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

**-----\*\*\*-----**

****

**BÁO CÁO BÀI TẬP LỚN**

**HỌC PHẦN “THỊ GIÁC MÁY TÍNH”**

***Đề tài:***

***SELFIE CAPTURE: “XÂY DỰNG CHƯƠNG TRÌNH TỰ ĐỘNG CHỤP ẢNH KHI PHÁT HIỆN NGƯỜI TRONG ẢNH MỈM CƯỜI”***

***GVHD: TS. Nguyễn Hữu Tuân***

***Sinh viên thực hiện: Vũ Thị Mùi– Mã SV: 93917***

**HẢI PHÒNG 20****24**

**LỜI CẢM ƠN**

Trong thời gian nghiên cứu tìm hiểu đề tài bài tập lớn tuy gặp nhiều khó khăn trong việc nghiên cứu, nhưng nhờ sự giúp đỡ, giảng dạy tận tình của thầy ở các tiết học trên lớp, em đã hoàn thành được đề tài theo đúng tiến độ giao.

Lời đầu tiên cho em gửi lời cảm ơn đến Thầy TS. Nguyễn Hữu Tuân người đã trực tiếp hướng dẫn, giảng dạy về kiến thức, phương pháp để em có thể hoàn thành thành công đề tài bài tập lớn của học phần Thị giác máy tính.

Mặc dù đã có nhiều cố gắng trong suốt quá trình thực hiện đề tài, song có thể còn những mặt hạn chế, thiếu sót. Em mong nhận được ý kiến đóng góp và sự chỉ dẫn của các thầy– người có chuyên môn và kiến thức sâu rộng để em có thể rút kinh nghiệm hoàn thành tốt hơn các đề tài sau này.

**TRƯỜNG ĐẠI HỌC HÀNG HẢI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BỘ MÔN KHOA HỌC MÁY TÍNH**

**-----\*\*\*-----**

**BÀI TẬP LỚN**

**HỌC PHẦN: THỊ GIÁC MÁY TÍNH**

1. **Tên đề tài**

Đề tài: **“*Xây dựng chương trình tự động chụp ảnh khi phát hiện người trong ảnh mỉm cười”***

1. **Mục đích**

Xây dựng chương trình tự đông chụp ảnh khi phát hiện người trong ảnh mỉm cười nhằm ứng dụng những kiến thức đã học của học phần.

1. **Công việc cần thực hiện**

* Xây dựng chương trình tự động chụp ảnh khi phát hiện người trong ảnh mìm cười
* Làm báo cáo bài tập lớn của học phần
* Bảo vệ bài tập lớn

1. **Yêu cầu**

* Kết quả làm bài tập lớn: Một chương trình tự động chụp ảnh khi phát hiện người trong ảnh mỉm cười
* Báo cáo bài tập lớn phải được trình bày theo mẫu quy định (kèm theo), báo cáo có thể kết xuất thành tệp định dạng PDF và nộp qua email (không bắt buộc phải in ấn)
* Hạn nộp báo cáo bài tập lớn: …/12/2024.

***Hải Phòng, tháng năm 2024***

**NGƯỜI HƯỚNG DẪN**

MỤC LỤC

[CHƯƠNG 1. GIỚI THIỆU 1](#_Toc183823330)

[1.1. Bài toán 1](#_Toc183823331)

[1.2. Mục tiêu của đề tài 1](#_Toc183823332)

[1.3. Giới hạn đề tài 2](#_Toc183823333)

[1.4. Phương pháp nghiên cứu 2](#_Toc183823334)

[CHƯƠNG 2: SỬ DỤNG THUẬT TOÁN HAAR CASCADE TRONG OPENCV CHO BÀI TOÁN PHÁT HIỆN NGƯỜI TRONG ẢNH CƯỜI. 4](#_Toc183823335)

[2.1. Tổng quan 4](#_Toc183823336)

[*2.1.1. Nhận dạng phát hiện đối tượng* 4](#_Toc183823337)

[*2.1.2. Nhận diện khuôn mặt người* 4](#_Toc183823338)

[*2.1.2. Khái quát quy trình chụp ảnh khi phát hiện người mỉm cười* 6](#_Toc183823339)

[2.2. Phương pháp nhận diện khuôn mặt sử dụng đặc trưng Haar-Like 10](#_Toc183823340)

[*2.2.1. Đăc trưng Haar-Like* 10](#_Toc183823341)

[2.3. *Nhận diện người trong ảnh mỉm cười sau khi phát hiện khuôn mặt* 16](#_Toc183823342)

[*2.4. Cơ chế phát hiện nụ cười thời gian thực sử dụng OpenCV* 19](#_Toc183823343)

[CHƯƠNG 3: XÂY DỰNG CHƯƠNG TRÌNH TỰ ĐỘNG CHỤP ẢNH KHI PHÁT HIỆN NGƯỜI TRONG ẢNH MỈM CƯỜI 21](#_Toc183823344)

[3.1. Môi trường phát triển 21](#_Toc183823345)

[*3.1.1. Ngôn ngữ lập trình* 21](#_Toc183823346)

[*3.1.2. Thư viện sử dụng và cài đặt* 21](#_Toc183823347)

[3.2. Quá trình thực hiện 25](#_Toc183823348)

[*3.2.1. Mô tả các bước thực hiện* 25](#_Toc183823349)

[3.4. Kết quả 25](#_Toc183823350)

[KẾT LUẬN 29](#_Toc183823351)

[4.1. Kết luận 29](#_Toc183823352)

[*4.1.1 Ưu điểm* 29](#_Toc183823353)

[*4.1.2 Nhược điểm* 29](#_Toc183823354)

[*4.2. Định hướng tương lai* 30](#_Toc183823355)

[TÀI LIỆU THAM KHẢO 31](#_Toc183823356)

**DANH MỤC HÌNH**

|  |  |  |
| --- | --- | --- |
| **[Hình](#_Toc163552185)** | **[Tên hình](#_Toc163552185)** | **[Trang](#_Toc163552185)** |
| [1.1](#_Toc163552185) | [Mô hình minh họa nhận diện người trong ảnh mỉm cười](#_Toc163552185) | [1](#_Toc163552185) |
| [2.1](#_Toc163552185) | [Các mẫu phát hiện yếu tố khuôn mặt cho chuỗi được trích xuất](#_Toc163552185) | [5](#_Toc163552185) |
| [2.2](#_Toc163552185) | [Quy trình AdaBoost cho các máy dò đối tượng khác nhau](#_Toc163552185) | [6](#_Toc163552185) |
| [2.3](#_Toc163552185) | [Sơ đồ hệ thống chương trình tự động chụp ảnh khi phát hiện người trong ảnh mỉm cười](#_Toc163552185) | [8](#_Toc163552185) |
| [2.4](#_Toc163552185) | [Sơ đồ hình ảnh hệ thống chương trình](#_Toc163552185) | [8](#_Toc163552185) |
| [2.5](#_Toc163552185) | [Các đặc trưng Haar-Like](#_Toc163552185) | [9](#_Toc163552185) |
| [2.6](#_Toc163552185) | [Bốn đặc trưng Haar-Like cơ bản](#_Toc163552185) | [9](#_Toc163552185) |
| [2.7](#_Toc163552185) | [Mô hình phân tầng cascade](#_Toc163552185) | [11](#_Toc163552185) |
| [2.8](#_Toc163552185) | [Hàng đầu tiên hiển thị hình ảnh từ cơ sở dữ liệu hình ảnh trái tim và khuôn mặt phía trước của con người. Hàng thứ hai chứa hình ảnh điển hình từ cơ sở dữ liệu lộn xộn.](#_Toc163552185) | [12](#_Toc163552185) |
| [2.9](#_Toc163552185) | [Phân bố các điểm đối tượng cho các tính năng Haar-Like](#_Toc163552185) | [12](#_Toc163552185) |
| [2.10](#_Toc163552185) | [Phân phối các điểm lộn xộn cho các tính năng Haar khác](#_Toc163552185) | [13](#_Toc163552185) |
| [2.11](#_Toc163552185) | [Chuẩn hóa cường độ](#_Toc163552185) | [14](#_Toc163552185) |
| [2.12](#_Toc163552185) | [Sơ đồ nhận diện khuôn mặt](#_Toc163552185) | [14](#_Toc163552185) |
| [2.13](#_Toc163552185) | [Đặc trưng Haar-Like](#_Toc163552185) | [15](#_Toc163552185) |
| [2.14](#_Toc163552185) | [Quá trình training nhận dạng phát hiện người mỉm cười](#_Toc163552185) | [16](#_Toc163552185) |
| [2.15](#_Toc163552185) | [Cơ chế hệ thống nhận dạng người trong ảnh mỉm cười sử dụng thời gian thực](#_Toc163552185) | [17](#_Toc163552185) |
| [3.1](#_Toc163552185) | [Hình ảnh minh họa ngôn ngữ Python](#_Toc163552185) | [19](#_Toc163552185) |
| [3.2](#_Toc163552185) | [Hình ảnh minh họa thư viện OpenCV](#_Toc163552185) | [20](#_Toc163552185) |
| [3.3](#_Toc163552185) | [Hình ảnh minh họa thư viện Tkinter trong Python](#_Toc163552185) | [21](#_Toc163552185) |
| [3.4](#_Toc163552185) | [Thư viện Pillow](#_Toc163552185) | [21](#_Toc163552185) |
| [3.5](#_Toc163552185) | [Cách chạy file chương trình trên cmd](#_Toc163552185) | [23](#_Toc163552185) |
| [3.6](#_Toc163552185) | [Hai Button chọn file và mở cam để nhận diện](#_Toc163552185) | [24](#_Toc163552185) |
| [3.7](#_Toc163552185) | [Nút tạm dừng và xem ảnh](#_Toc163552185) | [24](#_Toc163552185) |
| [3.8](#_Toc163552185) | [Dữ liệu video test đầu vào](#_Toc163552185) | [24](#_Toc163552185) |
| [3.9](#_Toc163552185) | [Kết quả nhận diện từ video](#_Toc163552185) | [25](#_Toc163552185) |
| [3.10](#_Toc163552185) | [Kết quả nhận diện từ video](#_Toc163552185) | [25](#_Toc163552185) |
| [3.11](#_Toc163552185) | [Kết quả nhận diện từ webcam](#_Toc163552185) | [26](#_Toc163552185) |

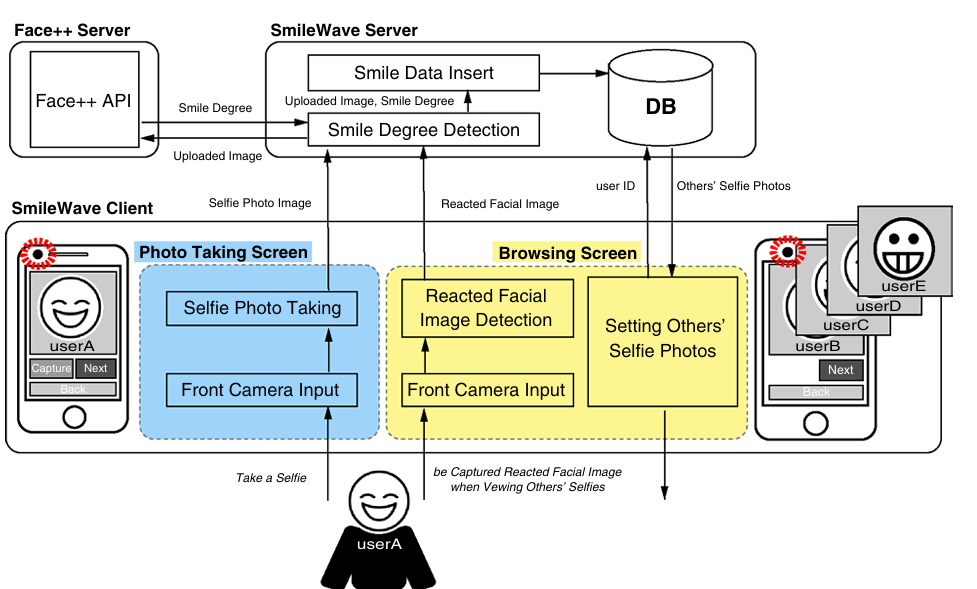
**DANH MỤC CÁC TỪ VIẾT TẮT**

|  |  |
| --- | --- |
| **[Từ](#_Toc163552185)** | **[Ý nghĩa](#_Toc163552185)** |
| [SSD](#_Toc163552185) | [Single Shot MultiBox Detector](#_Toc163552185) |
| [BFS](#_Toc163552185) | [Breadth-First Search](#_Toc163552185) |
| [GA](#_Toc163552185) | [Genetic Algorithm](#_Toc163552185) |
| [FLDA](#_Toc163552185) | [Fisher Linear Discriminant Analysis](#_Toc163552185) |
| [CNN](#_Toc163552185) | [Convolutional Neural Network](#_Toc163552185) |
| [YOLO](#_Toc163552185) | [You Only Look Once](#_Toc163552185) |
| [OpenCV](#_Toc163552185) | [Open Source Computer Vision Library](#_Toc163552185) |
| [PIL](#_Toc163552185) | [Python Imaging Library](#_Toc163552185) |

# CHƯƠNG 1. GIỚI THIỆU

## 1.1. Bài toán

Trong thời đại hiện nay, nhu cầu ghi lại những khoảnh khắc tự nhiên, vui vẻ của con người, đặc biệt là trong các sự kiện xã hội và không gian giải trí, ngày càng tăng cao. Tuy nhiên, việc chụp ảnh thủ công thường đòi hỏi sự tham gia của người dùng, gây gián đoạn và đôi khi bỏ lỡ những khoảnh khắc quan trọng. Để giải quyết vấn đề này và nâng cao trải nghiệm người dùng, công nghệ nhận diện khuôn mặt và cảm xúc đã được phát triển mạnh mẽ.

**

*Hình 1.1. Mô hình minh họa nhận diện người trong ảnh mỉm cười*

Một trong những ứng dụng nổi bật của công nghệ này là hệ thống tự động chụp ảnh khi phát hiện nụ cười trên khuôn mặt. Với sự hỗ trợ của các mô hình nhận diện nụ cười thông minh, hệ thống có khả năng xác định chính xác khi nào người dùng mỉm cười và tự động chụp ảnh ngay lập tức, ghi lại những khoảnh khắc chân thật nhất.

Hệ thống này không chỉ giảm thiểu công sức và thời gian của người dùng mà còn loại bỏ sự cần thiết phải nhấn nút chụp thủ công, giúp tạo ra trải nghiệm chụp ảnh tự nhiên và tiện lợi hơn trong nhiều bối cảnh khác nhau, từ các buổi tiệc đến các khu vực chụp ảnh tự động trong trung tâm thương mại.

## 1.2. Mục tiêu của đề tài

Trong đề tài này, em đã đề xuất một hệ thống phát hiện và nhận diện nụ cười trong thời gian thực từ hình ảnh và video. Hệ thống có khả năng nhận diện nụ cười qua webcam hoặc từ các video đầu vào, cụ thể là phát hiện khuôn mặt và nhận diện nụ cười của người trong khung hình. Em đã sử dụng thư viện OpenCV để xử lý và phát hiện khuôn mặt cùng nụ cười trong ảnh. Cụ thể, em sử dụng mô hình CascadeClassifier của OpenCV để phát hiện các vùng khuôn mặt và sau đó là các đặc điểm của nụ cười trong ảnh đã được xử lý.Từ đó giúp:

Nắm vững cơ sở lý thuyết và sử dụng các bộ phân loại Cascade cho xử lý ảnh: giúp em hiểu rõ về cách hoạt động của các bộ phân loại Haar Cascade – một phương pháp học máy để phát hiện đối tượng trong ảnh. Em đã áp dụng bộ phân loại Cascade để phát hiện khuôn mặt và nụ cười từ các khung hình, từ đó nhận diện nụ cười của người dùng.

Khám phá phương pháp phát hiện đối tượng thời gian thực: em đã tích hợp các hàm xử lý thời gian thực trong Python với giao diện Tkinter, cho phép hiển thị video trực tiếp và kết quả phát hiện nụ cười. Hệ thống không chỉ xử lý và phân tích ảnh từ webcam mà còn từ các video đầu vào với kích thước cố định.

Thực hiện các thử nghiệm và chạy mô hình trên môi trường giao diện người dùng: em đã xây dựng giao diện với Tkinter để dễ dàng tương tác và thử nghiệm. Các nút lệnh cho phép chọn video đầu vào, mở webcam, và dừng quá trình nhận diện. Hệ thống còn tự động lưu lại các ảnh khi nụ cười được phát hiện, giúp lưu trữ kết quả.

Sử dụng các thư viện hỗ trợ để thực thi mô hình: em đã tìm hiểu về các thư viện hỗ trợ như OpenCV và Tkinter để xây dựng giao diện người dùng và thực thi quá trình xử lý video. Em cũng làm quen với các thao tác quản lý tệp và lưu ảnh tự động, giúp tăng hiệu quả lưu trữ dữ liệu đầu ra cho hệ thống.

## 1.3. Giới hạn đề tài

Trong đề tài là chương trình tự động chụp ảnh khi phát hiện cười ở đây bài toán cụ thể là khi cười tươi (hay là cười nhe răng) và trong điều kiện ánh sáng thỏa mãn (không quá yếu hoặc sáng quá, sáng không đều). Hệ thống chỉ dừng lại ở việc nghiên cứu và giới hạn kiến thức, chưa thể áp dụng ra thị trường.

## 1.4. Phương pháp nghiên cứu

* Thu thập tài liệu và tham khảo ứng dụng liên quan đã có trước đó:
* Điều này giúp em hiểu rõ hơn về các thuật toán phát hiện khuôn mặt và nụ cười, đồng thời nắm bắt các ứng dụng hiện có trong lĩnh vực nhận diện cảm xúc từ khuôn mặt.
* Tìm hiểu về các công trình nghiên cứu, bài báo khoa học và ứng dụng đã sử dụng các phương pháp nhận diện khuôn mặt và nụ cười trong các môi trường thực tế như an ninh, chăm sóc sức khỏe hoặc các ứng dụng giải trí.
* Đọc thêm tài liệu và tra cứu trên mạng:
* Tìm hiểu về các công nghệ nhận diện khuôn mặt và nụ cười hiện đại, như sử dụng thuật toán Haar Cascade trong OpenCV và các kỹ thuật học sâu như CNN và YOLO cho nhận diện khuôn mặt và cảm xúc.
* Tra cứu các tài liệu, bài báo và tài liệu học tập về xử lý ảnh, nhận diện khuôn mặt, nụ cười và các hệ thống tự động chụp ảnh
* Thực hiện thử nghiệm và áp dụng thuật toán nhận diện nụ cười:
* Sử dụng các thư viện hỗ trợ như OpenCV để phát hiện khuôn mặt và nhận diện nụ cười trong các hình ảnh và video thử nghiệm.
* Điều chỉnh các tham số của thuật toán Haar Cascade để tối ưu hóa việc phát hiện khuôn mặt và nụ cười trong các điều kiện ánh sáng khác nhau và với các đối tượng có nhiều góc độ.
* Tạo ra một hệ thống tự động chụp ảnh khi phát hiện nụ cười, đồng thời lưu ảnh vào thư mục với tên độc nhất.
* Tìm hiểu về cách xử lý ảnh và cải tiến hệ thống nhận diện:
* Thực hiện các thử nghiệm với các video từ webcam và video tệp để kiểm tra khả năng nhận diện nụ cười trong thời gian thực.
* Hiểu về cách cải thiện độ chính xác của hệ thống nhận diện nụ cười qua webcam và video bằng cách sử dụng các thuật toán cải tiến và thử nghiệm với các bộ dữ liệu khác nhau.

# CHƯƠNG 2: SỬ DỤNG THUẬT TOÁN HAAR CASCADE TRONG OPENCV CHO BÀI TOÁN PHÁT HIỆN NGƯỜI TRONG ẢNH CƯỜI.

## 2.1. Tổng quan

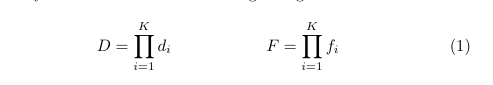
### *2.1.1. Nhận dạng phát hiện đối tượng*

Nhận dạng đối tượng nhằm mục đích tìm các tiểu vùng của hình ảnh chứa các trường hợp của một đối tượng quan tâm. Nhiều ứng dụng phát hiện đối tượng là thách thức vì độ chính xác cao được yêu cầu trong khi hình ảnh được đánh giá ở tốc độ thời gian thực. Các ứng dụng như vậy bao gồm phát hiện phương tiện [1], trong đó người ta cần cảnh báo người lái xe ô tô về các tai nạn có thể xảy ra càng sớm càng tốt và các nhiệm vụ giám sát trong đó mọi khung hình video cần được kiểm tra trong thời gian thực để biết sự hiện diện của những kẻ xâm nhập [2].

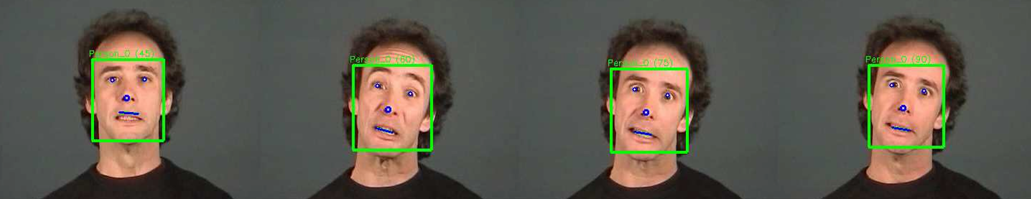
Các bộ phân loại dựa trên các tính năng Haar-Like [3] đã được chứng minh là thành công đáng kể để sử dụng trong các nhiệm vụ phát hiện đối tượng. Điều này chủ yếu là do thực tế là chúng cung cấp một sự đánh đổi hấp dẫn giữa độ chính xác và tốc độ đánh giá. Viola và Jones đã đề xuất một hệ thống phát hiện đối tượng phổ biến dựa trên các tính năng này. Sau sự đóng góp của Viola và Jones, nhiều hệ thống thành công dựa trên các tính năng Haar-Like đã được đề xuất [4,5,6,7]. Trong đề tài này em đề xuất các tính năng giống Haar để phát hiện đâu là người .Tương tự như các tính năng Haar-Like truyền thống, các tính năng được đề xuất có thể được sử dụng để tạo thành các bộ phân loại yếu, do đó có thể được tăng cường và được sắp xếp theo kiến trúc tầng loại bỏ để tạo thành một hệ thống phát hiện đối tượng.

### *2.1.2. Nhận diện khuôn mặt người*

Hệ thống phát hiện khuôn mặt được sử dụng trong đề tài này tích hợp, trong số các tín hiệu khác, các phân loại khác nhau dựa trên khung phát hiện đối tượng chung của Viola và Jones , màu da, theo dõi đa cấp, v.v. Máy dò đối tượng Viola-Jones là một loạt các phân loại. Mỗi bộ phân loại sử dụng một tập hợp các tính năng của Haar. Các bộ phân loại yếu: mỗi bộ phân loại có tỷ lệ phát hiện rất cao, với tỷ lệ loại bỏ thực sự nhỏ. Bằng cách này, chúng hoạt động như một chuỗi lọc. Chỉ những vùng hình ảnh quản lý để đi qua tất cả các giai đoạn của máy dò mới được coi là có chứa khuôn mặt, xem [8]. Đối với một loạt các bộ phân loại K, tỷ lệ phát hiện kết quả, D và tỷ lệ dương tính giả, F, của tầng được cho bởi sự kết hợp của từng tỷ lệ phân loại giai đoạn đơn:



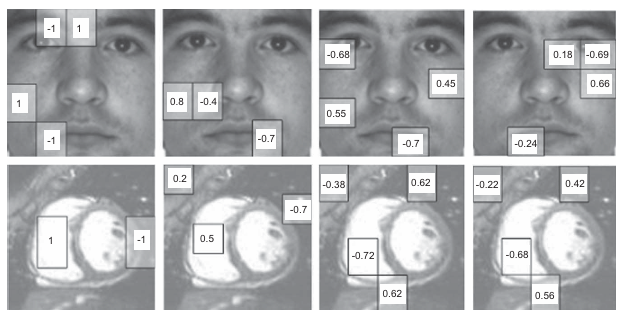
Mặt khác, khung này cho phép tốc độ xử lý hình ảnh cao, do thực tế là các vùng nền của hình ảnh nhanh chóng bị loại bỏ trong khi dành nhiều thời gian hơn cho các vùng giống như khuôn mặt đầy hứa hẹn. Để giảm thiểu hơn nữa ảnh hưởng của báo động sai, khả năng phát hiện đặc điểm khuôn mặt đã được mở rộng, định vị không chỉ khuôn mặt mà còn cả mắt, mũi và miệng. Điều này làm giảm số lượng báo động sai, vì ít có khả năng nhiều máy dò, tức là khuôn mặt và các tính năng bên trong của nó, được kích hoạt đồng thời với báo động sai.



Hình 2.1. Các mẫu phát hiện yếu tố khuôn mặt cho chuỗi được trích xuất

Các mẫu dương tính cho các bộ đào tạo các đặc điểm bên trong đã thu được bằng cách ghi chú thủ công các vị trí mắt, mũi và miệng trong 7000 hình ảnh khuôn mặt được lấy ngẫu nhiên từ Internet. Các hình ảnh sau đó đã được chuẩn hóa bằng thông tin mắt đến 59×65 pixel. Năm máy dò khác nhau đã được tính toán: 1-2) Mắt trái và mắt phải (18× 12 pixel), 3) cặp mắt (22 × 5), 4) mũi (22× 15) và 5) miệng (22×15). Các máy dò này đã được công bố công khai. Quy trình phát hiện yếu tố khuôn mặt chỉ được áp dụng ở những khu vực có bằng chứng chứa khuôn mặt. Điều này đúng với các vùng trong khung hình hiện tại, nơi khuôn mặt đã được phát hiện hoặc ở những khu vực có khuôn mặt được phát hiện trong khung hình trước đó. Để xử lý luồng video, với diện tích ước tính cho từng tính năng bên trong, các ứng cử viên được tìm kiếm ở những khu vực đó không chỉ bằng máy dò đặc điểm khuôn mặt dựa trên Viola Jones mà còn bằng cách theo dõi các yếu tố khuôn mặt trước đó bằng SSD. Khi tất cả các ứng cử viên đã thu được, khả năng dựa trên các vị trí chuẩn hóa cho mũi và miệng được tính toán cho mỗi kết hợp, chọn một vị trí có khả năng cao nhất.

Đối với mỗi vấn đề phát hiện đối tượng bốn máy dò đã được xây dựng. Bốn máy dò này về cơ bản khác nhau về loại tính năng được sử dụng để xây dựng chúng. Các tính năng được sử dụng là:



Hình 2.2. Quy trình AdaBoost cho các máy dò đối tượng khác nhau

Minh họa các tính năng Haar-Like, chồng lên một hình ảnh đào tạo điển hình, được chọn trong lần lặp đầu tiên của quy trình AdaBoost cho các máy dò đối tượng khác nhau. Các cột, từ trái sang phải, hiển thị các tính năng được chọn bằng các chiến lược trọng số hình chữ nhật khác nhau: trọng số mặc định, trọng số tối ưu (BFS), trọng số tối ưu (GA) và trọng số tối ưu (FLDA), tương ứng

### *2.1.2. Khái quát quy trình chụp ảnh khi phát hiện người mỉm cười*

Hệ thống sử dụng thư viện OpenCV với thuật toán HaarCascade để phát hiện và nhận diện khuôn mặt cũng như nụ cười trong video hoặc luồng webcam thời gian thực, kết hợp giao diện người dùng với Tkinter để quản lý và hiển thị kết quả.

Lý do chọn thuật toán Haar Cascade thay vì các thuật toán khác

Haar Cascade là một thuật toán đã được tích hợp sẵn trong thư viện OpenCV, rất dễ sử dụng và không đòi hỏi quá nhiều kiến thức về lập trình học máy hoặc xử lý ảnh nâng cao.Chỉ cần một mô hình huấn luyện sẵn (định dạng .xml) là có thể áp dụng trực tiếp mà không cần huấn luyện từ đầu.

Đối với các bài toán nhận diện đơn giản như phát hiện khuôn mặt hoặc nụ cười trong môi trường có ánh sáng tốt, Haar Cascade hoạt động nhanh và hiệu quả mà không cần phần cứng mạnh.

Haar Cascade sử dụng các đặc trưng Haar, kết hợp với thuật toán "cascade classifier", giúp xử lý dữ liệu đầu vào một cách nhanh chóng và tiết kiệm tài nguyên, phù hợp với hệ thống xử lý thời gian thực. Khác với các thuật toán phức tạp như YOLO, SSD, hoặc CNN, Haar Cascade không yêu cầu người dùng phải có một bộ dữ liệu lớn để huấn luyện lại mô hình.

Thư viện OpenCV cung cấp sẵn các mô hình Haar Cascade được huấn luyện trên các bộ dữ liệu khuôn mặt và nụ cười phổ biến, đảm bảo độ chính xác tốt với các ứng dụng cơ bản.

Dưới đây là các bước khái quát của quy trình:

1. *Bước 1: Khởi động nguồn cấp video*

Hệ thống bắt đầu với việc cung cấp hai tùy chọn cho người dùng: mở webcam hoặc chọn video từ tệp có sẵn. Qua giao diện đồ họa được xây dựng bằng thư viện Tkinter, người dùng có thể khởi chạy webcam hoặc tải lên file video. Hệ thống sau đó sử dụng cv2.VideoCapture() để khởi tạo nguồn cấp video và đọc từng khung hình. Quá trình này hoạt động liên tục theo thời gian thực, giúp hệ thống giám sát hình ảnh một cách chính xác và nhanh chóng.

1. *Bước 2: Tiền xử lý hình ảnh*

Sau khi nhận được khung hình từ nguồn cấp video, hệ thống tiến hành tiền xử lý để chuẩn bị cho việc nhận diện:

Chuyển đổi sang thang độ xám: Mỗi khung hình được chuyển đổi từ ảnh màu RGB sang ảnh thang độ xám bằng cv2.cvtColor(). Điều này giảm thiểu độ phức tạp tính toán, bởi thông tin màu sắc không cần thiết cho việc nhận diện khuôn mặt và nụ cười.

Xử lý kích thước: Khung hình có thể được thay đổi kích thước để phù hợp với tốc độ xử lý và hiển thị trong giao diện.Người dùng có thể chọn mở webcam hoặc tải video từ file thông qua giao diện.

1. *Bước 3: Phát hiện khuôn mặt*

Trong mỗi khung hình, hệ thống sử dụng thuật toán Haar Cascade thông qua tệp phân loại sẵn có haarcascade\_frontalface\_default.xml. Thuật toán này phát hiện các vùng chứa khuôn mặt trong khung hình và trả về tọa độ (x, y, w, h) của các khung bao quanh khuôn mặt. Các khu vực này được xem là vùng quan tâm (ROI), nơi hệ thống tiếp tục thực hiện các bước nhận diện tiếp theo.

Quá trình phát hiện khuôn mặt hoạt động với độ chính xác cao nhưng yêu cầu xử lý liên tục theo thời gian thực. Để tối ưu, các tham số như tỷ lệ thay đổi kích thước và số lần lặp lại phát hiện được tinh chỉnh để cân bằng giữa độ chính xác và hiệu năng.

1. *Bước 4: Phát hiện nụ cười*

Sau khi xác định được các vùng khuôn mặt, hệ thống tiếp tục áp dụng thuật toán Haar Cascade khác (haarcascade\_smile.xml) để phát hiện nụ cười. Trong mỗi vùng khuôn mặt, hệ thống kiểm tra xem người trong ảnh có mỉm cười hay không. Nếu phát hiện được nụ cười, hệ thống đánh dấu vùng nụ cười bằng một khung hình chữ nhật màu đỏ và tiến hành các bước xử lý tiếp theo.

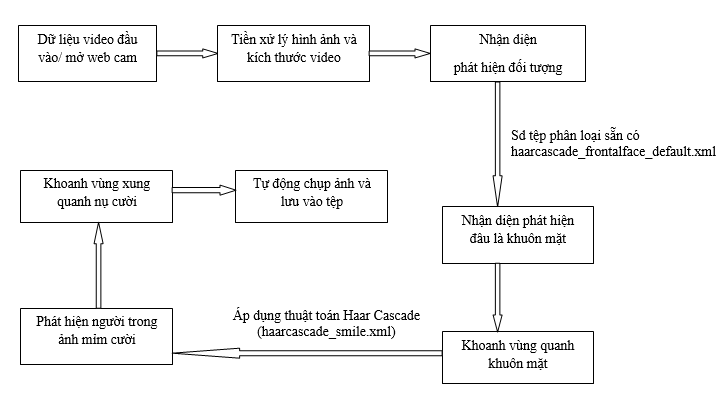
Thuật toán Haar Cascade được thiết kế để nhận diện các đặc trưng cụ thể của nụ cười, dựa trên sự thay đổi độ sáng và cường độ trong ảnh thang độ xám.

1. *Bước 5:Tự động chụp và lưu ảnh*

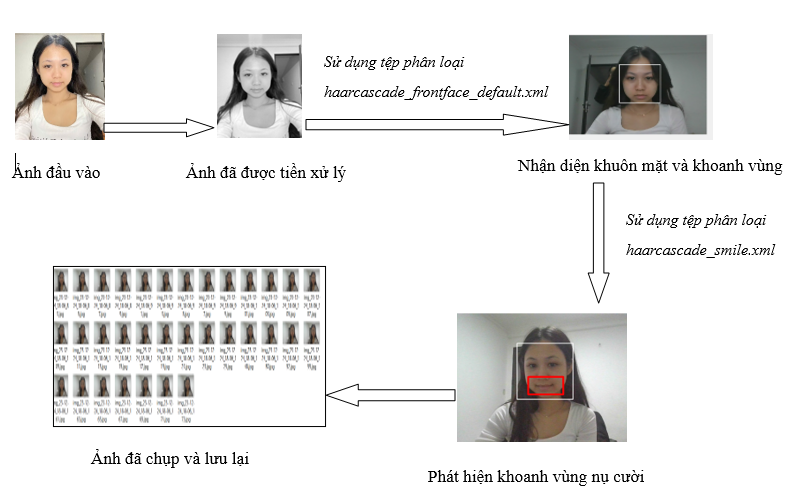
Khi phát hiện nụ cười, hệ thống tự động chụp và lưu khung hình hiện tại vào thư mục được định sẵn. Quá trình này bao gồm:

* Ghi lại thời gian thực tại và số thứ tự của ảnh.
* Sử dụng cv2.imwrite() để lưu ảnh dưới định dạng .jpg.
* Đảm bảo mỗi ảnh được đặt tên duy nhất dựa trên thời gian và thứ tự, giúp quản lý dữ liệu dễ dàng hơn.
* Ảnh được đặt tên theo thời gian thực và số thứ tự, đảm bảo quản lý file dễ dàng.

1. *Bước 6: Kết thúc và quản lý ảnh:* Khi dừng hệ thống, người dùng có thể mở thư mục chứa ảnh đã chụp trực tiếp từ giao diện và xem những bức ảnh đã chụp



Hình 2.3. Sơ đồ hệ thống chương trình tự động chụp ảnh khi phát hiện người trong ảnh mỉm cười

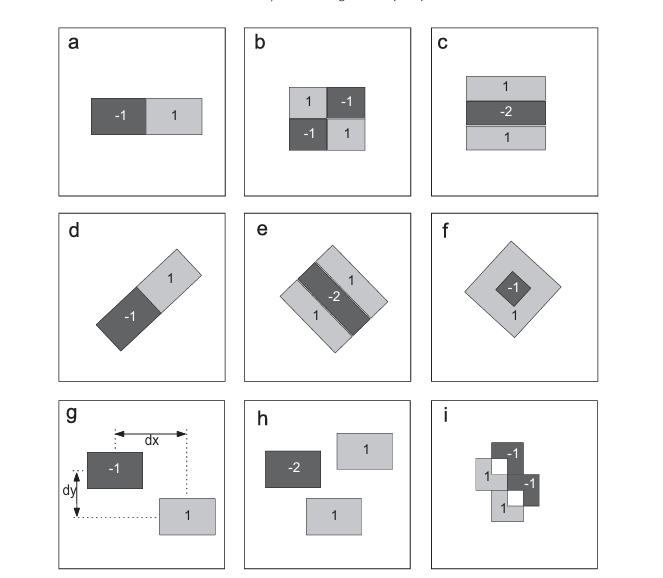


Hình 2.4. Sơ đồ hình ảnh hệ thống chương trình

## 2.2. Phương pháp nhận diện khuôn mặt sử dụng đặc trưng Haar-Like

### *2.2.1. Đặc trưng Haar-Like*

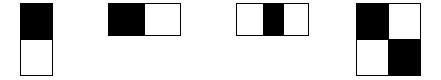
Các tính năng Haar-Like là một tập hợp hoàn chỉnh các hàm Haar hai chiều (2D), có thể được sử dụng để mã hóa sự xuất hiện cục bộ của các đối tượng. Chúng bao gồm hai hoặc nhiều vùng hình chữ nhật được đặt trong một mẫu. Giá trị tính năng f của một đối tượng địa lý giống Haar có k hình chữ nhật thu được như sau:



Hình 2.6. Các đặc trưng Haar-Like

Các tính năng Haar-Like được hiển thị với trọng số mặc định được gán cho các hình chữ nhật của nó. (a) và (b) hiển thị các đặc điểm giống Haar do Papageogiou et al. (c) đề xuất cho thấy một đặc điểm giống Haar với ba hình chữ nhật được giới thiệu bởi Viola và Jones. (d–f) cho thấy các đặc điểm xoay của Leinhardt. (g) và (h) cho thấy các đặc điểm giống Haar rời rạc của Li và cộng sự. (i) cho thấy đặc điểm đường chéo của Jones và Viola được thiết kế để chụp các cấu trúc đường chéo trong hình dạng của vật thể.

Đặc trưng do Viola và Jones công bố gồm 4 đặc trưng cơ bản để xác định khuôn mặt người. Mỗi đặc trưng Haar-Like là sự kết hợp của hai hay ba hình chữ nhật trắng hay đen như trong hình sau:

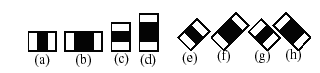


Hình 2.5. Bốn đặc trưng Haar-Like cơ bản

* Đặc trưng cạnh(edge feature)



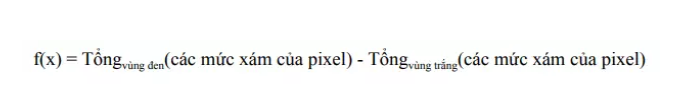
* Đặc trưng đường(line feature)



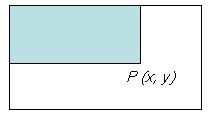
* Đặc trưng xung quanh tâm(center-surround features)



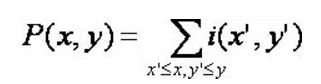
Dùng các đặc trưng trên, ta có thể tính được các giá trị của đặc trưng Haar-Like là sự chênh lệch giữa tổng của các pixel của vùng đen và vùng trắng như trong công thức sau:



Viola và Joines đưa ra một khái niệm gọi là Integral Image, là một mảng 2 chiều với kích thước bằng với kích thước của ảnh cần tính 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.



Công thức tính Intergral Image:

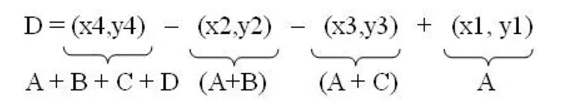


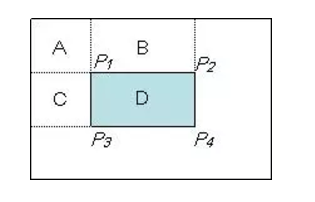
Sau khi tính được Integral Image, việc tính tổng các giá trị mức xám của một vùng bất kỳ nào đó trên ảnh thực hiện rất đơn giản theo cách sau:

Giả sử ta cần tính tổng giá trị mức xám của vùng D như hình dưới, ta có thể tính được như sau:

D = A + B + C + D – (A+B) – (A+C) + A

Với A + B + C + D chính là giá trị tại điểm P4 trên Integral Image, tương tự như vậy A+B là giá trị tại điểm P2, A+C là giá trị tại điểm P3, và A là giá trị tại điểm P1. Vậy ta có thể viết lại biểu thức tính D ở trên như sau:

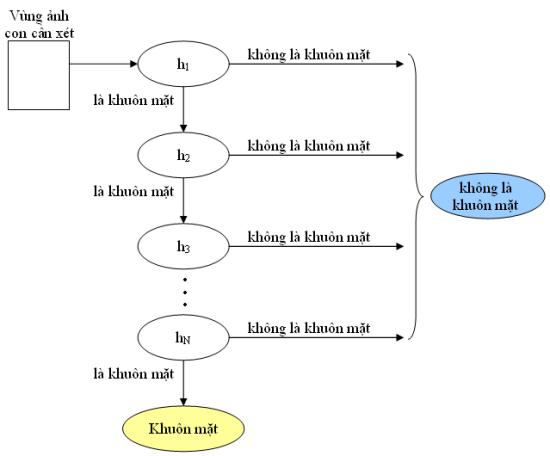




Một trong những lý do chính cho sự phổ biến Haar-like là chúng cung cấp một sự đánh đổi rất hấp dẫn giữa tốc độ đánh giá và độ chính xác. Với một bộ phân loại yếu đơn giản dựa trên các tính năng Haar-like chỉ tốn 60 lệnh vi xử lý, Viola và Jones đã đạt được 1% âm tính giả và 40% dương tính giả cho vấn đề phát hiện khuôn mặt. Tốc độ đánh giá cao chủ yếu là do sử dụng các hình ảnh tích phân, một khi được tính toán, có thể được sử dụng để đánh giá nhanh.

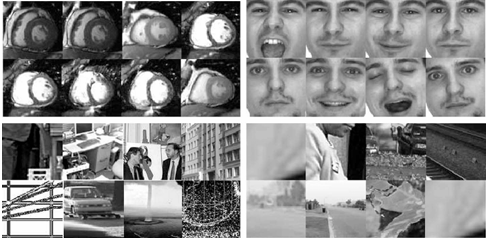
#### **2.2.2. AdaBoost**

AdaBoost là một bộ phân loại mạnh phi tuyến phức dựa trên hướng tiếp cận boosting được Freund và Schapire đưa ra vào năm 1995. Adaboost cũng hoạt động trên nguyên tắc kết hợp tuyến tính các weak classifiers để hình thành một trong các classifiers. Viola và Jones dùng AdaBoost kết hợp các bộ phân loại yếu sử dụng các đặc trưng Haar-like theo mô hình phân tầng (cascade) như

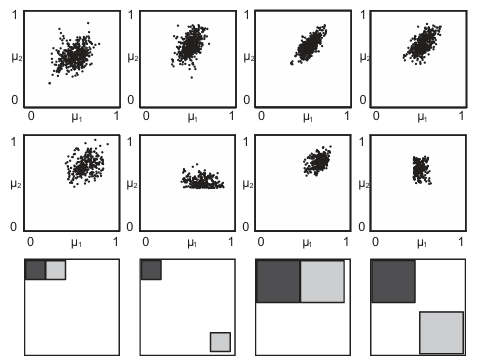


Hình 2.7. Mô hình phân tầng cascade

Các bộ phân loại gắn nhãn một tiểu vùng của hình ảnh, x, thuộc về lớp đối tượng hoặc lớp lộn xộn bằng cách so sánh f với một threshold 0 như hình dưới đây Họ được gọi là yếu vì họ dự kiến sẽ chỉ thực hiện tốt hơn một chút so với một người đoán ngẫu nhiên:

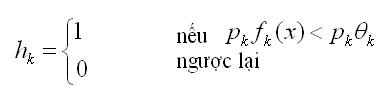
\

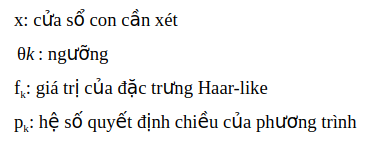
Hình 2.8. Hàng đầu tiên hiển thị hình ảnh từ cơ sở dữ liệu hình ảnh trái tim và khuôn mặt phía trước của con người. Hàng thứ hai chứa hình ảnh điển hình từ cơ sở dữ liệu lộn xộn.



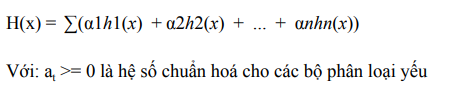
Hình 2.9. Phân bố các điểm đối tượng cho các tính năng Haar-Like

Trong đó, **h(k)** là các bộ phân loại yếu, được biểu diễn như sau:

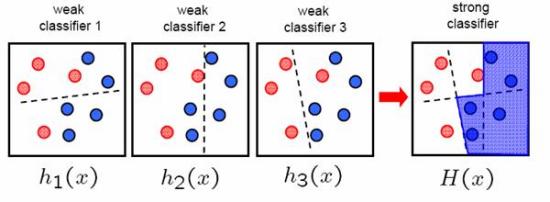




AdaBoost sẽ kết hợp các bộ phân loại yếu thành bộ phân loại mạnh như sau:

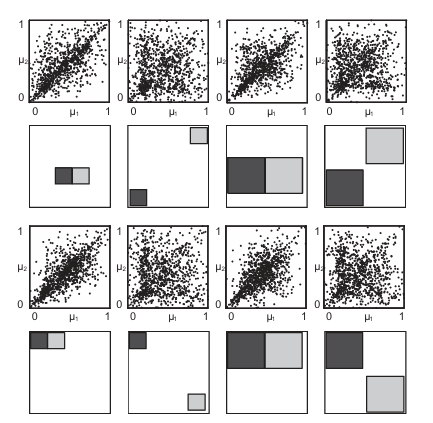


Đây là hình ảnh minh họa việc kết hợp các bộ phân loại yếu thành bộ phân loại mạnh:



#### **2.2.3. Không gian đặc trưng hình chữ nhật đơn (SRFS)**

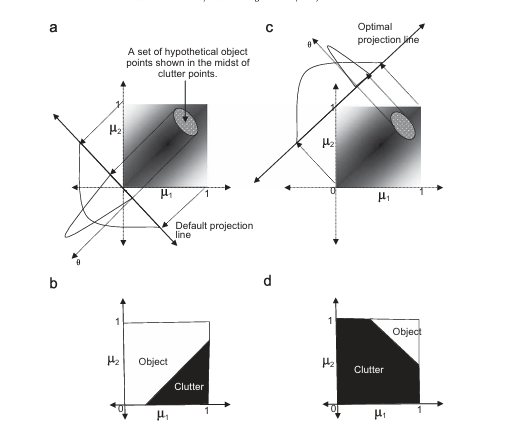
Khi hình ảnh được đánh giá với một tính năng Haar-Like với k góc chữ nhật, một vectơ có chiều dài k chứa các phép đo trung bình hình chữ nhật có thể được tạo ra. Sử dụng các phép đo này, hình ảnh có thể được lặp lại như một điểm trong không gian tính năng k chiều- gọi là không gian tính năng hình chữ nhật đơn. Hãy để biểu diễn của một hình ảnh thuộc lớp đối tượng được gọi là điểm đối tượng và simi larly, biểu diễn của một hình ảnh thuộc lớp lộn xộn

**

Hình 2.10. Phân phối các điểm lộn xộn cho các tính năng Haar khác

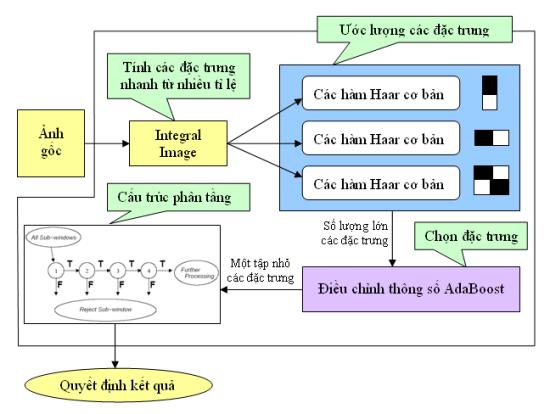
Dựa trên một tập hợp các hình ảnh thuộc lớp đối tượng và sự lộn xộn đã được đánh giá trên tính năng Haar-Like với hai hình chữ nhật, do đó tạo ra một đám mây đối tượng và các điểm lộn xộn trong SRFS hai chiều. Số lượng hình chữ nhật trong tính năng Haar-Like bị giới hạn ở hai để hiển thị rõ ràng. Hai cơ sở dữ liệu đối tượng khác nhau đã được chọn cho nghiên cứu này. Một chứa hình ảnh mặt trước của con người từ cơ sở dữ liệu AR [10] và cái còn lại chứa hình ảnh cộng hưởng từ trục ngắn của tim.

Mỗi hình ảnh được chuẩn hóa cường độ bằng cách chia mọi giá trị pixel cho 2n 1, trong đó nis số bit được sử dụng để đại diện cho mỗi giá trị pixel. Điều này làm cho các giá trị pixel từ hai hình ảnh bất kỳ so sánh với nhau vì tất cả chúng đều khác nhau từ 0 đến 1.



Hình 2.11. Chuẩn hóa cường độ

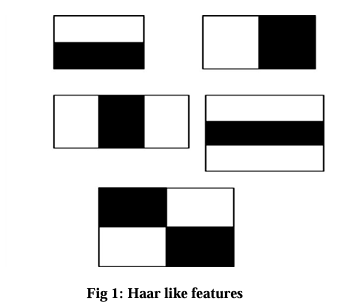
Từ đó có thể tóm tắt qua sơ đồ nhận diện khuôn mặt sau:



Hình 2.12. Sơ đồ nhận diện khuôn mặt

## 2.3. *Nhận diện người trong ảnh mỉm cười sau khi phát hiện khuôn mặt*

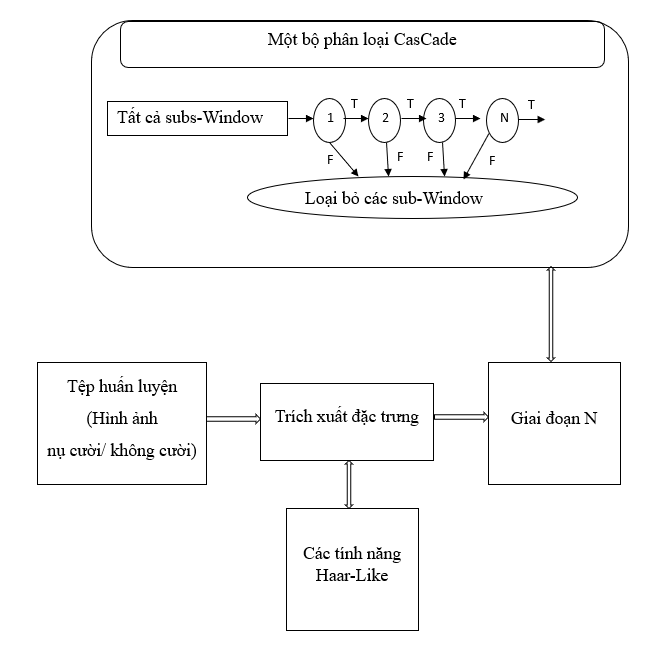
Sau khi phát hiện đối tượng và nhận diện khuôn mặt bằng cách sử dụng bộ phân loại tầng dựa trên tính năng để xác định xem người đó có đang cười trên khuôn mặt hay không, em đã sử dụng cùng một nguyên tắc mà đã sử dụng để phát hiện nụ cười trên khuôn mặt con người. Hệ thống tiếp tục áp dụng thuật toán Haar Cascade khác ( với file haarcascade\_smile.xml) để phát hiện nụ cười dựa trên tính năng Haar-Like



Hình 2.13. Đặc trưng Haar-Like

Ở đây em sẽ làm việc với phát hiện nụ cười. Ban đầu, thuật toán cần rất nhiều hình ảnh tích cực (hình ảnh khuôn mặt cười) và hình ảnh tiêu cực (hình ảnh không có khuôn mặt cười) để đào tạo bộ phân loại. Sau đó, cần phải trích xuất các tính năng từ nó. Đối với điều này, các tính năng haar được hiển thị trong hình ảnh được sử dụng. Chúng giống như quản lý tài nguyên tích chập của chúng tôi. Mỗi tính năng là một giá trị duy nhất thu được bằng cách trừ tổng pixel dưới hình chữ nhật màu trắng khỏi tổng pixel dưới hình chữ nhật màu đen. Bây giờ tất cả các kích thước và vị trí có thể có của mỗi hạt nhân được sử dụng để tính toán nhiều tính năng

Để tìm các tính năng phù hợp nhất trong hình ảnh nhất định, em sử dụng thuật toán Adaboost để lựa chọn sử dụng đầy đủ các tính năng. Đối với điều này, em áp dụng từng tính năng haar trên tất cả các hình ảnh đào tạo. Đối với mỗi và mọi tính năng, nó tìm ra ngưỡng tốt nhất sẽ phân loại khuôn mặt cười. Nhưng rõ ràng, sẽ có lỗi hoặc phân loại sai. Em chọn các tính năng có tỷ lệ lỗi tối thiểu, có nghĩa là chúng là những tính năng phân loại tốt nhất hình ảnh nụ cười và hình ảnh không cười. Trong quá trình này, mỗi hình ảnh được đưa ra một trọng số bằng nhau ngay từ đầu. Sau mỗi lần phân loại, trọng số của hình ảnh phân loại sai được tăng lên. Sau đó, một lần nữa quá trình tương tự được thực hiện. Tỷ lệ lỗi mới được tính toán và trọng số mới.

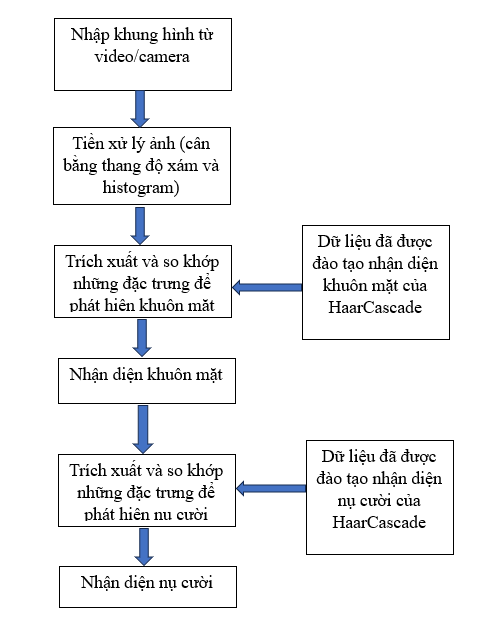


Hình 2.14.Quá trình training nhận dạng phát hiện người mỉm cười

Quá trình này được tiếp tục cho đến khi đạt được độ chính xác hoặc tỷ lệ lỗi cần thiết hoặc yêu cầu số lượng tính năng được tìm thấy. Một số phân loại tính năng Haar tạo thành một giai đoạn. Bộ so sánh giai đoạn tính tổng tất cả các kết quả phân loại tính năng Haar trong một giai đoạn và so sánh tổng hợp này với ngưỡng giai đoạn. Ngưỡng cũng là hằng số thu được từ thuật toán AdaBoost. Mỗi giai đoạn không có một số tính năng Haar nhất định. Tùy thuộc vào các thông số của dữ liệu đào tạo, các giai đoạn riêng lẻ có thể có số lượng Haar featurs khác nhau. Phân loại cuối cùng là tổng trọng số của các phân loại yếu này. Nó được gọi là yếu vì một mình nó không thể phân loại hình ảnh, nhưng cùng với những người khác tạo thành một bộ phân loại mạnh. Toàn bộ quá trình được thể hiện qua hình 2.13.

### *2.4. Cơ chế phát hiện nụ cười thời gian thực sử dụng OpenCV*

Đối với cơ chế phát hiện nụ cười, chúng tôi đã sử dụng thuật toán phát hiện đối tượng nhanh do Viola đề xuất và Jones được sử dụng làm cơ sở cho thiết kế của chúng tôi. Những đặc điểm haar cụ thể của vùng miệng cười được sử dụng để phát hiện nụ cười. Hình dưới đây hiển thị ở trên cho thấy cơ chế phát hiện nụ cười thời gian thực được sử dụng trong nguyên mẫu của chúng tôi. Trong quá trình thực cơ chế phát hiện nụ cười, hình ảnh hoặc video nhất định được chụp từng khung hình và chuyển đổi thành thang màu xám. Ánh sáng thay đổi là một trong những khó khăn để phát hiện nụ cười trong khuôn mặt thực. Chúng tôi đã sử dụng cân bằng biểu đồ tần suất cho hình ảnh thang màu xám vì HE là một đơn giản và kỹ thuật được sử dụng rộng rãi để chuẩn hóa hiệu ứng chiếu sáng.



Hình 2.15. Cơ chế hệ thống nhận dạng người trong ảnh mỉm cười sử dụng thời gian thực

Hệ thống có khả năng phát hiện nụ cười từ hình ảnh được chụp cho mục đích nguyên mẫu. Từ sơ đồ trên, hệ thống đầu tiên tìm kiếm nhận diện khuôn mặt sau đó quá trình phát hiện nụ cười diễn ra. Phát hiện khuôn mặt xác định vị trí của khuôn mặt trong hình ảnh và nó được thực hiện bằng cách quét các tỷ lệ hình ảnh khác nhau và trích xuất các mẫu chính xác để phát hiện khuôn mặt. Ứng cử viên khuôn mặt là một phần hình chữ nhật của hình ảnh gốc được gọi là cửa sổ phụ. Nói chung, các cửa sổ phụ này có kích thước cố định (thường là 24×24 pixel). Phát hiện phân loại Haar được sử dụng để tạo một cửa sổ tìm kiếm trượt qua hình ảnh và kiểm tra xem một vùng nhất định của hình ảnh có giống khuôn mặt hay không. Các tính năng của Haar-Like và một tập hợp lớn các phân loại rất yếu sử dụng một tính năng duy nhất để xác định một hình ảnh nhất định là khuôn mặt hoặc không khuôn mặt. Dữ liệu được lưu trữ tại dữ liệu haar. Mỗi đối tượng địa lý được mô tả theo mẫu, tọa độ của nó liên quan đến nguồn gốc cửa sổ tìm kiếm và kích thước của đối tượng địa lý. Cửa sổ tìm kiếm quét nhanh bộ phân loại đầu tiên trên thác như thể hiện trong sơ đồ trình phân loại trả về false thì tính toán trên cửa sổ đó cũng kết thúc và kết quả không có khuôn mặt nào được phát hiện (false). Hơn nữa, nếu trình phân loại trả về true, thì cửa sổ sẽ được chuyển xuống bộ phân loại tiếp theo trong tầng để thực hiện chính xác điều tương tự. Khi tất cả các trình phân loại trả về true cho cửa sổ đó, thì kết quả cũng sẽ trả về true cho mặt cửa sổ nhất định được phát hiện.

Sau quá trình phát hiện khuôn mặt, phát hiện nụ cười diễn ra. Phát hiện nụ cười xác định vị trí của nụ cười trong hình ảnh khuôn mặt và nó được thực hiện bằng cách quét các thang hình ảnh khuôn mặt khác nhau và trích xuất các mẫu chính xác để phát hiện nụ cười. Để phát hiện nụ cười, chúng tôi đã sử dụng các tính năng và chức năng Haar-Like từ OpenCV. Trong phân loại haar này, phát hiện được sử dụng để tạo một cửa sổ tìm kiếm (43 \* 19 pixel) trượt qua hình ảnh khuôn mặt và kiểm tra thời tiết, một vùng nhất định của khuôn mặt, tìm kiếm người có cười hay không. Một nụ cười tạo dáng là một phần hình chữ nhật của hình ảnh gốc được gọi là cửa sổ phụ. Các tính năng Haar-Like và một bộ phân loại rất yếu sử dụng một tính năng duy nhất để xác định một hình ảnh khuôn mặt nhất định là nụ cười hoặc không nụ cười được lưu trữ tại dữ liệu haar cascade smiled. Mỗi đối tượng địa lý được mô tả theo mẫu, tọa độ của nó liên quan đến nguồn gốc cửa sổ tìm kiếm và kích thước của đối tượng địa lý. Cửa sổ tìm kiếm nhanh chóng quét trình phân loại đầu tiên trên dòng thác phân loại. Nếu trình phân loại trả về false thì tính toán trên cửa sổ đó cũng kết thúc và kết quả không có nụ cười nào được phát hiện (false).

# CHƯƠNG 3: XÂY DỰNG CHƯƠNG TRÌNH TỰ ĐỘNG CHỤP ẢNH KHI PHÁT HIỆN NGƯỜI TRONG ẢNH MỈM CƯỜI

## 3.1. Môi trường phát triển

### *3.1.1. Ngôn ngữ lập trình*

Python là một ngôn ngữ lập trình cao cấp, được phát triển vào năm 1991 bởi Guido van Rossum trong một dự án mã mở (open source). Python được thiết kế để có cú pháp đơn giản, dễ đọc và dễ hiểu, cũng như có tính linh hoạt và mở rộng. Python là ngôn ngữ lập trình đa năng và được sử dụng trong nhiều lĩnh vực, bao gồm:

* Phát triển web: Python có các framework như Django và Flask cho phép phát triển các ứng dụng web hiệu quả và nhanh chóng.
* Khoa học dữ liệu: Python là ngôn ngữ phổ biến trong việc phân tích dữ liệu, máy học và trí tuệ nhân tạo.
* Game và đồ họa: Python có thể được sử dụng để phát triển các trò chơi và ứng dụng đồ họa, nhờ vào các thư viện như Pygame và PyOpenGL.
* Hệ thống: Python cung cấp các công cụ hữu ích để viết các ứng dụng hệ thống và quản lý cơ sở dữ liệu.



Hình 3.1. Hình ảnh minh họa ngôn ngữ Python

Phiên bản Python 3.10 được cải thiện hiệu năng với việc tối ưu hóa cấu trúc dữ liệu Dictionary. Dictionary là một trong những cấu trúc dữ liệu phổ biến và quan trọng trong Python, được sử dụng rộng rãi trong nhiều ứng dụng. Việc tối ưu hóa này giúp giảm thời gian truy cập và thao tác trên Dictionary, từ đó em sử dụng phiên bản Python 3.10 để cải thiện hiệu suất của chương trình.

### *3.1.2. Thư viện sử dụng và cài đặt*

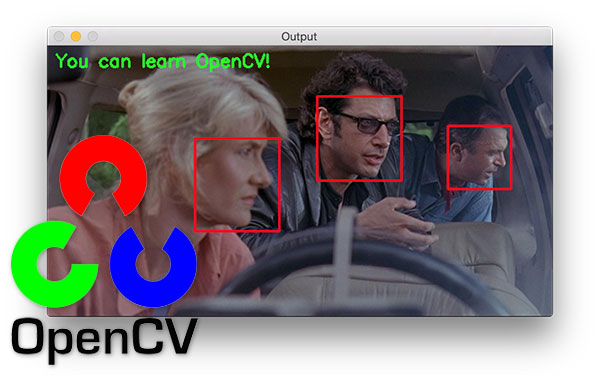
1. *OpenCV*

OpenCV (Open Source Computer Vision Library) là một thư viện mã nguồn mở được phát triển từ Intel vào năm 1999 bởi Gary Bradsky và Vadim Pisarevsky. Thư viện sử dụng rộng rãi cho xử lý ảnh và video trong các ứng dụng thị giác máy tính. Nó được viết bằng ngôn ngữ C ++ và có thể sử dụng được với các ngôn ngữ như Python, Java, C# và MATLAB.

OpenCV được sử dụng cho đa dạng nhiều mục đích và ứng dụng khác nhau bao gồm :

* Hình ảnh street view
* Kiểm tra và giám sát tự động
* Robot và xe hơi tự lái
* Phân tích hình ảnh y học
* Tìm kiếm và phục hồi hình ảnh/video
* Phim – cấu trúc 3D từ chuyển động
* Nghệ thuật sắp đặt tương tác

Trong Python, OpenCV được tích hợp thông qua thư viện cv2. Để sử dụng thư viện cv2, bạn cần cài đặt nó trên máy tính của mình. Sau đó, bạn có thể import thư viện cv2 vào chương trình Python của mình và sử dụng các chức năng và tính năng có sẵn trong thư viện để xử lý ảnh và video.



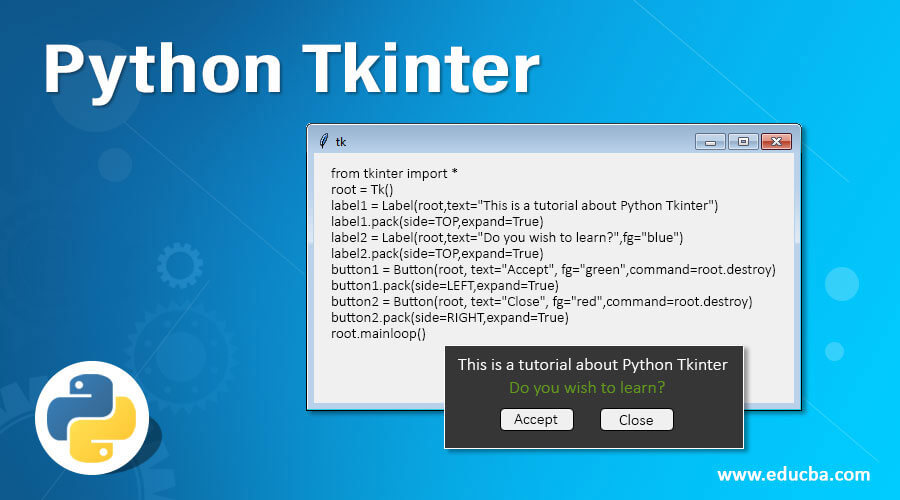
Hình 3.2. Hình ảnh minh họa thư viện OpenCV

Tiếp đến cách để thực hiện cài đặt chúng thì cách đơn giản nhất là sử dụng lệnh PIP nhưng phải đảm bảo bạn đã cài đặt Python nên có thể kiểm tra bằng cách mở terminal (cmd trên Windows hoặc terminal trên macOS/Linux) và chạy lệnh: ***python –version.***

Nếu chưa cài thì truy cập [python.org](https://www.python.org/downloads/) để cài đặt Python từ trang web chính thức, Sau đó, chỉ cần chạy lệnh sau để cài đặt thư viện opencv-python: ***pip install opencv-python***

1. *Python Tkinter*

Tkinter là một thư viện đồ họa cho Python, cung cấp các công cụ để xây dựng các ứng dụng GUI (Giao diện người dùng đồ họa) trên nền tảng desktop. Tkinter là một phần của Python được phân phối hầu hết các phiên bản Python.



Hình 3.3. Hình ảnh minh họa thư viện Tkinter trong Python

Tkinter là một thư viện giao diện người dùng đơn giản và dễ sử dụng, với các thành phần giao diện người dùng phổ biến như hộp văn bản, nút bấm, danh sách thả xuống,… Nó cho phép người dùng tạo các ứng dụng GUI đơn giản hoặc phức tạp, được sử dụng rộng rãi trong các ứng dụng desktop của Python.

Và để cài đặt thư viện này ta chỉ cần sử dụng pip để cài đặt các thư viện này: ***pip install opencv-python pillow tk***

1. *PIL (Python Imaging Library)*

PIL là một thư viện mã nguồn mở cho phép xử lý ảnh trong Python. Thư viện này cung cấp các phương thức và lớp để mở, tạo, chỉnh sửa và lưu các tệp ảnh. PIL cũng hỗ trợ các định dạng ảnh phổ biến như JPEG, PNG, GIF và BMP.

Tuy nhiên, PIL đã không được bảo trì kể từ năm 2011 và được thay thế bởi Pillow, một thư viện phát triển dựa trên PIL. Pillow giữ lại toàn bộ API của PIL, cùng với việc cải thiện hiệu suất và hỗ trợ các phiên bản Python mới hơn.



Hình 3.4. Thư viện Pillow

Với Pillow, các lập trình viên có thể thực hiện các tác vụ xử lý ảnh như chuyển đổi định dạng ảnh, cắt ảnh, thay đổi kích thước, xoay ảnh, thêm đường viền và phông chữ, và nhiều hơn nữa.

Pillow là phiên bản thay thế cho thư viện **PIL** cũ, và bạn có thể cài đặt nó bằng **pip**. Để cài đặt Pillow, bạn chỉ cần mở terminal (hoặc cmd trên Windows) và chạy lệnh sau: ***pip install Pillow***

1. *Thư viện OS và Datetime*

Thư viện **os** là một thư viện tiêu chuẩn trong Python cung cấp các công cụ để tương tác với hệ điều hành. Với os, bạn có thể dễ dàng làm việc với hệ thống tệp, thư mục, và thực hiện các thao tác như tạo, xóa thư mục, liệt kê các tệp, thay đổi thư mục làm việc, kiểm tra sự tồn tại của tệp hoặc thư mục, và nhiều tác vụ khác liên quan đến hệ thống tệp.

Các chức năng chính của thư viện os:

* Quản lý thư mục và tệp: Tạo, xóa, thay đổi, và liệt kê các tệp và thư mục.
* Tương tác với hệ điều hành: Chạy các lệnh hệ thống và làm việc với các biến môi trường.
* Xử lý đường dẫn tệp: Thực hiện các thao tác liên quan đến đường dẫn, chẳng hạn như phân tách hoặc nối các phần của đường dẫn.

Thư viện **datetime** là một phần của thư viện chuẩn trong Python, cung cấp các công cụ mạnh mẽ để làm việc với ngày và giờ. Với datetime, bạn có thể thực hiện các tác vụ như lấy thời gian hiện tại, tính toán khoảng thời gian, chuyển đổi giữa các định dạng ngày giờ, và nhiều thao tác khác liên quan đến thời gian.

Các chức năng chính của thư viện datetime:

* Lấy thông tin về ngày và giờ: Truy xuất thời gian hiện tại, năm, tháng, ngày, giờ, phút, giây.
* Chuyển đổi giữa các định dạng ngày giờ: Chuyển đổi giữa chuỗi và đối tượng datetime.
* Tính toán khoảng thời gian: Thực hiện phép cộng hoặc trừ giữa các đối tượng datetime hoặc timedelta.
* So sánh các đối tượng datetime: So sánh thời gian giữa các đối tượng datetime.

Để sử dụng các thư viện **os** và **datetime** trong Python, bạn không cần phải cài đặt chúng vì chúng là các thư viện tiêu chuẩn (standard libraries), có sẵn ngay khi bạn cài đặt Python. Bạn chỉ cần import chúng vào chương trình

## 3.2. Quá trình thực hiện

### *3.2.1. Mô tả các bước thực hiện*

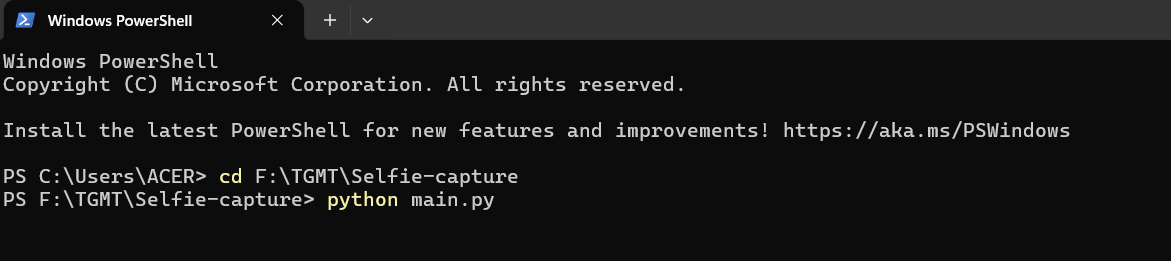
Khi đã cài đặt thành công các thư viện cần thiết để chạy chương trình tự động chụp ảnh khi phát hiện nhận diện nụ cười qua video hoặc webcam thì cần chuẩn bị thư mục và các tệp cần thiết.

Đầu tiên là sẵn các tệp Haar Cascade Classifier để nhận diện khuôn mặt và nụ cười. Những tệp này có thể tải từ thư viện Haar Cascade của OpenCV và bao gồm haarcascade\_frontalface\_default.xml và haarcascade\_smile.xml. Đảm bảo rằng bạn đã có sẵn các tệp này trong máy tính của mình.

Chương trình cũng sẽ lưu các bức ảnh có nụ cười phát hiện được vào thư mục *F:/TGMT/Selfie-capture/res.* Trước khi chạy chương trình, hãy chắc chắn rằng thư mục này tồn tại và có quyền ghi ảnh.

Sau đó, trước khi chạy chương trình, bạn cần kiểm tra mã nguồn để đảm bảo rằng các tệp và thư mục được tham chiếu chính xác. Đặc biệt, hãy kiểm tra đường dẫn đến các tệp Haar Cascade

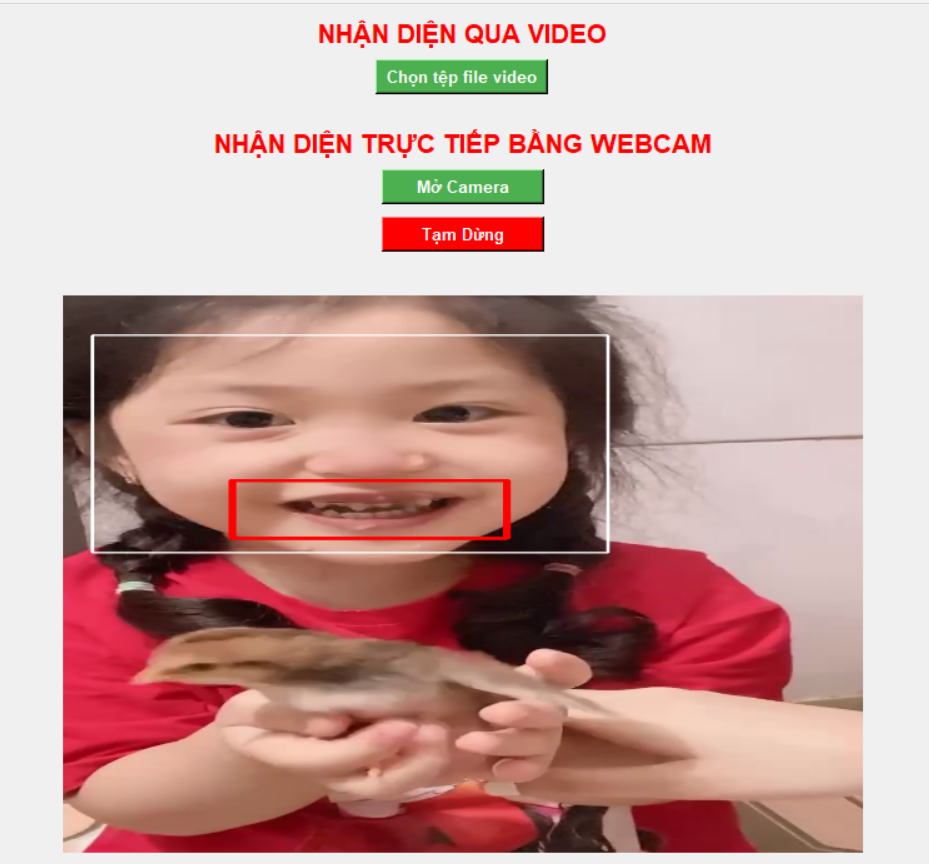
Khi đã hoàn tất các bước chuẩn bị, bạn có thể chạy chương trình. Nếu bạn sử dụng một IDE như PyCharm, VSCode hoặc một trình biên dịch Python khác, chỉ cần mở tệp .py và nhấn nút Run. Nếu bạn muốn chạy từ dòng lệnh, mở terminal hoặc command prompt, điều hướng đến thư mục chứa tệp Python và sử dụng lệnh sau:



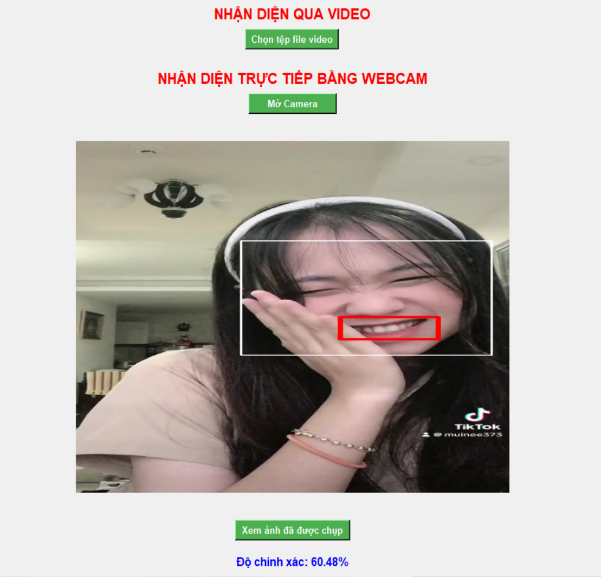
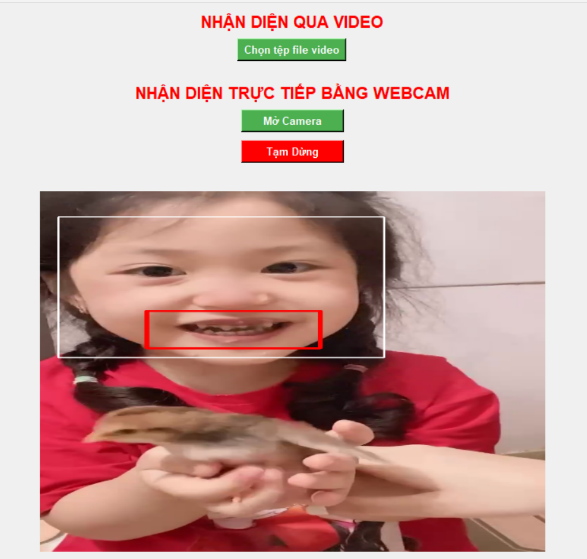
Hình 3.5. Cách chạy file chương trình trên cmd

## 3.4. Kết quả

Khi bắt đầu khởi chạy chương trình, một cửa sổ giao diện Tkinter sẽ xuất hiện. Giao diện này cung cấp các nút để bạn tương tác với chương trình. Bạn có thể chọn video từ tệp bằng cách nhấn nút Chọn tệp file video. Chương trình sẽ phát video và nhận diện nụ cười trong video đó. Nếu bạn muốn nhận diện trực tiếp từ webcam, chỉ cần nhấn Mở Camera để chương trình nhận diện nụ cười ngay từ webcam.

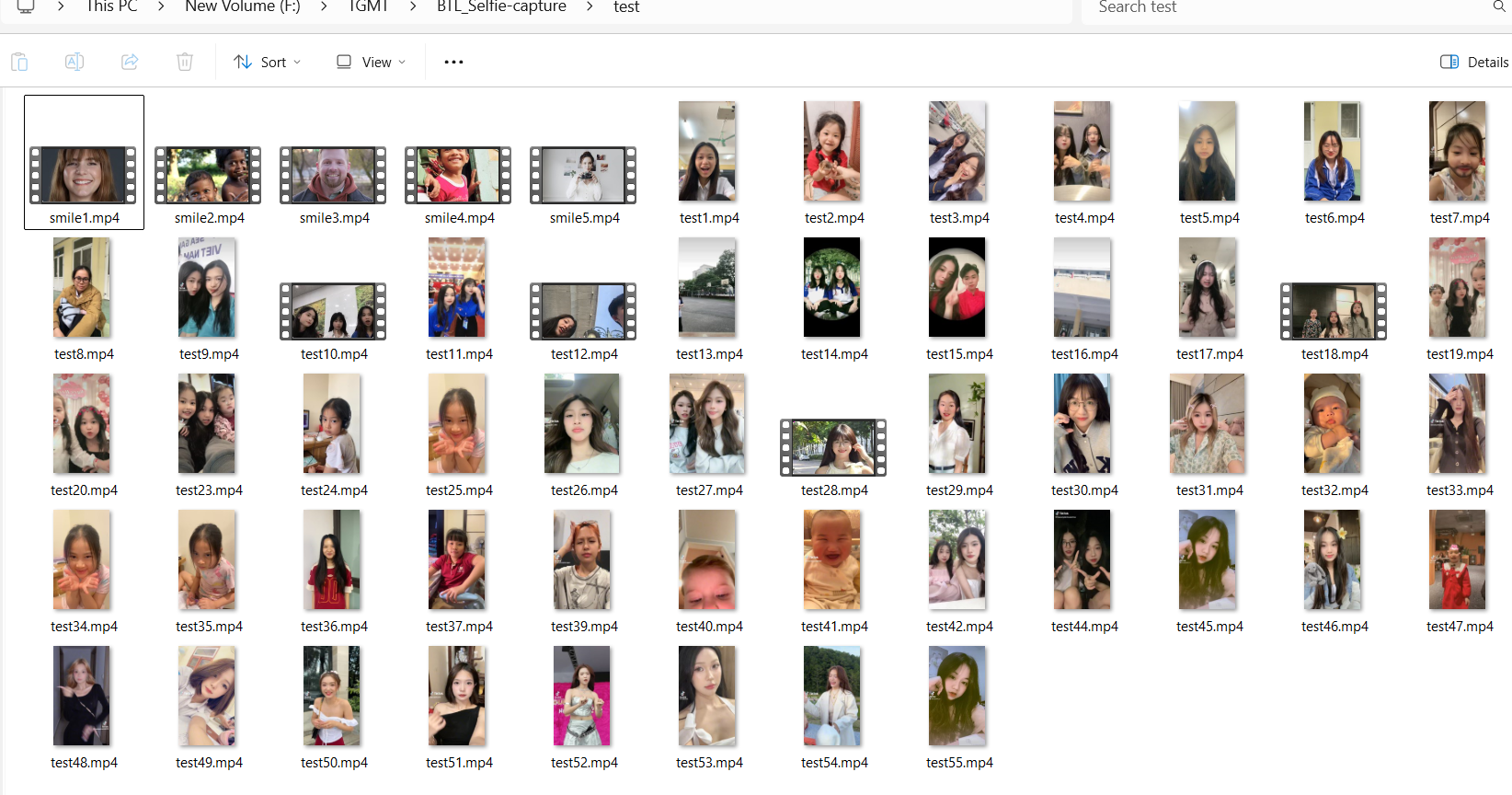


Hình 3.6. Hai Button chọn file và mở cam để nhận diện

Ngoài ra, sau khi chương trình phát hiện và lưu ảnh có nụ cười, bạn có thể nhấn Xem ảnh đã được chụp để mở thư mục chứa ảnh đã lưu. Nếu muốn dừng chương trình, bạn có thể nhấn Tạm Dừng.

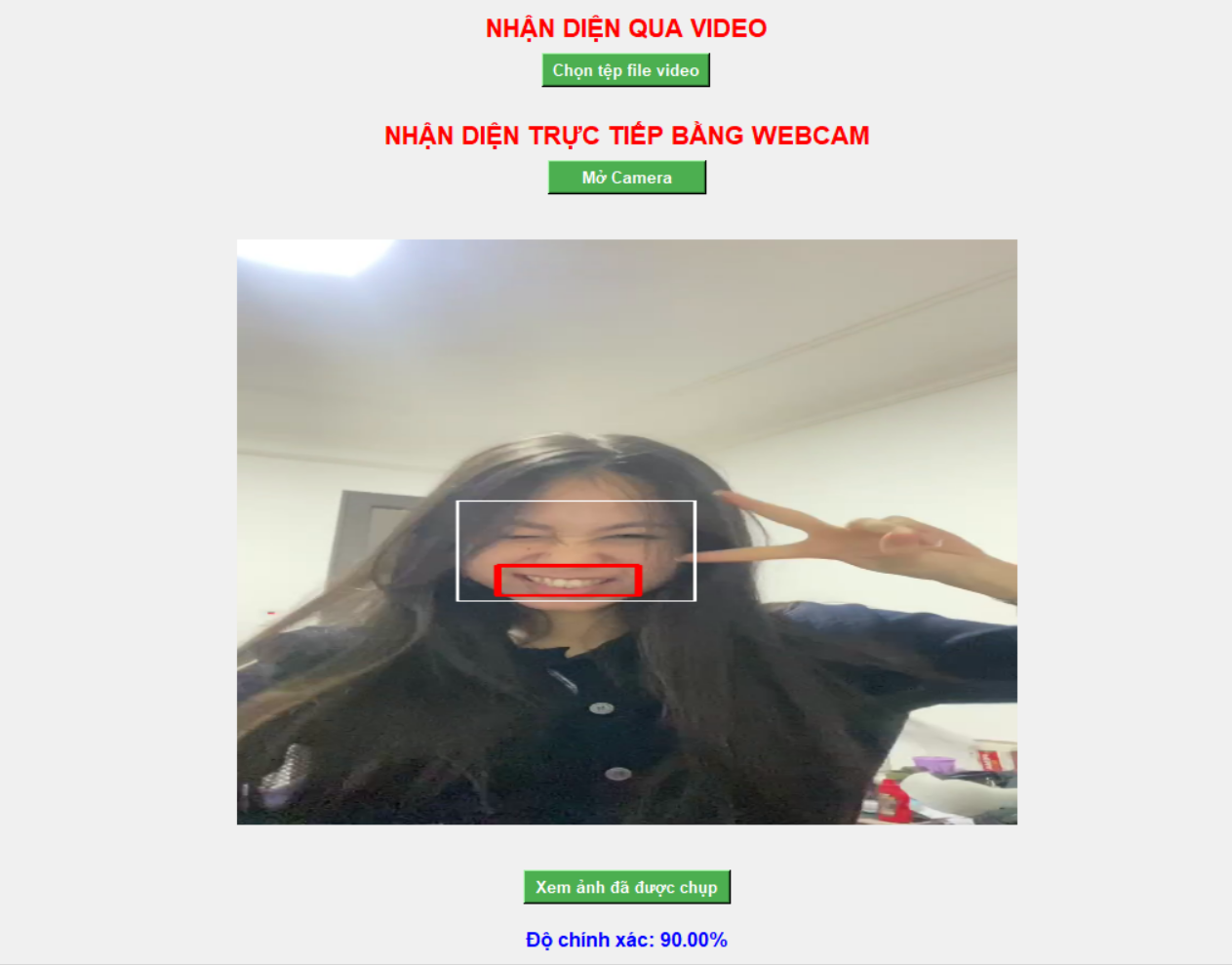
Hình 3.7. Nút tạm dừng và xem ảnh

:



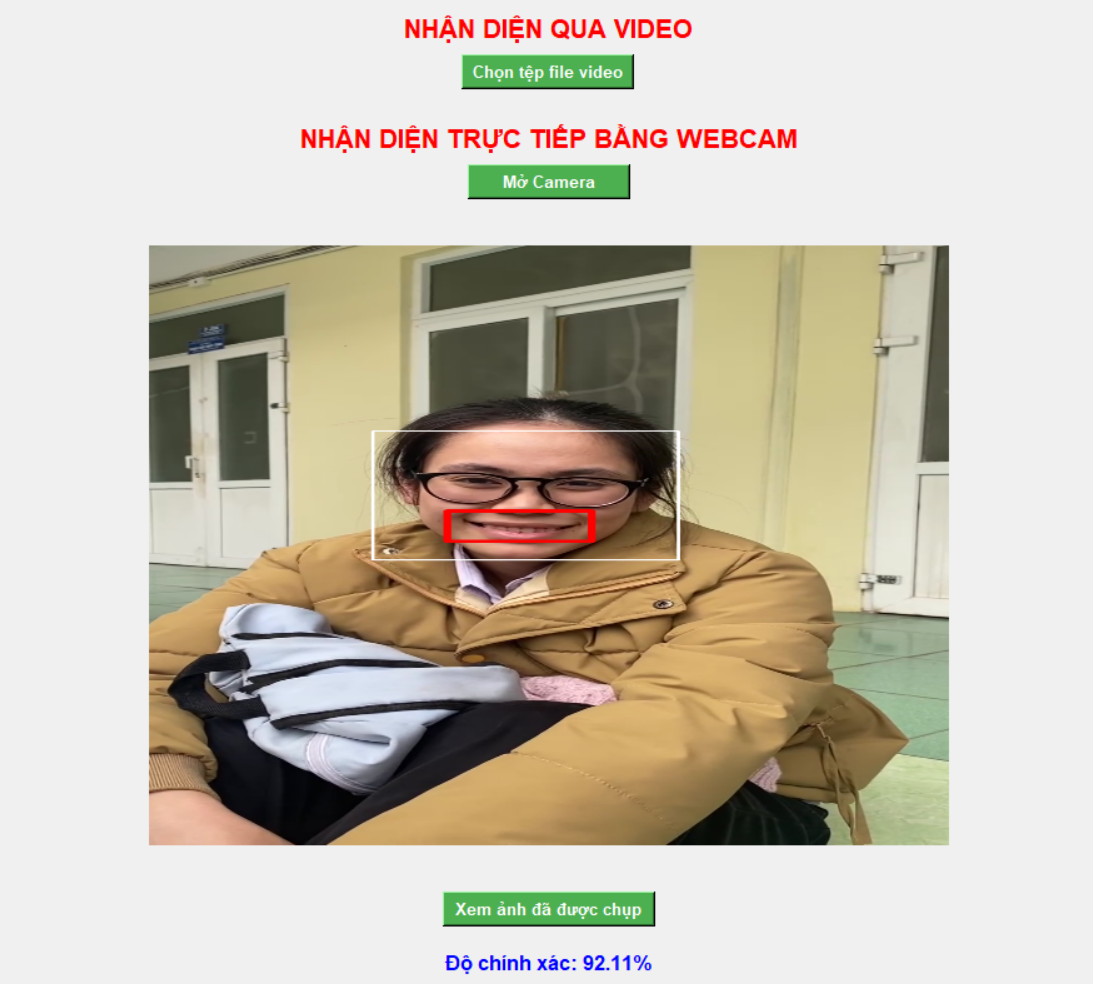
Hình 3.8. Dữ liệu video test đầu vào

Sau khi chạy chương trình, bạn sẽ thấy các ảnh có nụ cười được lưu vào thư mục F:/TGMT/Selfie-capture/res dưới tên img\_<ngày-gio>\_<số thứ tự>.jpg. Độ chính xác của việc nhận diện nụ cười sẽ được hiển thị trực tiếp trên giao diện để bạn theo dõi hiệu suất của chương trình.



Hình 3.9. Kết quả nhận diện từ video

Nhìn chung mô hình nhận diện nụ cười trong chương trình đã hoạt động khá ổn, với khả năng phát hiện khuôn mặt và nụ cười trong video hoặc qua webcam.



Hình 3.10.Kết quả nhận dạng từ video

Chương trình sử dụng bộ thư viện OpenCV để nhận diện khuôn mặt và nụ cười, giúp cho quá trình nhận diện có thể diễn ra trong thời gian thực mà không gặp phải quá nhiều lỗi nghiêm trọng. Dù có những trường hợp nụ cười nhẹ hoặc bị che khuất vẫn được nhận diện chính xác.

Tuy nhiên, một số hạn chế vẫn còn tồn tại trong chương trình. Ví dụ, với những video có độ phân giải thấp hoặc ánh sáng không tốt, chương trình có thể gặp khó khăn trong việc phát hiện khuôn mặt và nụ cười. Đồng thời, chương trình vẫn có thể đưa ra các kết quả không chính xác, như việc nhận diện nụ cười không thật sự rõ ràng hoặc có thể xảy ra các nhầm lẫn giữa các biểu cảm khác nhau, làm cho độ chính xác đôi khi không đạt được mức cao như mong đợi.



Hình 3.11. Kết quả nhận từ webcam

Ngoài ra, việc tính toán độ chính xác trong thời gian thực có thể bị ảnh hưởng bởi tốc độ video đầu vào. Khi video được chạy nhanh, độ chính xác có thể giảm do không đủ thông tin để nhận diện chính xác trong mỗi khung hình. Vì vậy, cần thêm các cơ chế tối ưu hóa để cải thiện tính chính xác khi làm việc với video chất lượng thấp khi nhận diện trong môi trường ánh sáng thay đổi.

# KẾT LUẬN

## 4.1. Kết luận

Trong công việc này, em đã xây dựng thành công một chương trình nhận diện nụ cười tự động qua video và webcam, sử dụng thư viện OpenCV kết hợp với các bộ phân loại Haar Cascade để phát hiện khuôn mặt và nụ cười. Trong đó test thử bộ dữ liệu do e thu thập thì kếtb quả đúng 55/60 còn lại do tác động ánh sáng, độ phân giải, video bị khuất hoặc che mặt hoặc nụ cười. Chương trình có thể nhận diện nụ cười trong các video đầu vào hoặc từ webcam và tự động lưu ảnh khi phát hiện nụ cười. Thuật toán nhận diện nụ cười đã được cải tiến để tính toán độ chính xác của việc nhận diện trong thời gian thực, mang lại một hệ thống có khả năng phát hiện nụ cười khá tốt. Dù đã đạt được một số kết quả đáng khích lệ, hệ thống vẫn cần được tối ưu hóa thêm để nâng cao độ chính xác trong các tình huống thực tế, chẳng hạn như nhận diện trong điều kiện ánh sáng yếu hoặc video có độ phân giải thấp. Hệ thống được đề xuất có thể được áp dụng vào các ứng dụng như kiểm tra cảm xúc trong các cuộc họp trực tuyến hoặc làm công cụ hỗ trợ trong việc phát triển các ứng dụng nhận diện cảm xúc.

### *4.1.1 Ưu điểm*

Hệ thống nhận diện nụ cười đã hoạt động tự động, giảm thiểu sự can thiệp của người dùng, từ việc phát hiện khuôn mặt cho đến nhận diện nụ cười.

Việc sử dụng OpenCV giúp phát hiện nhanh chóng các đặc điểm của khuôn mặt và nụ cười, đồng thời tính toán độ chính xác giúp người dùng có cái nhìn rõ ràng về hiệu quả của chương trình.

Các phương pháp nhận diện đơn giản như Haar Cascade, mặc dù có thể gặp một số khó khăn trong việc nhận diện nụ cười trong các điều kiện không tối ưu, nhưng vẫn mang lại kết quả ổn định trong nhiều tình huống.

Hệ thống này có tiềm năng được phát triển để áp dụng trong các lĩnh vực như tâm lý học, giáo dục hoặc làm công cụ hỗ trợ trong các cuộc gọi video.rãi trong nhiều lĩnh vực khác nhau của quản lý giao thông và thu thập thông tin.

### *4.1.2 Nhược điểm*

Dù hệ thống đã có thể nhận diện nụ cười trong phần lớn các tình huống, nhưng trong một số trường hợp như khi người đối diện cười mờ, bị che khuất hoặc khi ánh sáng không đủ, độ chính xác của chương trình có thể giảm sút. Bên cạnh đó, việc phát hiện nụ cười trong các video có chất lượng thấp hoặc độ phân giải không cao cũng làm giảm khả năng nhận diện chính xác.

Hệ thống đôi khi có thể không phát hiện được nụ cười trong các tình huống ánh sáng không đồng đều hoặc video có độ rung lắc mạnh, gây khó khăn cho thuật toán trong việc phân tích.

Thêm vào đó, chương trình vẫn chưa được tối ưu hóa hoàn toàn và còn một số hạn chế về khả năng hoạt động trong thời gian thực, đặc biệt là với các video dài

### *4.2. Định hướng tương lai*

Trong tương lai, tôi sẽ tìm cách cải thiện chương trình bằng cách chuyển sang sử dụng các mô hình học sâu mạnh mẽ hơn như YOLO hoặc các mô hình học máy khác như CNN để nhận diện chính xác hơn các khuôn mặt và nụ cười trong các điều kiện ánh sáng kém hoặc video có chất lượng thấp. Điều này sẽ giúp hệ thống hoạt động hiệu quả hơn trong các ứng dụng thực tế, chẳng hạn như trong các cuộc gọi video trực tuyến hoặc trong các cuộc khảo sát cảm xúc.

Đồng thời, việc tích hợp các thuật toán nhận diện khuôn mặt và nụ cười phức tạp hơn sẽ mở ra cơ hội phát triển hệ thống để nhận diện nhiều cảm xúc khác nhau, từ đó hỗ trợ các ứng dụng trong các lĩnh vực như nghiên cứu tâm lý học, giải trí hoặc marketing.

Hệ thống nhận diện này có thể được mở rộng ra các ứng dụng khác như nhận diện cảm xúc trong môi trường công việc hoặc môi trường giáo dục, giúp tăng cường hiệu quả giao tiếp và đánh giá cảm xúc.

# TÀI LIỆU THAM KHẢO

[1] Z. Sun, G. Bebis, R. Miller, On-road vehicle detection: a review, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (5) (2006) 694–711.

[2] P. Viola, M.J. Jones, D. Snow, Detecting pedestrians using patterns of motion and appearance, in: ICCV '03: Proceedings of the International Conference on Computer Vision, Washington, DC, USA, 2003, pp. 734–741.

[3] C.P. Papageorgiou, M. Oren, T. Poggio, A general framework for object detection, in: ICCV '98: Proceedings of the International Conference on Computer Vision, Washington, DC, USA, 1998, pp. 555–562.

[4] R. Lienhart, J. Maydt, An extended set of Haar-like features for rapid object detection, in: ICIP '02: Proceedings of the International Conference on Image Processing, 2002, pp. 900–903.

[5] T. Mita, T. Kaneko, O. Hori, Joint Haar-like features for face detection, in: ICCV '05: Proceedings of the International Conference on Computer Vision, Washington, DC, USA, 2005, pp. 1619–1626.

[6] J. Šochman, J. Matas, Waldboost—learning for time constrained sequential detection, in: CVPR '05: Proceedings of Conference on Computer Vision and Pattern Recognition, Los Alamitos, USA, vol. 2, June 2005, pp. 150–157.

[7] D. Zhang, S.Z. Li, D. Gatica-Perez, Real-time face detection using boosting in hierarchical feature spaces, in: ICPR '04: Proceedings of the International Conference on Pattern Recognition, Washington, DC, USA, 2004, pp. 411–414.

[8] Hewitt, R.: Seeing with OpenCV, part 2. Servo, 4852 (2007)

[9]<http://kdientu.duytan.edu.vn/media/49682/le-dac-thinh-bao-cao-nckh.pdf>

[10] A. Martínez, The AR face database, Technical Report #24, Computer