# PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

**1. Thông tin về sinh viên**

Họ và tên sinh viên: Võ Hoàng Lan Ngọc

Điện thoại liên lạc: 0906882899 Email: ngoc20091921@gmail.com

Lớp: CNTT3-K54 Hệ đào tạo: Chính quy

Đồ án tốt nghiệp được thực hiện tại: trung tâm nghiên cứu và phát triển SVMC của công ty TNHH Samsung Electronics Việt Nam

Thời gian làm ĐATN: Từ ngày 10/02/2014 đến 28/05/2014

**2. Mục đích nội dung của ĐATN**

Xây dựng hệ thống nhận dạng khuôn mặt nhằm bảo mật truy cập ứng dụng trên điện thoại Android

**3. Các nhiệm vụ cụ thể của ĐATN**

* Tìm hiểu các bước xây dựng hệ thống nhận dạng khuôn mặt trên điện thoại di động
* Tìm hiểu phương pháp tăng cường an ninh cho hệ thống nhằm tránh bị qua mặt bởi thủ thuật sử dụng ảnh thay thế cho người thật
* Xây dựng ứng dụng khóa để bảo mật truy cập trên điện thoại Android bằng cách nhận dạng khuôn mặt.

**4. Lời cam đoan của sinh viên:**

Tôi – *Võ Hoàng Lan Ngọc* - cam kết ĐATN là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *TS. Cao Tuấn Dũng*.

Các kết quả nêu trong ĐATN là trung thực, không phải là sao chép toàn văn của bất kỳ công trình nào khác.

|  |  |
| --- | --- |
|  | *Hà Nội, ngày 28 tháng 05 năm 2014*  Tác giả ĐATN  *Võ Hoàng Lan Ngọc* |

**5. Xác nhận của giáo viên hướng dẫn về mức độ hoàn thành của ĐATN và cho phép bảo vệ:**

|  |  |
| --- | --- |
|  | *Hà Nội, ngày 28 tháng 05 năm 2014*  Giáo viên hướng dẫn  *TS. Cao Tuấn Dũng* |

# TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP

Nhận dạng mặt người là một lĩnh vực nghiên cứu của ngành Computer Vision, và cũng được xem là một lĩnh vực của ngành sinh trắc học, tương tự như nhận dạng vân tay. Các thuật toán nhận dạng khuôn mặt đã được nghiên cứu như là một giải pháp mạnh mẽ để xác thực sinh trắc học. Trong bối cảnh điện thoại thông minh đang dần trở nên phổ biến, việc áp dụng phương pháp nhận dạng gương mặt để bảo mật thay cho phương pháp mật khẩu thông thường cũng đang rất được quan tâm.

Do đó, trong đồ án này, chúng tôi tiến hành nghiên cứu và xây dựng hệ thống nhận dạng khuôn mặt phù hợp để thực thi trên điện thoại di động, từ đó áp dụng làm một ứng dụng khóa để bảo vệ việc truy cập trên điện thoại. Song song với việc nhận dạng, chúng tôi còn nghiên cứu thêm phương pháp dùng để tránh bị vượt mặt bằng các thủ thuật giả mạo người truy cập, nhằm tăng cường an ninh cho hệ thống nhận dạng.

Kết quả, đồ án đã xây dựng được một hệ thống nhận dạng khuôn mặt áp dụng để làm ứng dụng khóa trên điện thoại Android, và tìm hiểu được một số phương pháp để ngăn chặn thủ thuật sử dụng ảnh chụp thay thế người thật

# LỜI CẢM ƠN

Để có thể thực hiện đồ án này, trước tiên em xin gửi lời cảm ơn đến các thầy, cô của Viện Công nghệ Thông Tin và Truyền thông cũng như tất cả các thầy cô của trường đại học Bách Khoa Hà Nội đã tận tình dạy dỗ, dìu dắt em trong suốt thời gian em học tập tại trường.

Em xin gửi lời biết ơn sâu sắc đến thầy giáo hướng dẫn, TS. Cao Tuấn Dũng đã trực tiếp hướng dẫn, tận tình giúp đỡ và chỉ bảo em trong suốt quá trình thực hiện đồ án này.

Đồng thời em xin cảm ơn trung tâm nghiên cứu và phát triển SVMC của công ty Samsung Electronic Việt Nam đã tạo điều kiện về môi trường và thiết bị cho em trong quá trình thực hiện hiện đồ án.

Em xin gửi lời cảm ơn đến anh Phạm Quang Vinh và anh Đỗ Đức An của trung tâm SVMC đã tận tình giúp đỡ về trang thiết bị và chỉ bảo em trong thời gian thực tập và làm đồ án trên trung tâm SVMC. Cũng xin gửi lời cảm ơn đến các anh chị trên trung tâm SVMC và các bạn sinh viên STP đã nhiệt tình giúp đỡ trong quá trình thu thập dữ liệu cho việc thực hiện đồ án.

Tôi cũng xin cảm ơn bạn Phạm Đình Hoàng, người đã cùng tôi thực hiện đồ án này. Và cũng xin cảm ơn các bạn sinh viên thuộc chương trình STP đã giúp đỡ và cho tôi những góp ý hữu ích để hoàn thiện đồ án.

Đặc biệt con xin cảm ơn Bố, Mẹ vì đã luôn yêu thương động viên con trong suốt quá trình thực hiện đồ án này.

*Hà Nội, ngày 28 tháng 5 năm 2014*

Tác giả ĐATN

***Võ Hoàng Lan Ngọc***

MỤC LỤC

[PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP 2](#_Toc389135412)

[TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP 3](#_Toc389135413)

[LỜI CẢM ƠN 4](#_Toc389135414)

[MỤC LỤC 5](#_Toc389135415)

[DANH MỤC CÁC HÌNH 7](#_Toc389135416)

[DANH MỤC CÁC BẢNG 8](#_Toc389135417)

[DANH MỤC CÁC TỪ VIẾT TẮT VÀ THUẬT NGỮ 9](#_Toc389135418)

[LỜI NÓI ĐẦU 10](#_Toc389135419)

[PHẦN 1: ĐẶT VẤN ĐỀ, ĐỊNH HƯỚNG VÀ CƠ SỞ LÝ THUYẾT 11](#_Toc389135420)

[1. VẤN ĐỀ ĐẶT RA VÀ ĐỊNH HƯỚNG GIẢI PHÁP 11](#_Toc389135421)

[1.1 Tổng quan về đề tài 11](#_Toc389135422)

[1.2 Vấn đề cần giải quyết 12](#_Toc389135423)

[1.2.1 Hệ thống nhận dạng gương mặt 12](#_Toc389135424)

[1.2.2 Vấn đề phân biệt người thật với hình ảnh để tránh bị qua mặt 13](#_Toc389135425)

[1.2.3 Ứng dụng bảo mật truy cập trên điện thoại Android 14](#_Toc389135426)

[1.3 Định hướng giải pháp 14](#_Toc389135427)

[1.3.1 Quy trình thực hiện nhận dạng khuôn mặt trong hệ thống 14](#_Toc389135428)

[1.3.2 Cấu trúc tổng quan của hệ thống bảo mật truy cập bằng nhận dạng 16](#_Toc389135429)

[2. CƠ SỞ LÝ THUYẾT VÀ CÁC NGHIÊN CỨU LIÊN QUAN 17](#_Toc389135430)

[2.1 Các phương pháp phát hiện khuôn mặt 17](#_Toc389135431)

[2.1.1 Dựa trên đặc trưng Haar-like 17](#_Toc389135432)

[2.1.2 Dựa trên đặc trưng LBP 19](#_Toc389135433)

[2.1.3 Áp dụng Adaboost cho các bộ phát hiện khuôn mặt 21](#_Toc389135434)

[2.2 Các thuật toán nhận dạng gương mặt 23](#_Toc389135435)

[2.2.1 Eigenfaces 23](#_Toc389135436)

[2.2.2 Local binary patterns histograms 28](#_Toc389135437)

[2.3 Các phương pháp nhận diện vật thể sống 29](#_Toc389135438)

[2.3.1 Thực hiện nháy mắt 29](#_Toc389135439)

[2.3.2 Dựa trên dữ liệu dòng quang – optical flow 30](#_Toc389135440)

[2.3.3 Learning Multispectral Reflectance Distributons 31](#_Toc389135441)

[2.3.4 Dựa trên việc phân tích phổ Fourier 31](#_Toc389135442)

[PHẦN 2: CÁC KẾT QUẢ ĐẠT ĐƯỢC 34](#_Toc389135443)

[3. LỰA CHỌN GIẢI PHÁP CHO QUY TRÌNH NHẬN DẠNG KHUÔN MẶT 34](#_Toc389135444)

[3.1 Mục tiêu hướng tới 34](#_Toc389135445)

[3.2 Cơ sở dữ liệu sử dụng để đánh giá 34](#_Toc389135446)

[3.3 Triển khai thực nghiệm và lựa chọn phương pháp cho các pha thực hiện 35](#_Toc389135447)

[3.3.1 Phát hiện khuôn mặt 35](#_Toc389135448)

[3.3.2 Tiền xử lý 36](#_Toc389135449)

[3.3.3 Nhận dạng khuôn mặt 37](#_Toc389135450)

[3.3.4 Tổng kết các phương pháp lựa chọn 41](#_Toc389135451)

[4. ỨNG DỤNG BẢO MẬT TRUY CẬP BẰNG NHẬN DẠNG KHUÔN MẶT 42](#_Toc389135452)

[4.1 Tổng quan về ứng dụng 42](#_Toc389135453)

[4.2 Phân tích ứng dụng 43](#_Toc389135454)

[4.3 Thiết kế ứng dụng 46](#_Toc389135455)

[4.4 Kết quả 48](#_Toc389135456)

[KẾT LUẬN 54](#_Toc389135457)

[TÀI LIỆU THAM KHẢO 55](#_Toc389135458)

# DANH MỤC CÁC HÌNH

[Hình 1: Các module chính của hệ thống 12](#_Toc389133379)

[Hình 2: Hệ thống nhận dạng gương mặt tổng quát 12](#_Toc389133380)

[Hình 3: Mô hình của quy trình nhận dạng khuôn mặt 15](#_Toc389133381)

[Hình 4: Mô hình của quy trình huấn luyện – face training 15](#_Toc389133382)

[Hình 5: Cấu trúc tổng quan của hệ thống 16](#_Toc389133383)

[Hình 6: Đặc trưng Haar-like cơ bản 17](#_Toc389133384)

[Hình 7: Các đặc trưng mở rộng của các đặc trưng Haar-like cơ sở 17](#_Toc389133385)

[Hình 8:Minh họa sự tính toán LBP 19](#_Toc389133386)

[Hình 9: Minh họa toán tử LBP mở rộng với các P,R khác nhau 20](#_Toc389133387)

[Hình 10:Từ trái sang phải, các mẫu vân cơ bản 20](#_Toc389133388)

[Hình 11: Biểu diễn gương mặt bằng LBP 21](#_Toc389133389)

[Hình 12: Mô tả thuật toán boosting 21](#_Toc389133390)

[Hình 13:Mô hình phân tầng kết hợp các bộ phân loại yếu để xác định khuôn mặt 23](#_Toc389133391)

[Hình 14: Biểu diễn các vector riêng thành hình ảnh 26](#_Toc389133392)

[Hình 15: Minh họa không gian con PCA 27](#_Toc389133393)

[Hình 16:Toán tử LBP có thể làm giảm sự khác biệt khi biến đổi độ sáng 28](#_Toc389133394)

[Hình 17:Phương pháp này dễ dàng vượt qua bởi thao tác nháy mắt giả 30](#_Toc389133395)

[Hình 18: Dữ liệu dòng quang của ảnh và người thật 30](#_Toc389133396)

[Hình 19: Phương án đòi hỏi phần cứng mở rộng, 31](#_Toc389133397)

[Hình 20:Phổ Fourier của ảnh và người thật 32](#_Toc389133398)

[Hình 21: Biểu đồ so sánh giá trị giữa ảnh chụp (fake) và người thật (live) 33](#_Toc389133399)

[Hình 22: Một set hình ảnh trong Yale Face Database A 34](#_Toc389133400)

[Hình 23:Kết quả phát hiện gương mặt 35](#_Toc389133401)

[Hình 25:Biểu đồ so sánh tốc độ khi sử dựng đặc trưng Haar like và LBP 35](#_Toc389133402)

[Hình 26: Ví dụ cho một hình ảnh tiền xử lý 36](#_Toc389133403)

[Hình 28: Sơ đồ các bước trong khâu nhận dạng khuôn mặt 37](#_Toc389133404)

[Hình 31: Độ dao động giá trị predict của LBPH 39](#_Toc389133405)

[Hình 32: Độ dao động giá trị predict của Eigenfaces 39](#_Toc389133406)

[Hình 36: Biểu đồ hoạt động cho "Khóa ứng dụng" 43](#_Toc389133407)

[Hình 37: Biểu đồ hoạt động cho "Mở ứng dụng" 44](#_Toc389133408)

[Hình 38: Biểu đồ Usa Case tổng quan 45](#_Toc389133409)

[Hình 39: Màn hình chính của ứng dụng 47](#_Toc389133410)

[Hình 40: Giao diện setting và thay mật khẩu 47](#_Toc389133411)

[Hình 41: Giao diện huấn luyện và mở khóa 48](#_Toc389133412)

# DANH MỤC CÁC BẢNG

[Bảng 1:Tốc độ thực hiện phát hiện mặt trong môi trường smartphone 35](#_Toc389133413)

[Bảng 2:Tốc độ tiền xử lý trên các kích thước ảnh 37](#_Toc389133414)

[Bảng 3:Thời gian thực hiện 7 triệu lần các phép tính cộng trừ nhân chia với các kiểu số khác nhau thực hiện môi trường máy tính (a) và smartphone (b) 38](#_Toc389133415)

[Bảng 4:Tốc độ xử lý thuật toán của Eigenfaces và LBPH 38](#_Toc389133416)

[Bảng 5:Số ảnh thông báo nhận dạng được (chưa biết đúng sai) 40](#_Toc389133417)

[Bảng 6:Số ảnh nhận dạng sai trong số được cho là nhận dạng được ở trên 40](#_Toc389133418)

[Bảng 7:Thời gian nhận dạng đo được trên máy Samsung Galaxy Mega 41](#_Toc389133419)

[Bảng 8: Tổng kết các phương án lựa chọn cho quy trình nhận dạng 41](#_Toc389133420)

[Bảng 9: Đặc tả UC Khóa ứng dụng 45](#_Toc389133421)

[Bảng 10: Đặc tả UC Mở ứng dụng 46](#_Toc389133422)

[Bảng 11: Phân chia các nhóm lớp 46](#_Toc389133423)

[Bảng 12: Kịch bản vận hành của hệ thống 52](#_Toc389133424)

[Bảng 13: Thử nghiệm nhận dạng với điều kiện ánh sáng khác nhau 52](#_Toc389133425)

[Bảng 14: Thử nghiệm nhận dạng với biểu cảm khác nhau 52](#_Toc389133426)

# DANH MỤC CÁC TỪ VIẾT TẮT VÀ THUẬT NGỮ

|  |  |  |
| --- | --- | --- |
| **Số TT** | **Từ** | **Giải nghĩa** |
|  | PCA | Principal component analysis |
|  | LBP | Local binary patterns |
|  | LBPH | Local binary patterns histograms |
|  | Cascade | Phân tầng |
|  | Classifier | Bộ phân loại |
|  | Face Detection | Phát hiện khuôn mặt |
|  | Face Recogniton | Nhận dạng khuôn mặt |
|  | Training | Huấn luyện |
|  | Preprocessing | Tiền xử lý |

# LỜI NÓI ĐẦU

Nhận dạng mặt người (Face Recognition) là một lĩnh vực nghiên cứu của ngành Computer Vision, và cũng được xem là một lĩnh vực của ngành sinh trắc học, tương tự như nhận dạng vân tay. Trong khi nhận dạng vân tay đã áp dụng một cách rộng rãi trên thực tế thì nhận dạng mặt vẫn còn nhiều thách thức và vẫn là một lĩnh vực nghiên cứu thú vị với nhiều người bởi nó được thực hiện trong các điều kiện môi trường không được kiểm soát.

Các thuật toán nhận dạng khuôn mặt đã được nghiên cứu như là một giải pháp mạnh mẽ để xác thực sinh trắc học. Nó được sử dụng trong một loạt các ứng dụng như giám sát, kiểm soát truy cập, hệ thống nhập cư sân bay, các hệ thống này đều đòi hỏi một khả năng xử lý mạnh mẽ, đi kèm với nó là camera có chất lượng cao.

Tuy nhiên, ngày nay máy tính không còn là công cụ trợ giúp độc tôn dành cho con người. Hầu hết chúng ta ai cũng biết đến sự phát triển mạnh mẽ của các loại thiết bị giải trí cầm tay nhỏ gọn, đó chính là smartphone. Với ưu điểm là kích thước bé, đi kèm với nó là các chíp xử lý thông minh tốc độ cao, smartphone có khả năng đảm đương rất nhiều tác vụ giống như đang thao tác trên máy tính. Phát triển phần mềm cho smartphone hiện cũng là xu thế tất yếu.

Ứng dụng công nghệ nhận dạng gương mặt trên smartphone là ý tưởng có tính ứng dụng cao trong bối cảnh điện thoại smartphone đang ngày càng trở nên phổ biến đối với con người. Tôi – sinh viên Võ Hoàng Lan Ngọc – viện công nghệ thông tin và truyền thông, dưới sự hướng dẫn của TS. Cao Tuấn Dũng, đã chọn đề tài cho đồ án tốt nghiệp của mình là “Sử dụng phương pháp nhận dạng khuôn mặt để bảo mật smartphone”. Đề tài này được thực hiện dưới dự bảo trợ của trung tâm nghiên cứu và phát triển SVMC của công ty TNHH Samsung Electronics Việt Nam cùng với một thành viên khác là sinh viên Phạm Đình Hoàng - viện điện tử viễn thông – khóa 54. Nội dung đồ án bao gồm:

* Chương 1: Giới thiệu sơ lược về vấn đề, các bài toán đặt ra
* Chương 2: Tìm hiểu các nghiên cứu liên quan trước đó và cơ sở lý thuyết để thực hiện đồ án
* Chương 3: Trình bày các bước thực hiện cụ thể, đánh giá và lựa chọn các giải pháp đưa ra để đưa vào áp dụng cho việc xây dựng hệ thống nhận dạng
* Chương 4: Thiết kế ứng dụng bảo mật truy cập trên điện thoại Android dựa trên hệ thống nhận dạng gương mặt. Đánh giá quá trình sử dụng của ứng dụng

# PHẦN 1: ĐẶT VẤN ĐỀ, ĐỊNH HƯỚNG VÀ CƠ SỞ LÝ THUYẾT

Phần này sẽ nêu ra bài toán cùng các vấn đề cần giải quyết, định hướng giải pháp cho các vấn đề đó và cơ sở lý thuyết liên quan. Nội dung phần 1 bao gồm

* Đặt vấn đề và đưa ra hướng giải quyết
* Cơ sở lý thuyết và các nghiên cứu liên quan tìm hiểu được

# VẤN ĐỀ ĐẶT RA VÀ ĐỊNH HƯỚNG GIẢI PHÁP

## Tổng quan về đề tài

**Bối cảnh và nhu cầu của người dùng**

Ngày nay, những tiến bộ mới trong khoa học kỹ thuật công nghệ đã giúp ích rất nhiều cho cuộc sống của con người. Một trong những công nghệ tiên tiến đang được áp dụng rộng rãi trong đời sống chính là công nghệ nhận dạng.Các đối tượng của bài toán nhận dạng thì rất phong phú, ví dụ như nhận dạng tiếng nói, nhận dạng vân tay, nhận dạng mã vạch, và nhận dạng khuôn mặt cũng nằm trong số đó.

Trong bối cảnh smartphone đang ngày càng trở bên phổ biến, việc xây dựng hệ thống nhận dạng trên smartphone cũng đang rất được quan tâm. Ứng dụng công nghệ nhận dạng gương mặt trên smartphone chính là ý tưởng được hướng tới khi thực hiện đề tài này. Bài toán cần giải quyết là làm thế nào xây dựng một hệ thống thông minh cho phép phát hiện và nhận dạng gương mặt trên thiết bị di động để từ đó áp dụng để làm phương thức bảo mật thay cho phương thức nhập mật khẩu thông dụng. Bên cạnh đó, còn phải đưa ra phương pháp để tăng cường tính bảo mật cho quá trình nhận dạng này, tránh để bị người dùng qua mặt bằng một số kỹ thuật đơn giản như sử dụng hình ảnh 2D.

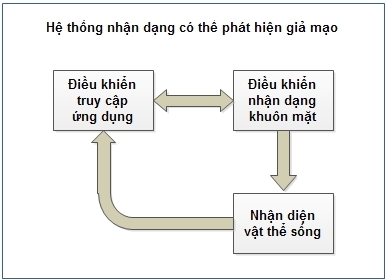
**Mục tiêu của ứng dụng**

Theo như đề tài nhận được từ phía Samsung, mục tiêu của ứng dụng là xây dựng một hệ thống nhận dạng gương mặt dùng để bảo mật truy cập. Trong quá trình nhận dạng, hệ thống sẽ đồng thời thực hiện kiểm tra để tránh bị qua mặt.

Cụ thể các vấn đề cần giải quyết bao gồm:

* Xây dựng một hệ thống nhận dạng khuôn mặt để bảo mật truy cập trên điện thoại di động
* Đưa ra phương án kiểm tra và phát hiện vật thể sống để phân biệt hình ảnh và người thật

Các giải pháp được đưa ra sẽ được sử dụng để xây dựng một ứng dụng khóa, có thể đáp ứng được việc xử lý nhận dạng bằng camera thời gian thực trên smartphone, sử dụng để khóa các ứng dụng đã được cài sẵn trong điện thoại và thực hiện mở khóa bằng cách nhận dạng người dùng đã được ghi nhớ trước đó.



Hình 1: Các module chính của hệ thống

**Nhiệm vụ của đồ án**

Trong đồ án này, tôi chủ yếu thực hiện nhiệm vụ điều khiển nhận dạng khuôn mặt và một phần của điều khiển truy cập ứng dụng, cụ thể các nhiệm vụ cần giải quyết là

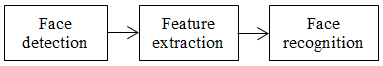
* Tìm hiểu các bước để thực hiện quy trình nhận dạng khuôn mặt, đánh giá, lựa chọn thuật toán được sử dụng để giải quyết vấn đề phù hợp với môi trường di động.
* Áp dụng để xây dựng hệ thống bảo mật truy cập, mở khóa bằng nhận dạng khuôn mặt

Ngoài ra, tôi còn tìm hiểu sơ qua một số phương pháp tăng cường an ninh để tránh bị qua mặt bởi thủ thuật sử dụng hình ảnh giả, tuy nhiên sẽ không đi sâu vào cài đặt module này.

## Vấn đề cần giải quyết

### Hệ thống nhận dạng gương mặt

Một số các tiếp cận [1,2] định nghĩa hệ thống nhận dạng gương mặt là một quy trình xử lý gồm 3 bước



Hình 2: Hệ thống nhận dạng gương mặt tổng quát

Trong đó:

* **Face detection** – phát hiện gương mặt: là quá trình trích rút gương mặt từ hình ảnh
* **Feature extraction** – trích xuất đặc trưng: những đặc trưng có thể một số vùng của gương mặt, hoặc độ thay đổi giữa các vùng,…
* **Face recognition** – nhận dạng gương mặt: là quá trình nhận dạng; so sánh, phân loại, trả về định danh có trong cơ sở dữ liệu của gương mặt đầu vào, dựa vào các đặc trưng đã được trích xuất.

Tuy nhiên, quá trình trích xuất đặc trưng phụ thuộc chủ yếu vào thuật toán được chọn để nhận dạng. Do đó để xây dựng hệ thống nhận dạng gương mặt, bài toán cần giải quyết là:

* Phát hiện gương mặt (face detection)
* Nhận dạng gương mặt (face recogntion).

Ngoài ra, theo nghiên cứu [4] đối với bất kỳ hệ thống nhận dạng nào, các dữ liệu tạp nhiễu chính là yếu tố hàng đầu gây suy giảm độ chính xác. Do đó, quá trình loại bỏ các dữ liệu thừa luôn cần thiết để cải thiện độ chính xác nhận dạng. Quá trình này được gọi là tiền xử lý.

Cơ sở lý thuyết cụ thể được đưa ra để giải quyết sẽ được trình bày trong mục 2.

### Vấn đề phân biệt người thật với hình ảnh để tránh bị qua mặt

Hệ thống nhận dạng khuôn mặt thực hiện đánh giá dựa trên sự tương đồng giữa ảnh đầu vào và tập ảnh huấn luyện, nên các hệ thống nhận dạng khuôn mặt cơ bản đều dễ dàng bị vượt qua khi người truy cập trái phép tạo ra các hình ảnh đầu vào mô phỏng người chủ sở hữu.

Người truy cập trái phép có thể sử dụng một số thủ thuật đơn giản để dễ dàng vượt qua hệ thống này. Điển hình nhất là ba thủ thuật chính:

* Sử dụng ảnh chân dung của chủ sở hữu: Với sự bùng nổ của internet và mạng xã hội, có thể dễ dàng thu thập được ảnh có chứa khuôn mặt của chủ sở hữu. Đây là thủ thuật phổ biến nhất khi đối tượng muốn đánh lừa hệ thống nhận dạng
* Sử dụng video có hình ảnh của chủ sở hữu: Không phổ biến và khó thu thập dữ liệu hơn so với cách sử dụng hình ảnh trên nhưng sử dụng video chứa chân dung của chủ sở hữu vẫn được dùng để qua mặt hệ thống.
* Sử dụng mặt nạ: tạo ra các mô hình 3D như các bức tượng, các mặt nạ với nhiều chất liệu khác nhau để mô phỏng đối tượng sở hữu, là thủ thuật hết sức tinh vi, đòi hỏi nhiều công sức và có khả năng đánh lừa hệ thống nhận dạng khuôn mặt ở mức độ cao tuy nhiên hạn chế khả năng sử dụng bởi người dùng phổ thông.

Trong khuôn khổ đồ án này, ta sẽ tìm hiểu một số giải pháp phát hiện dấu hiệu sống để tăng cường bảo mật cho hệ thống nhận dạng khuôn mặt, giải quyết được tình huống khi kẻ gian sử dụng ảnh chụp của chủ sở hữu để đánh lừa hệ thống. Tuy nhiên, ta sẽ không đi sâu vào vấn đề cài đặt module này

### Ứng dụng bảo mật truy cập trên điện thoại Android

Như đã trình bày ở trên, quy trình nhận dạng gương mặt sẽ được sử dụng để xây dựng một ứng dụng khóa, áp dụng phương thức bảo mật là nhận dạng khuôn mặt thay cho phương thức nhập mật khẩu thông dụng. Ứng dụng này phải đáp ứng được việc xử lý nhận dạng bằng camera thời gian thực trên smartphone, dùng để khóa các ứng dụng đã được cài sẵn trong điện thoại và thực hiện mở khóa bằng cách nhận dạng người dùng đã được ghi nhớ trước đó.

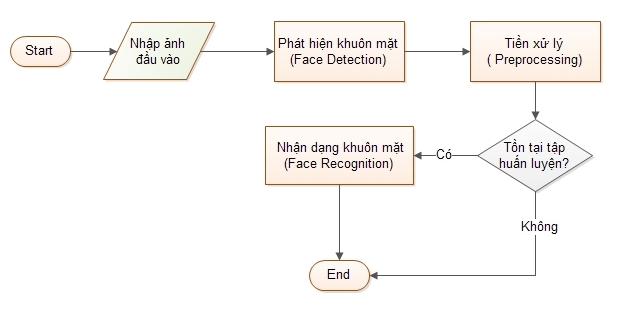
Ứng dụng được thực hiện trên nền tảng Android, với sự hỗ trợ của thư viện xử lý ảnh mã nguồn mở OpenCV

## Định hướng giải pháp

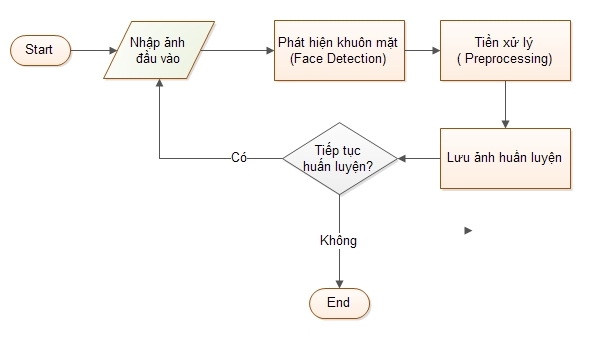
### Quy trình thực hiện nhận dạng khuôn mặt trong hệ thống

Trong hệ thống nhận dạng được xây dựng, quá trình nhận dạng sẽ liên quan đến các bước sau:

* **Face detection**: là quá trình xác định vùng có gương mặt trên một hình ảnh, ko quan tâm đó là ai, chỉ cần là mặt người
* **Face preprocessing**: tiền xử lý - là quá trình điều chỉnh hình ảnh gương mặt nhìn rõ ràng hơn và tương đồng với những gương mặt khác
* **Face training**: thu thập và học biết gương mặt - là quá trình thu thập lưu giữ nhiều gương mặt đã được tiền xử lý và học cách để nhận ra họ
* **Face recognition**: nhận diện gương mặt - là quá trình kiểm tra trong những gương mặt đã được thu thập, gương mặt nào giống với mặt trong camera nhất

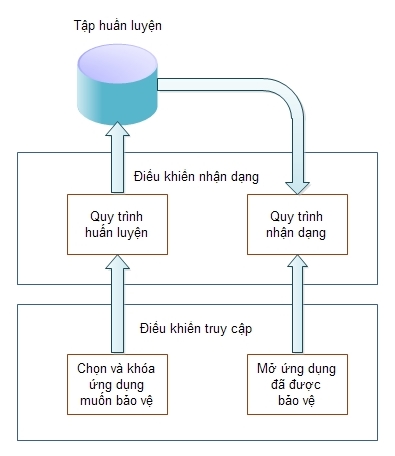


Hình 3: Mô hình của quy trình nhận dạng khuôn mặt



Hình 4: Mô hình của quy trình huấn luyện – face training

### Cấu trúc tổng quan của hệ thống bảo mật truy cập bằng nhận dạng



Hình 5: Cấu trúc tổng quan của hệ thống

* Từ đây, ta có các chức năng chính cần cài đặt của hệ thống bao gồm
  + Xác định các ứng dụ ng được lựa chọn bảo vệ bởi người dùng
  + Huấn luyện khuôn mặt: lưu trữ các khuôn mặt của chủ sở hữu để hệ thống có thể nhận dạng
  + Mở khóa ứng dụng: thực hiện việc nhận dạng để mở khóa khi có ứng dụng đã bảo vệ được người dùng kích hoạt.

# CƠ SỞ LÝ THUYẾT VÀ CÁC NGHIÊN CỨU LIÊN QUAN

## Các phương pháp phát hiện khuôn mặt

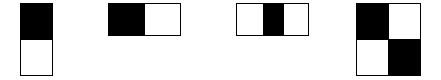
### Dựa trên đặc trưng Haar-like

Bộ phát hiện đối tượng sử dụng bộ phân loại cascade dựa trên đặc trưng Haar là một phương phát phát hiện vật thể hiệu quả được đề xuất bởi Paul Viola và Michael Jones.

Các phát hiện gương mặt này là một hướng tiếp cận dựa trên diện mạo, sử dụng bộ phân loại mạnh Adaboost là sự kết hợp của các bộ phân loại yếu dựa trên các đặc trưng Haar-like để, huấn luyện từ rất nhiều hình ảnh positive (có chứa đối tượng) và negative (không chứa đối tượng), sau đó dùng để phát hiện đối tượng trong một bức hình khác

* **Đặc trưng Haar-like**

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:



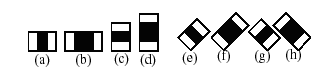
Hình 6: Đặc trưng Haar-like cơ bản

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

1. . Đặc trưng cạnh (edge features):

http://1.bp.blogspot.com/-VfVkuegsiwU/TzI414grUcI/AAAAAAAAAew/yWbLYpIyYeQ/s1600/image002.png

1. Đặc trưng đường (line features):



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

http://4.bp.blogspot.com/-J3I13L81GDw/TzI5PBONVRI/AAAAAAAAAfA/lcj_rZ2k8e4/s1600/Haarlike-3.png

Hình 7: Các đặc trưng mở rộng của các đặc trưng Haar-like cơ sở

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

**f(x) = Tổngvùng đen(pixel) - Tổngvùng trắng(pixel)**

Như vậy ta có thể thấy rằng, để tính các giá trị của đặc trưng Haar-like, ta phải tính tổng của các vùng pixel trên ảnh. Nhưng để tính toán các giá trị của các đặc trưng Haar-like cho tất cả các vị trí trên ảnh đòi hỏi chi phí tính toán khá lớn, không đáp ứng được cho các ứng dụng đòi hỏi tính run-time. Do đó Viola và Jones đưa ra một khái niệm gọi là *Integral Image* để tính toán nhanh cho các đặc trưng cơ bản. Sau này, Lienhart kế thừa gọi *Integral Image* là SAT – *Summed Area Table* và đưa ra thêm khái niệm RSAT – *Rotated Summed Area Table* dùng để tính toán nhanh cho các đặc trưng xoay 1 góc 45o*.*

***Integral Image*** là một mảng 2 chiều với kích thước bằng với kích của ảnh cần tính các đặc trưng Haar-like, với mỗi phần tử của mảng này được tính bằng cách tính tổng của điểm ảnh phía trên (dòng-1) và bên trái (cột-1) của nó. Bắt đầu từ vị trí trên-bên trái đến vị trí dưới-phải của ảnh, việc tính toán này đơn thuần chỉ đựa trên phép cộng số nguyên đơn giản, do đó tốc độ thực hiện rất nhanh

http://2.bp.blogspot.com/-_sc_-nfrj1s/TzI5hpdtXFI/AAAAAAAAAfQ/LAntPDhSXCI/s1600/IntegralImag-2-new.gif.jpg

P(x,y)

**Cách tính Integral Image của ảnh**

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

A

C

B

D

P1

P2

P3

P4

**Ví dụ cách tính nhanh tổng các điểm ảnh của vùng D trên ảnh**

Giả sử ta cần tính tổng các giá trị điểm ảnh của vùng D như trong hình trên, ta có thể tính 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:

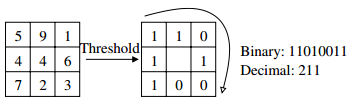
Tiếp theo, để chọn các đặc trưng Haar-like dùng cho việc thiết lập ngưỡng, Viola và Jones sử dụng một phương pháp máy học được gọi là AdaBoost. AdaBoost sẽ kết hợp các bộ phân loại yếu để tạo thành một bộ phân loại mạnh. Với bộ phân loại yếu chỉ cho ra câu trả lời chính xác chỉ hơn viện đoán một cách ngẫn nhiên một chút, còn bộ phân loại mạnh có thể đưa ra câu trả lời chính xác trên 60%.

### Dựa trên đặc trưng LBP

Ý tưởng cơ bản của LBP - Local binary patterns cũng gần giống như Haar, tuy nhiên nó sử dụng các biểu đồ so sánh cường độ điểm ảnh, như cạnh, góc, khu vực bằng phẳng.

* **Tổng quát LBP**

LBP là một toán tử kernel 3×3, nó tổng quát hóa cấu trúc không gian cục bộ của một ảnh. Ojala và các đồng nghiệp đã giới thiệu phương pháp LBP và chỉ ra khả năng phân tách cao của chúng cho sự phân lớp vân. Tại một vị trí pixel (xc, yc) cho trước, LBP được định nghĩa như một chuỗi nhị phân có trật tự dựa trên sự so sánh giá trị điểm ảnh của pixel trung tâm (xc, yc) và 8 pixel lân cận của nó. Như vậy mỗi pixel sẽ được biểu diễn bởi một chuỗi nhị phân, giá trị thập phân của chuỗi nhị phân này chính là giá trị của pixel trung tâm trong sự biểu diễn bởi toán tử LBP.



Hình 8:Minh họa sự tính toán LBP

Giá trị thập phân của chuỗi LBP có thể được biểu diễn như sau

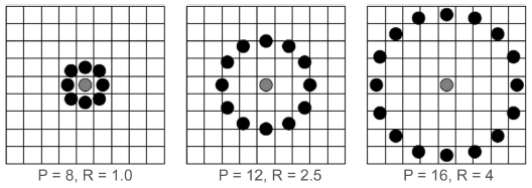
Với gc là giá trị điểm ảnh của pixel trung tâm (xc, yc), gn là giá trị điểm ảnh của pixel thứ n trong 8 pixel lân cận của pixel trung tâm. Hàm sn được định nghĩa như sau:

s(x) =

Sau đó, có thể mở rộng toán tử LBP cũ đến một lân cận tròn với các bán kính khác nhau.

* **Toán tử LBP mở rộng**

Với việc sử dụng toán tử LBP mở rộng, LBPP, R kí hiệu sự xem xét đến P pixels lân cận trên một vòng tròn có bán kính.



Hình 9: Minh họa toán tử LBP mở rộng với các P,R khác nhau

Giá trị các pixel được nội suy cho các điểm không nằm trong tâm của một pixel

Nếu tọa độ của pixel tâm là (xc, yc) thì tọa độ của P pixel lân cận trên đường tròn tâm (xc, yc) bán kính R (tính theo đường tròn lượng giác) là:

xp = xc + Rcos(2p/P)

yp = yc + Rsin(2p/P), p = {0,1,…,P-1}

Một mở rộng khác là cái được gọi là các mẫu đồng nhất. Một LBP được gọi là đồng nhất nếu nó chứa tối đa hai sự chuyển đổi bit từ 0 đến 1 hoặc từ 1 đến 0. Ví dụ như 00000000, 00011110 và 10000011 là các mẫu đồng nhất. Ojala và các đồng nghiệp đã kết luận : trong thông tin cấu trúc của một bức ảnh, các mẫu đồng nhất chiếm 90% khi dùng vùng lân cận (8,1) và khoảng 70% với vùng (16,2)

Ký hiệu cho toán tử LBP đồng nhất: .

Việc sử dụng LBP đồng nhất có 2 điểm lợi. Thứ nhất là tiết kiệm bộ nhớ, vì nếu xét LBP tổng quát ta có 2p mẫu có thể, nhưng nếu chỉ xét mẫu đồng nhất, ta có tối đa P(P+1)+2 mẫu có thể. Lợi điểm thứ 2 là chỉ phát hiện những mẫu vân cục bộ quan trọng như các điểm, điểm cuối đường thẳng, biên cạnh và các góc



Hình 10:Từ trái sang phải, các mẫu vân cơ bản

điểm chấm, điểm chấm nhạt, điểm cuối đường thẳng, biên cạnh, góc được phát hiện bởi LBPu2

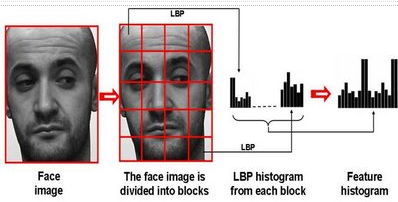
* **Biểu diễn gương mặt trên LBP**

Mỗi gương mặt đều có thể xem như được cấu tạo từ các mẫu nhỏ, dễ dàng được phát hiện bởi toán tử LBP. Ahonen và các đồng nghiệp đã giới thiệu một cách biểu diễn khuôn mặt dựa trên LBP để dùng cho nhận diện gương mặt. Để chiếu đến các thông tin hình dạng của mặt, họ chia ảnh chụp gương mặt ra làm nhiều vùng nhỏ không chồng lên nhau R0, R1,…RM. Biểu đồ LBP được trích xuất từ từng vùng con, sau đó nối vào nhau trở thành một biểu đồ mở rộng được định nghĩa như sau

Trong đó n là số các nhãn khác nhau được tạo ra bởi toán tử LBP và:

*I*

Biểu đồ sẽ bao gồm thông tin về phân phối của các mẫu cục bộ như góc, điểm, vùng biên, trên toàn bộ ảnh

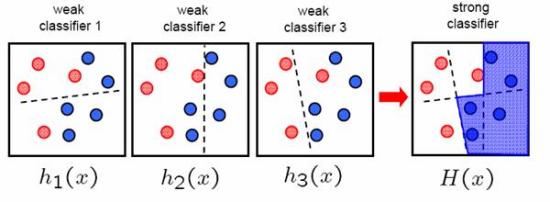


Hình 11: Biểu diễn gương mặt bằng LBP

### Áp dụng Adaboost cho các bộ phát hiện khuôn mặt

Trong quá trình huấn luyện, bộ phân loại phải duyệt qua tất cả các đặc trưng của các mẫu trong tập *huấn luyện*. 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.

* ***Hướng tiếp cận boosting***



Hình 12: Mô tả thuật toán boosting

Để hiểu cách hoạt động của thuật toán boosting, ta xét một bài toán phân loại 2 lớp (mẫu cần nhận dạng chỉ thuộc một trong hai lớp) với *D* là tập huấn luyện gồm có *n* mẫu. Trước tiên, chúng ta sẽ chọn ngẫu nhiên ra *n1* mẫu từ tập *D* (*n1<n*) để tạo tập *D1*. Sau đó, chúng ta sẽ xây dựng bộ phân loại yếu đầu tiên *C1* từ tập *D1*. Tiếp theo, chúng ta xây dựng tập *D2* để huấn luyện bộ phân loại *C2*. *D2* sẽ được xây dựng sao cho một nửa số mẫu của nó được phân loại đúng bởi *C1* và nửa còn lại bị phân loại sai bởi *C1*. Bằng cách này, *D2* chứa đựng những thông tin bổ sung cho *C1*. Bây giờ chúng ta sẽ xây huấn luyện *C2* từ *D2*.

Tiếp theo, chúng ta sẽ xây dựng tập *D3* từ những mẫu không được phân loại tốt bởi sự kết hợp giữa *C1* và *C2*: những mẫu còn lại trong *D* mà *C1* và *C2* cho kết quả khác nhau. Như vậy, *D3* sẽ gồm những mẫu mà *C1 và C2* hoạt động không hiệu quả. Sau cùng, chúng ta sẽ huấn luyện bộ phân loại *C3* từ *D3*.

Bây giờ chúng ta đã có một bộ phân loại mạnh: sự kết hợp *C1*, *C2* và *C3*. Khi tiến hành nhận dạng một mẫu X, kết quả sẽ được quyết định bởi sự thỏa thuận của 3 bộ *C1*, *C2* và *C3*: Nếu cả *C1* và *C2* đều phân *X* vào cùng một lớp thì lớp này chính là kết quả phân loại của *X*; ngược lại, nếu *C1* và *C2* phân X vào 2 lớp khác nhau, C3 sẽ quyết định *X* thuộc về lớp nào

* ***Adaboost***

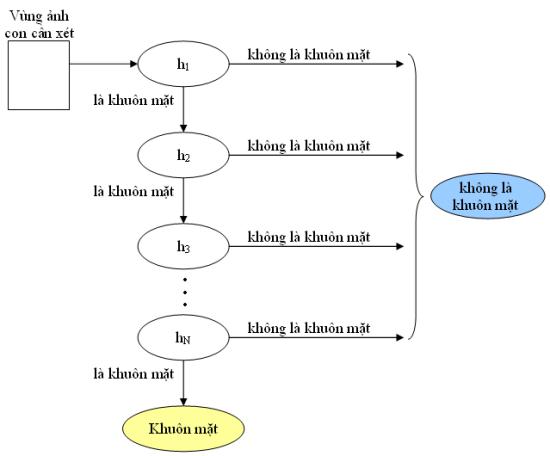
AdaBoost là một bộ phân loại mạnh phi tuyến phức dựa trên hướng tiếp cận boosting được Freund và Schapire đưa ra vào năm 1995. Adaboost cũng hoạt động trên nguyên tắc kết hợp tuyến tính các bộ phân loại yếu để hình thành một bộ phân loại mạnh.

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

Hầu hết các bộ phân lớp tốt cần nhiều thời gian để có các kết quả phân lớp bởi vị chúng cần xem xét đến một lượng lớn các đặc trưng của mẫu. Cấu trúc cascade – phân tầng của các bộ phân lớp mạnh đã được đề xuất để giảm thời gian thực thi và giảm tỷ lệ phát hiện sai

Thư viện OpenCV hỗ trợ cả hai bộ phân loại phân tầng (Cascade Classifier) dựa trên đặc trưng Haar like và đặc trưng LBP

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 sau:



Hình 13:Mô hình phân tầng kết hợp các bộ phân loại yếu để xác định khuôn mặt

## Các thuật toán nhận dạng gương mặt

Trên thực tế người ta hay chia các phương pháp nhận dạng mặt ra làm 3 loại: phương pháp tiếp cận toàn cục (như Eigenfaces-PCA, Fisherfaces-LDA), phương pháp tiếp cận dựa trên các đặc điểm cục bộ (như LBP, Gabor wavelets) và phương pháp lai (hybrid, là sự kết hợp của hai phương pháp toàn cục và local feature). Phương pháp dựa trên các đặc điểm cục bộ đã được chứng minh là ưu việt hơn khi làm việc trong các điều kiện không có kiểm soát.

Trong đồ án này, ta chỉ đi vào xem xét phương pháp tiếp cận toàn cục và phương pháp dựa trên đặc điểm cục bộ

### Eigenfaces

Vấn đề gặp phải khi biểu diễn hình ảnh được đưa là chiều không gian. Một bức ảnh xám 2 chiều pxq trải lên một không gian vector m=pq chiều, một bức ảnh 100x100 pixel sẽ nằm trên một khôn gian 10000 chiều, nhưng không phải tất cả các chiều đều có ích. Ta chỉ có thể đưa ra quyết định khi có sự chênh lệch trong dữ liệu, vậy nên cần tìm ra thành phần chứa được nhiều thông tin nhất

Eigenfaces (khuôn mặt riêng) là tên được đặt cho một tập hợp vector riêng khi chúng được sử dụng cho vấn đề nhận dạng mặt người. Hướng tiếp cận sử dụng Eigenfaces – hay còn gọi là PCA (Principal Component Analysis) - cho việc nhận dạng được phát triển bởi Sirovich và Kirby và được sử dụng bởi Matthew Turk và Alex Pentland [5] trong phân loại khuôn mặt.

Các vector riêng có nguồn gốc từ ma trận hiệp phương sai của phân phối xác xuất dựa trên không gian vector nhiều chiều của các hình ảnh khuôn mặt. Điều này làm tiêu giảm kích thước bằng cách thiết lập lại một bộ hình ảnh cơ bản nhỏ hơn của các ảnh ban đầu. Sự phân loại có thể được thực hiện bằng cách so sánh cách mà khuôn mặt được biểu diễn bởi tập cơ bản

Phương án dựa trên một hướng tiếp cận lý thuyết thông tin về việc phân tách các hinh ảnh gương mặt thành một tập các hình ảnh đặc trưng nhỏ hơn được gọi là "eigenfaces" , thành phần chính của tập huấn luyện ban đầu của các hình ảnh mặt

Việc nhận dạng được thực hiện bằng cách chiếu một hình ảnh mới vào không gian con được trải đầy bằng các eigenface (gọi là không gian mặt - Face space), sau đó phân loại mặt bằng cách so sánh vị trí của nó trong không gian với những cá nhân đã biết

* **Thuật toán**

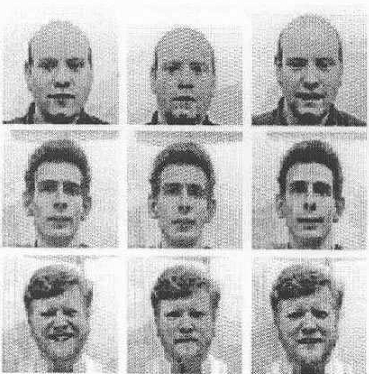
Về mặt toán học, phương pháp phân tích thành phần chính sẽ là xem mọi bức ảnh trong tập huấn luyện như một vector nhiều chiều. Vector riêng của ma trận hiệp phương sai của các vector này sẽ pha trộn sự biến đối giữa các hình ảnh khuôn mặt. Mỗi ảnh trong tập huấn luyện sẽ mang một vector riêng, nó có thể được hiển thị như một eigenface (khuôn mặt riêng) đại diện cho sự đóng góp của nó trong những biến đổi giữa các bức ảnh.

|  |
| --- |
| Cho X = {x1, x2,…, xn} là vector ngẫu nhiên với x1 Rd  (xi biểu diễn một ảnh i trong tập training)   1. Tính toán trung bình 2. Tính toán ma trận hiệp phương sai S 3. Tính toán giá trị riêng (eigenvalues ) và vector riêng vi (eigenvectors)   (Các hình ảnh mặt được biểu diễn bởi được gọi là eigenfaces)   1. Sắp xếp các vector riêng theo thứ tự giảm dần giá trị riêng. Chỉ giữ K vector riêng ứng với k giá trị lớn nhất   Vector x quan sát thấy được biểu diễn với trong đó W = {w1, w2,…, wk}  x có thể được tái tạo từ PCA với |

* **Tính toán các Eigenfaces**

**Bước 1:**

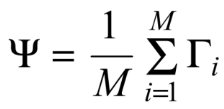
Sử dụng các ảnh khuôn mặt *I1, I2, … In*(tập các khuôn mặt huấn luyện) với khuôn mặt phải *chính diện* & tất cả ảnh phải *cùng kích thước.*



**Bước 2:** Biểu diễn n ảnh thành vector

**Bước 3:** Tính vector khuôn mặt trung bình theo công thức

Với M là số ảnh trong tập luyện, Γi là vector 1 chiều (N2×1) đại diện cho mỗi ảnh.



Ψ là vector trung bình (kích thước (N2×1) của tập tất cả các Γi trên; Ψ còn được gọi là vector trung bình mặt của tập luyện

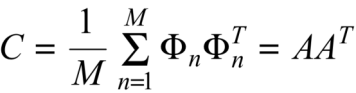
**Bước 4:** Trừ đi giá trị trung bình



Trong đó:

* Φi là vector sai số ứng với mỗi ảnh,
* Γi là vectot 1 chiều của các ảnh,
* Ψ là vector trung bình mặt.

**Bước 5:** Tính ma trận hiệp phương sai



Trong đó

* C là ma trận hiệp phương sai (kích thước N2xN2)
*  sẽ có kích thước là *N2xM*

**Bước 6:**  Tính các giá trị riêng và vector đặc trưng của ma trận hiệp phương sai

Để tính các trị riêng và vector đặc trưng của ma trận hiệp biến ta phải tính với ma trận C hay chính là tính cho ma trận AAT .Tuy nhiên điều này là không khả thi, do C là ma trận N2xN2 nên số chiều là quá lớn, khối lượng tính toán là vô cùng nhiều (đặc biệt khi kích thước ảnh luyện lớn thì N lớn, N và N2xN2 là cực kỳ lớn

Ta thấy 2 ma trận AAT và ATA luôn có chung trị triêng và vector riêng thì có liên hệ với nhau bởi biểu thức *ui = Avi* mà ma trận ATA lại có số chiều ít hơn hẳn (MxM) nên ta sẽ chuyển vè mà trận ATA

Sau khi tính toán ta được M vector riêng của AAT (*ui = Avi)* tương ứng với M giá trị riêng

Chuẩn hóa các vector riêng ui về vector đơn vị

()

**Bước 7:**  Sắp xếp các vector riêng theo thứ tự bằng các giá trị riêng, chỉ giữ K vector riêng ứng với K giá trị riêng lớn nhất



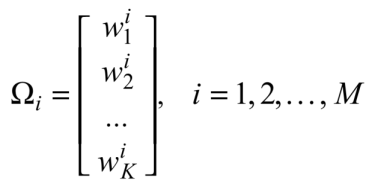
Hình 14: Biểu diễn các vector riêng thành hình ảnh

* **Lựa chọn các thành phần và xây dựng vector đặc trưng**

Để giảm thiểu tối đa số chiều cũng như giảm độ phức tạp tính toán, ta chỉ giữ lại K vector riêng (tương ứng K giá trị riêng lớn nhất). Khi đó mỗi khuôn mặt sau khi trừ đi giá trị trung bình sẽ được đại diện bởi K vector riêng:

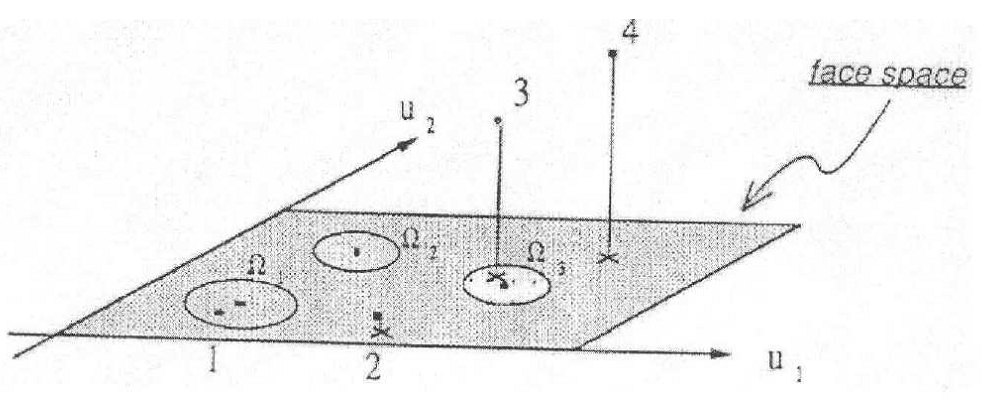
|  |  |  |
| --- | --- | --- |
| **𝛷**i– mean = |  |  |

Mỗi khuôn mặt training **𝛷**i sẽ được biểu diễn bởi một vector Ωi



Trong đó (j = 1..K) là các trọng số tương ứng với K vector riêng, Ωi là vector đại diện cho khuôn mặt thứ I trong huấn luyện

* **Nhận dạng gương mặt**
* Chiếu tất cả ảnh mẫu training vào không gian con PCA
* Chiếu ảnh truy vấn vào không gian con PCA



Đây là các vector x, ảnh khuôn mặt “gốc” ban đầu

Chiếu các vector x vào trong *“không gian khuôn mặt”* ta được các vector y

Đây là các vector cơ sở của không gian khuôn mặt (được gọi là *“EIGENFACE”)*

Hình 15: Minh họa không gian con PCA

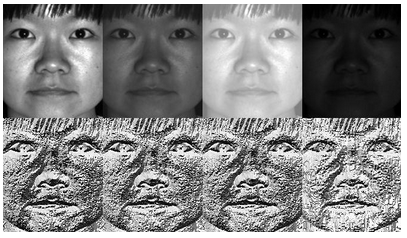
* Tìm khoảng cách gần nhất giữa các hình chiếu của ảnh huấn luyện với hình chiếu của ảnh truy vấn
  + - Bước này chúng ta sẽ tính khoảng cách Euclide của ma trận được đưa vào so với không gian mặt. Tức là ta sẽ đi tính khoảng cách tới từng bức ảnh trong tập ảnh luyện và tìm ra khoảng cách tới bức ảnh huấn luyện gần nhất trong không gian mặt.



* + - Tức ta tìm khuôn mặt thứ *l* trong tập mẫu có khoảng cách gần nhất với khuôn mặt cần nhận dạng
* Sau khi tính được khoảng cách Euclide, ta so sánh với khoảng cách ngưỡng Tr (khoảng cách ngưỡng này được xác định thực nghiệm tùy thuộc vào từng tập ảnh luyện và tùy thuộc vào số vector riêng được giữ lại).
* Nếu er < Tr  thì đó được nhận diện là khuôn mặt thứ l trong tập training còn không thì Γ không là khuôn mặt nào đã cho

### Local binary patterns histograms

Thuật toán LBPH bắt đầu từ việc không nhìn bức ảnh như một vector không gian nhiều chiều, mà chỉ mô tả các đặc trưng cục bộ của một đối tượng. Ý tưởng cơ bản của mẫu nhị phân cục bộ (Local Binary Pattern) là tóm gọn cấu trúc cục bộ của một hình ảnh bằng cách so sánh từng pixel với các lân cận của nó, xem điểm ảnh trung tâm như một ngưỡng. Mô tả cụ thể đã được nhắc đến ở phần 2.1.2. Bằng cách định nghĩa này, toán tử LBP bất biến đối với bất kỳ sự biến đổi độ sáng “đều” và bảo toàn trật tự mật độ các pixel trong một lân cận cục bộ



Hình 16:Toán tử LBP có thể làm giảm sự khác biệt khi biến đổi độ sáng

* **Nhận dạng bằng LBP histogram**

Sau khi biểu diễn gương mặt thu được thành biểu đồ LBP, ta sử dụng thuật toán so sánh 2 biểu đồ để tìm ra khoảng cách ngắn nhất giữa biểu đồ từ hình ảnh đầu vào với các biếu đồ có được từ tập trainning.

Để so sánh 2 histograms, (H1 và H2 ), ta phải có được một thông số d diễn tả độ giống nhau giữa 2 biểu đồ

OpenCV cung cấp 4 phương pháp so sánh để đưa ra thông số này

1. Correlation
2. Chi-Square
3. Intersection
4. Bhattacharyya

Kết quả thực tế [14] cho thấy với Correlation và Intersection, số liệu càng cao thực hiện càng chính xác hơn và Chi-Square và Bhattacharyya số liệu càng thấp thì càng đánh giá đúng

## Các phương pháp nhận diện vật thể sống

### Thực hiện nháy mắt

Đây là phương pháp phổ biến trên các hệ thống nhận diện khuôn mặt hiện nay:

* không đòi hỏi thêm phần cứng thực hiện chức năng
* yêu cầu sự tương tác của người sử dụng.

Cử động nháy mắt là hành vi tự nhiên và thường xuyên của con người, trung bình một phút con người nháy mắt từ 15 đến 30 lần và hành vi này thường kéo dài trung bình 250 mili giây. Với các camera được trang bị cho điện thoại hiện nay có tốc độ khung hình trên giây tối thiểu là 15 fps và thời gian của mỗi khung hình không quá 70 mili giây nên các camera này hoàn toàn có thể nhận biết được cử động nháy mắt.

Về cơ bản, phương pháp này sẽ thu một loạt ảnh( từ 5-8 bức ảnh) của người đang yêu cầu mở khóa. Sau đó, sẽ phân tích vùng ảnh chụp mắt của người sử dụng để phát hiện cử động nháy mắt.

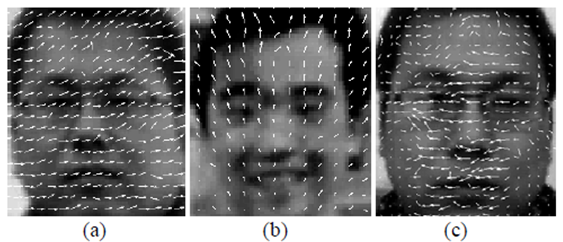
Tuy nhiên, phương pháp này đòi hỏi người dùng thực hiện thao tác nháy mắt nhiều lần khi mở khóa, điều khiến cho không ít người sử dụng khó chịu. Bên cạnh đó, người dùng mạo danh vẫn có thể vượt qua hệ thống bằng cách tạo thao tác nháy mắt giả mạo bằng cách khoét vùng mắt trên ảnh của người chủ thiết bị mà biến bức ảnh đó thành "mặt nạ biết nháy mắt"



Hình 17:Phương pháp này dễ dàng vượt qua bởi thao tác nháy mắt giả

### Dựa trên dữ liệu dòng quang – optical flow

Phương pháp này [7] dựa trên tính chất khác nhau của dữ liệu dòng quang sinh ra do một bề mặt phẳng và một vật thể có bề mặt gồ ghề sinh ra khi chuyển động. Phương pháp này thu thập một loạt ảnh (5-8 ảnh) của người sử dụng trước camera, xử lý dữ liệu dòng quang để phát hiện các dấu hiệu sống chuyển động của khuôn mặt (bề mặt không phẳng).



Hình 18: Dữ liệu dòng quang của ảnh và người thật

**a. dữ liệu dòng quang của ảnh 2D được dịch chuyển tịnh tiến trước camera**

**b. dữ liệu dòng quang của ảnh 2D được uốn cong, làm biến dạng bề mặt trước camera**

**c. dữ liệu dòng quang của người thật trước camera**

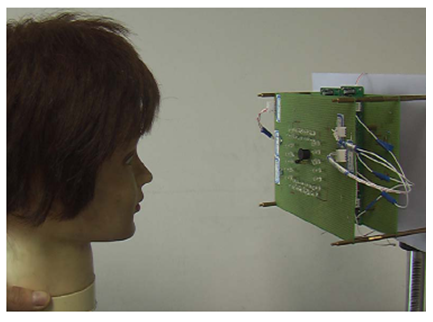
Do phương pháp này đưa ra quyết định chuỗi ảnh đầu vào được thu từ mặt phẳng 2D hay 3D nên nó cho kết quả tương đối tốt đồi với mục tiên phân biệt ảnh và người thật đã được đề ra. Tuy nhiên, việc tính toán giá trị optical flow cho chuỗi ảnh được xử lý khá lâu lại đem lại nhược điểm rất lớn cho phương pháp này.

Công việc tính toán này được áp dụng cho từng điểm ảnh của các ảnh trong chuỗi ảnh đầu vào. Hơn nữa, các thao tác tiền xử lý cho thuật toán này chưa được quan tâm đúng mức nên Optical Flow mới chỉ hiệu quả và phổ biến cho phát hiện chuyển động ở khoảng cách tương đối lớn so với các chuyển dịch của khuôn mặt. Khối lượng và thời gian tính toán lớn khiến cho phương pháp này không phù hợp để triển khai cho ứng dụng thời gian thực và cho thiết bị di động mà đề tài hướng đến.

### Learning Multispectral Reflectance Distributons

Phương pháp này [8] sẽ cho phép người sử dụng hoàn toàn thoải mái khi sử dụng hệ thống, hệ thống sẽ kiểm tra phân biệt giữa người thật và ảnh chụp mà không đòi hỏi người sử dụng phải tương tác với hệ thống. Nguyên tắc của phương pháp này là sử dụng tính chất đa phổ của da người và chất liệu không phải da người (chất liệu in ảnh) đối với các bước sóng khác nhau được cài đặt trước. Để làm điều này, hệ thống sử dụng hai bước sóng ánh sáng đặc biệt đã biết trước, lần lượt chiếu tới đối tượng trước camera sau đo thu và phân tích mức độ phản xạ của đối tượng với hai loại bước sóng ánh sáng này.

Phương pháp này hứa hẹn có thể phát hiện các trường hợp truy nhập trái phép ngay cả khi người dùng mạo danh sử dụng video hay mẫu vật 3D của người chủ thiết bị. Tuy nhiên hạn chế lớn nhất của phương pháp này đó là nó đòi hỏi thêm phần cứng thi hành việc phát ra hai bước sóng khác nhau. Điều này là bất khả thi với các thiết bị di động hiện hành trên thị trường cũng như không đáp ứng được sự tiện dụng cho người sử dụng.

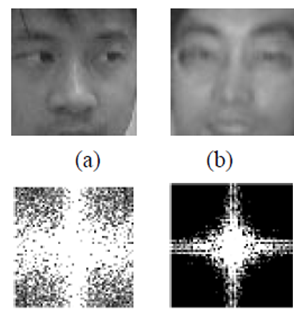


Hình 19: Phương án đòi hỏi phần cứng mở rộng,

hứa hẹn phát hiện được trường hợp mạo danh dùng video, mẫu vật 3D

### Dựa trên việc phân tích phổ Fourier

Phương pháp này dựa trên kết quả phân tích phổ Fourier [9] của ảnh 2D. Tương tự như phương pháp sử dụng optical flow, phương pháp này lợi dụng các biển đổi của gương mặt thật như cử động nháy mắt, cử động của miệng liên quan tới các biểu đạt cảm xúc để xác định sự biến đổi giá trị năng lượng của phổ Fourier trong chuỗi ảnh liên tiếp để đưa ra kết luận chuỗi ảnh đầu vào thu trực tiếp từ người thật hay từ một bức ảnh. Do chuỗi ảnh thu từ một ảnh chụp 2D không có các biến đổi như chuỗi ảnh chụp từ người thật nên độ chênh lệch của giá trị năng lượng phổ fourier của chuỗi ảnh này thấp hơn rất nhiều so với sự biến đổi năng lượng của chuỗi ảnh thật.



Hình 20:Phổ Fourier của ảnh và người thật

**a. Phổ Fourier của người thật trước Camera**

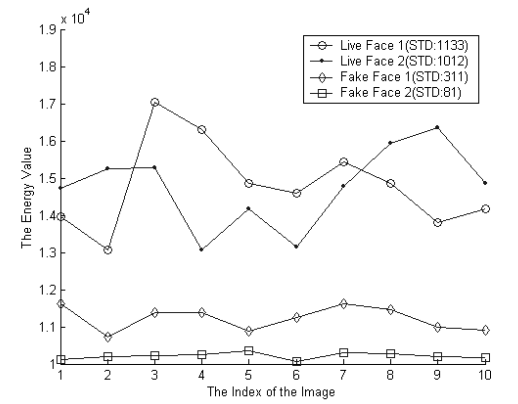
**b. Phổ Fourier của ảnh 2D trước Camera**

* Cách tính toán độ biến đối trên một loạt ảnh đầu vào
* Cho một dãy các ảnh đầu vào thu được từ camera
* Tính toán năng lượng Fourier của mỗi bức ảnh (tj)
* Tính năng lượng trung bình (tm)
* Tính toán FDD (frequency dynamics descriptor) – trị số miêu tả độ biến đổi – theo công thức

****

* Theo lý thuyết, FDD của ảnh chụp sẽ thấp hơn nhiều so với FDD của người thật.
* Việc chọn ngưỡng cho FDD phụ thuộc nhiều vào thực nghiệm

Trong nghiên cứu [9], kết quả thu được cho thấy giá trị FDD thu được từ ảnh và người thật có một sự khác biệt đáng kể.



Hình 21: Biểu đồ so sánh giá trị giữa ảnh chụp (fake) và người thật (live)

# PHẦN 2: CÁC KẾT QUẢ ĐẠT ĐƯỢC

Phần này sẽ trình bày các kết quả đạt được trong quá trình lựa chọn thuật toán sử dụng cho quy trình nhận dạng và xây dựng ứng dụng bảo mật truy cập. Nôi dung trong đó gồm có

* Trình bày các kết quả thực nghiệm và lựa chọn các phương pháp, thuật toán sử dụng cho quy trình nhận dạng gương mặt phù hợp với môi trường di động và tính toán thời gian thực.
* Phân tích thiết kế ứng dụng bảo mật truy cập đã được xây dựng trên môi trường điện thoại Android

1. **LỰA CHỌN GIẢI PHÁP CHO QUY TRÌNH NHẬN DẠNG KHUÔN MẶT**

## Mục tiêu hướng tới

Quy trình nhận dạng khuôn mặt này sẽ được áp dụng để thực hiện việc xử lý nhận dạng trên camera thời gian thực trong môi trường điện thoại di động, mà cụ thể ở đây là môi trường Android. Thế nên, các yếu tố cần xem xét để đánh giá sự phù hợp của phương pháp lựa chọn đối với yêu cầu của hệ thống bao gồm:

* Tốc độ (thời gian xử lý, thời gian nhận dạng) không được quá chậm, đảm bảo việc xử lý trên thời gian thực
* Bộ nhớ không được chiếm quá nhiều, tránh làm trì trệ hệ thống
* Độ chính xác: đảm bảo cho việc nhận dạng đúng với tỉ lệ cao nhất

## Cơ sở dữ liệu sử dụng để đánh giá

Một hệ thống nhận dạng gương mặt sẽ đòi hỏi dữ liệu hình ảnh có chứa mặt người để sử dụng cho việc huấn luyện, tập dữ liệu hình ảnh này được gọi là tập huấn luyện, những hình ảnh này sẽ được chuẩn hóa và được dùng trong quá trình nhận dạng.

Trong đề tài này, ta sử dụng bộ dữ liệu Yale Face Database A (6.4 MB) bao gồm 165 ảnh xám của 15 cá nhân. Mỗi người có 11 ảnh, được chụp dưới các điều kiện ánh sáng và cảm xúc khác nhau. Ở đây, ta chia 165 ảnh này thành 2 tập, tập huấn luyện (6 ảnh/người) và tập nhận dạng (5 ảnh/người)



Hình 22: Một set hình ảnh trong Yale Face Database A

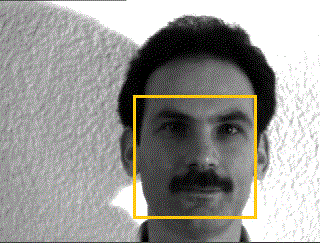
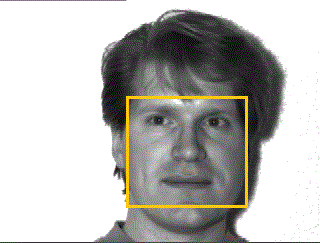
Ngoài ra, việc đánh giá còn dựa trên các số liệu thực tế thu được khi thực hiện các bước xử lý trực tiếp trên camera thời gian thực của một số dòng máy Android

## Triển khai thực nghiệm và lựa chọn phương pháp cho các pha thực hiện

### Phát hiện khuôn mặt

Phát hiện khuôn mặt – face detection : là quá trình xác định vùng có gương mặt trên một hình ảnh, ko quan tâm đó là ai, chỉ cần là mặt người

Những cascade classifier detector sử dụng để phát hiện khuôn mặt thường được huấn luyện với ít nhất 1000 hình ảnh có độc nhất một gương mặt và 10000 hình ảnh không có gương mặt. Quá trình huấn luyện có thể tốn nhiều thời gian, tuy nhiên, OpenCV đã có một số bộ phát hiện Haar và LBP ở dưới dạng xml đã được huấn luyện trước.

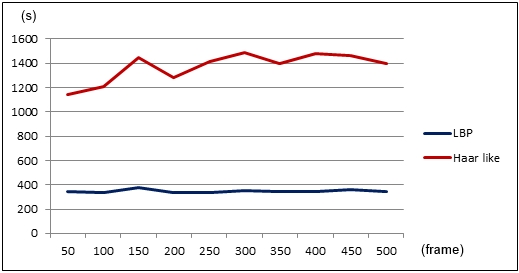
**

Hình 23:Kết quả phát hiện gương mặt

Kết quả bên dưới có được khi thực hiện phát hiện gương mặt với 100 bức ảnh có chứa mặt người và không có mặt người trên môi trường smartphone (máy HTC One S)

|  |  |  |
| --- | --- | --- |
|  | Tốc độ trung bình | Nhận dạng đúng |
| Haar-like | 31.5ms | 96% |
| LBP | 11.2ms | 97% |

Bảng 1:Tốc độ thực hiện phát hiện mặt trong môi trường smartphone



Hình 24:Biểu đồ so sánh tốc độ khi sử dựng đặc trưng Haar like và LBP

Cả hai bộ phát hiện đều cho ra kết quả tương đối tốt với thời gian ngắn (>95%), tuy nhiên xét về thời gian, LBP nhanh hơn Haar-like gấp nhiều lần.

### Tiền xử lý

Face preprocessing - tiền xử lý là quá trình điều chỉnh hình ảnh gương mặt nhìn rõ ràng hơn và tương đồng với những gương mặt khác

Đối với các hệ thống nhận dạng, độ nhiễu thông tin chính là nguyên nhân chính khiến việc nhận dạng bị sai lệch, đòi hỏi các gương mặt đầu vào và các gương mặt thuộc bộ huấn luyện phải được căn chỉnh đồng dạng với nhau. Tuy nhiên, vùng khuôn mặt đầu vào phát hiện được chỉ ở mức tương đối, không phải so khớp một cách chính xác.

Do đó, cần phải loại bỏ các thông tin thừa để cải thiện độ chính xác của nhận dạng. Điều này cũng sẽ góp phần giúp giảm tải khối lượng dữ liệu cần xử lý trong quá trình nhận dạng. Đối với hệ thống được cài đặt trên smartphone, đây là một bước quan trọng giúp cải thiện tốc độ tính toán và mức độ phức tạp của thuật toán.

Quá trình này được gọi là tiền xử lý, nó sẽ bắt đầu với một số biến đổi hình học trên gương mặt sau khi ta có được điểm trung gian giữa mắt và miệng

* Đưa hình ảnh mặt phát hiện được về một kích cỡ cố định.
* Cân bằng histogram để giảm bớt độ tương phản của ảnh
* Có thể thực hiện một số biến đổi hình học như xoay, căn chỉnh, xê dịch, sau khi có được điểm trung gian giữa mắt và miệng, nhằm để đôi mắt luôn ở trung tâm và ở một độ cao cố định
* Cắt bỏ các phần thừa của gương mặt và những yếu tố ngoại cảnh (tóc, trán, tai,..) bằng cách sử dụng một mặt nạ eclipse



Hình 25: Ví dụ cho một hình ảnh tiền xử lý

Gương mặt thu được sau khi tiền xử lý phải được sửa đổi để có kích thước bằng nhau bất kể kích thước đầu vào ban đầu. Ở đây, kích thước được chọn có tỷ lệ 1:1, ta sẽ xem xét với các kích thước đầu ra là 640x640, 128x128 , 64x64, kích thước nào phù hợp nhất để thực hiện nhận dạng

Tốc độ thực hiện tiền xử lý khi thực hiện trên môi trường smartphone của máy Samsung Galaxy Mega được đưa ra ở bảng dưới. Quá trình xử lý được thực hiện dựa trên sự hỗ trợ của bộ thư viện xử lý ảnh OpenCV

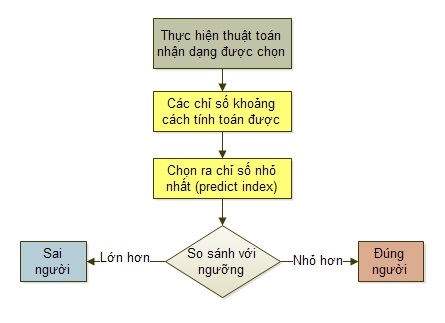
|  |  |  |  |
| --- | --- | --- | --- |
|  | 640x640 | 128x128 | 64x64 |
| **Tốc độ tiền xử lý** | ~824ms | ~255ms | ~220ms |

Bảng 2:Tốc độ tiền xử lý trên các kích thước ảnh

### Nhận dạng khuôn mặt

Face recognition - nhận diện gương mặt : là quá trình kiểm tra trong những gương mặt đã được thu thập, gương mặt nào giống với mặt trong camera nhất.

Đối với cả 2 thuật toán PCA và LBPH, đây là quá trình kiểm tra xem trong các chỉ số khoảng cách của ảnh đầu vào so với các ảnh trong tập huấn luyện, chỉ số nào bé nhất. Từ đó so sánh với một ngưỡng cho trước đến xác định xem đây có thật sự là gương mặt có trong tập huấn luyện hay không.

****

Hình 26: Sơ đồ các bước trong khâu nhận dạng khuôn mặt

1. **Tốc độ tính toán trên smartphone**

Nhìn chung, các thuật toán nhận dạng khuôn mặt đều được thể hiện bằng các giá trị số bình thường. Số thực thường được sử dụng cho các thuật toán đồ họa, xử lý ảnh. Tuy nhiên tổn tại một hiệu suất đáng kể giữa số thực và số nguyên trong quá trình tính toán. Hơn nữa, sự khác biệt này xuất hiện rõ rệt trên môi trường smartphone. Kết quả nghiên cứu của [4] đã chứng minh điều này

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Integer | | Float | | Double | |
|  | (a) | (b) | (a) | (b) | (a) | (b) |
| Cộng | 62.1 | 272.7 | 89.1 | 270.63 | 255.1 | 280.3 |
| Trừ | 78.2 | 267.7 | 87.6 | 269.54 | 277.6 | 282.80 |
| Nhân | 113.3 | 274.0 | 447.2 | 3702.76 | 997.1 | 3702.03 |
| Chia | 311.2 | 1379.0 | 490.5 | 4053.54 | 817.7 | 4022.88 |

Bảng 3:Thời gian thực hiện 7 triệu lần các phép tính cộng trừ nhân chia với các kiểu số khác nhau thực hiện môi trường máy tính (a) và smartphone (b)

Ta thấy thuật toán LBPH được thực hiện hầu hết với số nguyên còn Eigenfaces là số thực nên sẽ có một sự khác biệt lớn về tốc độ trung bình 1 lần xử lý thuật toán nhận dạng giữa 2 thuật toán.

Tiếp tục đo tốc độ xử lý trung bình của cả 2 thuật toán khi chạy trên các kích thước hình ảnh khác nhau, thuật toán được xử lý trên điện thoại smartphone Samsung Galaxy Mega, sử dụng 100 bức ảnh khác nhau.

|  |  |  |  |
| --- | --- | --- | --- |
|  | 640x640 | 128x128 | 64x64 |
| Eigenfaces | 551,44ms | 72,23ms | 60,71ms |
| LBPH | 473,12ms | 42,30ms | 40,42ms |

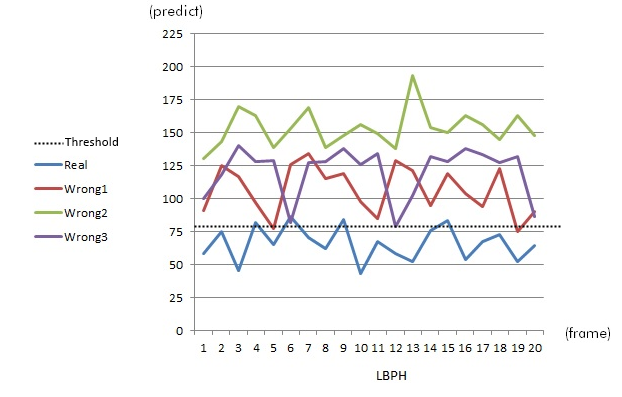
Bảng 4:Tốc độ xử lý thuật toán của Eigenfaces và LBPH

1. **Chọn ngưỡng cho thuật toán**

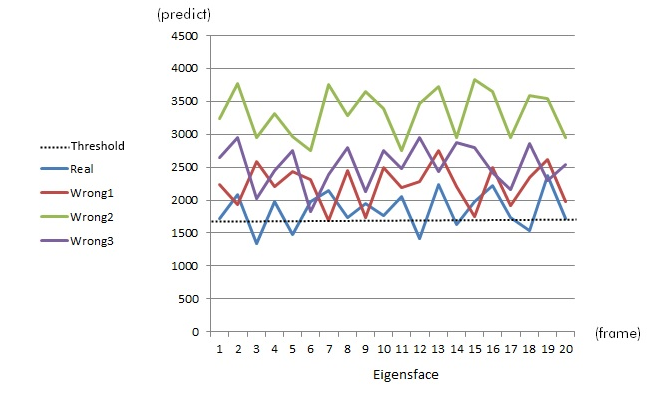
Cả 2 thuật toán đều trả về kết quả cuối cùng là chỉ số biểu hiện khoảng cách nhỏ nhất giữa bức ảnh đầu vào so với tập training trước đó. Theo như hình 13, ta gọi đây là chỉ số dự đoán (predict). Chỉ số này sẽ được so sánh với một mức ngưỡng được chọn ra trước đó để xác định mặt người trong bức ảnh đầu vào có nằm trong tập training không. Nếu bé hơn ngưỡng, ảnh đầu vào sẽ được xác định là đúng, lớn hơn sẽ là sai

Mức ngưỡng này được chọn ra từ thực nghiệm, đánh giá từ một loạt chỉ số đo được khi để ảnh đầu vào là người có mặt lẫn không có mặt trong tập training .

Dưới đây là biểu đồ thể hiện sự biến đối giá trị predict của người có mặt trong tập training (right) và những người không có mặt (wrong), khi thử nghiệm với cả 2 thuật toán PCA và LBPH với ảnh đầu vào thu được từ camera thời gian thực



Hình 27: Độ dao động giá trị predict của LBPH



Hình 28: Độ dao động giá trị predict của Eigenfaces

1. **Đánh giá độ chính xác và thời gian nhận dạng giữa 2 thuật toán**

* ***Với bộ cơ sở dữ liệu Yale Face A***

Thực hiện một phép đo độ chính xác bằng cách sử dụng tập trainning (66 ảnh) của Yale face database sau đó dùng các ảnh trong tập nhận dạng (55 ảnh) để kiểm tra.

Số ảnh nhận dạng được và nhận dạng sai giữa 2 thuật toán được cho ở 2 bảng sau.

|  |  |  |  |
| --- | --- | --- | --- |
|  | 640x640 | 128x128 | 64x64 |
| Eigenfaces | 28/55 | 28/55 | 27/55 |
| LBPH | 26/55 | 26/55 | 29/55 |

Bảng 5:Số ảnh thông báo nhận dạng được (chưa biết đúng sai)

|  |  |  |  |
| --- | --- | --- | --- |
|  | 640x640 | 128x128 | 64x64 |
| Eigenfaces | 0/28 | 0/28 | 0/27 |
| LBPH | 0/26 | 0/26 | 2/29 |

Bảng 6:Số ảnh nhận dạng sai trong số được cho là nhận dạng được ở trên

Theo như 2 bảng trên, với kích thước 64x64, lượng thông tin tiêu giảm dẫn đến nhận dạng sai ở một số trường hợp đối với LBPH

Cả 2 thuật toán đều đưa ra những đánh giá sai trong trường hợp tập huấn luyện không có chứa ảnh có cùng điều kiện ánh sáng hoặc có biểu cảm trên gương mặt quá khác so với ảnh đầu vào. Điều này chứng tỏ để có thể gia tăng độ chính xác cho quá trình nhận dạng, ta cần thu thập một bộ huấn luyện đa dạng trên nhiều phương diện như ánh sáng, cảm xúc, điệu bộ,….

* ***Với camera thời gian thực***

Tương tự như trên, việc nhận dạng trên thực tế chỉ diễn ra tốt khi điều kiện ngoại cảnh (ánh sáng, góc nhìn, biểu cảm) của ảnh đầu vào có nhiều sự tương đồng với ảnh trong tập huấn luyện, đặc biệt là với điều kiện ánh sáng.

Ngoài ra, khi sử dụng nhận dạng bằng camera thời gian thực, theo như biểu đồ giá trị predict ở trên hình 31-32, ta thấy

* Với thuật toán LBPH, giá trị của đúng và sai được phân bổ khá rõ ràng ở hai bên ngưỡng.
* Với thuật toán Eigenfaces, giá trị dự đoán dao động rộng, khó tìm được ngưỡng có thể phân tách hầu hết giá trị đúng và sai. Theo ngưỡng đã chọn trong hình 32, số lần đạt được giá trị bé hơn ngưỡng đã chọn cách nhau khá xa, dẫn đến thời gian kiểm tra người trước camera có đúng là người nằm trong tập training không sẽ tốn nhiều hơn so với LBPH.

Thực hiện đo thời gian nhận dạng bằng cách cho người đã được huấn luyện đứng trước camera và chờ đến khi các chỉ số thu được từ các ảnh đầu vào đạt đến ngưỡng cho trước được 3 lần, ta có

|  |  |  |  |
| --- | --- | --- | --- |
|  | 640x640 | 128x128 | 64x64 |
| Eigenfaces | 12,5s | 6.9s | 5.8s |
| LBPH | 5.7s | 2.1s | 2.2s |

Bảng 7:Thời gian nhận dạng đo được trên máy Samsung Galaxy Mega

Với ứng dụng được xử lý trên camera thời gian thực, đảm bảo tốc độ là yếu tố cần thiết, do đó thuật toán nhận dạng hợp lý để sử dụng trong trường hợp này là LBPH

### Tổng kết các phương pháp lựa chọn

Từ các bảng số liệu trên, chúng tôi lựa chọn phương pháp thực hiện các pha của quy trình nhận dạng như sau:

|  |  |  |
| --- | --- | --- |
|  | **Phương án lựa chọn** | **Lý do** |
| **Phát hiện khuôn mặt** | LBP cascade classifier | * Nhanh hơn nhiều lần so với bộ phân loại Haar Cascade * Độ chính xác cả 2 cao tương đương nhau (đều >95%) |
| **Kích thước tiền xử lý** | 128x128 | * 640x640 bị loại do tốn quá nhiều thời gian * 128x128 và 64x64 có thời gian xử lý ở các bước khác biệt không đáng kể * 64x64 có xảy ra một số trường hợp nhận dạng sai |
| **Nhận dạng khuôn mặt** | LBPH | * Có lợi thế trong việc chọn ngưỡng hơn so với PCA, dẫn đến tốc độ nhận dạng nhanh hơn |

Bảng 8: Tổng kết các phương án lựa chọn cho quy trình nhận dạng

# 4. ỨNG DỤNG BẢO MẬT TRUY CẬP BẰNG NHẬN DẠNG KHUÔN MẶT

### Tổng quan về ứng dụng

Mục đích là xây dựng một ứng dụng khóa, dùng để bảo vệ các ứng dụng được cài đặt trên điện thoại bằng các yêu cầu nhận diện gương mặt.

#### Yêu cầu của ứng dụng

* **Yêu cầu chức năng**

Ứng dụng được xây dựng nhằm giải quyết các yêu cầu chức năng sau

* Cho phép người dùng quyết định ứng dụng nào sẽ được bảo vệ
* Chức năng training để ghi nhớ gương mặt sẽ dùng để nhận dạng và mở khóa bảo vệ
* Hỗ trợ khóa bằng mật khẩu để dự phòng trong trường hợp ko thể nhận dạng được mặt
* Giao diện nhận dạng gương mặt xuất hiện khi một trong các ứng dụng được chọn để bảo vệ được mở lên
* Bảo vệ chính bản thân ứng dụng khóa khỏi việc xóa dữ liệu, hủy ứng dụng từ người ngoài
* **Yêu cầu phi chức năng**
* Chức năng phát hiện nhận dạng phải bảo đảm cho ra kết quả với độ chính xác cao nhất có thể.
* Công việc xử lý các frame thu được từ camera preview trong thời gian thực ko được quá lâu.

#### Công nghệ và thuật toán sử dụng

Ứng dụng được phát triển trên môi trường điện thoại Android, với sự hỗ trợ của bộ thư viện mã nguồn mở OpenCV

Các phương pháp được chọn để áp dụng cho quá trình nhận dạng là

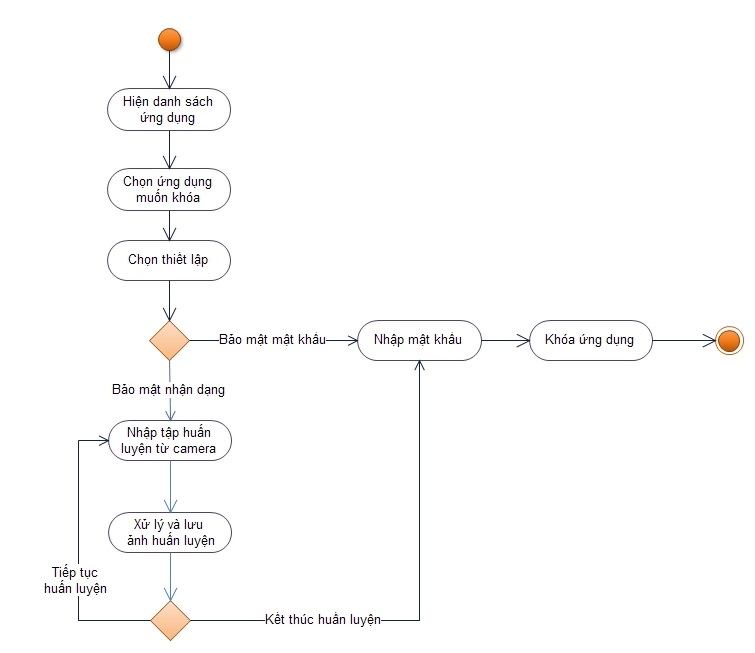
* LBP Cascade Classifier cho phát hiện gương mặt (face detection)
* Thuật toán LBPH cho nhận dạng gương mặt (face recognition)

Độ chính xác và đánh giá cho mỗi phương pháp đã được trình bày trong phần lựa chọn giải pháp cho quy trình nhận dạng.

### Phân tích ứng dụng

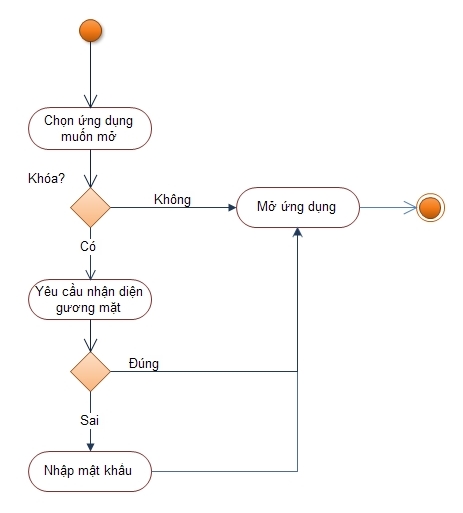
#### Biểu đồ hoạt động

* **Biểu đồ hoạt động cho quá trình “Khóa ứng dụng”**



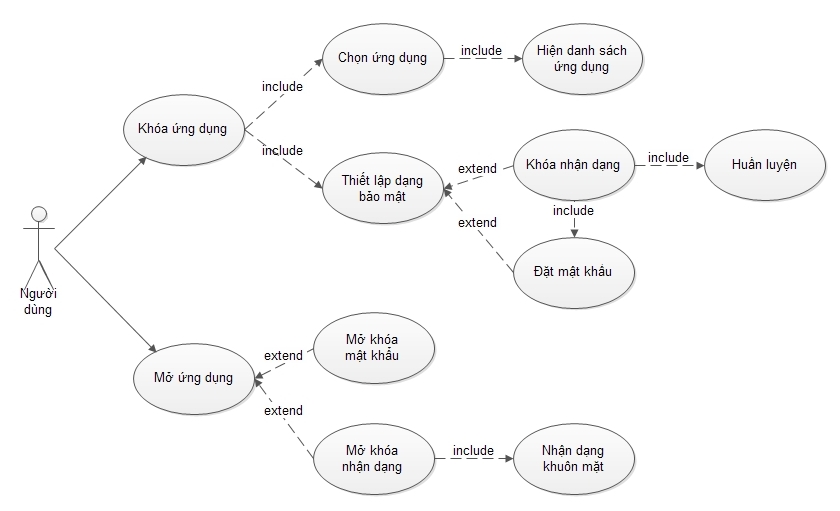
Hình 29: Biểu đồ hoạt động cho "Khóa ứng dụng"

* **Biểu đồ hoạt động cho quá trình “Mở ứng dụng”**

****

Hình 30: Biểu đồ hoạt động cho "Mở ứng dụng"

#### Biểu đồ Use Case



Hình 31: Biểu đồ Usa Case tổng quan

|  |  |
| --- | --- |
| Tên UC | **Khóa ứng dụng** |
| Tác nhân | Người dùng |
| Mục đích | Tiến hành khóa ứng dụng được chọn |
| UC liên quan | UC Chọn ứng dụng, UC Hiện ds ứng dụng, UC Thiết lập dạng bảo mật. UC Khóa nhận dạng, UC Đặt mật khẩu, UC Huấn luyện |
| Điều kiện | Điều kiện trước   * Người dùng mở ứng dụng * Danh sách ứng dụng có mặt trong hệ thống được hiển thị |
| Kịch bản | * Người dùng chọn app mình muốn bảo vệ trên danh sách * Người dùng muốn chọn chức năng khóa bằng nhận dạng khuôn mặt, thực hiện UC Khóa nhận dạng trong phần thiết lập trước tiên bắt buộc phải * Tham gia huấn luyện, thực hiện UC Huấn luyện * Đặt password dự phòng, thực hiện UC Đặt mật khẩu * Ngoài ra, phần thiết lập còn cho phép chỉ chọn bảo vệ bằng password * Việc huấn luyện và cài password chỉ bắt buộc thực hiện trong lần đầu tiên, các lần tiếp theo tùy vào người dùng |

Bảng 9: Đặc tả UC Khóa ứng dụng

|  |  |
| --- | --- |
| Tên UC | **Mở ứng dụng** |
| Tác nhân | Người dùng |
| Mục đích | Mở ứng dụng bất kỳ trên điện thoại |
| UC liên quan | UC Mở khóa nhận dạng, UC Mở khóa mật khẩu, UC Nhận dạng khuôn mặt |
| Điều kiện | Điều kiện trước   * Ứng dụng được mở có thể đã được chọn để bảo vệ trước đó hoặc không * Nếu ứng dụng được mở có bảo vệ, người dùng đã chọn thiết lập nhận dạng hoặc mật khẩu   Điều kiện sau   * Mở ứng dụng nếu xác định đúng người * Chuyển qua nhập mật khẩu nếu quá thời gian quy định * Chuyển qua nhập mật khẩu nếu xác định không phải vật sống |
| Kịch bản | * Nếu ứng dụng không được bảo vệ, UC Mở ứng dụng được thực hiện * Nếu ứng dụng được bảo vệ mật khẩu, người dùng phải * Nhập mật khẩu (UC Mở khóa mk) * Nếu ứng dụng được bảo vệ bằng nhận dạng, người dùng phải thực hiện UC Mở khóa nhận dạng, thực hiện * UC Nhận dạng khuôn mặt |

Bảng 10: Đặc tả UC Mở ứng dụng

### Thiết kế ứng dụng

#### Thiết kế lớp

Các lớp được chia thành các gói chính mang các chức năng như sau

|  |  |
| --- | --- |
| Gói lớp | Chức năng |
| com.activity | Chứa các activity hiển thị giao diện của hệ thống, bao gồm giao diện chính, huấn luyện, mở khóa, thiết lập, nhập password |
| com.widget | Một số thành phần giao diện nhỏ |
| com.model | Điều khiển hiển thị danh sách ứng dụng |
| com.facereconizer | Thực hiện công việc nhận dạng khuôn mặt |
| com.service | Vận hành service để bắt sự kiện mở app |

Bảng 11: Phân chia các nhóm lớp

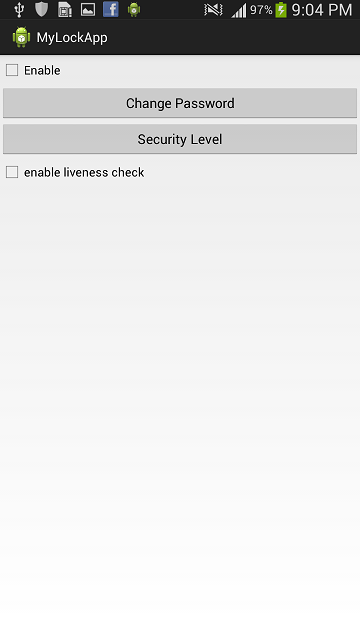
#### Thiết kế giao diện

* Giao diện chính của ứng dụng, hiển thị tất cả các ứng dụng đã được cài đặt trong điện thoại. Thanh bar phía trên có chứa 3 nút bấm
* Traning: dùng để nhập các ảnh huấn luyện
* Setting: thiết lập chức năng cho ứng dụng
* Finish: thoát ứng dụng



Hình 32: Màn hình chính của ứng dụng

* Giao diện setting và nhập mật khẩu

 ****

Hình 33: Giao diện setting và thay mật khẩu

* Giao diện huấn luyện
* Add Face : Nhập các hình ảnh huấn luyện
* Erase: Mở ra mục quản lý việc xóa các hình ảnh huấn luyện có sẵn
* Done: Hoàn thành huấn luyện, về lại giao diện cũ
* Giao diện mở khóa
* Có kèm theo nút “Unlock with password” trong trường hợp ko nhận dạng được

****

Hình 34: Giao diện huấn luyện và mở khóa

### Kết quả

#### Cài đặt hệ thống

* Lên google play cài đặt OpenCV Manager tương thích với dòng máy
* Tải file apk vào điện thoại
* Cài đặt file apk
* Mở ứng dụng lên và sử dụng theo kịch bản như dưới đây

#### Thử nghiệm hệ thống

* Thực hiện việc bảo mật truy cập của một hoặc nhiều ứng dụng bất kỳ bằng phương thức nhận dạng khuôn mặt, không sử dụng chức năng nhận dạng vật thể sống

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Kịch bản vận hành** | **Kết quả mong đợi** | **Kết quả thực tế** |
| 1 | - Mở ứng dụng  - Chọn ứng dụng cần bảo vệ bằng cách nhấn nút khóa ở bên cạnh ứng dụng | - Hiển thị danh sách các ứng dụng có sẵn trong điện thoại  - Các ứng dụng được chọn có nút khóa màu xanh bên phải | Hiển thị thành công |
| 2 | * Nhấn nút training * Bấm nút Add Face để bắt đầu thu thập ảnh huấn luyện. (Mỗi lần nhấn, sẽ lưu 6 bức ảnh liên tục theo thời gian) | * Hiển thị giao diện huấn luyện * Hình vuông phát hiện mặt được hiển thị | Thành công |
| 3 | * Nhấn nút Erase để vào giao diện quản lý các hình ảnh đã thu thập | * Giao diện hiển thị đã tồn tại ảnh => chứng tỏ ảnh huấn luyện đã được lưu |  |
| 4 | * Về lại giao diện chính, chọn Setting * Chọn Enable để kích hoạt việc bảo vệ nhận diện gương mặt * Chọn change password để tạo mật khẩu dự phòng (nếu không sẽ mặc định là 1111) * Về lại giao diện chính, kết thúc quá trình khóa | * Hiển thị dấu tích khi đã chọn * Bật màn hình nhập mật khẩu cũ sau đó đổi mật khẩu mới | Hiển thị thành công    C:\Users\KTC_Computer\Desktop\ScreenShot\Screenshot_2014-05-28-21-46-19.png |
| 5 | * Mở ứng dụng đã được chọn để bảo vệ trong bước 1 (Ở đây chọn chrome) * Đưa người đã được huấn luyện đứng trước camera | * Hiển thị giao diện mở khóa bằng nhận diện gương mặt * Một lúc sau màn hình mở khóa biến mất, ứng dụng được mở | Thành công    Màn hình chrome hiển thị |
| 6 | * Mở ứng dụng đã được chọn để bảo vệ trong bước 1 * Đưa người chưa được huấn luyện đứng trước camera khoảng 20s | * Màn hình mở khóa không biến mất | Thành công. Đã qua 20s |
| 7 | * Mở ứng dụng đã được chọn để bảo vệ trong bước 1 * Đưa người đã được huấn luyện đứng trước camera, chọn góc ánh sáng chưa được huấn luyện | * Ứng dụng được mở | Thất bại, màn hình mở khóa không biến mất, nhận dạng sai. |
| 8 | * Mở ứng dụng đã được chọn để bảo vệ trong bước 1 * Đưa “ảnh chụp” của người đã được huấn luyện đứng trước camera | * Màn hình mở khóa không biến mất, ứng dụng không được mở | Thất bại, ứng dụng được mở |

Bảng 12: Kịch bản vận hành của hệ thống

* Bước 5 được thử nghiệm với nhiều góc độ ánh sáng khác nhau của người đã được huấn luyện theo kịch bản sau:
* Chọn ra 10 vị trí đứng cố định (các vị trí này có điều kiện chiếu sang khác nhau)
* Chọn ngẫu nhiên 2 vị trí để tiến hành huấn luyện sau đó đứng ở 10 vị trí kia để nhận dạng. Biểu cảm không được thay đổi quá nhiều
* Tăng dần vị trí huấn luyện lên 5 – 8
* Lặp lại các bước trên với 3 người khác nhau làm người huấn luyện
* Kết quả: số lần nhận dạng được tăng dần theo số lần huấn luyện

|  |  |  |  |
| --- | --- | --- | --- |
| Số vị trí huấn luyện | 2 | 5 | 8 |
| Người số 1 | 5/10 | 8/10 | 10/10 |
| Người số 2 | 4/10 | 8/10 | 10/10 |
| Người số 3 | 6/10 | 7/10 | 10/10 |

Bảng 13: Thử nghiệm nhận dạng với điều kiện ánh sáng khác nhau

* Tiếp tục thử nghiệm bước 5 theo kịch bản
* Đứng ở một vị trí cố định
* Chỉ huấn luyện một lần với một biểu cảm nhất định trong số 5 biểu cảm sau: bình thường(1), nhíu mày(2), mếu(3). cười to(4), cười nhẹ(5)
* Nhận dạng với cả 5 biểu cảm nói trên
* Làm lại các bước với 3 người khác nhau
* Kết quả: Đối với những biểu cảm khiến cho cơ mặt thay đổi quá nhiều so với biểu cảm đã được huấn luyện, hệ thống không nhận dạng được.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Biểu cảm lúc huấn luyện | | Bình thường | Cười to | Nhíu mày |
| Biểu cảm  nhận dạng  được | Người số 1 | (1)(5) | (4)(3) | (1)(2)(5) |
| Người số 2 | (1)(1)(5) | (4) | (1)(2)(5) |
| Người số 3 | (1)(5) | (4) | (2) |

Bảng 14: Thử nghiệm nhận dạng với biểu cảm khác nhau

* Bước 6 được thử nghiệm lại theo kịch bản sau
* Chọn ra một người để huấn luyện
* Thử nghiệm với 10 người chưa được huấn luyện trong không gian thực.
* Lặp lại các bước trên với 3 người huấn luyện khác nhau
* Kết quả: trong một khoảng thời gian đủ lâu (8~10s), những người có khuôn mặt gần giống với người huấn luyện ban đầu đã lọt qua được hệ thống.

#### Đánh giá

Sau khi hoàn thành việc cài đặt những chức năng chính của ứng dụng và đưa ra thử nghiệm trên một số dòng điện thoại Android của Samsung, chúng tôi nhận thấy kết quá thu được tương đối tốt khi việc nhận dạng và huấn luyện diễn ra trong cùng một điều kiện ngoại cảnh. Độ chính xác và đánh giá cho mỗi phương pháp đã được trình bày cụ thể ở chương 3. Tuy nhiên ta có thể thấy rõ hơn khi thực hiện các bước thử nghiệm ứng dụng đã nêu trên

Nhìn chung, xét về phần nhận dạng, ứng dụng này vẫn còn một số nhược điểm như sau:

* Để đảm bảo cho khả năng nhận dạng ở nhiều vị trí, đòi hỏi người dùng phải thu thập một tập dữ liệu tốt, bao quát nhiều điều kiện liên quan đến độ sáng, góc chiếu, biểu cảm,….
* Đối với người có khuôn mặt tương đối giống với người được huấn luyện, hệ thống cũng trường hợp sẽ nhận nhầm và cho phép truy cập. Các trường hợp nhầm lẫn thường xảy ra với những người đeo kính
* Ảnh chụp của người sử dụng hoàn toàn có khả năng qua mặt được hệ thống nhận dạng. Tuy nhiên trong hệ thống hoàn chỉnh, module nhận diện vật thể sống đã khắc phục được điều này.
* Tốc độ xử lý tính toán phụ thuộc rất nhiều vào độ mạnh yếu của chip xử lý trong điện thoại. Đối với dòng máy tầm trung, tốc độ phát hiện cũng như nhận dạng diễn ra khá chậm.

KẾT LUẬN

Từ để tài ban đầu được phía công ty Samsung đưa ra trong chương trình hợp tác giữa công ty Samsung và trường Đại học Bách khoa Hà Nội là “Face Unlock System for Android smartphone”, chúng tôi đã tiến hành tìm hiểu và đi đến xây dựng hệ thống nhận dạng khuôn mặt để bảo mật truy cập ứng dụng trên điện thoại Android. Cụ thể nhóm thực hiện đồ án đã tìm hiểu để đưa ra cấu trúc hệ thống, đưa ra phương pháp giải quyết một số bài toán cần thiết cho việc xây dựng hệ thống.

Song song với việc tiến hành xây dựng ứng dụng trên bảo mật truy cập, nhóm đã tiến hành thực nghiệm để lựa chọn phương pháp phù hợp cho các pha của hệ thống nhận dạng gương mặt nhằm tìm cách đáp ứng yêu cầu về tốc độ và độ chính xác khi áp dụng trên môi trường di động

Trong quá trình thực hiện đồ án, qua phần nhiệm vụ của mình, tôi đã tìm hiểu được các bước cần thiết để xây dựng một hệ thống nhận dạng khuôn mặt. Đồng thời cũng có dịp để nghiên cứu thêm một số phương pháp trong lĩnh vực phát hiện vật thể sống khi đang tìm cách để tăng cường bảo mật cho hệ thống nhận dạng. Trong lúc đó, tôi cũng hiểu rõ hơn về cách lập trình, xây dựng ứng dụng trên Android và được tiếp cận với bộ thư viện xử lý ảnh OpenCV. Ngoài ra tôi học được cách thức nghiên cứu để giải quyết vấn đề cùng với kỹ năng thảo luận và làm việc nhóm.

Như vậy tôi đã đạt được phần lớn các mục tiêu và nội dung đã đặt ra trong đồ án. Tuy nhiên hệ thống xây dựng lên vẫn còn một số điểm chưa hoàn thiện và cần được khắc phục như nhận dạng khó khăn khi điều kiện ngoại cảnh thay đổi hay một số trường hợp xảy ra nhận nhầm. Do đó để cải thiện và xây dựng được một hệ thống hoàn hảo hơn, vẫn còn nhiều công việc tiếp theo cần thực hiện. Cụ thể đồ án sẽ có các hướng phát triển tiếp theo như sau:

* Thu thập nhiều dữ liệu ảnh người dùng trong thực tế dưới nhiều điều kiện hơn để đánh giá chính xác hơn về các thuật toán và các phương pháp đưa ra
* Tìm cách cải thiện thuật toán để tăng độ chính xác trong việc nhận dạng
* Tìm hiều và đưa ra thêm nhiều phương án tiền xử lý trước khi đưa vào nhận dạng nhằm cải thiện tốc độ và giảm thiểu hơn nữa những tác động của ngoại cảnh gây ảnh hưởng đến công việc nhận dạng
* Tìm cách cải thiện việc tính toán và phân bổ luồng hợp lý để tăng tốc độ xử lý trên những loại máy có chip xử lý trung bình và yếu

Cuối cùng, do thời gian thực hiện đồ án và kiến thức còn hạn chế, đồ án không tránh khỏi những thiếu sót, rất mong được sự đóng góp ý kiến của quí thầy cô và các bạn để đồ án được hoàn thiện hơn.

Tôi xin chân thành cảm ơn.

# TÀI LIỆU THAM KHẢO

1. W. Zhao, R. Chellappa, A. Rosenfeld, and P. Phillips. *“Face recognition: A literature survey”.* ACM Computing Surveys, pp. 399–458, 2003
2. P. Viola and M. Jones. *“Rapid object detection using a boosted cascade of simple feature”.* InComputer Vision and Pattern Recognition Conference 2001, vol.1, pp.511-518, 2001.
3. Rainer LienhartandJochen Maydt. “*An* Extended Set of Haar-like Features *for* Rapid Object Detection”. IEEE ICIP 2002, vol. 1, pp. 900-903, Sep. 2002
4. T. Ahonen, A. Hadid, and M. Pietikäinen. *“Face Description with Local Binary Patterns: Application to Face Recognition”*. In IEEE Trans. Pattern Analysis and Machine Intelligence, vol.28, no.12, Dec. 2006.
5. M. Turk and A. Pentland. *“Eigenfaces for Recognition”*, In Journal of Cognitive Neuroscience, vol.3, no.1, pp.71-86, 1991.
6. Kanghun Jeong, Dongil Han and Hyeonjoon Moon, *“Optimization of Face Recognition Algorithms for Smartphone Environment”*. In International Journal of Security and Its Applications, vol.7, no.6, pp.303-308, 2003
7. Wei Bao, Hong Li, Nan Li, Wei Jiang. *“A Liveness Detection Method for Face Recognition Based on Optical Flow Field”*. Proc. Int. Conf. on Image Analysis and Signal Processing (IASP 2009), pp.233-236, 2009.
8. Zhiwei Zhang, Dong Yi, Zhen Lei, and Stan Z. Li. FG. *“Face liveness detection by* learning multispectral reflectance distributions*”*, IEEE, pp.436-441, 2011.
9. J. Li, Y. Wang, T. Tan, and A. K. Jain, *“Live face detection based on the analysis of fourier spectra,”* In Defense and Security. International Society for Optics and Photonics, pp. 296–303, 2004.
10. Bruno Peixoto, Carolina Michelassi, Anderson Rocha. *“Face liveness detection under bad illumination conditions”*. ICIP 2011, pp. 3557-3560, 2011.
11. Adrian Kaehler, Gary Bradski, *"*Learning OpenCV*: Computer Vision with the OpenCV Library*", 2012.
12. OpenCV dev team, Face Recognition with OpenCV,

http://docs.opencv.org/modules/contrib/doc/facerec/facerec\_tutorial.html

1. OpenCV dev team, Histogram Comparison,

http://docs.opencv.org/doc/tutorials/imgproc/histograms/histogram\_comparison/histogram\_comparison.html

1. Shervin Emami, *“Mastering OpenCV with Practical Computer Vision Projects”*, Packt Publishing, 2012.