**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH**



**LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC**

**HỆ THỐNG QUẢN LÝ CHI PHÍ ĐI LẠI CỦA NHÂN VIÊN BÁN HÀNG TRONG CÔNG TY**

**HỘI ĐỒNG: 3**

**GVHD: ThS. Đặng Trần Trí**

**TS. Phan Trọng Nhân**

**GVPB: ThS. Nguyễn Thanh Tùng**

**Sinh viên thực hiện:**

|  |  |  |
| --- | --- | --- |
| STT | Họ và tên | MSSV |
| 1 | Đoàn Ngọc Tài | 51203208 |
| 2 | Nguyễn Công Bảy | 51200212 |
| 3 | Phạm Anh Nguyên Hồng | 51201293 |

*TP. HỒ CHÍ MINH, THÁNG 6 NĂM 2017*

# LỜI CAM ĐOAN

Chúng tôi cam đoan rằng các công việc và kết quả được trình bày trong luận văn này là do chính chúng tôi thực hiện, biên soạn và chưa hề được công bố ở bất kỳ đâu. Các thông tin, kết quả tham khảo được chúng tôi thu thập từ các nguồn khác nhau được ghi rõ trong mục Tài liệu tham khảo.

*TP.HCM, ngày 01 tháng 06 năm 2017*

Trân trọng

# LỜI CẢM ƠN

Với lòng biết ơn sâu sắc nhất, chúng tôi xin gửi đến quý Thầy Cô ở Trường Đại Học Bách Khoa – ĐHQG TP. Hồ Chí Minh đã truyền đạt tri thức và tâm huyết của mình cho chúng tôi trong suốt thời gian học tập tại trường.

Chúng tôi xin chân thành cảm ơn Thầy Đặng Trần Trí và Thầy Phan Trọng Nhân đã tận tâm theo sát chúng tôi trong suốt quá trình thực hiện luận văn, đóng góp những ý kiến quý báu cho chúng tôi qua từng buổi gặp mặt xuyên suốt quá trình tìm hiểu và thực hiện đề tài.

Chúng tôi cũng xin chân thành cảm ơn tác giả của các tài liệu mà chúng tôi đã tham khảo trong quá trình thực hiện luận văn.

*TP.HCM, ngày 01 tháng 06 năm 2017*

Trân trọng

# TÓM TẮT //TODO

Phương pháp xác thực là phương pháp kiểm tra xem người dùng đang tương tác với hệ thống có chính xác là chủ nhân thật sự hay không. Có rất nhiều hệ thống hiện nay dùng một chuỗi kí tự bí mật để làm mật khẩu xác thực. Đây là một phương pháp xác thực đơn giản. Tuy nhiên, nó cũng mang một số bất lợi. Thứ nhất, nó yêu cầu người dùng phải nhớ chính xác mật khẩu của mình tạo ra. Thứ hai, với những chuỗi mật khẩu quá đơn giản, ít kí tự thì hệ thống rất dễ bị tấn công. Chính vì vậy, để giải quyết những vấn đề trên, các phương pháp xác thực dùng đặc trưng sinh trắc ra đời. Một trong phương pháp xác thực người dùng rất được quan tâm hiện nay là xác thực dựa trên đặc trưng khuôn mặt.

Khuôn mặt đóng vai trò quan trọng trong quá trình giao tiếp giữa người với người, và cũng mang một lượng thông tin giàu có. Cùng với đó, sự phát triển của thiết bị di động ngày càng rộng rãi và phổ biến, các thông tin cá nhân được lưu trữ trên chúng càng nhiều dẫn đến việc bảo mật trở thành một vấn đề quan trọng. Do đó, xác thực khuôn mặt trên thiết bị di động là đề tài rất có tiềm năng và giúp ích được nhiều trong đời sống con người.

Mục tiêu của đề tài là nghiên cứu, tìm hiểu các cách rút trích đặc trưng gương mặt. Từ đó đánh giá ưu khuyết điểm của từng cách để áp dụng vào việc xác thực trên thiết bị di động. Đồng thời phải nâng cao tính an toàn khi sử dụng sinh trắc học bằng các phương pháp bảo mật.

Hệ thống đề xuất trong đề tài được phát triển trên nền tảng android với sự kết hợp của ba giải thuật PCA, Fisherface và Local binary pattern. Phương pháp bảo mật được tích hợp trong hệ thống là Secure Sketch. Hiệu suất của hệ thống được đo và thử nghiệm trên tập Face94 và hình ảnh thực tế thu được qua camera với kết quả giải thuật Eigenface nhanh nhất về tốc độ, giải thuật Fisherface có độ chính xác cao nhất với tỉ lệ FAR và FRR đều bằng 0% (không sử dụng bảo mật) và 2% (có sử dụng bảo mật)

# ABSTRACT //TODO

Authentication means verifying if the person interacting with the system is truly the device’s owner. Nowadays, there are many systems using a secret string as password, this is the simplest method. However, it has some drawbacks. First of all, users must remember exactly their password. Secondly, if the password is too simple, or contains few characters, the system will be vulnerable. Therefore, in order to solve the addressed problems, biometric authentication systems are born. One of the most concern methods right now is facial recognition.

Our faces play a critical role in human communication, moreover, they carry information about ourselves. And mobile devices have been growing rapidly and steadily for the last couple years, the more information we store on our devices, the stronger the need for security. Therefore, facial recognition on mobile devices is a promising approach that could contribute to human life.

The main objective of this thesis is researching on how to extract facial features, evaluate strengths and weaknesses of each approach to apply for facial recognition on mobile devices. Additionally, we have to keep the biometrics safe by template protection methods.

Our proposed system is based on Android platform with a combination of three algorithms PCA, Fisherface and Local Binary Pattern. The template protection technique used in our system is called Secure Sketch. The experiments are conducted on Faces94 database and photos taken by camera. According to the results, Eigenface gives us the best performance in term of running time, while Fisherface has the most accurate both rate FAR and FRR are 0% (no secure method), 2% (use secure method).

# MỤC LỤC//TODO

[LỜI CAM ĐOAN 2](#_Toc483507341)

[LỜI CẢM ƠN 3](#_Toc483507342)

[TÓM TẮT //TODO 4](#_Toc483507343)

[ABSTRACT //TODO 5](#_Toc483507344)

[MỤC LỤC//TODO 6](#_Toc483507345)

[DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT//TODO 9](#_Toc483507346)

[DANH MỤC CÁC HÌNH//TODO 10](#_Toc483507347)

[CHƯƠNG I: GIỚI THIỆU ĐỀ TÀI 12](#_Toc483507348)

[1. Giới thiệu đề tài 12](#_Toc483507349)

[2. Mục tiêu và phạm vi đề tài 12](#_Toc483507350)

[3. Phương pháp nghiên cứu và thực hiện //todo 13](#_Toc483507351)

[4. Cấu trúc báo cáo //todo 13](#_Toc483507352)

[CHƯƠNG II: CƠ SỞ LÝ THUYẾT //TODO 14](#_Toc483507353)

[1. Sinh trắc học 14](#_Toc483507354)

[2. Rút trích đặc trưng 20](#_Toc483507355)

[a. Principal component analysis (PCA) 20](#_Toc483507356)

[Giới thiệu 20](#_Toc483507357)

[Cơ sở lý thuyết 21](#_Toc483507358)

[Các vấn đề 25](#_Toc483507359)

[b. Fisherface 25](#_Toc483507360)

[Giới thiệu 25](#_Toc483507361)

[Cơ sở lý thuyết 26](#_Toc483507362)

[Các vấn đề 27](#_Toc483507363)

[c. Local binary pattern 28](#_Toc483507364)

[Giới thiệu 28](#_Toc483507365)

[Cơ sở lý thuyết 28](#_Toc483507366)

[Các vấn đề 32](#_Toc483507367)

[3. Bảo mật mẫu 32](#_Toc483507368)

[a. Secure Sketch 32](#_Toc483507369)

[b. Keyed-hash message authentication code (HMAC) 35](#_Toc483507370)

[4. Huấn luyện, đăng ký và xác thực 36](#_Toc483507371)

[a. Principal component analysis (PCA) 36](#_Toc483507372)

[Huấn luyện 36](#_Toc483507373)

[Đăng ký 38](#_Toc483507374)

[Xác thực 38](#_Toc483507375)

[b. Fisherface 39](#_Toc483507376)

[Huấn luyện 39](#_Toc483507377)

[Đăng ký 41](#_Toc483507378)

[Xác thực 41](#_Toc483507379)

[c. Local binary pattern 42](#_Toc483507380)

[Huấn luyện 42](#_Toc483507381)

[Đăng ký 44](#_Toc483507382)

[Xác thực 44](#_Toc483507383)

[d. Secure sketch-Fuzzy Extractor 45](#_Toc483507384)

[Huấn luyện và đăng ký 45](#_Toc483507385)

[Xác thực 46](#_Toc483507386)

[1. Nền tảng 47](#_Toc483507387)

[2. Các thư viện sử dụng 51](#_Toc483507388)

[a. JavaCV 51](#_Toc483507389)

[b. Google Mobile Vison API 51](#_Toc483507390)

[c. Sweet Alert Dialog 51](#_Toc483507391)

[CHƯƠNG III: HỆ THỐNG ĐỀ XUẤT 51](#_Toc483507392)

[1. Mô tả hệ thống 51](#_Toc483507393)

[a. Use case Scenario 54](#_Toc483507394)

[b. Sơ đồ sequence 57](#_Toc483507395)

[2. Các chức năng chính 60](#_Toc483507396)

[CHƯƠNG IV: HIỆN THỰC //TODO 62](#_Toc483507397)

[1. Website 62](#_Toc483507398)

[1.1. Workflow 62](#_Toc483507399)

[1.2. Giao diện Website 62](#_Toc483507400)

[1.2.1. Đăng nhập 62](#_Toc483507401)

[1.2.2. Trang chủ 62](#_Toc483507402)

[c. Chi tiết các khối 69](#_Toc483507403)

[3. Kiểm định sản phẩm 72](#_Toc483507404)

[a. Danh sách Unit Test Case. 72](#_Toc483507405)

[b. Chi tiết Unit Test Case 72](#_Toc483507406)

[c. Báo cáo Unit Test 75](#_Toc483507407)

[CHƯƠNG V: ĐÁNH GIÁ KẾT QUẢ //TODO 77](#_Toc483507408)

[1. Phương pháp đánh giá 77](#_Toc483507409)

[Tiêu chí đánh giá 77](#_Toc483507410)

[Xây dựng tập thí nghiệm sử dụng để huấn luyện, đăng kí và kiểm tra 77](#_Toc483507411)

[Phương pháp đánh giá thực nghiệm 78](#_Toc483507412)

[Môi trường thí nghiệm 78](#_Toc483507413)

[2. Các thí nghiệm 78](#_Toc483507414)

[- Đánh giá hiệu suất giải thuật PCA trên tập thí nghiệm 78](#_Toc483507415)

[-Đánh giá hiệu suất giải thuật Fisherface trên tập thí nghiệm 79](#_Toc483507416)

[-Đánh giá hiệu suất giải thuật LBP trên tập thí nghiệm 81](#_Toc483507417)

[-Đánh giá hiệu suất kết hợp 3 giải thuật trên tập thí nghiệm 82](#_Toc483507418)

[3. Kết quả thí nghiệm 83](#_Toc483507419)

[CHƯƠNG VI: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 84](#_Toc483507420)

[1. Kết luận 84](#_Toc483507421)

[2. Hướng phát triển 84](#_Toc483507422)

[THAM KHẢO 85](#_Toc483507423)

[PHỤ LỤC 87](#_Toc483507424)

[1. Trị riêng, vector riêng [15] 87](#_Toc483507425)

[a. Định nghĩa 87](#_Toc483507426)

[b. Tính chất 87](#_Toc483507427)

[c. Chứng minh 87](#_Toc483507428)

[d. Phương pháp giải tìm trị riêng, vectơ riêng 88](#_Toc483507429)

[e. Không gian con riêng ứng với GTR λ 88](#_Toc483507430)

[2. Chi-Squared [16] 88](#_Toc483507431)

# DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT//TODO

PCA: Principal component analysis

LDA: Linear discriminant analysis

LBP: Local binary pattern

FLD: Fisher’s linear discriminant

FAR : False accept ratio

FRR: False rejection ratio

OpenCV: Open source computer vision

HMAC: Hash message authentication code

SHA-1: Secure hash algorithm 1

API: Application programming interface

UTCID: Unit testcase identify

VTR: Vector riêng

GTR: Trị riêng

# DANH MỤC CÁC HÌNH//TODO

[Hình 1. Các loại sinh trắc học phổ biến 14](#_Toc470817981)

[Hình 2. Xác thực bằng giấy phép lái xe hoặc hộ chiếu 15](#_Toc470817982)

[Hình 3. Nhận dạng bằng password 15](#_Toc470817983)

[Hình 4. Nhận dạng bằng mã PIN, khó để ghi nhớ 16](#_Toc470817984)

[Hình 5. Hồ sơ sinh trắc học của một người 16](#_Toc470817985)

[Hình 6. Tương tác con người và máy tính 17](#_Toc470817986)

[Hình 7. Biểu cảm khuôn mặt 1 17](#_Toc470817987)

[Hình 8. Biểu cảm khuôn mặt 2 18](#_Toc470817988)

[Hình 9. Ví dụ về phương pháp PCA 21](#_Toc470817989)

[Hình 10. Biểu diễn khuôn mặt thành vector 22](#_Toc470817990)

[Hình 11. Hình minh họa giải thuật PCA 24](#_Toc470817991)

[Hình 12. Sự kết hợp tuyến tính của khuôn mặt 24](#_Toc470817992)

[Hình 13. Hình ảnh so sánh PCA và FLD trong bài toán hai lớp, nơi dữ liệu cho mỗi lớp nằm gần một không gian con tuyến tính 27](#_Toc470817993)

[Hình 14. Minh họa giải thuật LBP 29](#_Toc470817994)

[Hình 15. Ví dụ về quá trình tính toán đặc trưng LBP 31](#_Toc470817995)

[Hình 16. Mô hình Secure Sketch 32](#_Toc470817996)

[Hình 17. Mô hình Codebook Secure Sketch 33](#_Toc470817997)

[Hình 18. Quá trình sinh sketch và khôi phục 35](#_Toc470817998)

[Hình 19. Tập huấn luyện Eigenface 37](#_Toc470817999)

[Hình 20. Khuôn mặt trung bình 37](#_Toc470818000)

[Hình 21. Các eigen vector của tập huấn luyện 38](#_Toc470818001)

[Hình 22. Hình ảnh người cần xác thực 39](#_Toc470818002)

[Hình 23. Tập train Fisherface 40](#_Toc470818003)

[Hình 24. Khuôn mặt trung bình 40](#_Toc470818004)

[Hình 25. Các vector sau khi train Fisherface 41](#_Toc470818005)

[Hình 26. Testcase 41](#_Toc470818006)

[Hình 27. Tập huấn luyện của LBP 42](#_Toc470818007)

[Hình 28. Ảnh xám của tập huấn luyện 42](#_Toc470818008)

[Hình 29. Ảnh LBP thu được sau khi tính toán 43](#_Toc470818009)

[Hình 30. Biểu đồ minh họa Histogram 43](#_Toc470818010)

[Hình 31. Ảnh cần xác thực 44](#_Toc470818011)

[Hình 32. Ảnh xác thực đã chuyển qua ảnh xám 44](#_Toc470818012)

[Hình 33. Ảnh xác thực LBP 45](#_Toc470818013)

[Hình 34. Hệ thống đề xuất 47](#_Toc470818014)

[Hình 35. Chi tiết hệ thống 48](#_Toc470818015)

[Hình 36. Hình mô tả kết hợp 3 giải thuật 49](#_Toc470818016)

[Hình 37. Mô hình usecase của hệ thống 49](#_Toc470818017)

[Hình 38. Sequence đăng ký 53](#_Toc470818018)

[Hình 39. Sequence cập nhật sinh trắc 53](#_Toc470818019)

[Hình 40. Sequence xem hướng dẫn 54](#_Toc470818020)

[Hình 41. Sequence xác thực 55](#_Toc470818021)

[Hình 42. Sequence tùy chỉnh 55](#_Toc470818022)

[Hình 43. Quá trình đăng ký dùng Secure Sketch 56](file:///C:\Users\dohoa\AndroidStudioProjects\wowauthentication\document\PHÁT-TRIỂN-HỆ-THỐNG-XÁC-THỰC-BẰNG-ĐẶC-TRƯNG-GƯƠNG-MẶT-TRÊN-ANDROID-CÓ-BẢO-VỆ-MẪU-ĐẶC-TRƯNG-SINH-TRẮC.docx#_Toc470818023)

[Hình 44. Quá trình xác thực dùng Secure Sketch 57](#_Toc470818024)

[Hình 45. Kiến trúc của Android 59](#_Toc470818025)

[Hình 46. Sơ đồ miêu tả sự phổ biến của các phiên bản android 61](#_Toc470818026)

[Hình 47. Tỷ lệ kết quả theo testcase 70](#_Toc470818027)

[Hình 48. Số loại testcase 70](#_Toc470818028)

[Hình 49. Cấu trúc tập thí nghiệm 71](#_Toc470818029)

[Hình 50. Độ chính xác của PCA với các ngưỡng 73](#_Toc470818030)

[Hình 51. Độ chính xác của PCA với các DELTA 73](#_Toc470818031)

[Hình 52. Độ chính xác của Fisherface với các ngưỡng 74](#_Toc470818032)

[Hình 53. Độ chính xác của Fisherface với các DELTA 74](#_Toc470818033)

[Hình 54. Độ chính xác của LBP với các ngưỡng 75](#_Toc470818034)

[Hình 55. Độ chính xác của LBP với các DELTA 76](#_Toc470818035)

# GIỚI THIỆU ĐỀ TÀI

## Giới thiệu đề tài

Ngày nay, các công ty đều có chính sách hỗ trợ nhân viên chi phí đi lại từ nơi ở tới công ty, từ công ty tới gặp khách hàng, … nhất là đối với các nhân viên bán hàng, họ rất thường xuyên di chuyển do nhu cầu công việc. Hiện tại, việc quản lý chi phí đi lại của nhân viên thường thông qua các số liệu do nhân viên cung cấp (hóa đơn, vé xe, do nhân viên tự khai báo, …) mà chưa được quản lý một cách hệ thống, tự động.

Các thiết bị di động thông minh (Smartphone) đang được sử dụng ngày càng phổ biến và rộng rãi. Ngoài các tính năng nghe gọi, nhắn tin, Smartphone còn có khả năng định vị với độ chính xác cao. Do đó, các dịch vụ dựa trên vị trí (Location-based service) ngày càng phát triển mạnh.

Đề tài này được phát triển để hỗ trợ công ty quản lý chi phí đi lại của nhân viên trong các hoạt động hàng ngày. Dữ liệu đi lại của nhân viên sẽ được thu thập và lưu trữ. Dữ liệu này sẽ cho phép công ty tính toán được chi phí đi lại của các nhân viên để đưa ra mức hỗ trợ phù hợp.

## Mục tiêu đề tài

Các mục tiêu của đề tài:

* Định vị và quản lý lịch trình của nhân viên:
* Định vị tự động dùng Smartphone.
* Tính toán chi phí đi lại của nhân viên.
* Bảo vệ tính riêng tư về vị trí, đường đi:
* Bảo vệ tính riêng tư về vị trí.
* Quản lý truy xuất đến dữ liệu lịch sử di chuyển.

Hệ thống gồm các thành phần:

* Một ứng dụng di động (trên nền tảng Android) dành cho nhân viên, để thu thập dữ liệu về vị trí, lộ trình, chi phí đi lại của nhân viên.
* Một Website hỗ trợ những người quản trị hệ thống, để quản lý dữ liệu của công ty.

## Phạm vi đề tài

Phạm vi nghiên cứu của đề tài:

* Đề tài này được phát triển dựa trên giả định là nhân viên trung thực, nghĩa là các thông tin được nhân viên cung cấp thông qua ứng dụng di động là trung thực, không sử dụng những phần mềm làm sai lệch các thông tin đó nên đề tài bỏ qua việc chống gian lận vị trí.
* Công thức tính toán chi phí đi lại sẽ phụ thuộc vào chính sách của mỗi công ty. Do đó, đề tài sẽ không nghiên cứu, sử dụng một công thức tính toán cụ thể nào.

## Cấu trúc báo cáo //todo

Báo cáo gồm có 6 chương với các nội dung:

* Chương I: Giới thiệu đề tài: Trình bày tổng quát về đề tài tính cần thiết và tiềm năng ứng dụng của nó; Làm rõ mục tiêu và phạm vi đề tài; Phương pháp nghiên cứu và hiện thực của nhóm; Cấu trúc báo cáo.
* Chương II: Cơ sở lý thuyết: Giới thiệu về sinh trắc học, ứng dụng của sinh trắc học và các tiêu chí đánh giá một sinh trắc học; Lý thuyết về các phương pháp rút trích Principal component analysis, Fisherface, Local binary pattern. Phương pháp bảo mật mẫu sau khi đã rút trích được đặc trưng; Phương pháp huấn luyện và cách xác thực một khuôn mặt đầu vào.
* Chương III: Hệ thống đề xuất: Mô tả về hệ thống của nhóm với các sơ đồ usecase, activity. Chi tiết về các chức năng chính của phần mềm.
* Chương IV: Hiện thực: Giới thiệu về nền tảng android, các thư viện nhóm sử dụng. Các khối chức năng chính và việc kiểm định sản phẩm.
* Chương V: Đánh giá kết quả: Trình bày các phương pháp đánh giá, các thí nghiệm sử dụng để đánh giá và kết quả thí nghiệm.
* Chương VI: Kết luận và hướng phát triển

# CƠ SỞ LÝ THUYẾT

## Tổng quan về GPS

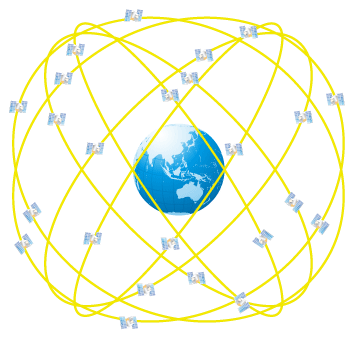
* + 1. **Giới thiệu về GPS**

GPS (viết tắt từ Global Positioning System – Hệ thống định vị toàn cầu) là hệ thống xác định vị trí toàn cầu được tài trợ và quản lý bởi Bộ Quốc phòng Mỹ (U. S. Department of Defense). Đây là dịch vụ định vị toàn cầu được cung cấp miễn phí, có khả năng cung cấp thông tin cần thiết để xác định vị trí của người dùng. [1]

* + 1. **Cơ chế hoạt động của GPS**

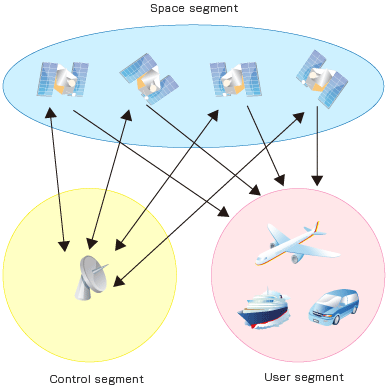
GPS bao gồm 3 thành phần: [3]

* Các vệ tinh GPS (Space segment): 24 vệ tinh GPS đang hoạt động được triển khai trên 6 quỹ đạo (khoảng 4 vệ tinh trên mỗi quỹ đạo) quay quanh trái đất với chu kỳ quỹ đạo khoảng 12 giờ ở độ cao khoảng 20 000 km so với mực nước biển.



Hình 2.1.2.1. Quỹ đạo của các vệ tinh GPS [2]

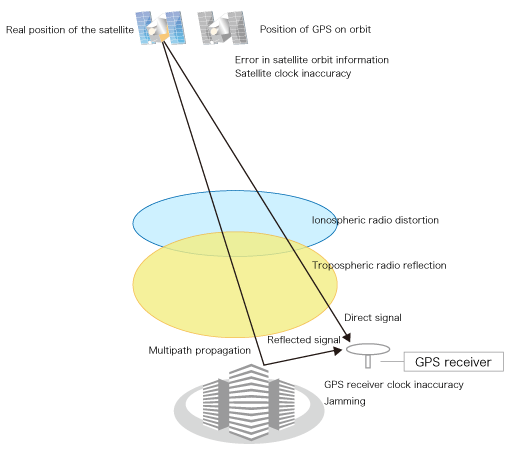
* Trạm điều khiển mặt đất (Control segment): Trạm điều khiển mặt đất đóng vai trò điều khiển, vận hành và duy trì hoạt động của các vệ tinh, đảm bảo chất lượng của tín hiệu GPS.
* Thiết bị thu GPS của người dùng (User segment): Thiết bị thu GPS của người dùng thu tín hiệu từ các vệ tinh GPS và chuyển đổi chúng thành những dữ liệu để ước tính vị trí người dùng. Cần kết hợp tín hiệu nhận được từ 4 vệ tinh GPS để tính toán vị trí người dùng (kinh độ, vĩ độ, cao độ và thời gian (time offset)).



Hình 2.1.2.2. Cấu trúc cơ bản của GPS [2]

* + 1. **Những hạn chế khi sử dụng GPS của thiết bị di động để định vị bản đồ**

Do nhiều yếu tố chủ quan và khách quan, sử dụng GPS của thiết bị di động để định vị bản đồ luôn có sai số không thể tránh khỏi (khoảng 1.4–7.1m ở những điều kiện bình thường). [3] Điều này ảnh hưởng đến độ chính xác của dữ liệu về vị trí và lộ trình của người dùng.



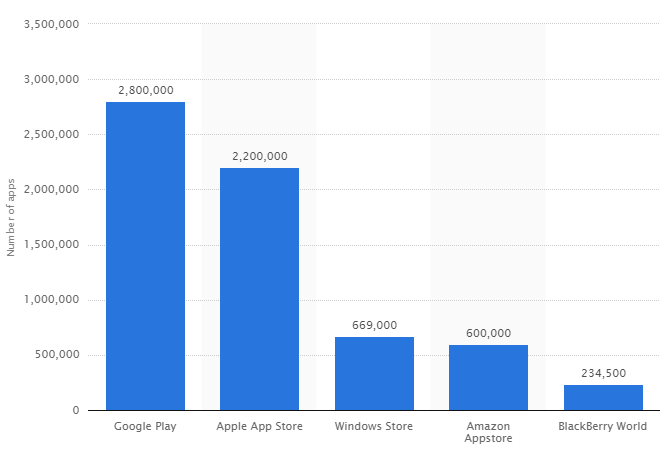
Hình 2.1.3.1. Một số yếu tố ảnh hưởng chất lượng định vị của GPS [2]

* 1. **Tổng quan về hệ điều hành Android**
     1. **Giới thiệu về hệ điều hành Android**

Android [4] là một hệ điều hành dựa trên nền tảng Linux được thiết kế dành cho các thiết bị di động có màn hình cảm ứng như điện thoại thông minh và máy tính bảng.

Ban đầu, Android được phát triển bởi công ty Android (Android, Inc.). Công ty Android được thành lập tại Palo Alto, California vào tháng 10 năm 2003 bởi Andy Rubin, Rich Miner, Nick Sears và Chris White để phát triển một hệ điều hành dành cho các thiết bị di động có thể nhận biết được ví trị và sở thích của người dùng. Google mua lại công ty Android vào ngày 17 tháng 8 năm 2005 và biến nó thành một bộ phận trực thuộc Google. Tại Google, nhóm do Rubin đứng đầu đã phát triển một nền tảng dành cho thiết bị di động trên nền nhân Linux. Hệ điều hành Android được ra mắt vào năm 2007. Năm 2008, chiếc điện thoại đầu tiên chạy hệ điều hành Android, HTC Dream, được bán ra.

Android là hệ điều hành mã nguồn mở, phát hành theo giấy phép Apache, một giấy phép không có nhiều ràng buộc. Điều này cho phép các nhà phát triển, các lập trình viên được tự do tinh chỉnh và phân phối Android. Bên cạnh đó, Android còn có một cộng đồng lập trình viên đông đảo, nhiệt huyết, chuyên viết các ứng dụng cho hệ điều hành này bằng ngôn ngữ lập trình Java. Theo thống kê từ Statista, đã có khoảng 2.8 triệu ứng dụng (tính đến tháng 3 năm 2017) [5] được phát hành trên Google Play, cửa hàng ứng dụng chính của Android, với 65 tỷ lượt tải về (từ tháng 8 năm 2010 đến tháng 5 năm 2016). [6]



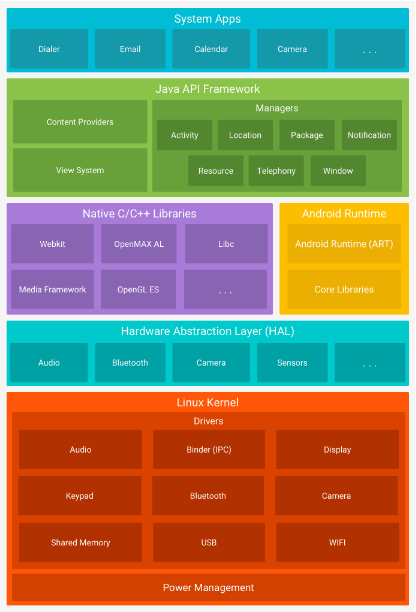
Hình 2.2.1.1. Thống kê số ứng dụng được phát hành trên các cửa hàng ứng dụng hàng đầu (tính đến tháng 3 năm 2017) [5]

Những yếu tố này đã giúp Android trở thành hệ điều hành điện thoại thông minh phổ biến nhất thế giới, và được các công ty công nghệ lựa chọn khi họ cần một hệ điều hành không nặng nề, có khả năng tinh chỉnh và giá rẻ chạy trên các thiết bị công nghệ cao thay vì tạo dựng từ đầu. Kết quả là mặc dù được thiết kế để chạy trên điện thoại và máy tính bảng, Android đã xuất hiện trên TV, máy chơi game và các thiết bị điện tử khác.

* + 1. **Kiến trúc của Android**

Kiến trúc của hệ điều hành Android bao gồm: [7]

* Linux Kernel : được dựa trên nhân Linux, là một thành phần cực kỳ quan trọng.
* Hardware Abstraction Layer (HAL): giúp các ứng dụng Android có thể tương tác với các thiết bị phần cứng.
* Native C/C++ Libraries: bao gồm các thư viện (viết bằng C/C++) hỗ trợ cho các ứng dụng thực hiện các chức năng mở rộng.
* Android Runtime: đảm nhận công việc biên dịch các tập tin ứng dụng, để chúng có thể chạy được trên nền tảng Android.
  + Core Libraries : tập các thư viện cơ bản của Android .
  + Dalvik Vitual Machine : bộ máy ảo của android, là nơi thực hiện các ứng dụng mà người dùng sử dụng trên thiết bị Android.
* Java API Framework: cung cấp toàn bộ những tính năng cần thiết của hệ điều hành Android cho lập trình viên sử dụng thông qua các API được viết bằng ngôn ngữ Java.
* System Apps: là tập hợp các ứng dụng và chức năng hệ thống cơ bản để tương tác với người sử dụng, hoặc để các lập trình viên phát triển ứng dụng.



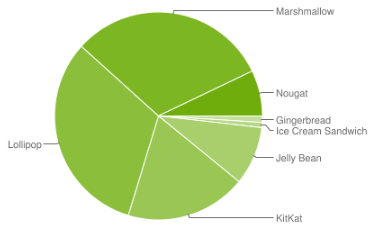
Hình 2.2.2.1. Sơ đồ kiến trúc của hệ điều hành Android [7]

* + 1. **Các phiên bản phổ biến của hệ điều hành Android**

Bảng sau đây sẽ liệt kê các phiên bản phổ biến nhất của hệ điều hành Android. Bảng sẽ không liệt kê những phiên bản chiếm ít hơn 0.1% thị phần của các thiết bị chạy hệ điều hành Android tính tới thời điểm được khảo sát (ngày 2 tháng 5 năm 2017).

| Phiên bản | Tên phiên bản | API level | Phân bố |
| --- | --- | --- | --- |
| 2.3.3 – 2.3.7 | Gingerbread | 10 | 1.0% |
| 4.0.3 – 4.0.4 | Ice Cream Sandwich | 15 | 0.8% |
| 4.1.x | Jelly Bean | 16 | 3.2% |
| 4.2.x | 17 | 4.6% |
| 4.3 | 18 | 1.3% |
| 4.4 | KitKat | 19 | 18.8% |
| 5.0 | Lollipop | 21 | 8.7% |
| 5.1 | 22 | 23.3% |
| 6.0 | Marshmallow | 23 | 31.2% |
| 7.0 | Nougat | 24 | 6.6% |
| 7.1 | 25 | 0.5% |

Bảng 2.2.3.1. Bảng liệt kê các phiên bản phổ biến của hệ điều hành Android [8]



Hình 2.2.3.2. Các phiên bản phổ biến của hệ điều hành Android

Nhóm quyết định chọn phiên bản Jelly Bean (API level: 16) là phiên bản thấp nhất được hệ thống quản lý chi phí đi lại của nhân viên bán hàng trong công ty hỗ trợ để cho phép được nhiều người dùng có thể sử dụng được hệ thống.

Hình . Các loại sinh trắc học phổ biến

Định danh sinh trắc học được sử dụng để đặt tên và mô tả cá nhân. Định danh sinh trắc học thường được phân loại dựa trên đặc điểm sinh lý và đặc điểm hành vi:

* Đặc điểm sinh lý là những gì có liên quan đến hình dạng của cơ thể. Ví dụ như vân tay, tĩnh mạch lòng bàn tay, xác thực khuôn mặt, DNA, mống mắt, võng mạc và mùi hương...
* Đặc điểm hành vi có liên quan đến mô hình của hành vi của một người, như gõ nhịp, dáng đi, và giọng nói...

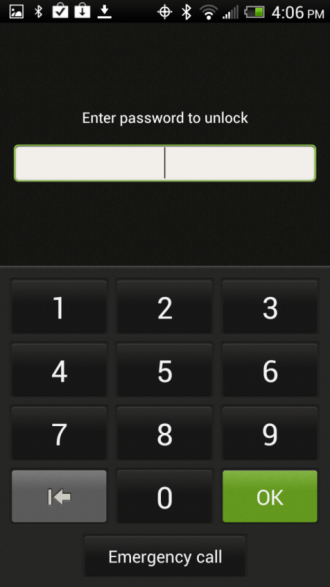
Nhiều phương tiện truyền thống kiểm soát truy cập bao gồm:

* Hệ thống xác thực thẻ, chẳng hạn như giấy phép lái xe hoặc hộ chiếu.



Hình . Xác thực bằng giấy phép lái xe hoặc hộ chiếu

* Hệ thống xác thực dựa trên tri thức, chẳng hạn như một mật khẩu hoặc số xác thực cá nhân.



Hình . Nhận dạng bằng password



Hình . Nhận dạng bằng mã PIN, khó để ghi nhớ

Định danh sinh trắc học là duy nhất cho các cá nhân, đáng tin cậy hơn trong việc xác minh danh tính hơn các phương pháp truyền thống do nó mang tính cá nhân và khó để làm giả. Tuy nhiên, việc thu xác thực sinh trắc học làm tăng mối quan tâm riêng tư về việc sử dụng cuối cùng của thông tin này.

Từ những thông tin rút trích được, có thể ứng dụng trong các lĩnh vực:

* An ninh (chẳng hạn tìm ra những tên trộm trong hồ sơ lưu trữ thông tin con người).



Hình . Hồ sơ sinh trắc học của một người

* Tương tác giữa con người và máy tính.



Hình . Tương tác con người và máy tính

* Nhiều công ty xe, có thể ứng dụng công nghệ nhận dạng để dò tìm những biểu cảm của tài xế để đưa ra những cảnh báo hợp lý.



Hình . Biểu cảm khuôn mặt 1

* Hay những nhà quảng cáo trên web cần biết những thông tin về khuôn mặt xem xét sự hiệu quả của những logo có thể đặt ở nơi nào trên cổng thông tin web đó. Thêm nữa, những công ty làm phim 3D cũng quan tâm đến những thông tin biểu cảm của khuôn mặt cho những sản phẩm của họ.



Hình . Biểu cảm khuôn mặt 2

Các đặc trưng sinh trắc học của cơ thể người được sử dụng phải đảm bảo các tiêu chuẩn sau đây [[1]](#ref10):

* Tính rộng rãi: là tính chất cho biết thông thường mọi người đều có đặc trưng này, có thể tạo ra khả năng sử dụng hệ thống an ninh sinh trắc học cho một số lượng người lớn.
* Tính phân biệt: là tính chất phân biệt đặc trưng sinh trắc học giữa hai người bất kỳ phải khác nhau, đảm bảo sự duy nhất của chủ thể sinh trắc.
* Tính ổn định: là tính chất mà đặc trưng sinh trắc phải có tính ổn định trong một thời gian tương đối dài.
* Tính dễ thu thập: là tính chất đặc trưng sinh trắc học phải dễ dàng thu nhận mẫu khi đăng ký, kiểm tra xác thực, nâng cao tính khả thi trong sử dụng.
* Tính hiệu quả: là tính chất mà việc xác thực sinh trắc phải chính xác, nhanh chóng và tài nguyên cần sử dụng được chấp nhận.
* Tính chấp nhận được: là tính chất mà quá trình thu thập mẫu sinh trắc phải được sự đồng ý của người người dùng.
* Chống giả mạo: là tính chất ưu việt của việc sử dụng đặc trưng sinh trắc khả năng mẫu sinh trắc khó bị giả mạo cao, ...

Mỗi loại đặc trưng sinh trắc có điểm mạnh và điểm yếu riêng. Tuy nhiên không một đặc trưng

sinh trắc nào thỏa mãn đầy đủ và tốt hết các yêu cầu nên trên. Qua quá trình nghiên cứu, các chuyên gia đã đưa ra bảng so sánh khái quát các tiêu chuẩn đánh giá tương ứng các đặc trưng sinh trắc học sau:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sinh trắc học | Vân tay | Khuôn mặt | Vân bàn tay | Móng mắt | Giọng nói |
| Tính rộng rãi | M | H | M | H | M |
| Tính phân biệt | H | L | M | H | L |
| Tính ổn định | H | M | M | H | L |
| Tính dễ thu nhập | M | H | M | M | M |
| Tính hiệu quả | H | L | M | H | L |
| Tính chấp nhận | M | H | M | L | H |
| Chống giả mạo | M | H | L | L | H |

Trong đó: H (cao), M (trung bình) và L (thấp).

Những phương pháp xác thực truyền thống (dựa trên mật khẩu hay ký hiệu) thì quá dễ để phá vỡ. Những phương pháp sinh trắc học là một lựa chọn hợp lý nhưng cũng có những mặt hạn chế :

* Phương pháp quét mống mắt rất tin cậy nhưng áp đặt người sử dụng quá nhiều, chi phí mắc để thực hiện và không được chấp nhận bởi nhiều người.
* Xác thực vân tay được chấp nhận mang tính chất xã hội, nhưng không thể ứng dụng đối với những người không tán thành.

Ngược lại, xác thực bằng khuôn mặt tương ứng với một sự thoả hiệp giữa sự chấp nhận có tính chất xã hội và tin cậy.

## Rút trích đặc trưng

### Principal component analysis (PCA)

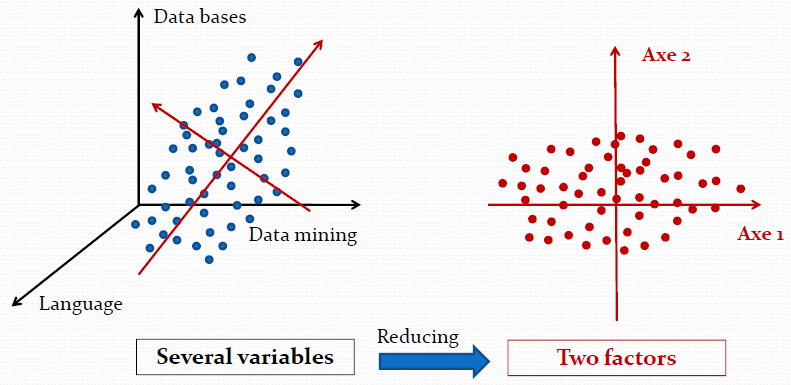
#### Giới thiệu

Phân tích thành phần chính (Principal Component Analysis – PCA) [[2]](#ref4) là một trong những phương pháp phân tích dữ liệu nhiều biến đơn giản nhất. Phương pháp PCA được phát minh vào năm 1901 bởi Karl Pearson. Phương pháp này thường dùng để giảm số lượng biến và hiệu quả nhất khi tập dữ liệu ban đầu có sự dư thừa, trùng lặp. Kết quả của phương pháp này cho ta số lượng biến nhỏ hơn mà ta gọi chúng là các thành phần chính.

Đối với các dữ liệu cần phân tích ban đầu phụ thuộc vào nhiều biến, các biến này thường có tương quan với nhau. Điều đó sẽ gây nên sự bất lợi cho việc xây dựng các mô hình tính toán khi dựa trên các biến này, và với số lượng biến quá lớn chúng ta sẽ rất khó để có cái nhìn trực quan về dữ liệu.

Để khắc phục điều này, phương pháp PCA sẽ biễu diễn lại dữ liệu lên một không gian mới có cơ sở trực giao, tức là nếu ta xem mỗi cơ sở trong không gian mới là một biến thì hình ảnh của dữ liệu gốc trong không gian mới này sẽ được biểu diễn thông qua các biến độc lập tuyến tính. Một vấn đề phát sinh ở đây chính là nếu chuyển dữ liệu ban đầu sang không gian mới thì những dữ liệu quan trọng, đáng quan tâm của dữ liệu ban đầu liệu có bị mất theo hay không? Để giải quyết vấn đề này, phương pháp PCA sẽ tìm không gian mới với tiêu chí cố gắng phản ánh được càng nhiều dữ liệu gốc càng tốt. Không gian tốt nhất với số chiều nhỏ nhất được xác định bởi những “thành phần chính” tốt nhất. Thước đo để tìm ra các “thành phần chính” tốt nhất chính là phương sai. Do các biến cơ sở của không gian mới độc lập, nên ta có thể dễ dàng tính toán phương sai của từng biến đối với dữ liệu, chính điều này giúp ta có thể lựa chọn các biến hợp lý để mô tả dữ liệu mà vẫn dữ được các thông tin chính quan trọng của tập dữ liệu đó.

Nói một cách ngắn gọn, mục tiêu của PCA là tìm một không gian mới (với số chiều nhỏ hơn không gian cũ). Các trục tọa độ trong không gian mới được xây dựng sao cho trên mỗi trục, độ biến thiên của dữ liệu trên đó là lớn nhất có thể.



Hình . Ví dụ về phương pháp PCA

Hình bên trái cho ta hình ảnh của tập dữ liệu ban đầu (tập điểm màu xanh). Tập dữ liệu này được mô tả trong không gian 3 chiều (trục màu đen). Dễ dàng thấy được, không gian này không biểu diễn tốt nhất độ biến thiên của dữ liệu. Bằng cách áp dụng phương pháp PCA, ta sẽ tìm ra được không gian mới với số chiều nhỏ hơn (trục màu đỏ). Dữ liệu sẽ ban đầu sẽ được biểu diễn lại trong không gian mới như hình bên phải (tập điểm màu đỏ). Rõ ràng, sau khi biểu diễn lại, độ biến thiên dữ liệu theo 2 trục của không gian mới này tốt hơn hẳn so với ban đầu.

Ưu điểm chính của PCA được sử dụng vào trong cách tiếp cận Eigenface, giúp ích trong việc giảm kích thước của cơ sở dữ liệu để nhận dạng hay xác thực một hình ảnh thử nghiệm nào đó. Các hình ảnh được lưu trữ thông qua vectơ đặc trưng của chúng trong cơ sở dữ liệu được tạo ra bằng cách chiếu chúng vào không gian Eigenface thu được khi tính toán tập huấn luyện. PCA được áp dụng vào cách tiếp cận Eigenface để làm giảm số chiều của một tập dữ liệu lớn.

#### Cơ sở lý thuyết

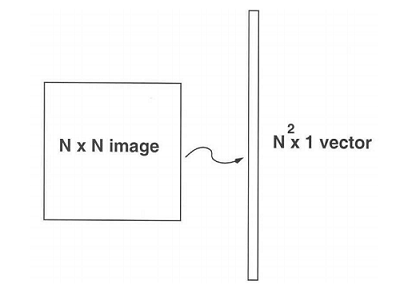
* Cách tiếp cận Eigenface:

Đây là phương pháp thích hợp và hiệu quả trong nhận dạng/xác thực khuôn mặt bởi sự đơn giản, nhanh chóng và khả năng học tập của nó. Eigenface là một tập các vectơ được sử dụng trong các vấn đề về thị giác máy tính như trong nhận dạng khuôn mặt hay xác thực con người. Eigenface có thể được coi là thành phần chủ yếu của phân phối khuôn mặt, các eigen vector của ma trận hiệp phương sai (covariance) của tập hình ảnh khuôn mặt, nơi mà hình ảnh NxN pixel được coi là một điểm trong không gian N2 chiều.

Mỗi hình ảnh khuôn mặt có thể được đại diện xấp xỉ bằng sự kết hợp tuyến tính của các vector trong tập Eigenface, và các vector hệ số chính là vector đặc trưng của khuôn mặt hiện tại. Số lượng tối đa của các vector trong tập Eigenface bằng với số hình ảnh khuôn mặt trong tập huấn luyện. Như vậy, bằng cách sử dụng tập Eigenface tốt nhất, ta sẽ có được hình ảnh tương đương với khuôn mặt ban đầu. Càng sử dụng tập Eigenface với số lượng vector nhỏ, thì hiệu quả tính toán sẽ càng cao.

* Ý tưởng chính của cách tiếp cận Eigenface :

Giả sử Γ là 1 vector N2x 1, tương ứng với một hình ảnh khuôn mặt N x N.



Hình . Biểu diễn khuôn mặt thành vector

Ý tưởng là biểu diễn Γ (Φ=Γ -mean face) trong không gian mới có chiều là K nhỏ hơn không gian ban đầu.

Φ = w1u1 + w2u2 + ... + wKuK (K << N2)

Tính toán Eigenface:

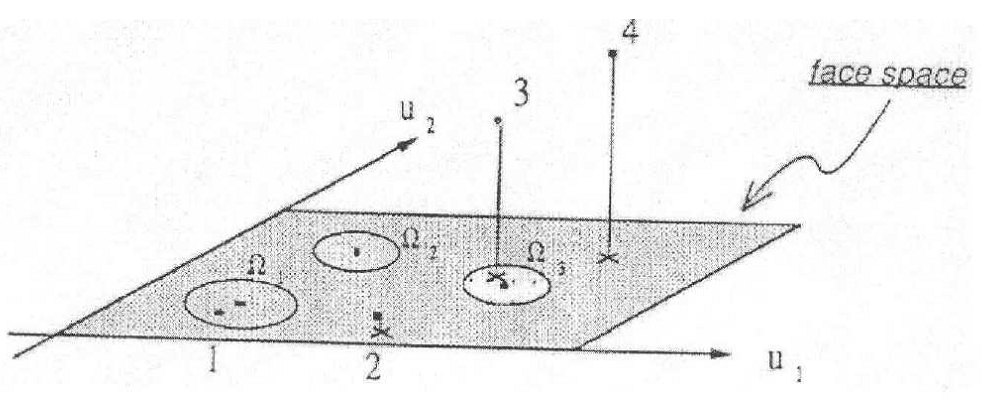
* Bước 1: Thu thập tập các hình ảnh khuôn mặt ( I1, I2, .., IM). Các hình ảnh này phải được canh giữa và có kích thước giống nhau.
* Bước 2: Biểu diễn mỗi hình ảnh Ii thành 1 vector Γi
* Bước 3: Tính vector khuôn mặt trung bình Ψ:
* ΨΓi
* Bước 4: Trừ đi khuôn mặt trung bình
* Φi=Γi−Ψ
* Bước 5: Tính ma trận hiệp phương sai (covariance) C:
* 
* Trong đó, 
* Bước 6: Tính các eigen vector ui của ma trận C = AAT
* Bước 6.1: Xem xét ma trận ATA ( ma trận MxM) (Do kích thước ma trận AAT rất lớn)
* Bước 6.2: Tính eigen vector vi của ATA



\*

Như vậy, ATA và AAT có cùng eigen value và các eigen vector có quan hệ : 

* Bước 6.3: Tính M eigen vector tốt nhất của AAT :  ( phải chuẩn hóa ui)
* Bước 7: Chỉ giữ K eigen vector tương ứng với K eigen value lớn nhất. (Cách tính trị riêng, vector riêng được đề cập ở phần Phụ lục).



Các vector (ảnh khuôn mặt “gốc” ban đầu)

Chiếu các vào trong *“không gian khuôn mặt”* ta được các vector  . Tọa độ của  trong không gian này là vector 

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

Hình . Hình minh họa giải thuật PCA

Biểu diễn khuôn mặt lên không gian EIGENFACE:

Mỗi khuôn mặt (trừ mean) Φi có thể được biểu diễn bởi sự kết hợp tuyến tính của K eigen vector:





Hình . Sự kết hợp tuyến tính của khuôn mặt

Như vậy, mỗi khuôn mặt đã chuẩn hóa Φi được biểu diễn trên không gian EIGENFACE bởi 1 vector:



Vector  trên chính là vector đặc trưng của hình ảnh ban đầu.

#### Các vấn đề

Ngoài những ưu điểm của Eigenface như dễ cài đặt, độ chính xác tương đối cao, và tốc độ nhanh…Eigenface còn tồn tại trong nó nhiều hạn chế. Đầu tiên phải nói tới sự phụ thuộc cứng nhắc vào tập huấn luyện, có nghĩa là các ảnh khuôn mặt kiểm tra phải giống với các ảnh đã đăng ký về kích thước, tư thế, ánh sáng…

Kích thước của ảnh kiểm tra cần phải có cùng kích thước với các ảnh trong tập huấn luyện. Vì vậy, chỉ cần sai lệch nhỏ về kích thước, thuật toán sẽ không tiến hành được. Để giải quyết điều này, đòi hỏi ta cần phải tiến hành tiền xử lý các ảnh, kể cả các ảnh trong tập huấn luyện. Hay nói cách khác, ta cần có hàm xử lý cắt khuôn mặt từ ảnh gốc ban đầu và làm sao cho khuôn mặt phải được canh giữa hình, đồng thời kích thước của các ảnh cắt này phải bằng nhau.

Sự thay đổi điều kiện ánh sáng, ảnh hưởng khá lớn tới hiệu quả của thuật toán này. Khi cường độ sáng quá lớn, hay có chênh lệch đáng kể về điều kiện sáng giữa ảnh kiểm tra và tập huấn luyện. Tỷ lệ sai sẽ tăng lên rõ rệt. Đây là một trong những hạn chế lớn nhất khi sử dụng thuật toán này.

Hướng chụp khuôn mặt khác nhau cũng ảnh hưởng tới thuật toán. Với hình ảnh khuôn mặt có góc chụp khác nhau, sự sai lệch giữa các vector đặc trưng sẽ lớn hơn. Chính vì thế, hiệu suất nhận dạng hay xác thực sẽ giảm. Biểu cảm khuôn mặt cũng là một trong những nguyên nhân gây ra việc giảm hiệu suất.

Ngoài ra, còn có một số các yếu tố khác tác động đến kết quả thuật toán như ảnh nền của khuôn mặt, chất lượng của ảnh đầu vào…Đối với ảnh nền, những chi tiết quá rõ gần khuôn mặt sẽ gây ra độ nhiễu lớn, ảnh hưởng tới việc rút trích đặc trưng và cho ta kết quả không mong muốn. Vì vậy, trong quá trình tiền xử lý, ta nên làm mờ phông nền, để giảm sự ảnh hưởng trên. Chất lượng ảnh đầu vào cũng là yếu tố ta cần nhắc tới, những hình ảnh có cùng chất lượng và rõ nét sẽ cho ta hiệu suất cao hơn những ảnh kém chất lượng, chi tiết mờ nhạt…

### Fisherface

#### Giới thiệu

Vì phương pháp Eigenface chỉ đơn thuần sử dụng “Phân tích thành phần chính” – Principal Components Analysis (PCA), không xem xét đến các lớp khác nhau, dẫn đến một số thông tin phân lớp có thể bị bỏ qua. Khắc phục điều đó, phương pháp Fisherface sử dụng thêm “Phân tích phân biệt tuyến tính” – Linear Discriminant Analysis (LDA), không tập trung vào việc tối đa phân hóa các cá thể, mà là tối đa phân hóa các lớp, gom cụm các cá thể thành lớp [[3]](#ref1).

#### Cơ sở lý thuyết

Cho X là một vector ngẫu nhiên trong tập mẫu từ c lớp

Các ma trận phân tán SB và SW được tính như sau:

trong đó µ là tổng bình quân:

Và µi là bình quân của lớp i {1, …, c}:

Ma trận chiếu W, với mục tiêu tối đa hóa phân chia lớp, có dạng như sau:

Có một vấn đề đó là bậc tối đa của ma trận SW là (N - c), với N mẫu và c lớp. Trong vấn đề xác thực mẫu, số lượng mẫu N luôn luôn nhỏ hơn số chiều của dữ liệu đầu vào (số lượng điểm ảnh), do đó ma trận phân tán SW là ma trận đơn. Điều này được giải quyết bằng sử dụng PCA trên dữ liệu gốc và chiếu các mẫu vào không gian (N-c) chiều. Sau đó, sử dụng LDA trên phần dữ liệu đã được thu giảm, bởi vì S­W không còn là ma trận đơn nữa.

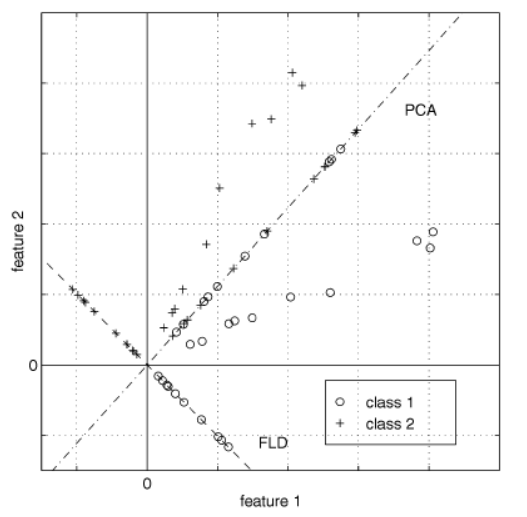
Vấn đề tối ưu hóa được viết lại như sau:

Ma trận W cần tìm, sau khi chiếu mẫu vào không gian (c-1) chiều, lúc này là:

W=WTfld W Tpca

#### Các vấn đề

Fisherface về cơ bản là tương tự như Eigenface, nhưng có cải tiến trong phân loại các lớp khác nhau. Với Fisher’s Linear Discriminant (FLD), chúng ta có thể phân loại tập huấn luyện để xử lý với những người khác nhau và biểu hiện trên khuôn mặt khác nhau, từ đó cải thiện độ chính xác. Bên cạnh đó, Fisherface loại bỏ ba thành phần chính đầu tiên chịu trách nhiệm những thay đổi cường độ ánh sáng, nó là bất biến hơn với cường độ ánh sáng.



Hình . Hình ảnh so sánh PCA và FLD trong bài toán hai lớp, nơi dữ liệu cho mỗi lớp nằm gần một không gian con tuyến tính

Eigenface không xét đến sự khác nhau giữa các lớp mà chỉ tính cục bộ trong mỗi lớp khi xây dựng không gian vector, nên không gian của Eigenface (PCA) không có sự phân biệt giữa lớp 1 và lớp 2. Còn Fisherface sử dụng thông tin cụ thể của mỗi lớp để tính toán, xác định các lớp khác nhau với cách thống kê khác nhau, nên những hình ảnh trong tập huấn luyện được chia thành các lớp tương ứng. Ta có thể thấy không gian của Fisherface (FLD) phân biệt riêng theo từng lớp.

Ưu điểm:

* FLD đạt được sự phân bố giữa các lớp tốt hơn PCA
* Sử dụng ít bộ nhớ
* Không bị ảnh hưởng nhiều bởi ánh sáng và sự thay đổi biểu cảm khuôn mặt
* Chính xác hơn phương pháp Eigenface

Nhược điểm:

* Mất khả năng tái tạo khuôn mặt của Eigenface
* Thời gian chạy lâu

### Local binary pattern

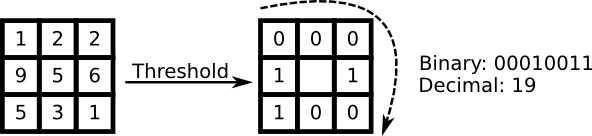
#### Giới thiệu

LBP là viết tắt của Local Binary Pattern hay là mẫu nhị phân địa phương được Ojala trình bày vào năm 1996 như là một cách đo độ tương phản cục bộ của ảnh [[4]](#ref11). Ý tưởng cơ bản của Local Binary Patterns là để tóm tắt các cấu trúc cục bộ trong một hình ảnh bằng cách so sánh mỗi pixel với các pixel lân cận. Lấy một pixel như là trung tâm và ngưỡng của các pixel láng giềng.Nếu độ xám của pixel trung tâm lớn hơn của các pixel láng giềng, biểu thị nó với 1 và 0 nếu ngược lại. Sau khi kết thúc chúng ta sẽ thu được một số nhị phân cho mỗi pixel, vd như :11001111. Vì vậy với 8 pixel bao quanh, bạn sẽ có 28 trường hợp có thể, được gọi là Local Binary Patterns hoặc LBP codes.

#### Cơ sở lý thuyết

Đầu tiên ta phải chuyển ảnh màu thành các ảnh xám để có thể tính được giá trị LBP.

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 độ xám 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 . Minh họa giải thuật LBP

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

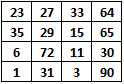
Với ic là độ xám của pixel trung tâm, ip là giá trị độ xám của pixel thứ p trong lân cận 8 của pixel trung tâm. Hàm s được định nghĩa như sau:

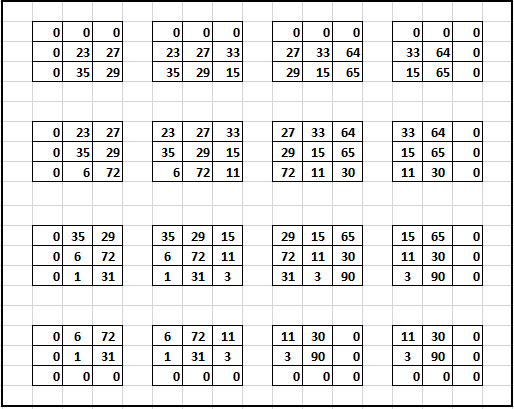
Bằng cách định nghĩa này, toán tử LBP bất biến đối với bất kỳ sự biến đổi độ sáng đều và bảo toàn trật tự mật độ các pixel trong một lân cận cục bộ.

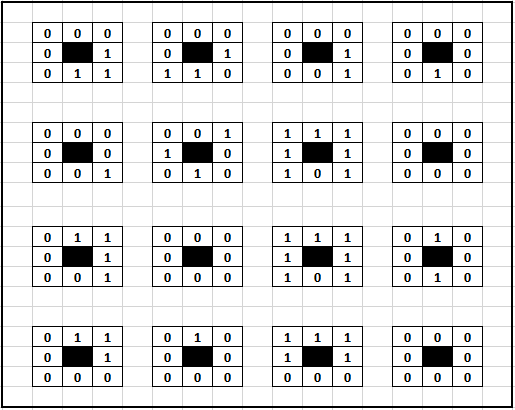
Ví dụ [[5]](#ref12) với một ảnh có kích thước 4x4 :

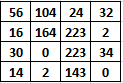
|  |  |  |  |
| --- | --- | --- | --- |
| 23 | 27 | 33 | 64 |
| 35 | 29 | 15 | 65 |
| 6 | 72 | 11 | 30 |
| 1 | 31 | 3 | 90 |

Chúng ta sẽ tính được giá trị đặc trưng LBP như sau:









Hình . Ví dụ về quá trình tính toán đặc trưng LBP

**Histogram :**

Sau khi ảnh được chuyển thành ảnh LBP, mỗi pixel sẽ có một mức xám khác nhau. Số mức xám nằm trong khoảng 0-255 (256 màu xám).

Biểu đồ histogram được sử dụng để lưu trữ thông tin cục bộ trong hình ảnh. Mỗi phần tử trong biểu đồ histogram có giá trị là số lần mức xám xuất hiện trong bức ảnh LBP. Như vậy, ta sẽ tạo một vector đặc trưng histogram với kích thước 1x256. Ta sẽ lặp qua từng pixel của ảnh và đếm số lượng các pixel có mức xám từ 0-255 để lưu vào vector tương ứng trên.

Từ 2 vector đặc trưng, ta có thể tính khoảng cách để đo sự giống và khác nhau giữa 2 bức ảnh sử dụng khoảng cách Chi-Squared.

Từ đó ta sẽ lựa chọn ngưỡng để xác định FAR và FRR thông qua khoảng cách histogram và suy ngược lại ngưỡng phù hợp. Với ngưỡng phù hợp này, ta sẽ sử dụng để xác thực một bức ảnh.

#### Các vấn đề

Ưu điểm:

* Thuật toán trích rút đặc trưng LBP cài đặt đơn giản.
* Hoạt động tốt với ảnh có kích thước khác nhau.
* Thời gian tính toán giá trị đặc trưng nhanh vì nó làm việc với giá trị nguyên.

Nhược điểm: Tuy nhiên độ chính xác không cao.

Ứng dụng: Được ứng dụng trong bài toán phát hiện mặt người.

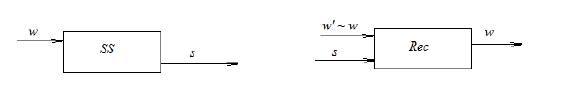
## Bảo mật mẫu

Xu hướng sử dụng sinh trắc học trong việc xác thực ngày càng phát triển. Tuy nhiên nếu những thiết bị này bị mất hoặc bị tấn công, thì những mẫu sinh trắc (được lưu dưới dạng thô hoặc biến đổi) sẽ bị người khác đánh cắp và sử dụng trái phép. Đặc tính của sinh trắc học là tính nhiễu do tác động của điều kiện ngoại cảnh, thường thay đổi theo thời gian, dẫn đến việc không thể rút trích một cách hoàn toàn chính xác [[6]](#ref3).

Nội dung của phần này sẽ lần lượt trình bày về phương pháp để bảo vệ mẫu sinh trắc đó là Fuzzy Extractor bao gồm Secure Sketch và Keyed-hash message authentication code (HMAC). Secure Sketch sẽ giúp việc lấy dữ liệu đáng tin cậy từ mẫu sinh trắc có nhiễu, còn HMAC sẽ đảm nhận bảo mật cho phần dữ liệu đó.

### Secure Sketch

Secure Sketch là phương pháp giúp phục hồi lại mẫu sinh trắc gốc khi được cung cấp một mẫu khác gần giống với mẫu gốc. Hai mẫu sinh trắc được xem là gần giống nhau khi khoảng cách của chúng nằm trong một ngưỡng cho phép.



Hình . Mô hình Secure Sketch

**Chú thích:**

SS: secure sketch s: sketch – thông tin mở

Rec: Recover w’: mẫu xác thực mới

w: mẫu bí mật ban đầu.

**Tính chất của Secure Sketch:**

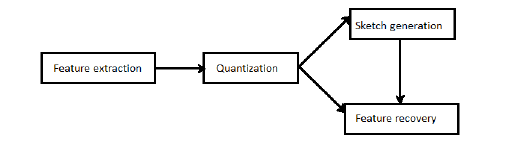
1. s là một chuỗi nhị phân {0, 1}
2. w chỉ được khôi phục khi
3. s không được tiết lộ quá nhiều thông tin về w.

Tổng quan, Secure Sketch gồm có hai giai đoạn:

* **Giai đoạn tạo sketch**: dữ liệu đầu vào là mẫu sinh trắc học cần được bảo vệ, và đầu ra là thông tin mở được rút trích từ mẫu gọi là sketch. Tuy sketch là thông tin mở (public information), nhưng không được tiết lộ quá nhiều thông tin về mẫu ban đầu, người dùng cũng không thể khôi phục được mẫu dựa trên sketch.
* **Giai đoạn khôi phục mẫu ban đầu**: dữ liệu đầu vào là mẫu xác thực mới và sketch. Nếu mẫu xác thực mới gần giống với mẫu ban đầu thì mô hình sẽ khôi phục và trả về mẫu sinh trắc ban đầu.

Về hiện thực, nhóm sử dụng mô hình Codebook Secure Sketch do Tran Tri Dang, Quynh Chi Truong và Tran Khanh Dang thuộc nhóm Data Security Applied Research (D-STAR Lab) phát triển trong “Practical Construction of Face-Based Authentication Systems with Template Protection Using Secure Sketch”. Lý do nhóm chọn mô hình này vì mô hình cho kết quả khá tốt trên tập dữ liệu Faces94.

Codebook Secure Sketch dùng để sửa những lỗi gây ra bởi “nhiễu giá trị”. Nếu như phần tử bị nhiễu làm thay đổi giá trị trong một ngưỡng δ định trước so với phần tử gốc, thì giá trị của phần tử này sẽ được hồi phục.



Hình . Mô hình Codebook Secure Sketch

Mô hình Codebook Secure Sketch có thể chia thành bốn giai đoạn: Rút trích đặc trưng (feature extraction), lượng tử hóa (quantization), tạo sketch (sketch generation) và khôi phục đặc trưng (feature recovery). Tuy nhiên, ở đây ta chỉ quan tâm tới vấn đề bảo mật mẫu, nên ta sẽ không nói đến giai đoạn đầu tiên là rút trích đặc trưng.

**Lượng tử hóa**

Sau quá trình rút trích, giá trị của các đặc trưng sẽ nằm trong miền liên tục (continuous domain). Vì mô hình Codebook chỉ hoạt động trên miền rời rạc (discrete domain), nên ta cần thực hiện lượng tử hóa. Lượng tử hóa cũng có lợi ích là giá trị rời rạc sẽ giúp việc hồi phục mẫu gốc chính xác và dễ dàng hơn.

Thoạt đầu, với việc giảm không gian tìm kiếm liên tục với vô hạn phần tử thành không gian rời rạc với hữu hạn phần tử có vẻ sẽ ảnh hưởng đến tính bảo mật. Tuy nhiên, việc xác thực bằng sinh trắc học không phải là một quá trình trùng khớp hoàn toàn, nên không cần phải thử tất cả các giá trị để tấn công. Hơn nữa, ta cũng có thể kiểm soát kích thước của miền lượng tử, bằng cách điều chỉnh phạm vi của các giá trị liên tục khi được ánh xạ sang giá trị lượng tử. Vì thế, lượng tử hóa không ảnh hưởng đến tính bảo mật.

Quá trình lượng tử hóa:

* Chuẩn hóa: sau khi chuẩn hóa, các giá trị đặc trưng sẽ là số thực nằm trong [0, 1]
* Chuyển từ miền [0, 1] sang miền [0, N].
* Làm tròn giá trị để chuyển về miền số nguyên.

Gọi x là giá trị trước lượng tử hóa, x’ là giá trị sau khi lượng tử hóa. x’ được tính như sau:

x’ = round(xN)

**Tạo Sketch**

Gọi w là vector đặc trưng sau khi lượng tử hóa. Sau khi lượng tử hóa thì miền giá trị của w là [0, N].

* Ta tạo ra một codebook với nhiều codeword, các codeword được phân bố đều trong [0, N]. Khoảng cách giữa các codeword kề nhau (ci, ci+1) là bằng nhau và bằng 2, với là giá trị nguyên dương.
* Định nghĩa hàm ánh xạ M(w), khi giá trị w nằm trong [ci - , ci + ], hàm sẽ trả về codeword ci gần nhất.
* Sketch được tính theo công thức:

SS(w) = w – M(w) (1)

**Khôi phục đặc trưng**

Dữ liệu đầu vào là mẫu xác thực w’, ta sử dụng sketch SS(w) để khôi phục lại mẫu w ban đầu nếu

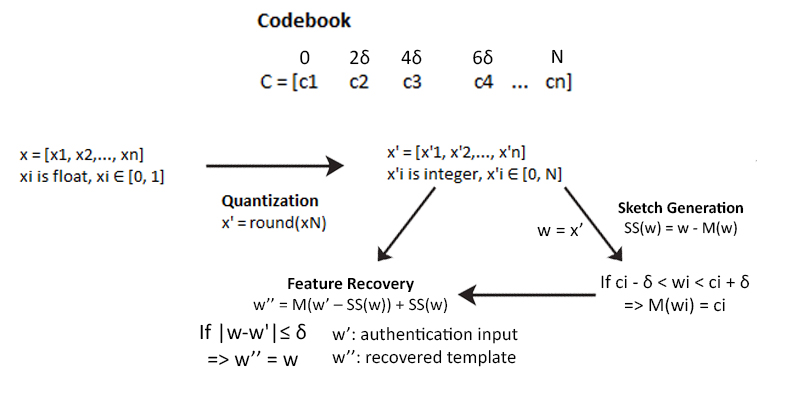
Gọi w’’ là mẫu được khôi phục, w’’ được tính như sau:

w’’ = M(w’ – SS(w)) + SS(w) (2)

Để chứng minh rằng mẫu ban đầu w sẽ được khôi phục đúng nếu , thay (1) vào (2), ta có:

w’’ = M(w’ – w + M(w)) + w – M(w) (3).

Nếu , thì w’ – w – M(w) nằm trong [M(w) –, M(w) + ]. Theo quy tắc xây dựng codebook, áp dụng hàm ánh xạ với bất kì giá trị trong khoảng này, đều sẽ trả về codeword gần nhất, cũng tương tự như M(w). Vậy, thay thế M(w’ – w + M(w)) bằng M(w) trong (3), ta sẽ có w’’ = w.



Hình . Quá trình sinh sketch và khôi phục

**Hiệu quả bảo mật**

Ta sẽ xem xét vấn đề bảo mật đối với phương thức tấn công cơ bản nhất: brute-force. Mỗi mẫu đặc trưng w có thể được xem là một vector gồm N phần tử. Để khôi phục thành công w, mỗi phần tử của vector phải được sửa chính xác. Ta thấy rằng mỗi codeword đều phải được thử để có thể tấn công brute-force.

 Hiện nay, ta sử dụng chung lượng tử hóa và codebook cho tất cả các phần tử bất chấp phân bố giá trị của chúng. Giả sử ta có S codeword trong codebook, vậy trung bình, kẻ tấn công phải thử trường hợp để lấy được mẫu w chính xác.

### Keyed-hash message authentication code (HMAC)

Hàm băm (hash) là giải thuật nhằm tạo ra một bản nhận dạng (fingerprint) từ một thông điệp, chuỗi ký tự hoặc tập tin… nhằm bảo đảm tính toàn vẹn dữ liệu.

Hàm băm có các đặc điểm như sau [[7]](#ref2):

* Đầu vào là dữ liệu có độ dài bất kì.
* Đầu ra là dữ liệu có độ dài cố định.
* Hàm băm phải là hàm một chiều. Tức với bất kỳ giá trị băm h, không thể tìm được x sao cho H(x)=h.
* Biết x1, rất khó tìm được x2 sao cho H(x1) = H(x2)
* Rất khó tìm được cặp (x1, x2) sao cho H(x1) = H(x2)

Để xác thực xem mẫu khôi phục lại w’’ có giống với mẫu gốc w không, ban đầu ta sử dụng HMAC trên mẫu w, và lưu giữ kết quả này. Sau đó, ta cũng sử dụng HMAC trên mẫu w’’, và so sánh hai giá trị.

Mô hình HMAC nhận giá trị đầu vào là một khóa K và thông điệp m, khóa K được sử dụng khi băm thông điệp. Hiện nay có rất nhiều giải thuật băm khác nhau, trong đề tài này, nhóm lựa chọn sử dụng giải thuật HMAC SHA-1.

## Huấn luyện, đăng ký và xác thực

### Principal component analysis (PCA)

#### Huấn luyện

Để có được không gian EIGENFACE, trước hết chúng ta cần có tập ảnh huấn luyện ban đầu. Tập hình ảnh dùng để huấn luyện sẽ gồm các hình ảnh khuôn mặt của nhiều người. Những hình ảnh này có thể có góc chụp hay biểu cảm khuôn mặt khác nhau, đặc điểm ngoài của khuôn mặt thay đổi như râu, tóc, mắt kiếng…

Dưới đây là hình ảnh ví dụ cho tập ảnh huấn luyện, tập ảnh này gồm 10 tấm hình của một người.



Hình . Tập huấn luyện Eigenface

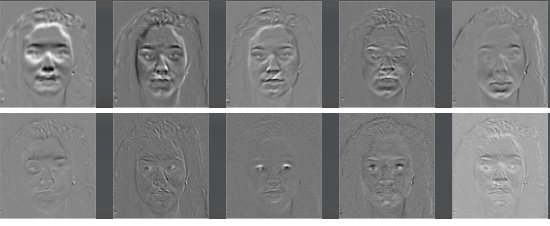
Khi đã có được tập ảnh huấn luyện, chúng ta tiến hành tiền xử lý ảnh. Các bước tiền xử lý ảnh bao gồm làm mờ nền, cân bằng sáng, làm rõ chi tiết khuôn mặt… Ta tiến hành huấn luyện tập ảnh trên với thuật toán Eigenface.

Bước đầu tiên của thuật toán sẽ là biểu diễn lại các hình ảnh trong tập huấn luyện (kích thước N x N) thành dạng vector (kích thước N2 x 1). Sau đó tìm khuôn mặt trung bình của các ảnh trong tập huấn luyện.



Hình . Khuôn mặt trung bình

Sau khi đã có khuôn mặt trung bình, ta lấy từng ảnh trong tập huấn luyện trừ đi mặt trung bình. Tiếp theo tính ma trận hiệp phương sai của tập các ảnh đã trừ cho mặt trung bình. Khi đã có ma trận hiệp phương sai, ta tiến hành tìm các eigen value và eigen vector của ma trận này. Tập hợp các eigen vector ứng với eigen value lớn nhất tương ứng tạo nên các cơ sở cho không gian khuôn mặt EIGENFACE của ta. Việc cần làm tiếp theo chỉ là chiếu một hình ảnh khuôn mặt nào đó lên không gian khuôn mặt EIGENFACE này ta sẽ thu được vector đặc trưng của khuôn mặt.



Hình . Các eigen vector của tập huấn luyện

#### Đăng ký

Để có dữ liệu xác thực, yêu cầu người dùng phải tiến hành bước đăng ký. Ở bước này, người dùng sẽ cung cấp một hoặc một số hình ảnh khuôn mặt của mình cho hệ thống, từ đó hệ thống sẽ tiến hành rút trích đặc trưng khuôn mặt và lưu lại trong cơ sỡ dữ liệu.

Khi có được hình ảnh khuôn mặt của người dùng, việc đầu tiên cần làm là tiến hành tiền xử lý ảnh như đã nói trong giai đoạn huấn luyện.

Bước tiếp theo sẽ là chuẩn hóa và chiếu các hình ảnh này vào trong không gian EIGENFACE. Từ đó, ta sẽ có được các vector đặc trưng khuôn mặt của người dùng đăng ký. Các vector này được lưu lại và được sử dụng để đối chiếu so sánh với mẫu sinh trắc đầu vào của hệ thống khi xác thực.

#### Xác thực

Ở giai đoạn này, ta cần cung cấp một mẫu sinh trắc khuôn mặt để xác thực đầu vào sinh trắc có phải là người đã đăng ký với hệ thống trước đó không.



Hình . Hình ảnh người cần xác thực

Ta sẽ tiến hành tiền xử lý, chuẩn hóa (biểu diễn lại, trừ cho mặt trung bình…) và chiếu hình ảnh lên không gian khuôn mặt EIGENFACE để thu được vector đặc trưng.

Tiến hành tính toán và so sánh khoảng cách Euclide[8] giữa vector đặc trưng của hình ảnh cần xác thực lần lượt với các vector đặc trưng có được trong giai đoạn đăng ký, nếu khoảng cách này thỏa điều kiện dưới một ngưỡng (threshold) nào đó, ta sẽ chấp nhận xác thực. Nếu so sánh hết với các vector đặc trưng đã lưu mà vẫn không thỏa điều kiện này, thì việc xác thực bị từ chối.

Ở đây ta thấy, kết quả của việc xác thực phụ thuộc vào việc xác định giá trị của ngưỡng (threshold). Vì vậy, cần có giải pháp để lựa chọn giá trị này sao cho phù hợp nhất để hiệu quả của thuật toán cao hơn, đồng thời tỷ lệ sai xót giảm.

### Fisherface

#### Huấn luyện

Để huấn luyện ta cần có tập ảnh của một người để tính toán. Dưới đây là một tập ảnh dùng để huấn luyện trong giải thuật Fisherface:



Hình . Tập train Fisherface

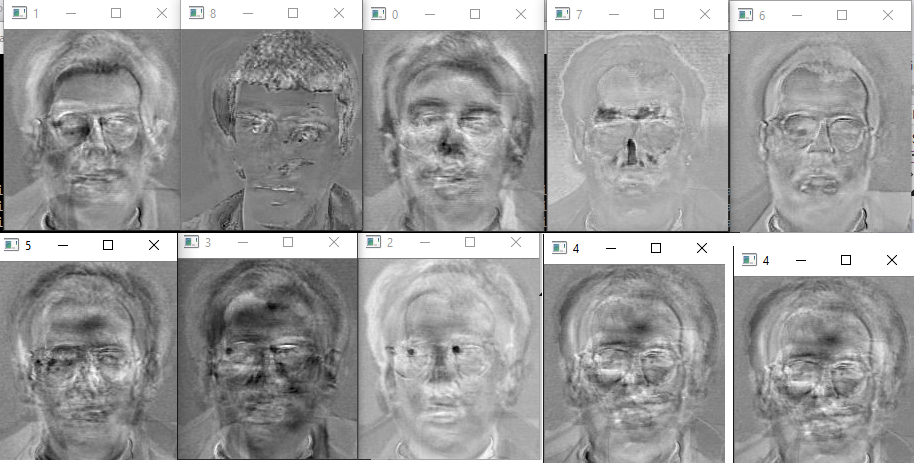
Ta cũng thực hiện bước tiền xử lý hình ảnh tương tự như Eigenface: làm mờ nền, cân bằng sáng, làm rõ chi tiết khuôn mặt…

Bước đầu tiên của thuật toán sẽ là biểu diễn lại các hình ảnh trên (N x N) thành dạng vector (N2x1). Sau đó tìm khuôn mặt trung bình của các ảnh trong tập huấn luyện. Ảnh sau đại diện cho khuôn mặt trung bình mà ta đã tính toán.



Hình . Khuôn mặt trung bình

Sau khi đã có mặt trung bình, ta lấy từng ảnh trong tập huấn luyện, kết hợp với các công thức và ảnh trung bình, để tính ma trận phân tán trong SB và phân tán ngoài SW. Tiếp theo tính các giá trị Wfld và Wpca rồi tính ma trận W sao cho có độ phân tán trong (SW) là nhỏ nhất.



Hình . Các vector sau khi train Fisherface

#### Đăng ký

Bước đăng ký của Fisherface cũng tương tự như của Eigenface. Người dùng sẽ cung cấp một hoặc một số hình ảnh khuôn mặt của mình cho hệ thống, từ đó hệ thống sẽ tiến hành rút trích đặc trưng khuôn mặt và lưu lại trong cơ sỡ dữ liệu.

Sau đó, ta sẽ tiến hành tiền xử lý hình ảnh, chuẩn hóa và chiếu các hình ảnh vào không gian Fisherface. Kết quả sau khi thực hiện phép chiếu chính là vector đặc trưng của người dùng. Các vector này được lưu lại và được sử dụng để đối chiếu so sánh với mẫu sinh trắc đầu vào của hệ thống khi xác thực.

#### Xác thực

Sau khi trải qua bước huấn luyện, ta có được không gian vector Fisherface. Ta chọn ảnh bất kì dùng xác thực, ví dụ như hình sau:



Hình . Testcase

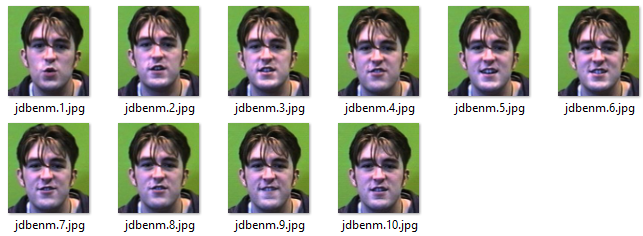
Qui trình xác thực cũng giống với Fisherface, tiến hành tiền xử lý, chuẩn hóa vector, chiếu vào không gian Fisherface để thu được vector đặc trưng.

Tiến hành tính toán và so sánh khoảng cách giữa vector đặc trưng của hình ảnh cần xác thực lần lượt với các vector đặc trưng có được trong giai đoạn đăng ký, nếu khoảng cách này thỏa điều kiện dưới một ngưỡng (threshold) nào đó, ta sẽ chấp nhận xác thực. Nếu so sánh hết với các vector đặc trưng đã lưu mà vẫn không thỏa điều kiện này, thì việc xác thực bị từ chối.

### Local binary pattern

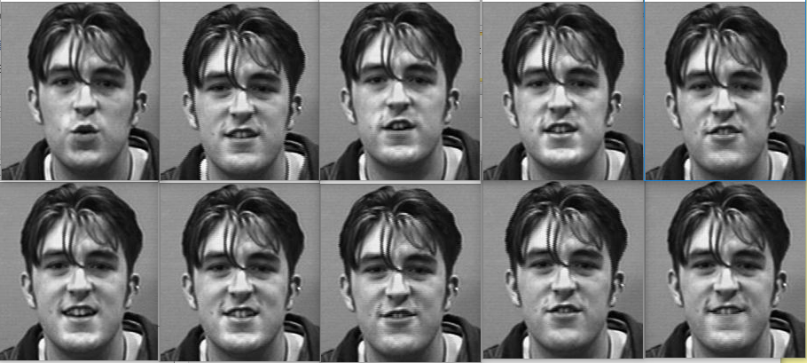
#### Huấn luyện

Tập hình ảnh dùng để huấn luyện nên có thể xét ở nhiều biến đổi và độ sáng tối khác nhau. Ví dụ một tập ảnh dùng để huấn luyện :



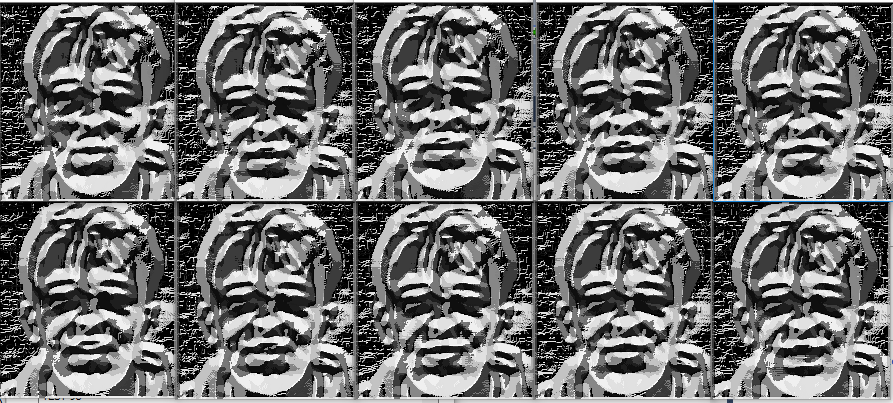
Hình . Tập huấn luyện của LBP

Trong phương pháp này, lần đầu tiên, một hình ảnh khuôn mặt được chuyển thành hình ảnh xám:



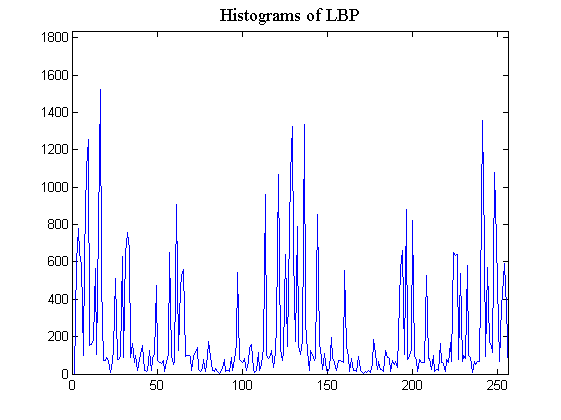
Hình . Ảnh xám của tập huấn luyện

Sau đó mỗi ảnh xám sẽ được tính toán giá trị và chuyền thành ảnh LBP. Mỗi giá trị pixel được tính bằng giá trị lân cận của mình. Nếu các điểm ảnh trung tâm là lớn hơn hoặc bằng giá trị lân cận nó, sau đó biểu thị nó bằng 1 và 0 nếu ngược lại. Các pixel xung quanh mang lại một số nhị phân cho một pixel trung tâm đó là giá trị xám của pixel trung tâm.



Hình . Ảnh LBP thu được sau khi tính toán

Biểu đồ histogram được sử dụng để lưu trữ thông tin cục bộ trong hình ảnh, ở gian đoạn này nhóm sử dụng toàn bộ ảnh LBP để tạo thành 1 biểu đồ histogram. Mỗi phần tử trong biểu đồ histogram có giá trị là số lần mức xám xuất hiện trong bức ảnh lbp. Như vậy, ta sẽ tạo một vector đặc trưng histogram với kích thước 1x256. Ta sẽ lặp qua từng pixel của ảnh và đếm số lượng các pixel có mức xám từ 0-255 để lưu vào vector tương ứng trên.



Hình . Biểu đồ minh họa Histogram

Vector đặc trưng được trình bày bởi một vector N chiều, trong trường hợp này N = 256 (256 màu xám), vector đặc trưng thu được bằng cách rút trích từ 1 biểu đồ histogram. Ta sẽ lặp qua từng giá trị của biểu đồ histogram của ảnh và lưu giá trị có mức xám từ 0-255 của biểu đồ vào vị trí tương ứng của vector.

#### Đăng ký

Bước đăng ký của LBP cũng tương tự như lúc huấn luyện. Người dùng sẽ cung cấp một hoặc một số hình ảnh khuôn mặt của mình cho hệ thống, từ đó hệ thống sẽ tiến hành tạo ảnh LBP và rút trích đặc trưng khuôn mặt từ đó và lưu lại trong cơ sỡ dữ liệu.

Kết quả sau khi tạo histogram chính là vector histogram đặc trưng của người dùng. Các vector này được lưu lại và được sử dụng để đối chiếu so sánh với mẫu sinh trắc đầu vào của hệ thống khi xác thực.

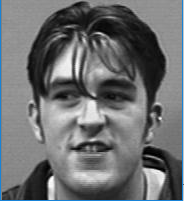
#### Xác thực

Để xác thực ta chọn hình ảnh cần xác thực :



Hình . Ảnh cần xác thực

Sau đó ta chuyển thành ảnh xám để thực hiện quá trình tính toán ra ảnh LBP :



Hình . Ảnh xác thực đã chuyển qua ảnh xám

Từ ảnh LBP ta thực hiện phép tính LBP để tạo ảnh LBP cho các pixel :



Hình . Ảnh xác thực LBP

Từ ảnh LBP ta sẽ có biểu đồ histogram và suy ra được vector đặc trưng của ảnh cần xác thực. Sau đó ta sử dụng vector đặc trưng của ảnh đầu vào và vector đặc trưng của tập huấn luyện để tính khoảng cách theo Chi-Squared[9]. Nếu khoảng cách Chi-Square nhỏ hơn ngưỡng thì xác thực là đúng. Để xác định ngưỡng, ta sẽ xem xét với các ngưỡng khác nhau thì tỉ lệ chấp nhận sai (FAR) và tỉ lệ từ chối sai (FRR) như thế nào, sau đó vẽ đồ thị của FAR và FRR để tìm ra ngưỡng phù hợp.

### Secure sketch-Fuzzy Extractor

#### Huấn luyện và đăng ký

Sau khi rút trích được vector đặc trưng từ các phương pháp PCA, Fisherface, LBP ta sẽ sử dụng Secure Sketch-Fuzzy Extractor để bảo mật mẫu và xác thực.

Từ mỗi vector đặc trưng ban đầu ta tiến hành hash từng phần tử theo mô hình HMAC và lưu vào cơ sở dữ liệu.

Từ đặc trưng sinh trắc ta tạo ra một codebook với nhiều codeword, các codeword được phân bố đều trong miền giá trị của sinh trắc. Khoảng cách giữa các codeword kề nhau là bằng nhau và bằng 2, với giá trị truyền vào. Sau khi tạo được codebook cho mỗi vector đặc trưng, sketch được tính theo công thức (1) và lưu vào cơ sở dữ liệu.

#### Xác thực

Vector sau khi rút trích từ ảnh cần xác thực sẽ được đưa vào phân tích để tái tạo ra các vector dựa vào sketch tương ứng trong cơ sở dữ liệu. Các vector được tái tạo sẽ được hash lại theo mô hình HMAC và so sánh với từng vector hash được lưu trong cơ sở dữ liệu. Nếu hai vector giống nhau thì xác thực thành công và ngược lại.

## Nền tảng

Android [[10]](#ref5) là một hệ điều hành dựa trên nền tảng Linux được thiết kế dành cho các thiết bị di động có màn hnh cảm ứng như điện thoại thông minh và máy tính bảng. Ban đầu, Android được phát triển bởi Tổng công ty Android, với sự hỗ trợ tài chính từ Google và sau này được chính Google mua lại vào năm 2005. Android ra mắt vào năm 2007 cùng với tuyên bố thành lập Liên minh thiết bị cầm tay mở: một hiệp hội gồm các công ty phần cứng, phần mềm, và viễn thông với mục tiêu đẩy mạnh các tiêu chuẩn mở cho các thiết bị di động.

Android có mã nguồn mở và Google phát hành mã nguồn theo Giấy phép Apache. Tính đến tháng 7 năm 2013, đã có hơn 1000000 ứng dụng được phát hành trên Google Play, và số lượt tải ứng dụng từ Google Play, cửa hàng ứng dụng chính của Android, ước tính khoảng 50 tỷ lượt.

Những yếu tố này đã giúp Android trở thành nền tảng điện thoại thông minh phổ biến nhất thế giới, và được các công ty công nghệ lựa chọn khi họ cần một hệ điều hành không nặng nề, có khả năng tinh chỉnh, và giá rẻ chạy trên các thiết bị công nghệ cao thay vì tạo dựng từ đầu. Kết quả là mặc dù được thiết kế để chạy trên điện thoại và máy tính bảng, Android đã xuất hiện trên TV, máy chơi game và các thiết bị điện tử khác.

Android chiếm 75% thị phần điện thoại thông minh trên toàn thế giới vào thời điểm quý 3 năm 2012, với tổng cộng 500 triệu thiết bị đã được kích hoạt và 1,3 triệu lượt kích hoạt mỗi ngày. Theo số liệu quý 2 năm 2014 của Strategy Analytics, thị phần của Android chiếm đến 85% tổng thị phần của nền tảng di động, áp đảo hoàn toàn so với các nền tảng nổi tiếng khác như iOS, WP ...

Một cuộc khảo sát về nhà phát triển ứng dụng di động vào 3-4/2013 cho thấy 71% nhà phát triển tạo ra ứng dụng Android và cuộc khảo sát năm 2015 cho thấy 40% nhà phát triển chuyên nghiệp lấy Android làm nền tảng mục tiêu trong khi với nền tảng iOS con số đó là 37%.

Android là một nền tảng dựa trên tập hợp các chương trình khác như một nhân Linux, các thư viện, API được viết bằng C và các ứng dụng gốc. Hình dưới đây sẽ cho chúng ta thấy rõ được kiến trúc của Android :



Hình . Kiến trúc của Android

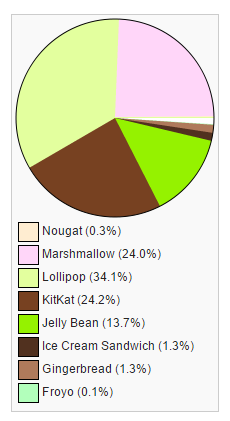
* Linux Kernel : một phần cực kì quan trọng trong kiến trúc, giúp các ứng dụng Android có thể tương tác với các thiết bị phần cứng.
* Libraries : bao gồm tập các thư viện (viết bằng C/C++) hỗ trợ cho các ứng dụng thực hiện các chức năng mở rộng.
* Android Runtime : đảm nhận việc biên dịch và thực thi các file ứng dụng.
  + Android Libraries : tập các thư viện cơ bản nhất của Android .
  + Dalvik Vitual Machine : bộ máy ảo của android, là nơi thực hiện các ứng dụng mà người dùng sử dụng trên thiết bị Android.
* Application Framework : tầng cung cấp cho ứng dụng những dịch vụ cấp cao hơn ở trạng thái class như dịch vụ tin nhắn, dịch vụ truy cập internet….
* Application Layer : tầng các ứng dụng để tương tác với người sử dụng, có thể là do các lập trình viên phát triển, ứng dụng gốc hoặc là ứng dụng của bên thứ 3.

Với kiến trúc trên, chúng ta có thể thấy Android có một kiến trúc hoạt động khá linh động và mạnh mẽ. Với việc sử dụng cơ chế máy ảo Dalvik VM sẽ giúp cho các ứng dụng của Android có thể chạy và hoạt động trên các thiết bị khác nhau có cùng nền tảng.

Dưới đây là bảng mô tả các phiên bản Android hiện có và sự phân bố người dùng của chúng tính tới tháng 7/2016

| Phiên bản | Tên phiên bản | Ngày phát hành | API level | Phân bố |
| --- | --- | --- | --- | --- |
| 7.1 | Nougat | 4/10/2016 | 25 | Nhỏ hơn 0.1% |
| 7.0 | Nougat | 22/8/2016 | 24 | 0.3% |
| 6.0 | Marshmallow | 5/10/2015 | 23 | 24.0% |
| 5.1 | Lollipop | 9/3/2015 | 22 | 22.8% |
| 5.0 | 3/11/2014 | 21 | 11.3% |
| 4.4 | KitKat | 31/10/2013 | 19 | 25.2% |
| 4.3 | Jelly Bean | 24/7/2013 | 18 | 2.0% |
| 4.2 | 13/11/2012 | 17 | 6.8% |
| 4.1 | 9/7/2012 | 16 | 4.9% |
| 4.0 | Ice Cream Sandwich | 16/12/2011 | 15 | 1.3% |
| 2.3.3+ | Gingerbread | 9/2/2011 | 10 | 1.3% |
| 2.2 | Froyo | 20/5/2010 | 8 | 0.1% |

Tính tới tháng 11/2016, phiên bản mới nhất của Android là 7.1 với tên Nougat. Tuy nhiên phiên bản đang được nhiều người sử dụng nhất lại là phiên bản Android Lollipop (34,1%). Phiên bản này, có nhiều chức năng mới, giao diện đẹp đẽ, sử dụng ổn định nên nhóm quyết định chọn phiên bản Lollipop để phát triển hệ thống.



Hình . Sơ đồ miêu tả sự phổ biến của các phiên bản android

## Các thư viện sử dụng

### JavaCV

JavaCV[[11]](#ref6) là một thư viện bao phủ (wrapper) các thư viện được sử dụng phổ biến bởi các nhà nghiên cứu trong lĩnh vực thị giác máy tính như OpenCV, FFmpeg, libdc1394, PGR FlyCapture, OpenKinect, CL PS3 Eye Driver, videoInput, ARToolKitPlus, flandmark…và cung cấp các class hữu ích cho việc sử dùng các chức năng của chúng dễ dàng hơn trên nền tảng Java, bao gồm cả Android.

JavaCV hỗ trợ đầy đủ các giải thuật phổ biến mà OpenCV [[12]](#ref7) hỗ trợ. Phiên bản mới nhất của JavaCV là 1.2, với sự ổn định và bổ sung nhiều cải tiến. Đây cũng là phiên bản mà nhóm quyết định sử dụng trong thực hiện đề tài này.

### Google Mobile Vison API

Mobile Vision API [[13]](#ref8) cung cấp một framework giúp ích cho việc tìm kiếm các đối tượng trong hình ảnh hoặc video. Framework bao gồm các detector dùng để định vị và mô tả các đối tượng thuộc về thị giác trong từng khung ảnh và video.

Hiện tại, Mobile Vison API bao gồm các chức năng liên quan đến khuôn mặt, barcode và phát hiện chữ. Chúng ta có thể sử dụng chúng riêng biệt hoặc kết hợp với nhau.

Trong đề tài này, nhóm chỉ sử dụng khối chức năng về khuôn mặt. Với sự trợ giúp của Mobile Vison API, việc phát hiện khuôn mặt được tiến hành dễ dàng và chính xác hơn, đồng thời giúp ta có thể định vị vị trí các đặc điểm cấu thành khuôn mặt.

Mobile Vision API được tích hợp trong các phiên bản Google Play Services 7.8 trở lên.

### Sweet Alert Dialog

Là thư viện hỗ trợ trong việc hiện các bảng thông báo với cung hiệu ứng. Dung lượng nhỏ nhẹ, hiệu ứng mượt mà, nhanh chóng, đẹp mắt. Có nhiều tùy chỉnh giúp người dùng thay đổi các thông báo theo ý muốn của mình [[14]](#ref9).

Phiên bản mời nhất của Sweet Alert Dialog là 1.3.

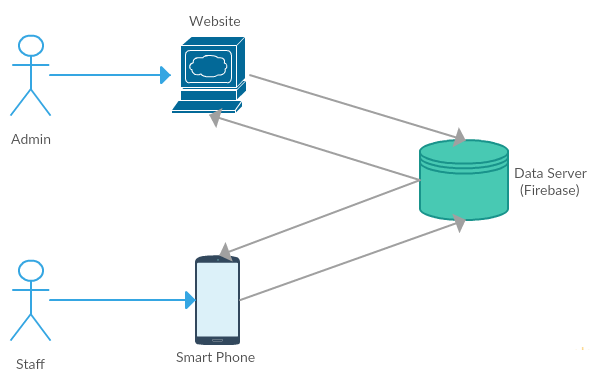
# HỆ THỐNG ĐỀ XUẤT

## Mô tả hệ thống

Hệ thống quản lý chi phí đi lại của nhân viên bán hàng trong công ty bao gồm một ứng dụng di động (trên nền tảng Android) dành cho nhân viên, để thu thập dữ liệu về vị trí, lộ trình, chi phí đi lại của nhân viên và một Website (trên nền tảng Web) hỗ trợ những người quản trị hệ thống, để quản lý dữ liệu của công ty (dữ liệu về thông tin nhân viên và vị trí, lộ trình, chi phí đi lại của nhân viên).

Ứng dụng di động được nhân viên cài đặt trên Smartphone cá nhân (hoặc Smartphone do công ty cấp). Nhân viên mỗi khi đi làm sẽ thực hiện đăng nhập vào hệ thống, lựa chọn phương tiện di chuyển và bắt đầu di chuyển. Hệ thống định vị tự động vị trí của nhân viên, vẽ lại lộ trình di chuyển của nhân viên, thu thập dữ liệu về quãng đường đi được và chi phí đi lại của nhân viên. Cứ sau 3 giây, hệ thống sẽ truy vấn vị trí của nhân viên một lần, nếu vị trí của nhân viên thay đổi so với vị trí lần trước (3 mét so với vị trí trước đó), hệ thống mới cập nhật vị trí cho nhân viên. Trước khi vị trí được lưu trên Data Server, dữ liệu vị trí sẽ được xử lý thông qua Web service Google Maps Snap to Roads (cứ nhận được 5 vị trí thì xử lý 1 lần) để giảm thiểu sai số do GPS gây ra. Mỗi khi nhân viên thay đổi phương tiện di chuyển, nhân viên sẽ thực hiện thay đổi phương tiện trên ứng dụng. Đối với các phương tiện công cộng có sử dụng vé cho mỗi chuyến đi, nhân viên cần nhập giá vé và chụp hình vé đối chứng. Nhân viên có thể xem thống kê về chi phí đi lại và quãng đường đi lại của bản thân trong khoảng thời gian mong muốn.

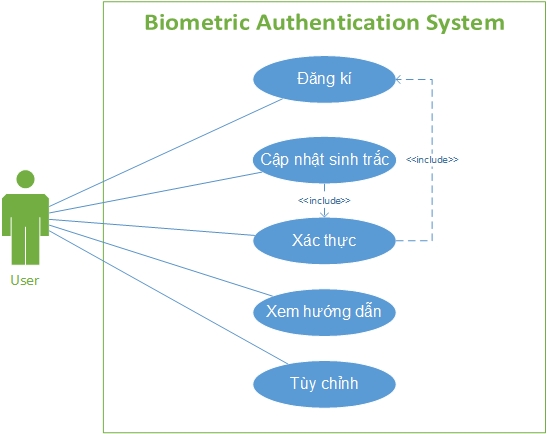
Website được những người quản trị hệ thống sử dụng để quản lý dữ liệu của công ty. Người quản trị sau khi đăng nhập có thể xem được danh sách toàn bộ nhân viên trong công ty và thống kê cá nhân của từng người (chi phí đi lại và quãng đường đi lại). Khi có nhân viên mới vào, người quản trị cần tạo tài khoản người dùng cho nhân viên đó dựa trên thông tin cá nhân của nhân viên cung cấp. Khi nhân viên nghỉ việc, người quản trị sẽ xóa dữ liệu người đó ra khỏi hệ thống (tài khoản đăng nhập, thông tin người dùng và các dữ liệu về vị trí, lộ trình, chi phí đi lại liên quan). Người quản trị có thể xem thống kê về tổng quãng đường và tổng chi phí đi lại của tất cả nhân viên công ty trong khoảng thời gian mong muốn (thống kê công ty). Tùy thuộc vào chính sách hỗ trợ của công ty, người quản trị sẽ nhập công thức tính toán chi phí cho từng phương tiện (là hàm số theo quãng đường đi được s và thời gian đi được t). Đây là những công thức được áp dụng trên ứng dụng di động của nhân viên để tính toán chi phí đi lại của nhân viên.



Hình . Tổng quan hệ thống

Hình . Chi tiết hệ thống

Dưới đây là mô hình use case tổng quát của hệ thống:



Hình . Mô hình usecase của hệ thống

Hệ thống có một actor là người dùng (User). Mỗi người dùng có thể tương tác với hệ thống với các chức năng sau:

* Chức năng đăng ký: người dùng cung cấp mẫu sinh trắc khuôn mặt của mình để đăng ký với hệ thống.
* Chức năng cập nhật sinh trắc: hay nói cách khác đây là chức năng đăng ký lại, sử dụng khi người dùng muốn thay đổi mẫu sinh trắc đã đăng ký.
* Chức năng xác thực: để truy cập và sử dụng các tài nguyên trong hệ thống, người dùng sẽ phải được cấp phép bằng cách xác thực chính mình là người đã đăng ký với hệ thống.
* Chức năng xem hướng dẫn: người dùng xem những hướng dẫn cơ bản, giải thích về hệ thống.
* Chức năng tùy chỉnh: người dùng cài đặt các thuật toán nào dùng để xác thực hay các ngưỡng xác thực…

### Use case Scenario

Các use case của hệ thống:

* Use case đăng ký:

|  |  |
| --- | --- |
| Use Case | Đăng kí |
| Actor | Người dùng |
| Brief Description | Đăng kí cho người dùng khi lần đầu sử dụng |
| Main Flow | - Người sử dụng ấn nút đăng kí để bật camera.  - Người dùng chụp 10 tấm hình khuôn mặt để đăng ký với hệ thống.  -Sau khi chụp đủ, người dùng chọn “DONE” để kết thúc đăng ký và chuyển về activity chính của app. |
| Alternative Flows | |  | | --- | | - Không. | |
| Pre-condition | |  | | --- | | - Lần đầu tiên sử dụng app. | |
| Special Requirements | |  | | --- | | - Hình ảnh chụp phải đúng là khuôn mặt. | |

* Use case cập nhật sinh trắc:

|  |  |
| --- | --- |
| Use Case | Cập nhật sinh trắc |
| Actor | Người dùng |
| Brief Description | Đăng kí lại sinh trắc học khi người dùng muốn thay đổi |
| Main Flow | - Khi chạy ứng dụng activity Chính sẽ được mở ra.  - Người dùng chọn cập nhật sinh trắc để đăng kí lại với hệ thống mẫu sinh trắc khác.  - Hệ thống sẽ chuyển qua camera để kiểm tra sinh trắc học cũ, nếu đúng là người dùng thì mới cho đăng kí mới. Nếu không sẽ hiện thông báo và yêu cầu kiểm tra lại.  - Khi xác thực đúng người dùng sẽ hiện thông báo và tiếp tục camera đăng kí mới.  - Sau khi đăng kí mới thành công, chuyển về activity chính của app. |
| Alternative Flows | |  | | --- | | - Không. | |
| Pre-condition | |  | | --- | | - Đã đăng kí trước đó rồi. | |
| Special Requirements | - Chức năng xác thực chính chủ phải chính xác |

* Use case xác thực:

|  |  |
| --- | --- |
| Use Case | Xác thực |
| Actor | Người dùng |
| Brief Description | Xác thực cho phép sử người dùng sử dụng điện thoại |
| Main Flow | |  | | --- | | - Sau khi mở màn hình khóa, activity xác thực sẽ mở lên**.**  - Camera sẽ mở và người dùng đưa khuôn mặt mình để quét.  - Ứng dụng chạy thuật toán, đối chiếu và cho kết quả.  - Nếu kết quả thành công, activity sẽ kết thúc và người dùng có thể sử dụng điện thoại. | |
| Alternative Flows | |  | | --- | | - Kết quả thất bại, thông báo cho người dùng biết kết quả.  - Người dùng nhập lại khuôn mặt.  - Ứng dụng chạy thuật toán, đối chiếu và cho kết quả. | |
| Pre-condition | |  | | --- | | - Người dùng phải đăng kí mẫu sinh trắc.  - Người dùng phải cài đặt kích hoạt chế độ màn hình khóa trong ứng  dụng.  - Người dùng phải mở màn hình chờ của điện thoại. | |
| Special Requirements | |  | | --- | | - Không | |

* Use case xem hướng dẫn:

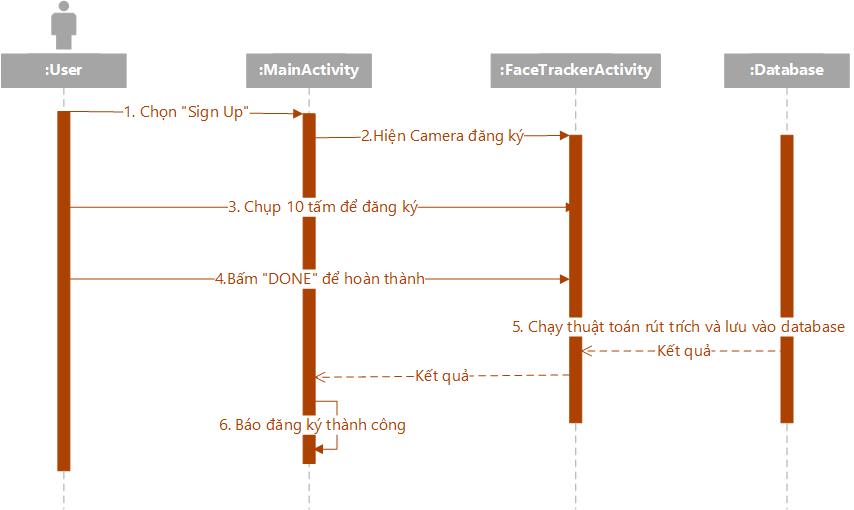
|  |  |
| --- | --- |
| Use Case | Xem hướng dẫn |
| Actor | Người dùng |
| Brief Description | Người dùng muốn xem cách sử dụng ứng dụng |
| Main Flow | |  | | --- | | - Khi chạy ứng dụng lần đầu sau khi cài đặt, activity Hướng dẫn sẽ được mở ra**.**  - Người dùng đọc hướng dẫn ở trang đầu (có thể bấm skip để bỏ qua  hướng dẫn) - Người dùng bấm Next để đọc các hướng dẫn ở trang kế, hoặc Back để  đọc lại trang trước đó.  - Khi đọc tới trang cuối cùng, bấm “GOT IT” để kết thúc hướng dẫn.  - Nếu người dùng muốn xem lại hướng dẫn, bấm vào biểu tượng dấu ba chấm bên góc phải, sau đó chọn Help, khi đó activity Hướng dẫn sẽ mở  ra. | |
| Alternative Flows | |  | | --- | | - Không | |
| Pre-condition | |  | | --- | | - Không | |
| Special Requirements | |  | | --- | | - Không | |

* Use case tùy chỉnh:

|  |  |
| --- | --- |
| Use Case | Tùy chỉnh |
| Actor | Người dùng |
| Brief Description | Người dùng thiết lập một số tùy chỉnh cho ứng dụng như chọn thuật toán, ngưỡng xác thực… |
| Main Flow | |  | | --- | | - Người dùng mở chương trình, chọn cài đặt (nút ba chấm ở góc phải  màn hình).  - Người dùng chọn các thuộc tính và thay đổi theo ý muốn. | |
| Alternative Flows | |  | | --- | | - Không. | |
| Pre-condition | |  | | --- | | - Người dùng phải đăng kí mẫu sinh trắc nếu chọn kích hoạt màn hình  khóa. | |
| Special Requirements | |  | | --- | | - Không | |

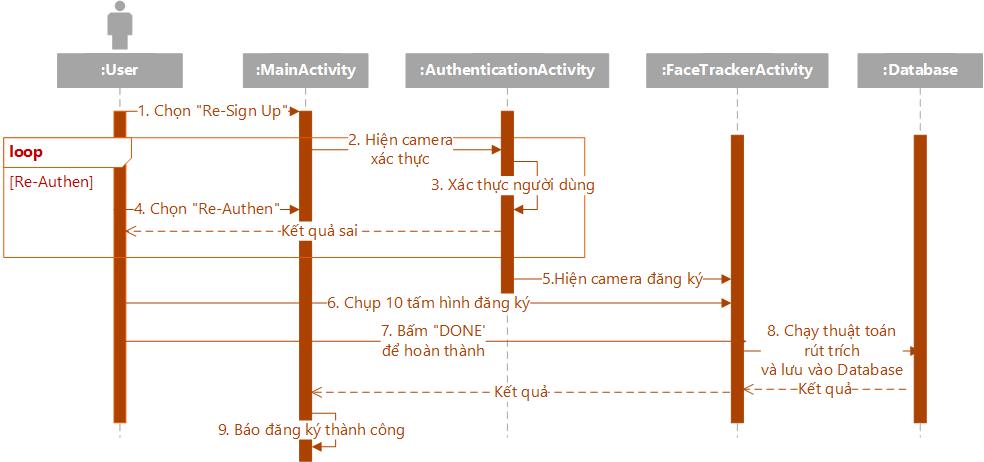
### Sơ đồ sequence

* Đăng ký:



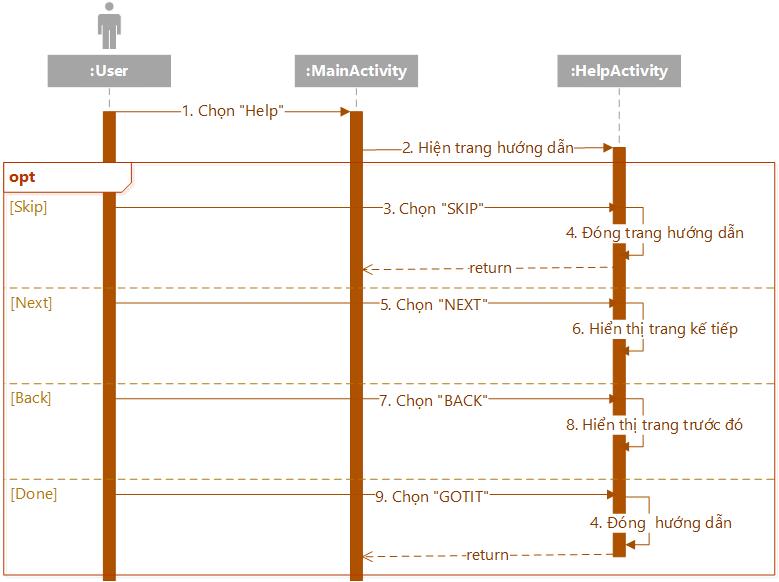
Hình . Sequence đăng ký

* Cập nhật sinh trắc



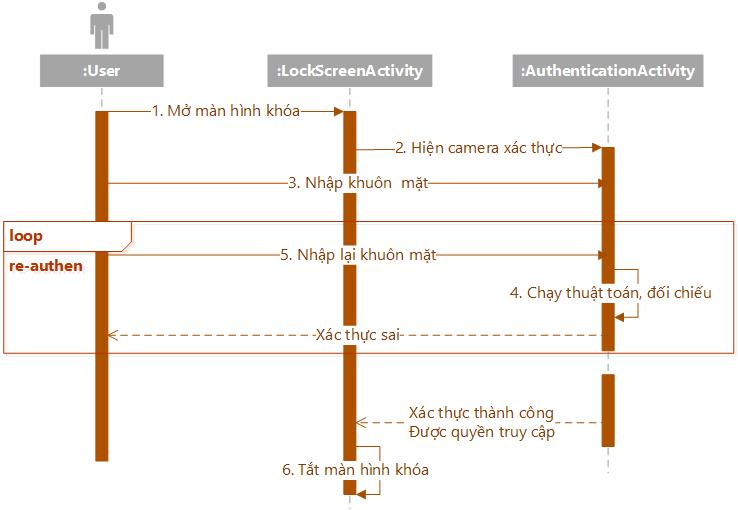
Hình . Sequence cập nhật sinh trắc

* Xem hướng dẫn



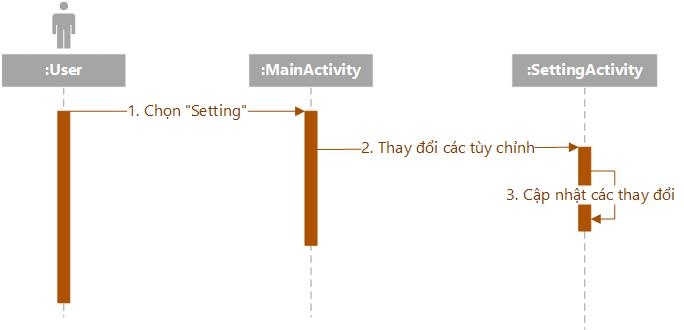
Hình . Sequence xem hướng dẫn

* Xác thực



Hình . Sequence xác thực

* Tùy chỉnh:



Hình . Sequence tùy chỉnh

## Các chức năng chính

Cấu trúc các thành phần của hệ thống bao gồm ba thành phần chính: Rút trích đặc trưng, đăng ký và xác thực.

* **Rút trích đặc trưng**: Là trọng tâm của đề tài. Sử dụng ba giải thuật khác nhau để rút trích các đặc trưng, dùng cho các bước đăng ký và xác thực sau này.
* **Đăng ký**: Sử dụng các đặc trưng được rút trích để đăng ký người dùng vào hệ thống. Bước này cũng bao gồm băm các mẫu sinh trắc và lưu lại các sketch.
* **Xác thực**: Lấy mẫu sinh trắc mới và tiến hành kiểm tra với dữ liệu đã được lưu lại để xác thực người dùng.

C:\Users\wicky\Downloads\Untitled Diagram.pngKhi sử dụng chương trình lần đầu tiên, người dùng sẽ phải đăng ký sinh trắc học (khuôn mặt) của mình với chương trình. Chương trình sẽ gọi lần lượt cả ba giải thuật để thực hiện các bước huấn luyện, rút trích đặc trưng, chiếu đặc trưng vào không gian huấn luyện. Sau đó, sử dụng các giải thuật bảo mật để sinh ra sketch từ tập mẫu của người dùng và lưu lại sketch để phục vụ cho giai đoạn xác thực. Ngoài ra, cũng cần phải lưu giá trị băm của tập mẫu để kiểm tra tính đúng đắn của mẫu được khôi phục sau này.

Hình . Quá trình đăng ký dùng Secure Sketch

Ở bước xác thực, quá trình diễn ra khá tương đồng với lúc đăng ký. Người dùng cũng sẽ được chụp lại khuôn mặt của mình, và giải thuật rút trích sẽ được hiện thực, và chiếu vào không gian huấn luyện tương tự ban đầu. Tiếp đến, chương trình sẽ đọc lại tập sketch đã được lưu ở bước đăng ký, và dùng tập sketch để khôi phục lại mẫu ban đầu dựa trên mẫu xác thực mới này. Vì mẫu ban đầu đã được áp dụng hàm băm, nên để so sánh, ta cũng áp dụng hàm băm lên tập mẫu khôi phục. Từ đó, so sánh sự tương đồng giữa hai mẫu và đưa ra kết luận.

C:\Users\wicky\Downloads\Untitled Diagram (1).png

Hình . Quá trình xác thực dùng Secure Sketch

Quy trình này đơn giản và thân thiện với người dùng, họ chỉ cần chụp lại khuôn mặt, tất cả những quá trình khác đều được chương trình thực hiện ở dưới. Hơn nữa, tính bảo mật của đặc trưng sinh trắc cũng được bảo đảm vì đã được áp dụng hàm hash. Nếu có bị tấn công, đặc trưng sinh trắc cũng sẽ không bị lộ, và ta có thể dễ dàng thay đổi hàm hash để sinh ra chuỗi giá trị khác.

# HIỆN THỰC //TODO

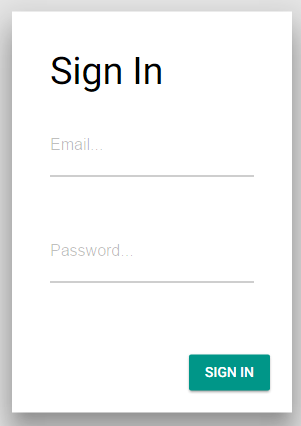
## Website

## Workflow

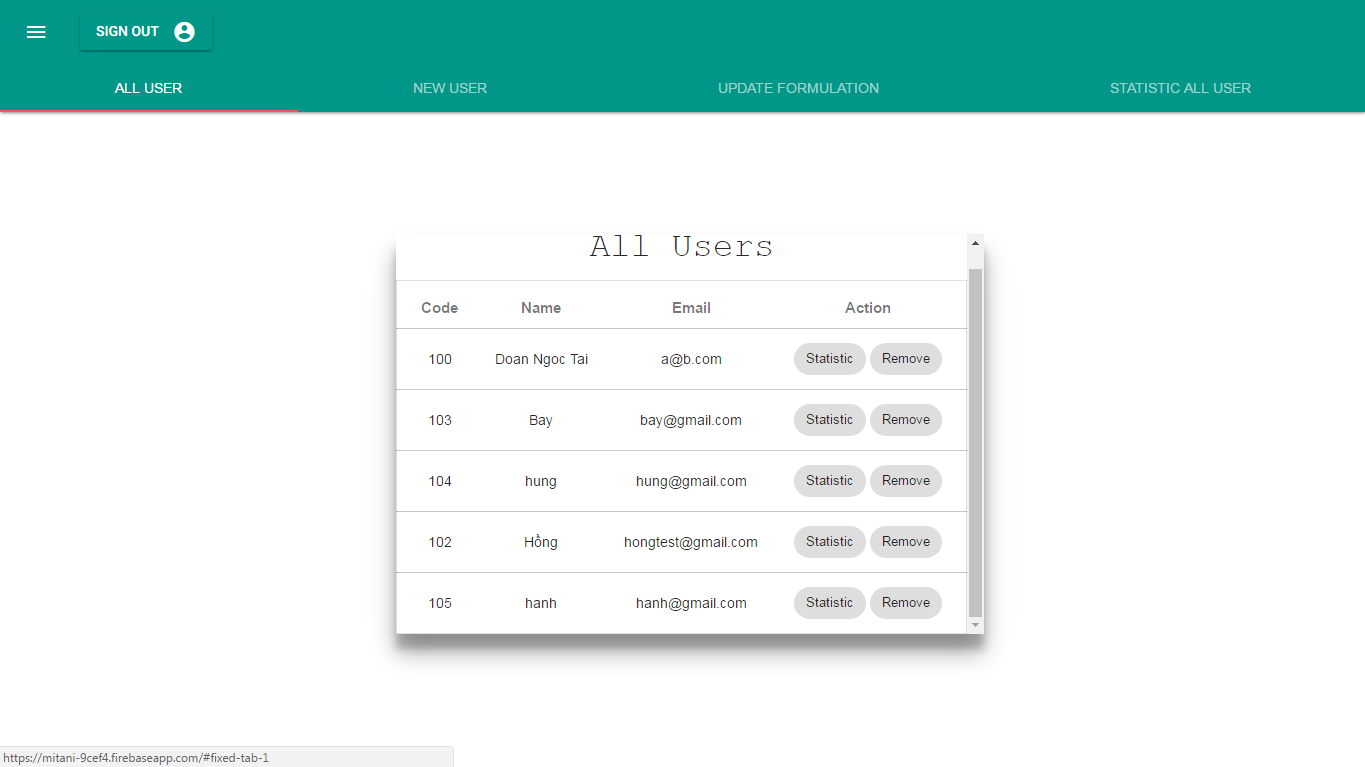
//Todo

## Giao diện Website

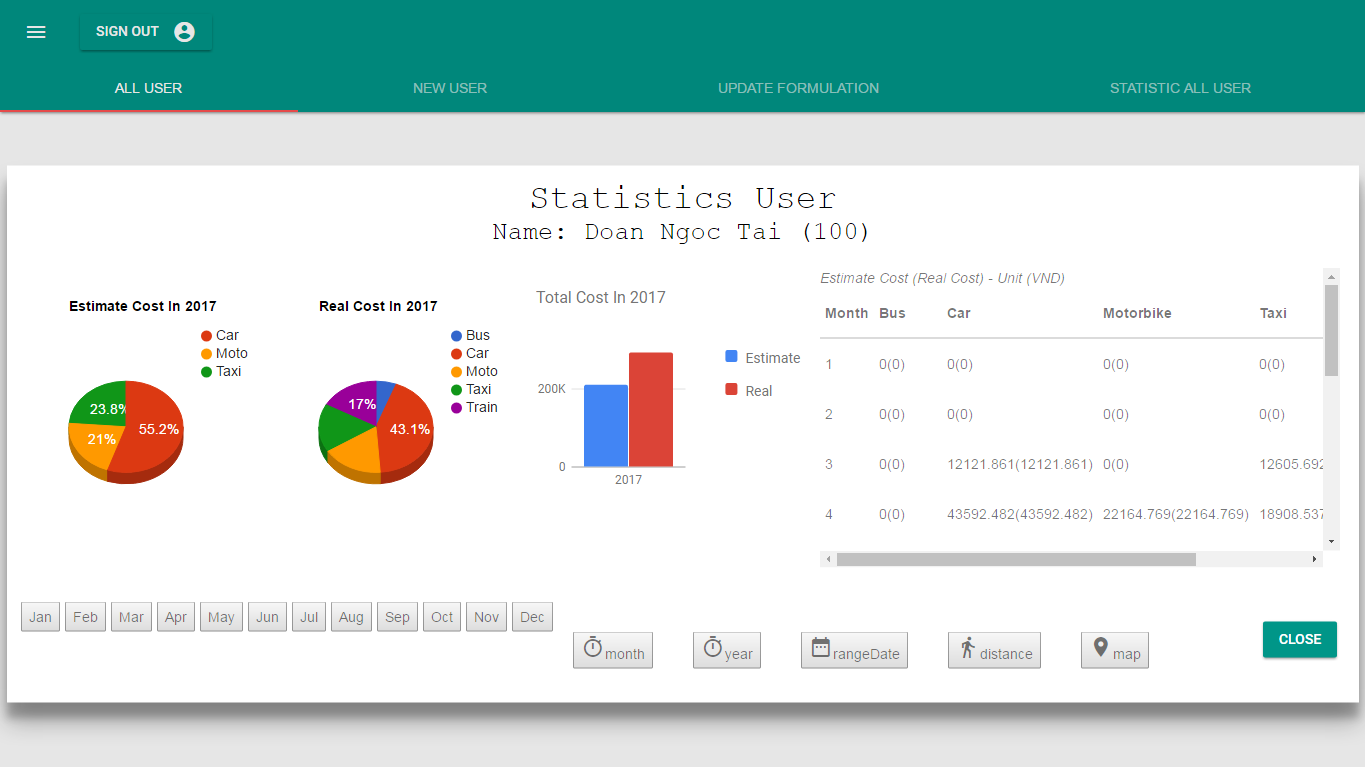
* + 1. **Đăng nhập**



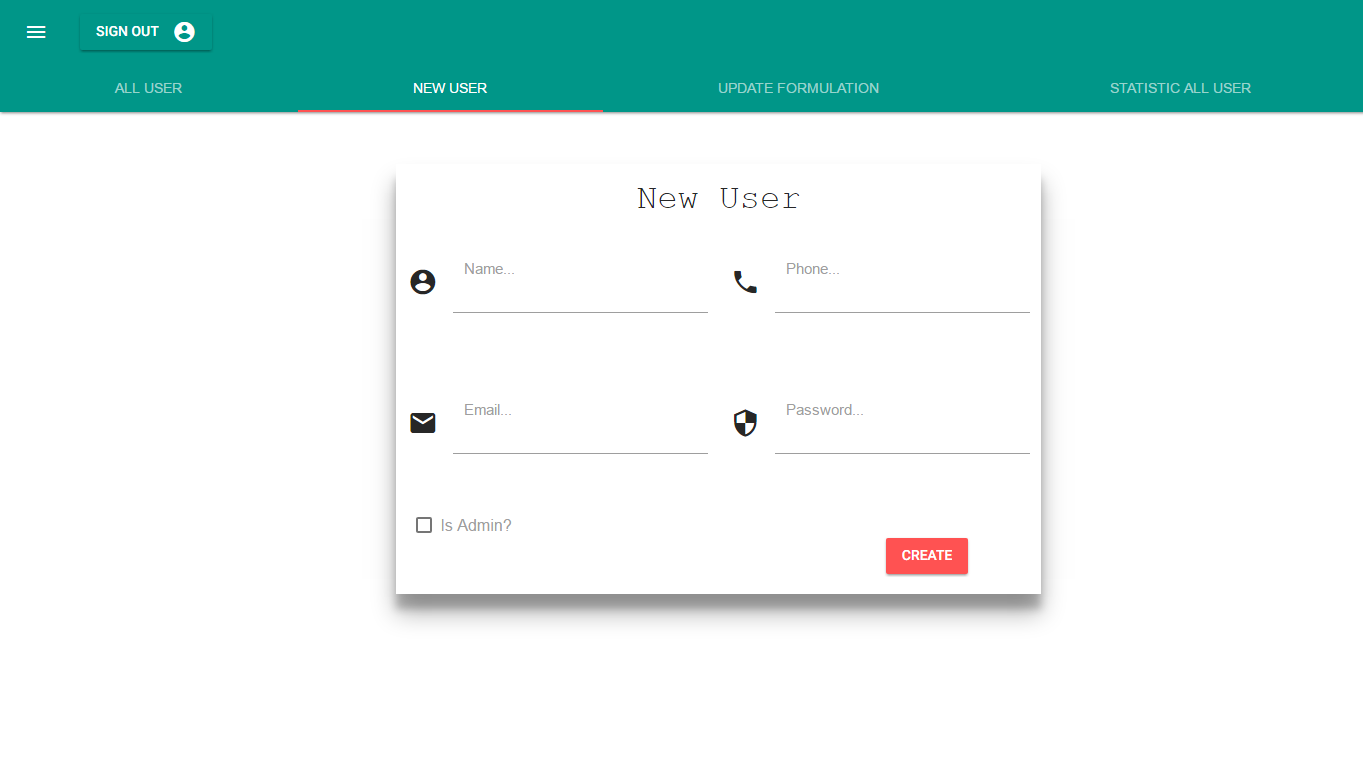
* + 1. **Trang chủ**



* + 1. Trang xem thống kê cá nhân



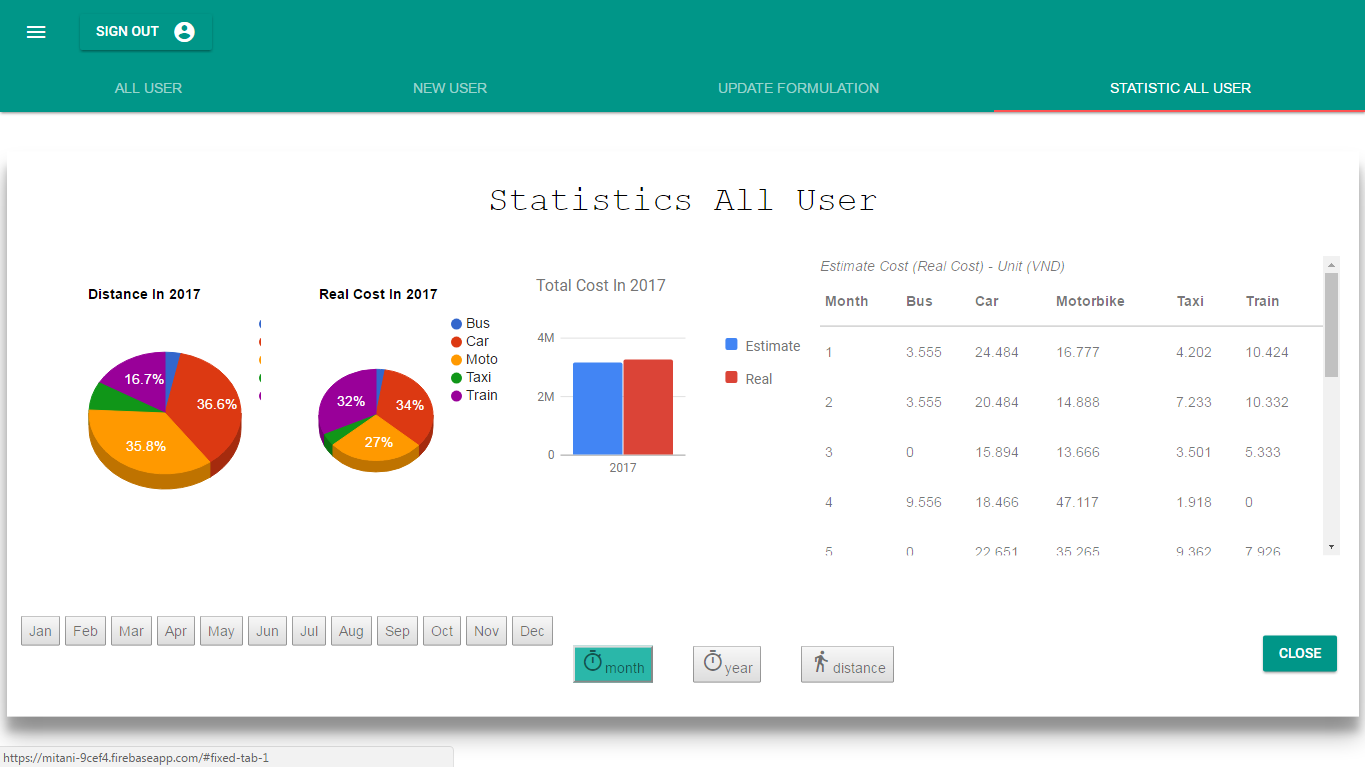
* + 1. Trang tạo user mới



* + 1. Trang cập nhật công thức



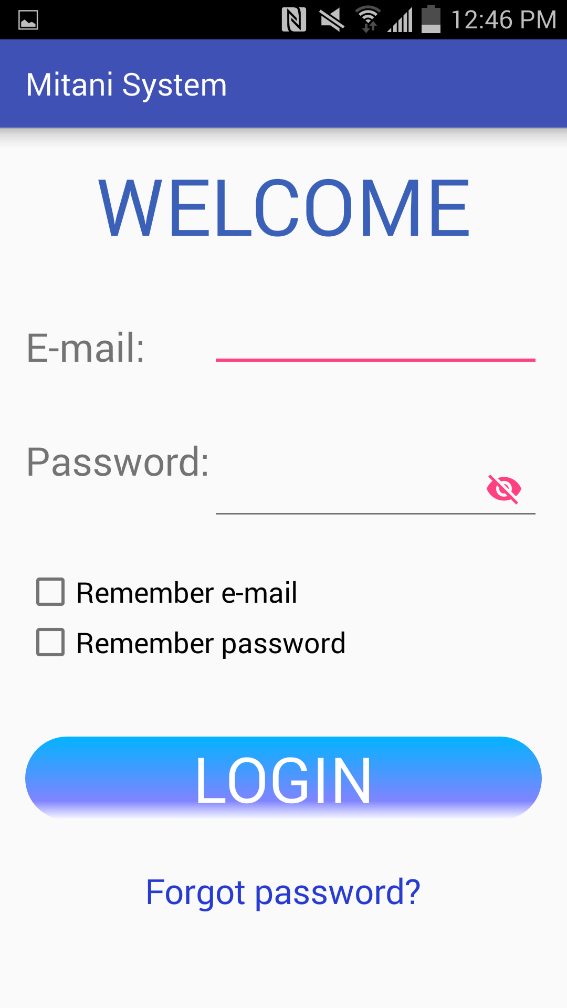
* + 1. Trang thống kê công ty



1. Ứng dụng di động
   1. Workflow

//todo

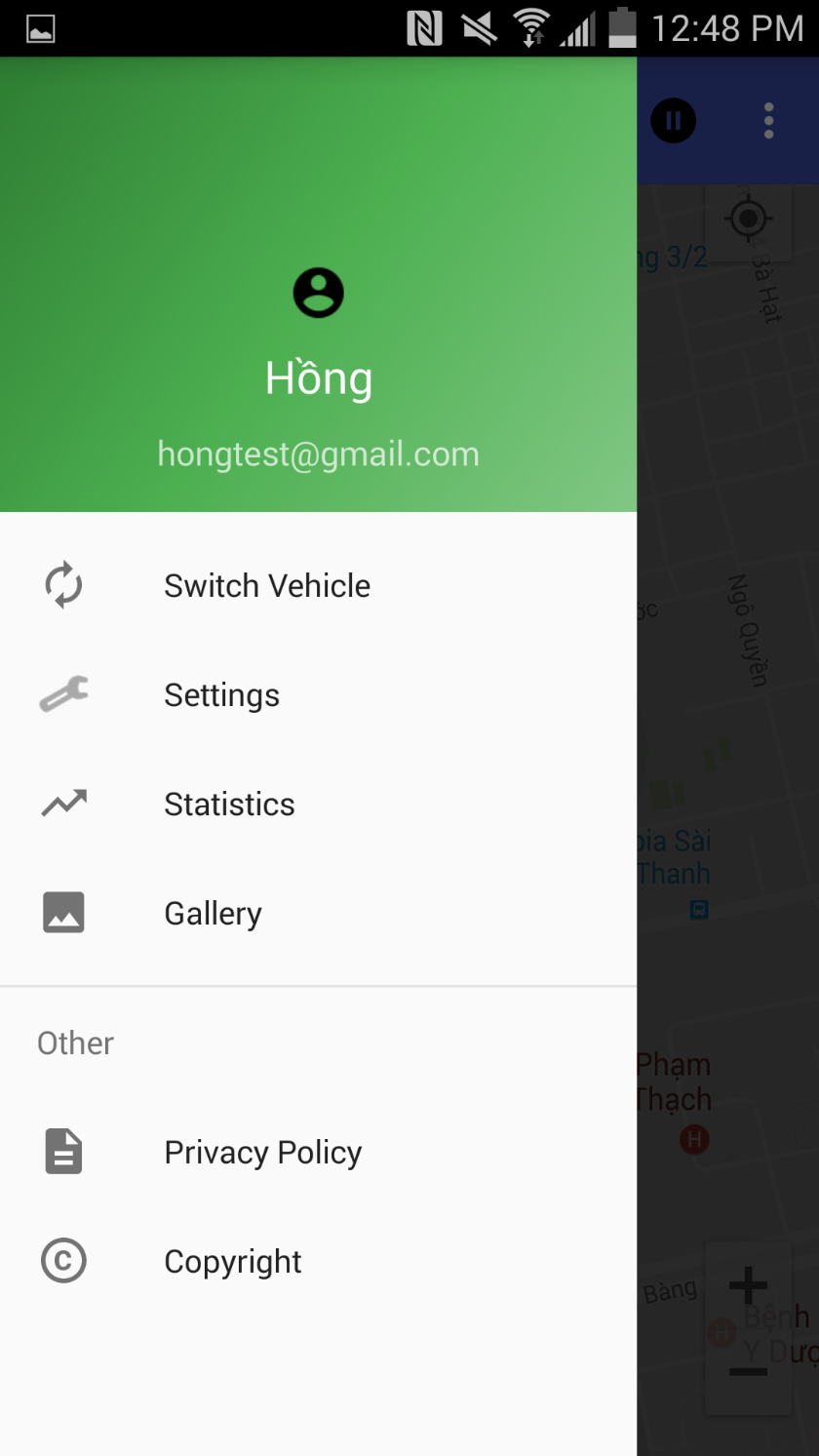
* 1. Giao diện ứng dụng di động
     1. Màn hình đăng nhập



* + 1. Màn hình



* + 1. Màn hình



Hệ thống có thể được chia thành các khối chức năng chính sau:

|  |  |
| --- | --- |
| Khối chức năng | Nhiệm vụ |
| UI | Gồm các Activity hiện thực giao diện để người dùng tương tác với phần mềm. |
| Camera | Hiện thực các hàm sử dụng camera, xác định khuôn mặt, tiền xử lý hình ảnh… |
| Feature Extraction | Hiện thực ba giải thuật rút trích đặc trưng Eigenface, Fisherface và Local Binary Pattern. |
| Security | Hiện thực chức năng bảo mật của chương trình: Codebook và HMAC. |
| Authentication | Hiện thực quá trình đăng ký và xác thực người dùng. |
| Utils | Gồm các hàm trợ giúp như xử lý dữ liệu đầu vào, lưu trữ các vector đặc trưng… |
| Settings | Cho phép người dùng xem hướng dẫn, chọn giải thuật xác thực. |

### Chi tiết các khối

**UI**

|  |  |
| --- | --- |
| Bảng tóm tắt các Activity | |
| WelcomeActivity | Activity khởi động của chương trình, hiển thị giới thiệu sơ lược về ứng dụng. Ở bước này, các tập tin chứa dữ liệu huấn luyện cũng sẽ được sao lưu vào máy để sử dụng cho những bước sau. |
| SignUpActivity | Activity hiển thị màn hình để người dùng đăng ký mới, đăng ký lại, hoặc kích hoạt tính năng xác thực khuôn mặt. |
| PictureActivity | Activity hỗ trợ chụp khuôn mặt của người dùng, và cắt hình ảnh theo kích thước định sẵn. |
| FaceTrackerActivity | Activity dùng cho giai đoạn đăng ký khuôn mặt, sau khi chụp ảnh, sẽ tiến hành huấn luyện lần lượt với cả ba giải thuật, và lưu sketch cũng như mẫu gốc đã được hash dưới dạng file xml. |
| AuthenticationActivity | Activity dùng cho giai đoạn xác thực khuôn mặt, sau khi chụp ảnh, sẽ tiến hành khôi phục mẫu gốc với mẫu vừa chụp và sketch được lưu lại ở bước trên, sau đó tiến hành so sánh kết quả. |
| SettingActivity | Activity cho phép người dùng xem hướng dẫn của chương trình và lựa chọn giữa các giải thuật xác thực. |

**Camera**

|  |  |
| --- | --- |
| Bảng tóm tắt các phương thức quan trọng | |
| CameraSourcePreview | Yêu cầu tính năng camera của điện thoại. |
| FaceGraphic | Xác định khuôn mặt của người dùng, vẽ khung hình chữ nhật bao quanh khuôn mặt trên camera. |
| PhotoHandle | Đọc, lưu, và cắt hình ảnh theo kích thước định sẵn. |
| PreprocessImage | Tiền xử lý hình ảnh trước khi áp dụng các giải thuật, làm mờ nền, chuyển sang Đa mức xám (Grayscale), cân chỉnh sáng. |

**Feature Extraction**

Ở đây, nhóm hiện thực rút trích đặc trưng với ba giải thuật Eigenface, Fisherface và LBP. Mỗi giải thuật có các đặc thù, tính toán khác nhau, nhưng có những điểm chung như sau:

|  |  |
| --- | --- |
| Bảng tóm tắt các phương thức quan trọng | |
| compute | Dữ liệu đầu vào là tập các hình ảnh và nhãn trong tập huấn luyện, ta sẽ tính toán ra hình ảnh trung bình và vector riêng (eigenvector). |
| doProject | Chiếu các mẫu đăng ký của người dùng vào không gian huấn luyện ở bước trên. |
| getAuthenMat | Sử dụng trong bước xác thực, chiếu mẫu xác thực vào không gian huấn luyện. |

**Security**

|  |  |
| --- | --- |
| Bảng tóm tắt các phương thức quan trọng | |
| calculateRFC2104HMAC | Dữ liệu đầu vào là nội dung thông điệp và khóa, dữ liệu đầu ra là thông điệp đã được mã hóa theo giải thuật HMAC SHA-1. |
| quantization | Dùng để chuyển giá trị của vector đặc trưng từ miền liên tục sang miền rời rạc. |
| sketchGeneration | Tạo ra sketch dựa trên vector đặc trưng. |
| featureRecovery | Dữ liệu đầu vào là mẫu xác thực và sketch, khôi phục lại mẫu ban đầu theo giải thuật codebook. |

**Authentication**

|  |  |
| --- | --- |
| Bảng tóm tắt các phương thức quan trọng | |
| codebookGenerator | Dữ liệu đầu vào là tập các vector đặc trưng sau khi đã chiếu vào không gian huấn luyện, áp dụng giải thuật codebook, dữ liệu đầu ra là tập các sketch ứng với các vector trên. |
| hashTemplate | Dữ liệu đầu vào cũng là tập các vector đặc trưng sau khi đã chiếu vào không gian huấn luyện, áp dụng HMAC, dữ liệu đầu ra là thông tin đã được mã hóa, dùng để so sánh trong bước xác thực. |
| eigenFace/fisherFace/LBPTraining | Đọc tập dữ liệu huấn luyện có sẵn và huấn luyện theo từng giải thuật, gọi hàm sinh sketch và băm mẫu ban đầu, lưu cả hai thông tin lại để dùng trong bước xác thực. |
| codebookRecover | Dữ liệu đầu vào là mẫu xác thực mới và tập các sketch đã lưu. Khôi phục lại mẫu ban đầu bằng cách áp dụng thuật toán tuần tự lên mẫu và từng sketch. Dữ liệu đầu ra là tập các mẫu được khôi phục. |
| predict | Dữ liệu đầu vào là tập các mẫu ban đầu và tập các mẫu vừa được khôi phục lại, vì lý do hiệu suất, nên không so sánh trùng khớp hoàn toàn, mà chỉ so sánh theo từng phần tử, |
| eigenFace/fisherFace/LBPAuthentication | Đọc tập sketch và mẫu ban đầu đã lưu, chiếu mẫu xác thực vào không gian huấn luyện, khôi phục lại mẫu ban đầu, và thực hiện so sánh. |

**Utils**

|  |  |  |
| --- | --- | --- |
| Bảng tóm tắt các phương thức quan trọng | | |
| InputHandle | Từ các hình được chụp, thực hiện giai đoạn tiền xử lý như cắt hình theo chuẩn, cân bằng sáng… |
| StoreModel | Hiện thực các hàm lưu và đọc những dữ liệu cần thiết cho các quá trình đăng ký và xác thực như tập huấn luyện, sketch, mẫu băm… |

**Settings**

Setting Actitivy cho phép người dùng lựa chọn một trong ba giải thuật để xác thực. Hoặc có thể kết hợp cả ba để tăng tính bảo mật.

## Kiểm định sản phẩm

### Danh sách Unit Test Case.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stt** | **Tên lớp** | **Tên hàm** | **Mã hàm** | **Mô tả** | **Tiền điều kiện** |
| 1 | FaceTracerActivity | doSignUp | SignUp | Đăng ký mẫu sinh trắc với hệ thống. | -Đăng ký lần đầu.  -Đã có các file cần thiết.  -Chụp đủ 10 tấm hình khuôn mặt. |
| 2 | SignUpActivity | doReSignUp | Re-SignUp | Cập nhật mẫu sinh trắc (Đăng ký lại). | -Đã đăng ký  -Đã xác thực thành công. |
| 3 | AuthenticationActivity | doAuthentication | Authentication | Thực hiện xác thực. | -Đã đăng ký.  -Đã có các file cần thiết.  -Đã có hình để xác thực. |
| 4 | SettingActivity | onSharedPreferenceChanged | Setting | Các tùy chỉnh thuật toán, ngưỡng… | -Không. |
| 5 | WelcomeActivity | launchWelcomeActivity | Help | Thông tin về ứng dụng. | -Không. |

### Chi tiết Unit Test Case

* SignUp

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | **UTCID01** |
| **Điều kiện** | **Tiền điều kiện** |  |  |  |
|  |  | **Đăng ký lần đầu** |  |  |
|  |  | **Có các file xml cần thiết** |  |  |
|  |  | **Có đủ 10 tấm hình khuôn mặt** |  |  |
| **Confirm** | **Return** |  |  |  |
|  | **Exception** |  |  |  |
|  | **Log message** |  |  |  |
|  |  |  | "Sign Up Successfully" | **O** |
| **Result** | Type(N : Normal, A : Abnormal, B : Boundary) | | | N | |
|  | Passed/Failed | | | P | |
|  | Executed Date | | | # | |
|  | Defect ID | | |  | |

* Re-SignUp

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  | **UTCID01** | |
| **Điều kiện** | **Tiền điều kiện** |  |  |  | |
|  |  | **Đã đăng ký** |  |  | |
|  |  | **Đã xác thực thành công** |  |  | |
| **Confirm** | **Return** |  |  |  | |
|  | **Exception** |  |  |  | |
|  | **Log message** |  |  |  | |
|  |  |  | "Re-SignUp Successfully" | **O** | |
| **Result** | Type(N : Normal, A : Abnormal, B : Boundary) | | | | N | |
|  | Passed/Failed | | | | P | |
|  | Executed Date | | | | # | |
|  | Defect ID | | | |  | |

* Authentication

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | **UTCID01** | | **UTCID02** | |
| **Điều kiện** | **Tiền điều kiện** |  |  |  | |  | |
|  |  | **Đã đăng ký** |  |  | |  | |
|  |  | **Đã có các file cần thiết** |  |  | |  | |
|  |  | **Đã có hình để xác thực** |  |  | |  | |
|  | **Hình xác thực đúng** |  |  | **O** | |  | |
|  | **Hình xác thực sai** |  |  |  | | **O** | |
| **Confirm** | **Return** |  |  |  | |  | |
|  |  |  | true | **O** | |  | |
|  |  |  | false |  | | **O** | |
|  | **Exception** |  |  |  | |  | |
|  | **Log message** |  |  |  | |  | |
| **Result** | Type(N : Normal, A : Abnormal, B : Boundary) | | | | N | | N | |
|  | Passed/Failed | | | | P | | P | |
|  | Executed Date | | | | # | | # | |
|  | Defect ID | | | |  | |  | |

* Setting

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  | **UTCID01** | **UTCID02** |
| **Điều kiện** | **Tiền điều kiện** |  |  |  |  |
|  | **Đổi thuật toán** |  |  | **O** |  |
|  | **Đổi ngưỡng** |  |  |  | **O** |
| **Confirm** | **Return** |  |  |  |  |
|  |  |  | true | **O** | **O** |
|  |  |  | false |  |  |
|  | **Exception** |  |  |  |  |
|  | **Log message** |  |  |  |  |
| **Result** | Type(N : Normal, A : Abnormal, B : Boundary) | | | N | N |
|  | Passed/Failed | | | P | P |
|  | Executed Date | | | # | # |
|  | Defect ID | | |  |  |

* Help

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  | **UTCID01** | |
| **Điều kiện** | **Tiền điều kiện** |  |  |  | |
| **Confirm** | **Return** |  |  |  | |
|  | **Exception** |  |  |  | |
|  | **Log message** |  |  |  | |
|  |  |  | "Show Welcome Successfully" | **O** | |
| **Result** | Type(N : Normal, A : Abnormal, B : Boundary) | | | | N | |
|  | Passed/Failed | | | | P | |
|  | Executed Date | | | | # | |
|  | Defect ID | | | |  | |

### Báo cáo Unit Test

(N : Normal, A : Abnormal, B : Boundary)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Stt** | **Mã hàm** | **Passed** | **Failed** | **Untested** | **N** | **A** | **B** | **Tổng cộng** |
| 1 | [SignUp](file:///C:\\Users\\Quang%20Huy\\Downloads\\Group03_Unit-Test-Case.xls" \l "HotStory!A1) | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 2 | [Re-SignUp](file:///C:\\Users\\Quang%20Huy\\Downloads\\Group03_Unit-Test-Case.xls" \l "ChooseStory!A1) | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 3 | [Authentication](file:///C:\\Users\\Quang%20Huy\\Downloads\\Group03_Unit-Test-Case.xls" \l "ChooseChapterToRead!A1) | 2 | 0 | 0 | 2 | 0 | 0 | 2 |
| 4 | [Setting](file:///C:\\Users\\Quang%20Huy\\Downloads\\Group03_Unit-Test-Case.xls" \l "Setting!A1) | 2 | 0 | 0 | 2 | 0 | 0 | 2 |
| 5 | [Help](file:///C:\\Users\\Quang%20Huy\\Downloads\\Group03_Unit-Test-Case.xls" \l "Login!A1) | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
|  | Sub total | 8 | 0 | 0 | 8 | 0 | 0 | 8 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test coverage** |  | **100** | % |
| **Test successful coverage** |  | **100** | % |
| **Normal case** |  | **100** | % |
| **Abnormal case** |  | **0** | % |
| **Boundary case** |  | **0** | % |

Hình . Tỷ lệ kết quả theo testcase

Hình . Số loại testcase

# ĐÁNH GIÁ KẾT QUẢ //TODO

## Phương pháp đánh giá

#### Tiêu chí đánh giá

Nhóm hiện thực và đánh giá theo các tiêu chí sau:

* Thời gian và độ chính xác khi xác thực không sử dụng giải thuật bảo mật
* Thời gian và độ chính xác khi xác thực sử dụng giải thuật bảo mật

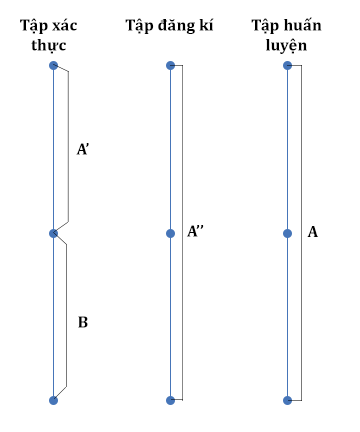
#### Xây dựng tập thí nghiệm sử dụng để huấn luyện, đăng kí và kiểm tra

Để đánh giá các tiêu chí trên, nhóm đã sử dụng tập dữ liệu dùng để nghiên cứ nhận dạng khuôn mặt là Face94. Từ tập Face94 nhóm chọn lọc các hình để tạo lên tập thí nghiệm gồm 3 phần chính:

- Tập huấn luyện A: Tập này dùng để làm không gian chiếu cho PCA và Fisherface. Nhóm chọn 100 người có cả đàn ông và phụ nữ. Mỗi người có 10 tấm hình khác nhau được đánh số theo cấu trúc tên [thứ tự của người đó]-[thứ tự ảnh của người đó]

- Tập đăng kí A’’: Tập này dùng để đăng ký 100 người với hệ thống để tiến hành xác thực. Nhóm chọn 100 người có cả đàn ông và phụ nữ, những người này là những người giống người ở tập A (về thứ tự nhưng khác hình). Mỗi người có 5 tấm hình khác nhau và khác hình ở tập A.

- Tập xác thực: Tập này là những hình đầu vào để xác thực và đo độ chính xác của giải thuật. Tập này gồm 100 người, mỗi người 1 tấm hình (ảnh đúng vị trí gồm 10 ảnh sáng, 10 ảnh tối, 30 ảnh bình thường và ảnh sai vị trí tương tự). Nhóm đã sắp thứ tự những hình này để tạo ra testcase so sánh. Testcase đúng sẽ là đăng nhập được với 50 người đầu (tập A’) và không đăng nhập được với 50 người sau (tập B).



Hình . Cấu trúc tập thí nghiệm

#### Phương pháp đánh giá thực nghiệm

Ta sẽ đánh giá dựa trên các tỷ lệ:

* FAR (False Accept Rate): số lần hệ thống trả về đăng nhập được khi kết quả chính xác của thí nghiệm là không đăng nhập được trên tổng số.
* FRR (False Reject Rate): số lần hệ thống trả về không đăng nhập được khi kết quả chính xác của thí nghiệm là đăng nhập được trên tổng số.

Từ FAR và FRR ta sẽ tìm ra điểm ngưỡng tối ưu mà ở đó tỉ lệ FAR và FRR chấp nhận được.

#### Môi trường thí nghiệm

Điện thoại gionee CTRL V4S với thông số phần cứng như sau:

-Camera trước 2.0 MP

-Hệ điều hành Android 5.0 (Lolillop)

-Chipset MTK 6582

-CPU Quad-core 1.3 GHz

-RAM 1GB

## Các thí nghiệm

#### - Đánh giá hiệu suất giải thuật PCA trên tập thí nghiệm

Hiệu suất giải thuật được đo về thời gian về độ chính xác của giải thuật PCA đối với tập thí nghiệm đã miêu tả ở trên.

Thời gian chạy (đơn vị mm:ss.sss):

|  |  |
| --- | --- |
| PCA | Tổng thời gian |
| Không bảo mật | 17:17.092 |
| Có bảo mật | 26:08.704 |

Độ chính xác:

Hình . Độ chính xác của PCA với các ngưỡng

Với lần chạy không sử dụng bảo mật trên tập thí nghiệm, giải thuật PCA đạt kết quả tốt nhất ở ngưỡng 2000 với tỉ lệ FAR bằng 0/50 và FRR bằng 1/50.

Hình . Độ chính xác của PCA với các DELTA

Với lần chạy sử dụng bảo mật Secure Sketch trên tập thí nghiệm, giải thuật PCA đạt kết quả tốt nhất ở giá trị DELTA bằng 50 với tỷ lệ FAR bằng 1/50 và FRR bằng 2/50.

#### -Đánh giá hiệu suất giải thuật Fisherface trên tập thí nghiệm

Hiệu suất giải thuật được đo về thời gian về độ chính xác của giải thuật Fisherface đối với tập thí nghiệm đã miêu tả ở trên

Thời gian chạy (đơn vị mm:ss.sss):

|  |  |
| --- | --- |
| Fisherface | Tổng thời gian |
| Không bảo mật | 21:27.597 |
| Có bảo mật | 29:56.123 |

Độ chính xác:

Hình . Độ chính xác của Fisherface với các ngưỡng

Với lần chạy không sử dụng bảo mật trên tập thí nghiệm, giải thuật Fisherface đạt hết quả tốt nhất ở ngưỡng 720 với tỉ lệ FAR bằng 0/50 và FRR bằng 0/50.

Hình . Độ chính xác của Fisherface với các DELTA

Với lần chạy sử dụng bảo mật Secure Sketch trên tập thí nghiệm, giải thuật Fisherface đạt kết quả tốt nhất ở giá trị DELTA bằng 42 với tỷ lệ FAR bằng 1/50 và FRR bằng 1/50.

#### -Đánh giá hiệu suất giải thuật LBP trên tập thí nghiệm

Hiệu suất giải thuật được đo về thời gian về độ chính xác của giải thuật LBP đối với tập thí nghiệm đã miêu tả ở trên

Thời gian chạy (đơn vị mm:ss.sss):

|  |  |
| --- | --- |
| PCA | Tổng thời gian |
| Không bảo mật | 15:12.466 |
| Có bảo mật | 21:10.141 |

Độ chính xác:

Hình . Độ chính xác của LBP với các ngưỡng

Với lần chạy không sử dụng bảo mật trên tập thí nghiệm, giải thuật LBP đạt kết quả tốt nhất ở ngưỡng 265 với tỉ lệ FAR bằng 0/50 và FRR bằng 6/50.

Hình . Độ chính xác của LBP với các DELTA

Với lần chạy sử dụng bảo mật Secure Sketch trên tập thí nghiệm, giải thuật LBP đạt kết quả tốt nhất ở giá trị DELTA bằng 3 với tỷ lệ FAR bằng 12/50 và FRR bằng 1/50.

#### -Đánh giá hiệu suất kết hợp 3 giải thuật trên tập thí nghiệm

Hiệu suất giải thuật được đo về thời gian về độ chính xác của 3 giải thuật đối với tập thí nghiệm đã miêu tả ở trên. Cụ thể, kết quả của sự kết hợp được tính là xác thực đúng nếu 2 trên 3 giải thuật cho kết quả xác thực đúng. Các thông số ngưỡng tối ưu và delta tối ưu của mỗi giải thuật được sử dụng để tiến hành thí nghiệm:

|  |  |  |
| --- | --- | --- |
| Thông số tối ưu | Ngưỡng | DELTA |
| PCA | 2000 | 50 |
| Fisherface | 720 | 42 |
| LBP | 265 | 3 |

Thời gian chạy (đơn vị mm:ss.sss) và độ chính xác:

|  |  |  |  |
| --- | --- | --- | --- |
| PCA, Fisherface, LBP | FAR | FRR | Tổng thời gian |
| Không bảo mật | 0/50 | 0/50 | 36:27.66 |
| Có bảo mật | 1/50 | 2/50 | 41:29.083 |

Với lần chạy không sử dụng bảo mật trên tập thí nghiệm, 3 giải thuật đạt kết quả tốt nhất với tỉ lệ FAR bằng 0/50 và FRR bằng 0/50.

Với lần chạy sử dụng Secure Sketch trên tập thí nghiệm, 3 giải thuật đạt kết quả tốt nhất với tỷ lệ FAR bằng 1/50 và FRR bằng 2/50.

## Kết quả thí nghiệm

Từ các kết quả thí nghiệm trên, ta rút ra một số kết luận sau:

* Thuật toán PCA hoạt động khá hiệu quả khi testcase có cùng điều kiện ánh sáng với hình ảnh trong tập huấn luyện, hướng của khuôn mặt là chính diện. Tốc độ của thuật toán khá nhanh trong việc huấn luyện, đăng ký và xác thực. Đối với những testcase có điều kiện ánh sáng thay đổi, cường độ sáng cao hoặc thấp. Tỷ lệ sai xót khá cao, việc xác thực cho kết quả không như mong đợi.
* Thuật toán Fisherface chạy tốt khi tập train và tập test có cùng điều kiện ánh sáng cũng như diện mạo. Đối với các ảnh sáng hoặc tối, giải thuật chạy không ổn định. Kết quả xác thực chỉ đúng và ổn định ở một số tập sáng đầu tiên.
* Giải thuật LBP chạy tốt ở các điều kiện sáng khác nhau. Độ chính xác của LBP không cao. Tốc độ chạy của LBP nhanh hơn so với PCA và Fisherface.
* Khi không sử dụng bảo mật, giải thuật Fisherface đạt độ chính xác cao nhất, sau đó tới PCA và cuối cùng là LBP. Khi sử dụng bảo mật Secure Sketch, giải thuật PCA đạt độ chính xác cao nhất, sau đó tới Fisherface và cuối cùng là LBP.
* Kết hơp 3 giải thuật cho kết quả tốt khi chạy trên tập thí nghiệm.

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

Đề tài của luận văn là “Phát triển hệ thống xác thực bằng đặc trưng gương mặt trên Android có bảo vệ mẫu đặc trưng sinh trắc”. Mục tiêu của hệ thống là thay thế màn hình khóa (lock screen) cổ điển bằng xác thực khuôn mặt người dùng, với trọng tâm là rút trích các đặc trưng sinh trắc, kết hợp với bảo mật.

Sau quá trình tìm hiểu và nghiên cứu, nhóm đã lựa chọn và hiện thực ba giải thuật rút trích đặc trưng sinh trắc là Eigenface, Fisherface và Local Binary Pattern. Mỗi giải thuật có ưu, nhược điểm khác nhau. Nhóm đã tiến hành phân tích, hiện thực, so sánh kết quả giữa các giải thuật này. Ngoài ra, nhóm cũng đề xuất sử dụng kết hợp giữa ba giải thuật để đạt được kết quả tốt nhất.

Tuy trọng tâm của đề tài là rút trích đặc trưng, để hoàn thiện hệ thống, nhóm đã hiện thực bảo mật mẫu sử dụng Secure Sketch theo mô hình Codebook và HMAC. Với tính năng này, cho dù thiết bị có bị đánh cắp hoặc tấn công, đặc trưng sinh trắc của người dùng cũng không bị lộ hoặc sử dụng trái phép. Ta cũng có thể dễ dàng đổi khóa của hàm băm để tạo ra mẫu mới, kẻ tấn công sẽ không đăng nhập vào hệ thống với mẫu cũ được.

Về kết quả, cả ba giải thuật có hiệu suất khá cao trên tập Faces94, cũng như khi sử dụng với các khuôn mặt ở ngoài. Điều này cho thấy các giải thuật rút trích được hiện thực khá tốt.

Điểm trừ của hệ thống đó là sau khi áp dụng thêm lớp bảo mật, hiệu suất giảm, đặc biệt là khi sử dụng trong thực tế. Lý do chính là vì các tác động bên ngoài như ánh sáng, góc mặt, kiểu tóc, phông nền… ảnh hưởng đến quá trình rút trích đặc trưng. Mặc dù Secure Sketch giúp sửa lỗi và khôi phục lại mẫu gốc, nhưng rất khó để khôi phục lại chính xác hoàn toàn mẫu gốc ban đầu. Tuy nhiên với tỷ lệ lỗi thấp, đặc biệt là FAR, chương trình vẫn đảm bảo sự an toàn tốt cho hệ thống.

## Hướng phát triển

Về hướng phát triển, có thể được chia thành hai hướng chính sau đây:

* Nâng cao giải thuật tiền xử lý cho ảnh: hiện tại, hệ thống cũng đã có chức năng giúp cân chỉnh sáng, xóa mờ phông nền cho ảnh sau khi chụp. Tuy nhiên, để cải thiện hiệu suất, cần có sự tìm hiểu kỹ hơn, và hiện thực các giải thuật tốt hơn.
* Bảo mật cho hệ thống: tính năng bảo mật cho hệ thống hiện tại còn đơn giản, tuy có khắc phục nhiễu nhưng chưa hoàn chỉnh. Tuy bảo mật không phải là trọng tâm của đề tài, nhưng nhóm cũng đã có tìm hiểu thêm các giải thuật sinh codebook khác nhau như Kekre’s Fast Code Book Generation kết hợp với K-Means. Do nhận thấy các giải thuật đó không phù hợp với các vector đặc trưng, nên nhóm đã không hiện thực. Do đó, cần giành nhiều thời gian nghiên cứu các giải thuật khác để đảm bảo cho hệ thống an toàn cũng như không làm giảm hiệu suất.

# THAM KHẢO

[1] Peter H. Dana (2000, May 2). *The Global Positioning System*. Retrieved from <http://www.colorado.edu/geography/gcraft/notes/gps/gps_f.html>

[2] *What is GPS?* (Data Accessed: 2017, May 26). Retrieved from <http://www.furuno.com/en/gnss/technical/tec_what_gps>

[3] Kaplan, E., & Hegarty, C. (2005). *Understanding GPS: principles and applications*. Artech house.

[4] *Android (Hệ điều hành)* (Data Accessed: 2017, May 26). Retrieved from <https://vi.wikipedia.org/wiki/Android_(hệ_điều_hành)>

[5] *App stores: number of apps in leading app stores 2017* (Data Accessed: 2017, May 26). Retrieved from <https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/>

[6] *Google Play: number of downloads 2010-2016* (Data Accessed: 2017, May 26). Retrieved from <https://www.statista.com/statistics/281106/number-of-android-app-downloads-from-google-play/>

[7] *Platform Architecture* (Data Accessed: 2017, May 26). Retrieved from <https://developer.android.com/guide/platform/index.html>

[8] *Dashboards* (Data Accessed: 2017, May 27). Retrieved from <https://developer.android.com/about/dashboards>

[1] PGS.TS. Nguyễn Thị Hoàng Lan – *Hệ thống an ninh thông tin dựa trên sinh trắc học - PKI (Bio-PKI Based Information Security System)*, 2009.

[2] A. Singh and S. Kumar, *Face Recognition Using PCA,* 2012.

[3] Face Recognition with OpenCV. (n.d.). Retrieved from OpenCV 2.4.13.1 documentation: http://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec\_tutorial.html#fisherfaces

[4] Tạ Tiến Thành- Quách Thu Thảo, *Một số phương pháp trích chọn đặc trưng khuôn mặt và ứng dụng*.

[5] Ths Nguyễn Quế Hương, *Nhận diện biển số xe sử dụng đặc trưng LBP (Local Binary Parttern)*

[6] Tran Tri Dang, Q. C. (n.d.), *Practical Construction of Face-Based Authentication Systems with Template Protection Using Secure Sketch.* ICT-EurAsia, 2013.

[7] Khang, P. N. (n.d.), *An toàn và Bảo mật Thông tin.* Retrieved from Khoa Công nghệ Thông tin và Truyền thông: http://www.cit.ctu.edu.vn/~pnkhang/cours/atbmtt/chuong%204-%20ham%20bam.pdf.

[8] Hussein Rady, *Face Recognition using Principle Component Analysis Face Recognition using Principle Component Analysis with Different ifferent ifferent Distance istance istance Classifiers*, 2011.

[9] Timo Ahonen, Student Member, IEEE, Abdenour Hadid, and Matti Pietikainen, ¨ Senior Member, IEEE. *Face Description with Local Binary Patterns: Application to Face Recognition*. 2006

[10] Mobile Vison API [Trực tuyến]. Available:   
https://developers.google.com/vision/.

[11] Thư viện JavaCV [Trực tuyến]. Available:   
https://github.com/bytedeco/javacv.

[12] Thư viện OpenCV [Trực tuyến]. Available:   
https://en.wikipedia.org/wiki/OpenCV.

[13] Mobile Vison API [Trực tuyến]. Available:   
https://developers.google.com/vision/.

[14] Thư viện Sweet Alert Dialog [Trực tuyến]. Available: https://github.com/pedant/sweet-alert-dialog.

[15] Trị riêng, vector riêng [Trực tuyến]. Available https://vi.wikipedia.org/wiki/Giá\_trị\_riêng.

[16] Chi-squared test [Trực tuyến]. Available: https://en.wikipedia.org/wiki/Chi-squared\_test.

# PHỤ LỤC

## Trị riêng, vector riêng [[15]](#ref14)

### Định nghĩa

Cho  A là ma trận vuông cấp n trên trường số K(K=R;C). Số λ ϵ K được gọi là giá trị riêng (gọi tắt là trị riêng – kí hiệu GTR) của ma trận A, nếu tồn tại một vectơ u ≠ 0, u ϵ Kn sao cho: Au = λu

Khi đó vectơ u được gọi là vectơ riêng (VTR) của ma trận A ứng với giá trị riêng λ.

### Tính chất

* Giá trị riêng λ chính là nghiệm của phương trình det(A – λI) = 0 (1) được gọi là phương trình đặc trưng của ma trận A.
* Một giá trị riêng có thể có nhiều vectơ riêng.
* Mỗi vectơ riêng chỉ ứng với một giá trị riêng duy nhất.
* Ma trận A là nghiệm của đa thức đặc trưng của chính nó (trong trường hợp này đa thức đặc trưng được coi là đa thức ma trận, nghĩa là biến số của nó không phải là biến số thực mà là biến ma trận)
* Nếu λ = 0 là giá trị riêng của ma trận A thì A không khả nghịch. Ngược lại, nếu mọi GTR của A đều khác không thì A khả nghịch.
* Nếu λ là GTR của ma trận A thì λk là giá trị riêng của ma trận Ak.

### Chứng minh

Số λ là trị riêng của A khi và chỉ khi Au = λu (u ≠ 0). Suy ra: hệ phương trình tuyến tính thuần nhất (A – λI)u = 0 có nghiệm u ≠ 0 ⬄ det(A – λI) = 0.

Điều này là hiển nhiên vì dựa vào định nghĩa và tính chất 1 thì hệ phương trình (A – λI)u = 0 có vô số nghiệm.

Giả sử vectơ riêng u1 ứng với 2 trị riêng λ1, λ2.

Ta cần chứng minh:  λ1 = λ2. Thật vậy, ta có:

Au1 = λ1u1 ; Au1 = λ2u1 => λ1u1 – λ2u1 = 0 => (λ1 - λ2) u1 = 0

Mà: u1 ≠ 0 Do đó: λ1 - λ2  = 0.

Ta có:

P(λ) = det(A – λI) => P(A) = det( A - A.I) = det(A – A) = 0

Do λ = 0  là GTR của ma trận A. Do đó:

P(0) = det(A – 0.I) = 0 => det(A) = 0.

Chứng tỏ A suy biến (không khả nghịch).

Ta có Au = λu. Do đó:

A2u = (A.A)u = A.(A.u) = A.(λu) = λ.Au =λ2u.

Từ đó, bằng cách chứng minh quy nạp, bạn sẽ có kết quả.

Nhận xét: từ kết quả trên, ta nhận thấy có 1 cách để tính nhanh |A – a.I |. Đó là ta tìm đa thức đặc trưng P(λ) = | A – λ.I | của ma trận A. Sau đó, tính giá trị của P(a).

### Phương pháp giải tìm trị riêng, vectơ riêng

* Bước 1: Giải phương trình đặc trưng det( A – λI) = 0 (1) tìm giá trị riêng.
* Bước 2: Tìm vectơ riêng ứng với giá trị riêng λ

Ứng với mỗi giá trị riêng λivừa tìm được, ta giải hệ phương trình tuyến tính thuần nhất

(A – λiI)u = 0 (2)

Lưu ý: theo tính chất trên, thì hpt (2) luôn luôn có vô số nghiệm. Do đó, nếu bạn giải pt (2) mà vô nghiệm hoặc có nghiệm duy nhất thì phải kiểm tra lại.

### Không gian con riêng ứng với GTR λ

* Các vetơ riêng của ma trận A ứng với giá trị riêng λ0 cùng với vectơ 0 tạo thành 1 không gian con được gọi là không gian con riêng ứng với λ0 .
* Ký hiệu: E(λ0) = { u ϵ Kn : Au = λu }

Nếu giá trị riêng λ0 là nghiệm bội k thì dim E(λ0) ≤ k.

## Chi-Squared [[16]](#ref13)

Kiểm định Chi bình phương thường được sử dụng khi chúng ta muốn xem liệu có mối quan hệ giữa hai biến phân loại (categorical variables) trong một tổng thể hay so sánh độ hớp của dữ liệu.

Dạng thống kê kiểm định Chi-Squared thông dụng nhất là:

Với là dữ liệu đo đạc (hay giá trị quan sát), là giá trị kì vọng.

Chi-Squared sẽ giúp cân bằng trọng số giữa các độ chênh lệch mà không bị phụ thuộc nhiều vào độ lớn của giá trị, do đó phản ánh rõ ràng hơn sự khác biệt của dữ liệu. Ví dụ : Ta tính chi square của 2 vector A[0]=0.001 và B[0]=0.011, đồng thời tính chi square của A’[0]=0.01 và B’[0]=0.11. Rõ ràng độ khác biệt của chúng đều là 0.01 nhưng khi chia cho sẽ phản ánh tốt hơn độ chênh lệch.

Phương pháp này kiểm tra tính độc lập thống kê và mức độ khớp của dữ liệu. Nếu nhỏ thì giá trị quan sát gần khớp với giá trị kì vọng.