong các bài toán nhận dạng mặt, dò tìm mặt, phát hiện mặt, lọc Kalman, …

Cấu trúc tổng quan của OpenCV bao gồm 5 phần chính. 4 trong 5 phần đó được chỉ ra trong hình vẽ dưới.



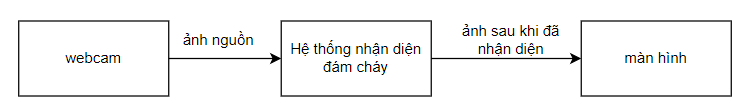
*Hình 3.1 Cấu trúc cơ bản của OpenCV*

Phần CV bao gồm các thư viện cơ bản về xử lý ảnh và các giải thuật về thị giác máy tính. ML là bộ thư viện về các thuật toán học máy, bao gồm rất nhiều bộ phân cụm và phân loại thống kê. HighGUI chứa đựng những thủ tục vào ra, các chức năng về lưu trữ cũng như đọc các file ảnh và video. Phần thứ 4, Cxcore chứa đựng các cấu trúc dữ liệu cơ bản ( ví dụ như cấu trúc XML, các cây dữ liệu …). Phần cuối cùng là CvAux, phần này bao gồm các thư viện cho việc phát hiện, theo dõi và nhận dạng đối tượng (khuôn mặt, mắt …).

## Phân tích

### Chức năng của hệ thông

Nhiệm vụ của chương trình đó là dò tìm đám cháy từ ảnh, video hay webcam, và đưa ra cảnh báo cho người dùng.



*Hình 3.2 Sơ đồ khối hệ thống*

Như vậy chức năng của chương trình bao gồm:

* Kết nối webcam, lấy ảnh đầu vào
* Phát hiện đám cháy
* Đưa ra cảnh báo cho người dùng

### Cách thức hoạt động của hệ thống

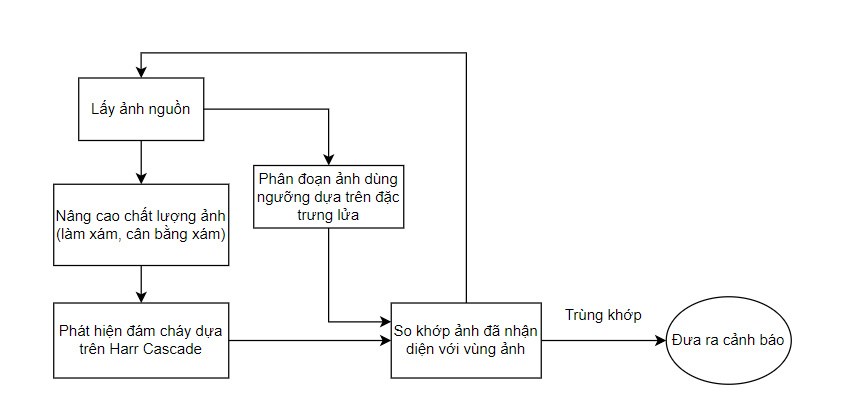
Hệ thống sẽ sử dụng thuật toán Harr Cascade (ở phần cơ sở lý thuyết để nhận diện đám cháy). Tuy nhiên, thuật toán Harr Cascade có những nhược điểm là dễ bị nhận diện nhầm các đối tượng giả - phát hiện đám cháy nhưng không có đám cháy nào. Để khắc phục độ chính xác cho hệ thống, chúng em sử dụng phân vùng ảnh dựa trên trên ngưỡng.

Đặc trưng thứ hai của màu sắc ngọn lửa được rút ra từ thực nghiệm. Đó là các giá trị trên các kênh RGB luôn nằm trong một ngưỡng nhất định:

P(x,y) =

Trong đó, P(x,y) là điểm ảnh (x,y); R(x,y), G(x,y) và B(x,y) lần lượt là giá trị trên kênh R, kênh G và kênh B của điểm ảnh tại tọa độ (x,y). Dựa vào đặc trưng của lửa như ở trên để áp dụng phân vùng đám cháy.

Với cách thức trên, ta có chương trình hệ thống hoạt động như sau:



*Hình 3.3 Khái quát quá trình thực hiện hệ thống*

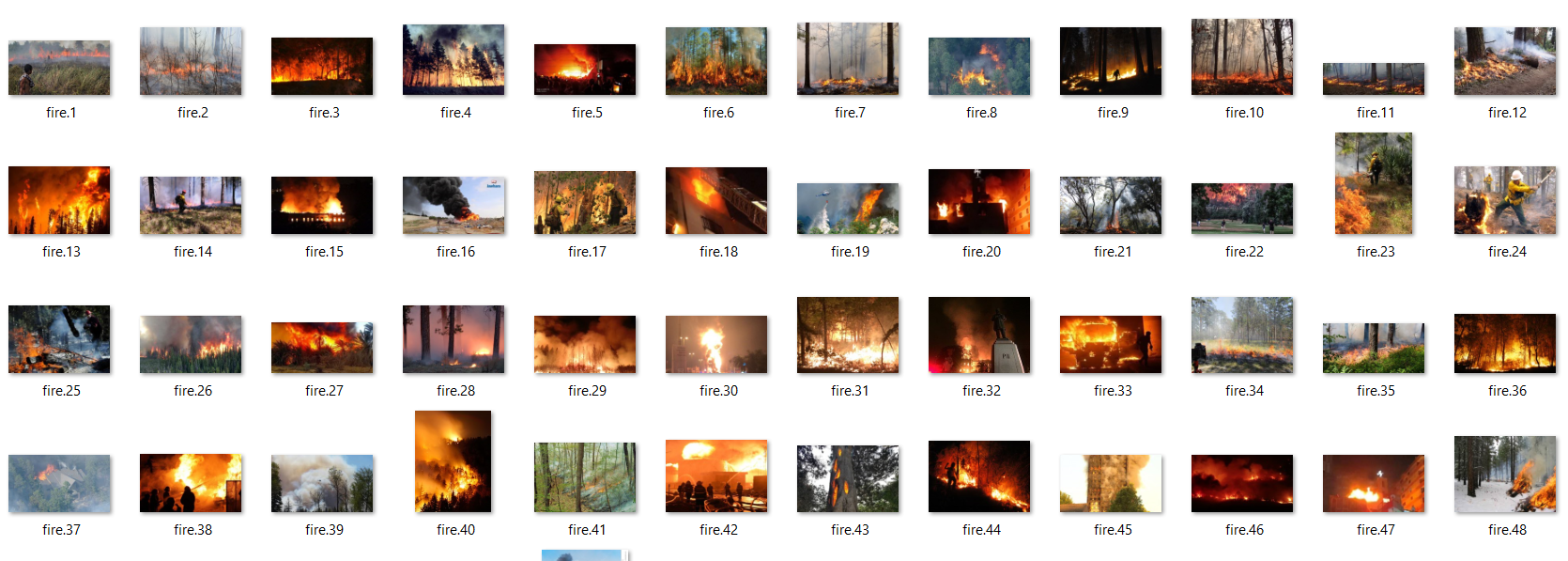
Đâu tiền hệ thống sẽ lấy ảnh vào từ webcam, sau đó tiến hành các bước tiền xử lý trước khi thực hiện thuật toán nhận diện nhằm nâng cấp ảnh. Các bước bao gồm biến đổi ảnh xám và thực hiện cân bằng xám. Tiếp theo sẽ thực hiện nhận diện đám cháy dựa trên thuật toán Harr Cascade cụ thể là sử dụng hàm **detectMultiScale()** thuộc lớp CascadeClassifier có sẵn trong OpenCV để nhận diện đối tượng dựa vào bộ phân lớp được lưu trong file .xml. Bởi vì độ chính xác của thuật toán Harr Cascade chưa cao nên hệ thống cũng sẽ tiến hành phân vùng ảnh có màu sắc phù hợp với đặc trưng màu sắc của lửa. Bước cuối cùng sẽ tiến hành so khớp ảnh đã nhận diện với ảnh đã phân vùng, đưa ra cảnh báo cho người dùng nếu đúng là có hỏa hoạn.

## Xây dựng hệ thống

### Xây dựng tập tin để huấn luyện

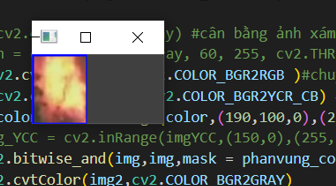
Chuẩn bị 2 bộ ảnh, một bộ ảnh có chứa đối tượng gọi là ảnh dương. Bộ còn lại là những ảnh bất kỳ không chứa đối tượng gọi là ảnh âm. Số lượng ảnh âm và dương càng nhiều thì độ chính xác càng cao và thời gian huấn luyện càng lâu. Khi training, chương trình sẽ rút ra đặc trưng rồi so sánh với các ảnh âm, nếu đặc trưng đó được tìm thấy trong ảnh âm thì đặc trưng đó không phù hợp. Cần phải chỉ định rõ vị trí hình chữ nhật chứa đối tượng cần tìm và lưu vào file chỉ mục.

Sử dụng OpenCV để tạo tập tin XML bằng cách dùng lệnh create\_samples của OpenCV để tạo các tệp mẫu.

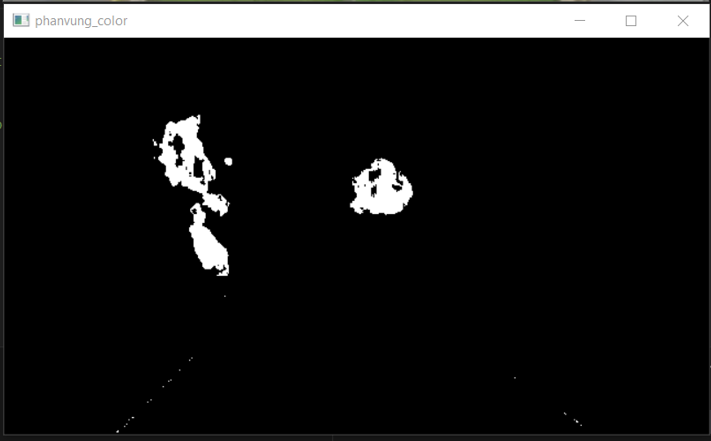


*Hình 3.4 Dữ liệu dùng để huấn luyện*

### Kết quả sau khi chạy chương trình



*Hình 3.5 Lửa được nhận diện*



*Hình 3.6 Lửa được tách ra khi sử dùng phân vùng ảnh*



*Hình 3.7 Cảnh báo có đám cháy*

# KẾT LUẬN VÀ HƯỚNG NGHIÊN CỨU

1. **Kết quả**

* Áp dụng thành công thuật toán Harr Cascadevào bài toán phát hiện đám cháy, nhận diện được real time camera.
* Hiểu thêm về khái quát xử lý ảnh.
* Hiểu thêm về bài toán xử lý nhận dạng đối tượng.
* Học hỏi thêm về ngôn ngữ lập trình Python và thư viện Opencv.

1. **Hướng nghiên cứu**

Hoàn thiện về phần giao diện. Tối ưu hóa việc sử dụng của người dùng và ứng dụng. Sử dụng các phương pháp học dữ liệu nhận diện khuôn mặt này để áp dụng vào điểm danh ở các nơi công sở, trường học để thuận lợi cho việc quản lý nhân sự.

**TÀI LIỆU THAM KHẢO**

| [1] | Nhà xuất bản truyền thông , "Thị giác máy tính" |
| --- | --- |
| [2] | Phạm Ngọc Pha , Nguyễn Trọng Hiếu , Nguyễn Việt Thắng , Nguyễn Trường Sơn , Quách Công Hoàng , Phạm Minh Triển, "THIẾT KẾ, CHẾ TẠO HỆ THỐNG CẢNH BÁO SỚM ĐÁM CHÁY TRONG TÒA NHÀ CAO TẦNG SỬ DỤNG CÔNG NGHỆ HỌC MÁY" |
| [3] | Nguyễn Hữu Phát, Trần Hoàng Vũ, "ĐỀ XUẤT MỘT THUẬT TOÁN PHÁT HIỆN CHÁY RỪNG DỰA TRÊN KHÓI/LỬA TRONG HÌNH ẢNH". |
| [4] | T. Celik, H. Demirel, H. Ozkaramanli, and M. Uyguroglu, “Fire detection using statistical color model in video sequences,” J. Visual Commun. and Image Representation, 2007. |