**BOR3 BỘ CÔNG THƯƠNG**

**TRƯỜNG CAO ĐẲNG KỸ THUẬT CAO THẮNG**

**KHOA ĐIỆN TỬ - TIN HỌC**

**...** 🙢 🕮 🙠 **...**

****

**ĐỒ ÁN TỐT NGHIỆP**

**Phát triển ứng dụng Android**

**Ứng dụng điểm danh thông minh**

*Giảng viên hướng dẫn:* **Vũ Đình Bảo**

*Sinh viên thực hiện:*

**Nguyễn Phước Hậu** MSSV**: 0306151238**

**Nguyễn Thanh Thế** MSSV**: 0306151420**

Lớp: **CĐTH 15PMC**

Khóa: **2015-2018**

TP.HỒ CHÍ MINH, ngày 03 tháng 07 năm 2018.

**LỜI CẢM ƠN**

Chúng em xin chân thành cảm ơn quý thầy cô trong khoa Điện tử - Tin học trường cao đẳng Kỹ Thuật Cao Thắng, để có thể hoàn thành được đồ án tốt nghiệp này là kết quả nền tảng từ kiến thức của sự giảng dạy tận tình của các quý thầy cô, trang bị cho chúng em những kiến thức quý báu và vận dụng những kiến thức đã học vào công việc khi ra trường.

Đặc biệt với tấm lòng biết ơn sâu sắc, em xin chân thành cảm ơn thầy Vũ Đình Bảo đã trực tiếp hướng dẫn, đóng góp ý kiến, giúp đỡ chúng em hoàn thành đồ án tốt nghiệp này.

Mặc dù đã cố gắng nổ lực thực hiện đồ án này nhưng với kiến thức và thời gian có hạn, chắc hẳn đồ án này sẽ không tránh khỏi sai sót. Chúng em rất mong nhận được sự thông cảm và đóng góp ý kiến của thầy cô để chúng em có thể hoàn thiện tốt hơn cho sau này.

Chúng em xin chân thành cám ơn!

Tp.Hồ Chí Minh, tháng 7 năm 2018

Nhóm sinh viên thực hiện

**Nguyễn Phước Hậu**

**Nguyễn Thanh Thế**

**LỜI NÓI ĐẦU**

Những năm gần đây, trí tuệ nhân tạo (AI - Artificial Intelligence), và cụ thể hơn là học máy (Machine Learning) nổi lên như một bằng chứng của cuộc cách mạng công nghiệp lần thứ tư (1 - động cơ hơi nước, 2 - năng lượng điện, 3 - công nghệ thông tin). Trí Tuệ Nhân Tạo đang len lỏi vào mọi lĩnh vực trong đời sống mà có thể chúng ta không nhận ra. Xe tự hành của Google và Tesla, hệ thống tự đánh dấu (tag) khuôn mặt trong ảnh của Facebook, trợ lý ảo Siri của Apple, hệ thống gợi ý sản phẩm của Amazon, hệ thống gợi ý phim của Netflix, máy chơi cờ vây AlphaGo của Google DeepMind,..., chỉ là một vài trong vô vàn những ứng dụng của AI/Machine Learning.

Gần đây trong thời đại công nghiệp 4.0 cho thấy mối quan tâm dành cho Machine Learning của cộng đồng lập trình viên ngày càng tăng và phổ biến hơn bao giờ hết. Để hòa nhập và trang bị cho bản thân kiến thức mới nên chúng em quyết định tìm hiểu về Machine Learning hay cụ thể hơn là nhận diện khuôn mặt (Face Recognition).

Trong khoảng thời gian đi học, chúng em nhận ra rằng việc điểm danh mỗi ngày là một việc rất cần thiết trong quá trình quản lý học viên. Nhưng trong thực thế, việc điểm danh học viên còn **thủ công** tức giảng viên phải gọi tên từng bạn học viên, sau đó ghi chép vào một cuốn sổ hay một tập tin trên máy tính để lưu lại. Công việc này làm ảnh hưởng không ít thời gian của giảng viên và sinh viên dẫn đến quỹ thời gian chung của bài giảng bị giảm xuống.

Xuất phát từ những nhu cầu thực tế trên, chúng em quyết định chọn đề tài “Phát triển ứng dụng Android - Ứng dụng điểm danh thông minh” với ngôn ngữ Java, nhằm xây dựng một ứng dụng hỗ trợ đầy đủ tính năng cần thiết nhất cho việc điểm danh trong lớp học.

Nhóm sinh viên thực hiện đề tài:

Nguyễn Phước Hậu

MSSV: 0306151238

Điện Thoại: 0974685100

Email: [phuochau409@gmail.com](file:///C:\Users\Simplify%20V2\Desktop\Hinh_DoAn\phuochau409@gmail.com)

Nguyễn Thanh Thế

MSSV: 0306151420

Điện Thoại: 0925003022

Email: [nguyenthe0202.1997@gmail.com](mailto:nguyenthe0202.1997@gmail.com)

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

TP.HỒ CHÍ MINH, ngày......tháng......năm 2018

Giáo viên hướng dẫn

**NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN**

TP.HỒ CHÍ MINH, ngày......tháng......năm 2018

Giáo viên phản biện

**MỤC LỤC**

[CHƯƠNG 1: GIỚI THIỆU 1](#_Toc518499607)

[1.1 Một số công nghệ được ứng dụng vào đồ án. 1](#_Toc518499608)

[1.1.1 Môi trường lập trình Android studio. 1](#_Toc518499609)

[1.1.1.1 Giới thiệu về Android Studio. 1](#_Toc518499610)

[1.1.1.2 Cài đặt Android Studio. 2](#_Toc518499611)

[1.1.2 Ngôn ngữ lập trình Java. 9](#_Toc518499612)

[1.1.2.1 Giới thiệu, lịch sử phát triển Java. 9](#_Toc518499613)

[1.1.2.2 Khả năng của ngôn ngữ Java 9](#_Toc518499614)

[1.1.2.3 Những đặc điểm của ngôn ngữ Java 10](#_Toc518499615)

[1.1.2.4 Máy ảo Java (JVM - Java Virtual Machine) 10](#_Toc518499616)

[1.1.2.5 Hai kiểu ứng dụng dưới ngôn ngữ java: 11](#_Toc518499617)

[1.1.2.6 Bộ phát triển ứng dụng Java (JDK- Java Development Kit) 11](#_Toc518499618)

[1.1.3 Framework OpenCV 12](#_Toc518499619)

[1.1.3.1 Giới thiệu Framework OpenCV 12](#_Toc518499620)

[1.1.3.2 Cài đặt OpenCV 12](#_Toc518499621)

[1.1.3.3 Tích hợp Framework OpenCV vào project Visual Studio 14](#_Toc518499622)

[1.1.3.4 Ví dụ minh họa Framework OpenCV 18](#_Toc518499623)

[1.1.4 Phát hiện và nhận diện khuôn mặt với Luxand FaceSDK. 21](#_Toc518499624)

[1.1.4.1 Giới thiệu Luxand FaceSDK 21](#_Toc518499625)

[1.1.4.2 Các nền tảng và cấu hình tối thiểu FaceSDK hỗ trợ 22](#_Toc518499626)

[1.1.4.3 FaceSDK với các thông số kỹ thuật 23](#_Toc518499627)

[1.1.4.4 Cài đặt Luxand FaceSDK. 24](#_Toc518499628)

[1.1.4.5 Cấu trúc thư mục Luxand FaceSDK 29](#_Toc518499629)

[1.1.4.6 Một số ứng dụng mẫu FaceSDK 30](#_Toc518499630)

[1.1.4.7 Các tình huống sử dụng 31](#_Toc518499631)

[1.2 Một số thuật toán rút trích đặc trưng và nhận dạng khuôn mặt. 32](#_Toc518499632)

[1.2.1 Giới thiệu một vài nhóm thuật toán Machine Learning 32](#_Toc518499633)

[1.2.2 Đặc trưng ảnh “ Histogram ”. 33](#_Toc518499634)

[1.2.2.1 Giới thiệu Histogram. 33](#_Toc518499635)

[1.2.2.2 Histogram của ảnh. 34](#_Toc518499636)

[1.2.3 Thuật toán rút trích đặt trưng Local Binary Patterns (LBP). 36](#_Toc518499637)

[1.2.3.1 Giới thiệu thuật toán rút trích đặt trưng LBP. 36](#_Toc518499638)

[1.2.3.2 Phân tích toán học. 36](#_Toc518499639)

[1.2.3.3 Thử nghiệm LBP. 37](#_Toc518499640)

[1.2.3.4 Thảo luận. 39](#_Toc518499641)

[1.2.4 Thuật toán K-Nearest Neighbors (KNN). 40](#_Toc518499642)

[1.2.4.1 Giới thiệu thuật toán K-Nearest Neighbors (KNN). 40](#_Toc518499643)

[1.2.4.2 Phân tích toán học 42](#_Toc518499644)

[1.2.4.3 Thử nghiệm KNN 43](#_Toc518499645)

[1.2.4.4 Thảo luận 44](#_Toc518499646)

[1.2.5 Thuật toán Support Vector Machines (SVM). 45](#_Toc518499647)

[1.2.5.1 Giới thiệu thuât toán SVM 45](#_Toc518499648)

[1.2.5.2 Phân tích toán học 46](#_Toc518499649)

[1.2.5.3 Thử nghiệm SVM 50](#_Toc518499650)

[1.2.5.4 Thảo luận 51](#_Toc518499651)

[CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ DỮ LIỆU 53](#_Toc518499652)

[2.1 Đặc tả chức năng. 53](#_Toc518499653)

[2.1.1 Quản lý lớp học. 53](#_Toc518499654)

[2.1.1.1 Thêm lớp học. 53](#_Toc518499655)

[2.1.1.2 Xóa lớp học. 53](#_Toc518499656)

[2.1.1.3 Sửa tên lớp học 53](#_Toc518499657)

[2.1.1.4 Cập nhật số lượng học viên 53](#_Toc518499658)

[2.1.2 Điểm danh. 53](#_Toc518499659)

[2.1.2.1 Quản lý ca điểm danh. 53](#_Toc518499660)

[2.1.2.2 Điểm danh bằng cách nhận dạng khuôn mặt. 53](#_Toc518499661)

[2.1.3 Thống kê. 53](#_Toc518499662)

[2.1.3.1 Thống kê theo lớp học. 54](#_Toc518499663)

[2.1.3.2 Thống kê theo học viên của từng lớp học. 54](#_Toc518499664)

[2.1.3.3 Thống kê ca điểm danh. 54](#_Toc518499665)

[2.1.3.4 Xuất file Excel. 54](#_Toc518499666)

[2.1.4 Ghi lịch sử 54](#_Toc518499667)

[2.2 Use Case. 54](#_Toc518499668)

[2.2.1 Thêm lớp học. 55](#_Toc518499669)

[2.2.2 Đổi tên lớp. 56](#_Toc518499670)

[2.2.3 Xóa lớp đang có. 55](#_Toc518499671)

[2.2.4 Tạo ca điểm danh. 56](#_Toc518499672)

[2.2.5 Điểm danh. 57](#_Toc518499673)

[2.2.6 Thống kê ca điểm danh hiện tại. 58](#_Toc518499674)

[2.2.7 Thống kê của lớp. 59](#_Toc518499675)

[2.2.7.1 Thống kê các lần điểm danh. 59](#_Toc518499676)

[2.2.7.2 Xem lịch sử thay đổi. 60](#_Toc518499677)

[2.3 Cơ sở dữ liệu. 61](#_Toc518499678)

[2.3.1 Danh mục các bảng. 61](#_Toc518499679)

[2.3.1.1 Bảng LOP\_HOC (Lớp Học). 61](#_Toc518499680)

[2.3.1.2 Bảng HOC\_VIEN (Học Viên). 61](#_Toc518499681)

[2.3.1.3 Bảng DIEM\_DANH (Điểm Danh). 62](#_Toc518499682)

[2.3.1.4 Bảng CHI\_TIET\_DIEM\_DANH (Chi Tiết Điểm Danh). 62](#_Toc518499683)

[2.3.1.5 Bảng LICH\_SU\_DIEM\_DANH (Lịch Sử Điểm Danh). 62](#_Toc518499684)

[2.3.2 Mô hình cơ sở dữ liệu 63](#_Toc518499685)

[CHƯƠNG 3: CÀI ĐẶT GIAO DIỆN VÀ CHƯƠNG TRÌNH 65](#_Toc518499686)

[3.1 Cấu trúc thư mục đồ án. 65](#_Toc518499689)

[3.2 Giao diện ứng dụng. 65](#_Toc518499690)

[3.2.1 Quản lý lớp học. 65](#_Toc518499691)

[3.2.1.1 Thêm lớp học. 65](#_Toc518499692)

[3.2.1.2 Xóa lớp học. 68](#_Toc518499693)

[3.2.1.3 Đổi tên lớp học. 70](#_Toc518499694)

[3.2.1.4 Danh sách lớp học. 72](#_Toc518499695)

[3.2.2 Điểm danh. 72](#_Toc518499696)

[3.2.2.1 Tạo ca điểm danh. 73](#_Toc518499697)

[3.2.2.2 Hủy ca điểm danh. 78](#_Toc518499698)

[3.2.2.3 Điểm danh bằng cách nhận diện khuôn mặt 80](#_Toc518499699)

[3.2.3 Thống kê. 80](#_Toc518499700)

[3.2.3.1 Thống kê theo lớp học. 81](#_Toc518499701)

[3.2.3.2 Thống kê theo học viên của từng lớp học. 83](#_Toc518499702)

[3.2.3.3 Thống kê ca điểm danh. 85](#_Toc518499703)

[3.2.3.4 Xuất file Excel 86](#_Toc518499704)

[3.2.4 Ghi lịch sử. 87](#_Toc518499705)

[CHƯƠNG 4: TỔNG KẾT VÀ HƯỚNG PHÁT TRIỄN 89](#_Toc518499706)

[4.1 Kết quả đạt được. 89](#_Toc518499707)

[4.2 Phần hạn chế của đề tài. 89](#_Toc518499708)

[4.3 Hướng phát triễn. 89](#_Toc518499709)

[TÀI LIỆU THAM KHẢO 91](#_Toc518499710)

**MỤC LỤC HÌNH ẢNH**

CHƯƠNG 1: GIỚI THIỆU

[Hình 1 - 1: Logo Android Studio 1](#_Toc518499734)

[Hình 1 - 2: Tải Android Studio 2](#_Toc518499735)

[Hình 1 - 3: Hộp thoại Welcome to Android Studio Setup 3](#_Toc518499736)

[Hình 1 - 4: Hộp thoại yêu cầu JDK 3](#_Toc518499737)

[Hình 1 - 5: Hộp thoại cấu hình cài đặt 4](#_Toc518499738)

[Hình 1 - 6: Hộp thoại thông báo cáo điều kiện sử dụng 4](#_Toc518499739)

[Hình 1 - 7: Hộp thoại chọn nơi cài dặt Android Studio và Android SDK 5](#_Toc518499740)

[Hình 1 - 8: Hộp thoại cài đặt Android Studio 5](#_Toc518499741)

[Hình 1 - 9: Tiến trình cài đặt 6](#_Toc518499742)

[Hình 1 - 10: Hộp thoại cài đặt hoàn tất 6](#_Toc518499743)

[Hình 1 - 11: Hộp thoai cài đặt hoàn tất 7](#_Toc518499744)

[Hình 1 - 12: Hộp thoại import Android Studio cũ 7](#_Toc518499745)

[Hình 1 - 13: Hộp thoại tải một số thành phần trong Android Studio 8](#_Toc518499746)

[Hình 1 - 14: Hộp thoại tải hoàn thành 9](#_Toc518499747)

[Hình 1 - 15: Logo Java 9](#_Toc518499748)

[Hình 1 - 16: Logo Framework OpenCV 12](#_Toc518499749)

[Hình 1 - 17: Tải Framework OpenCV 13](#_Toc518499750)

[Hình 1 - 18: Hộp thoại chọn đường dẫn lưu OpenCV 13](#_Toc518499751)

[Hình 1 - 19: Hộp thoại giải nén 14](#_Toc518499752)

[Hình 1 - 20: Nơi lưu trữ 2 folder Build & Sources 14](#_Toc518499753)

[Hình 1 - 21: Hộp thoại tạo project trên Visual Studio 15](#_Toc518499754)

[Hình 1 - 22: Cấu trúc cấu hình project 15](#_Toc518499755)

[Hình 1 - 23: Hộp thoại cấu hình project C/C++ 16](#_Toc518499756)

[Hình 1 - 24: Hộp thoại cấu hình project Linker 17](#_Toc518499757)

[Hình 1 - 25: Thêm thư viện 17](#_Toc518499758)

[Hình 1 - 26: Hộp thoại sau khi thêm thư viện 18](#_Toc518499759)

[Hình 1 - 27: Kết quả chạy chương trình 1 20](#_Toc518499760)

[Hình 1 - 28: Kết quả chạy chương trình 2 21](#_Toc518499761)

[Hình 1 - 29: Logo Luxand FaceSDK 21](#_Toc518499762)

[Hình 1 - 30: Download Luxand FaceSDK 25](#_Toc518499763)

[Hình 1 - 31: Hộp thoại Welcome to Luxand FaceSDK 25](#_Toc518499764)

[Hình 1 - 32: Hộp thoại thỏa thuận cấp phép sử dụng Luxand FaceSDK 26](#_Toc518499765)

[Hình 1 - 33: Hộp thoại chọn nơi lưu Luxand FaceSDK 27](#_Toc518499766)

[Hình 1 - 34: Hộp thoại cài đặt Luxand FaceSDK 27](#_Toc518499767)

[Hình 1 - 35: Hộp thoại diễn ra quá trình cài đặt 28](#_Toc518499768)

[Hình 1 - 36: Giao diện Luxand FaceSDK. 29](#_Toc518499769)

[Hình 1 - 37: Ví dụ đơn giản về histogram 34](#_Toc518499770)

[Hình 1 - 38: Ảnh minh họa histogram 35](#_Toc518499771)

[Hình 1 - 39: Histogram của ảnh 35](#_Toc518499772)

[Hình 1 - 40: Ví dụ minh họa về cách tính LBP và độ tương phản trực giao (C) 37](#_Toc518499773)

[Hình 1 - 41: Ảnh có kích thước 4x4 38](#_Toc518499774)

[Hình 1 - 42: Tách hình mẫu theo bán kính 3x3 38](#_Toc518499775)

[Hình 1 - 43: Kết quả sau khi tính giá trị nhị phân 39](#_Toc518499776)

[Hình 1 - 44: Kết quả rút trích đặc trưng LBP 39](#_Toc518499777)

[Hình 1 - 45: Ví dụ về KNN 41](#_Toc518499778)

[Hình 1 - 46: Bộ dữ liệu CĐTH 16A 43](#_Toc518499779)

[Hình 1 - 47: Bộ dữ liệu mẫu 44](#_Toc518499780)

[Hình 1 - 48: Support Vector Machine 46](#_Toc518499781)

[Hình 1 - 49: Các mặt phân cách hai classes lineary separable 48](#_Toc518499782)

[Hình 1 - 50: Margin của hai classes là bằng nhau và lớn nhất có thể 49](#_Toc518499783)

[Hình 1 - 51: Bộ dữ liệu CĐTH 16A 50](#_Toc518499784)

[Hình 1 - 52: Bộ dữ liệu mẫu 51](#_Toc518499785)

CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ DỮ LIỆU

[Hình 2 - 1: Mô hình cơ sở dữ liệu. 63](#_Toc518255343)

CHƯƠNG 3: CÀI ĐẶT GIAO DIỆN VÀ CHƯƠNG TRÌNH

[Hình 3 - 1: Cấu trúc thư mục đồ án 65](#_Toc518499711)

[Hình 3 - 2: Nút tạo lớp học 66](#_Toc518499712)

[Hình 3 - 3: Giao diện thêm lớp học 67](#_Toc518499713)

[Hình 3 - 4: Cách xóa lớp học 68](#_Toc518499714)

[Hình 3 - 5: Giao diện xóa lớp học 69](#_Toc518499715)

[Hình 3 - 6: Nút đổi tên lớp học 70](#_Toc518499716)

[Hình 3 - 7: Giao diện đổi tên lớp học 71](#_Toc518499717)

[Hình 3 - 8: Giao diện danh sách lớp học 72](#_Toc518499718)

[Hình 3 - 9: Bước 1 tạo ca điểm danh 73](#_Toc518499719)

[Hình 3 - 10: Bước 2 đặt giờ kết thúc 74](#_Toc518499720)

[Hình 3 - 11: Bước 3 chọn giờ 75](#_Toc518499721)

[Hình 3 - 12: Bước 4 xác nhận. 76](#_Toc518499722)

[Hình 3 - 13: Giao diện tạo ca điểm danh hoàn tất. 77](#_Toc518499723)

[Hình 3 - 14: Nút hủy ca điểm danh 78](#_Toc518499724)

[Hình 3 - 15: Giao diện hủy ca điểm danh. 79](#_Toc518499725)

[Hình 3 - 16: Giao diện nhận diện khuôn mặt. 80](#_Toc518499726)

[Hình 3 - 17: Thống kê tỉ lệ có mặt theo thời gian điểm danh và ai vắng nhiều nhất 81](#_Toc518499727)

[Hình 3 - 18: Danh sách học viên của lớp 82](#_Toc518499728)

[Hình 3 - 19: Thống kê tỉ lệ có mặt trong các lần điểm danh 83](#_Toc518499729)

[Hình 3 - 20: Thống kê tỉ lệ có mặt vào các thứ trong tuần 84](#_Toc518499730)

[Hình 3 - 21: Xem lại các ca điểm danh trước đó 85](#_Toc518499731)

[Hình 3 - 22: Nút xuất file Excel 86](#_Toc518499732)

[Hình 3 - 23: Giao diện lịch sử điểm danh 87](#_Toc518499733)

**CÁC BẢNG DỮ LIỆU**

[Bảng 1: Thử nghiệm KNN với bộ dữ liệu CĐTH 16A 43](#_Toc518320097)

[Bảng 2: Thủ nghiệm KNN với bộ dữ liệu mẫu 44](#_Toc518320098)

[Bảng 3: Thử nghiệm SVM với bộ dữ liệu CĐTH 16A 50](#_Toc518320099)

[Bảng 4: Thử nghiệm SVM với bộ dữ liệu mẫu 51](#_Toc518320100)

[Bảng 5: Bảng LOP\_HOC 61](#_Toc518320101)

[Bảng 6: Bảng HOC\_VIEN. 62](#_Toc518320102)

[Bảng 7: Bảng DIEM\_DANH 62](#_Toc518320103)

[Bảng 8: Bảng CHI\_TIET\_DIEM\_DANH 62](#_Toc518320104)

[Bảng 9: Bảng LICH\_SU\_DIEM\_DANH 63](#_Toc518320105)

**DANH MỤC CÁC KÝ TỰ, CÁC TỪ VIẾT TẮT**

|  |  |  |
| --- | --- | --- |
| **STT** | **Từ Viết Tắt** | **Ý Nghĩa** |
| 1 | SDK | Software Development Kit |
| 2 | JDK | Java Development Kit |
| 3 | RMI | Remote Method Invocation |
| 4 | JVM | Java Virtual Machine – Máy ảo Java |
| 5 | WWW | World Wide Web |
| 6 | AI | Artificial Intelligence – Trí tuệ nhân tạo |
| 7 | Transition | Chuyển đổi |
| 8 | Uniform | Đồng dạng |
| 9 | LBP | Local Binary Patterns |
| 10 | KNN | K-Nearest Neighbors |
| 11 | SVM | Support Vector Machines |
| 12 | OpenCV | Open Computer Vision |
| 13 | API | Application Programming Interface |
| 14 | Lazy Learning | Lười học |
| 15 | Overfitting | Quá mức |
| 16 | Classification | Phân loại |
| 17 | Regression | Hồi quy |
| 18 | Label | Nhãn |
| 19 | Euclid | Khoảng cách giữa 2 điểm trong không gian |
| 20 | PLA | Perceptron Learning Algorithm – Là một thuật toán phân loại đơn giản trong Machine Learning |
| 21 | Machine Learning | Máy học |
|  |  |  |

# **CHƯƠNG 1: GIỚI THIỆU**

## **Một số công nghệ được ứng dụng vào đồ án.**

* + 1. **Môi trường lập trình Android studio.**
       1. Giới thiệu về Android Studio.

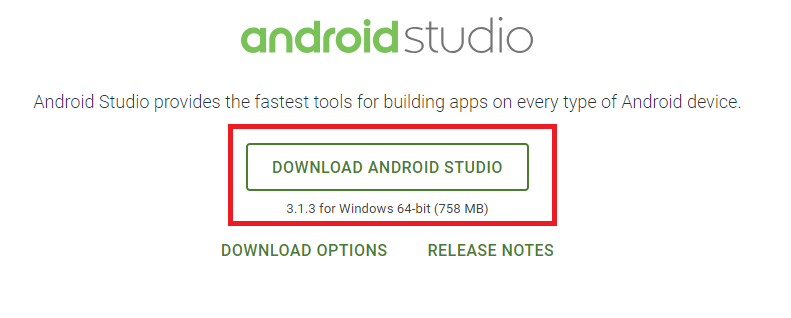
****

Hình 1 - 1: Logo Android Studio

**Android Studio** là một phần mềm bao gồm các bộ công cụ khác nhau dùng để phát triễn ứng dụng chạy trên thiết bị sử dụng hệ điều hành Android như các loại điện thoại smartphone, các tablet... Android Studio được đóng gói với một bộ code editor, debugger, các công cụ performance tool và một hệ thống build/deploy (trong đó có trình giả lập simulator để giả lập môi trường của thiết bị điện thoại hoặc tablet trên máy tính) cho phép các lập trình viên có thể nhanh chóng phát triển các ứng dụng từ đơn giản tới phức tạp.

Việc xây dựng một ứng dụng mobile (hoặc tablet) bao gồm rất nhiều các công đoạn khác nhau. Đầu tiên chúng ta sẽ viết code ứng dụng sử dụng máy tính cá nhân hoặc laptop. Sau đó chúng ta cần build ứng dụng để tạo file cài đặt. Sau khi build xong thì chúng ta cần copy file cài đặt này vào thiét bị mobile (hoặc table) để tiến hành cài đặt ứng dụng và chạy kiểm thử (testing). Bạn thử tưởng tượng nếu với mỗi lần viết một dòng code bạn lại phải build ứng dụng, cài đặt trên điện thoại hoặc tablet và sau đó chạy thử thì sẽ vô cùng tốn thời gian và công sức. Android Studio được phát triển để giải quyết các vấn đề này. Với Android Studio tất cả các công đoạn trên được thực hiện trên cùng một máy tính và các quy trình được tinh gọn tới mức tối giản nhất.

* + - 1. Cài đặt Android Studio.
* Tải bộ cài đặt đầy đủ Android Studio bằng đường dẫn sau: <https://developer.android.com/studio/>

****

Hình 1 - 2: Tải Android Studio

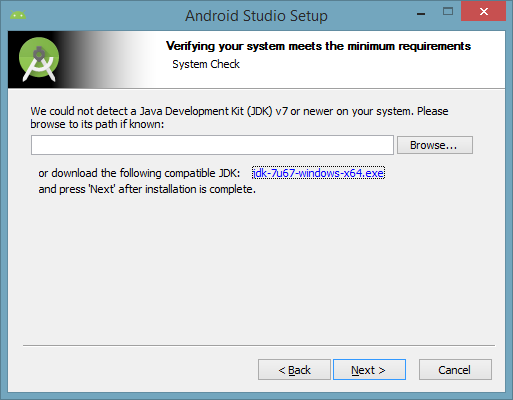
* Tiến hành cài đặt:

Sau khi tải về thành công, bạn chạy tập tin **android-studio-bundle,** hộp thoại Welcome to Setup Android Studio sẽ xuất hiện



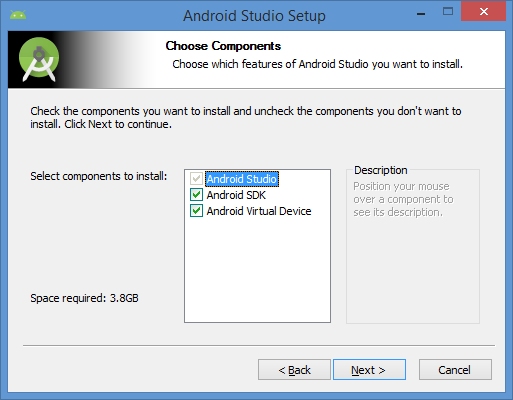
Hình 1 - 3: Hộp thoại Welcome to Android Studio Setup

Nhấn Next để tiếp tục. Nếu hệ thống không phát hiện ra JDK trong máy tính của bạn, một hộp thoại yêu cầu chỉ rõ đường dẫn hoặc cài đặt JDK xuất hiện.



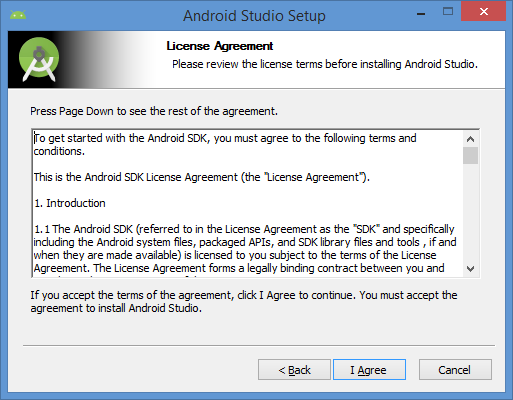
Hình 1 - 4: Hộp thoại yêu cầu JDK

Nếu đã có JDK, bạn vui long trỏ đường dẫn đến nơi cài đặt. Sau đó nhấn Next. Hộp thoại lựa chọn cấu hình cài đặt sẽ mở ra.



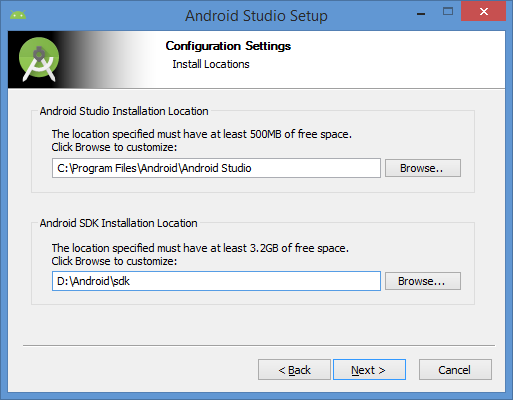
Hình 1 - 5: Hộp thoại cấu hình cài đặt

Bạn vui lòng kiểm tra như hình trên để tránh thiếu sót về sau. Sau đó nhấn Next để tiếp tục. Hộp loại thông báo các điều khoản và một số lưu ý khi sử dụng Android Studio xuất hiện.



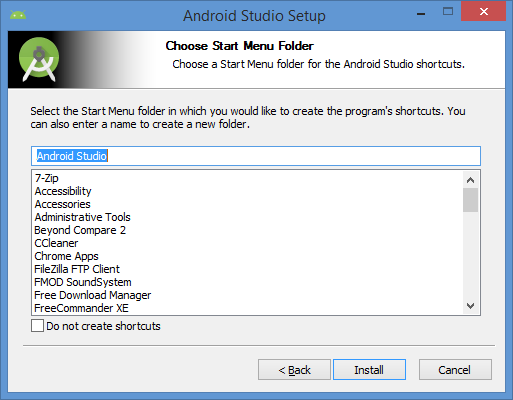
Hình 1 - 6: Hộp thoại thông báo cáo điều kiện sử dụng

Nếu có thể, bạn nên đọc kỹ các điều khoản, sau đó nhấn I Agree để tiếp tục. Hộp thoại yêu cầu chọn nơi cài đặt Android Studio và Android SDK xuất hiện.



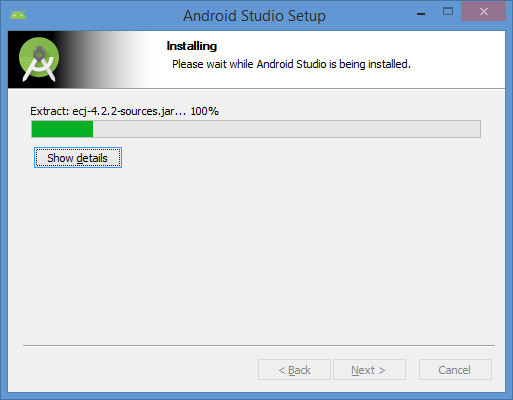
Hình 1 - 7: Hộp thoại chọn nơi cài dặt Android Studio và Android SDK

Bạn nên để đường dẫn mặc định như trên, sau đó nhấn Next để tiếp tục quá trình cài đặt. Hộp thoại chọn Start Menu xuất hiện, chọn Android Studio và nhấn vào Installl để tiếp tục quá trình cài đặt.



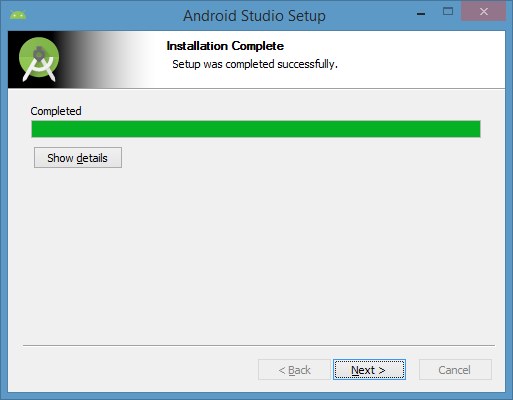
Hình 1 - 8: Hộp thoại cài đặt Android Studio

Chờ đợi quá trình cài đặt.



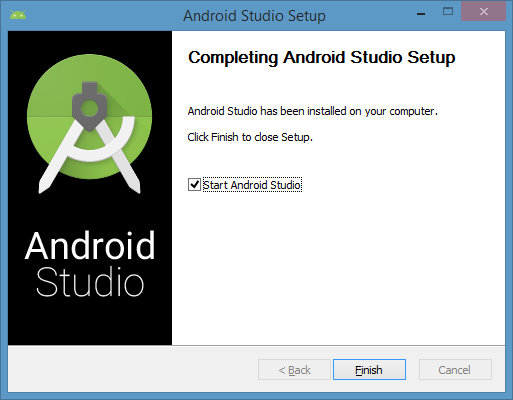
Hình 1 - 9: Tiến trình cài đặt

Khi quá trình cài đặt hoàn tất, nhấn Next để tiếp tục.



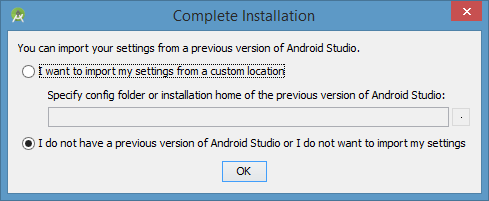
Hình 1 - 10: Hộp thoại cài đặt hoàn tất

Hộp thoại cuối cùng của quá trình cài đặt xuất hiện.



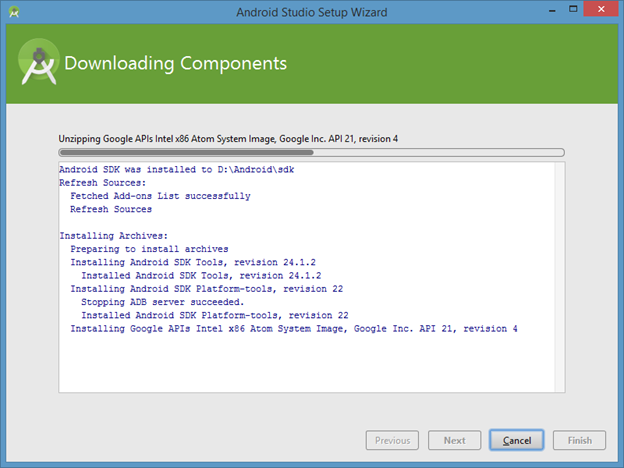
Hình 1 - 11: Hộp thoai cài đặt hoàn tất

Nhấn check vào Start Android Studio, và Finish.Trong quá trình khởi động, nếu có thông báo thiếu JDK, bạn vui lòng cài JDK 7 bản x64 và khởi động lại Android Studio.Ở lần khởi động đầu tiên, một hộp thoại import bản Android Studio cũ xuất hiện, nếu bạn cài đặt mới, vui lòng chọn như hình dưới.



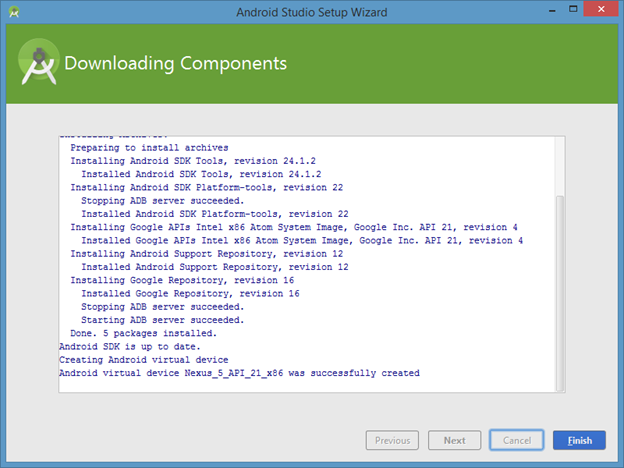
Hình 1 - 12: Hộp thoại import Android Studio cũ

Nhấn OK và tiếp tục. Ở lần đầu tiên, bạn sẽ phải update một vài thứ cho quá trình làm việc sau này. Vui lòng chờ đợi cho đến khi hoàn tất.



Hình 1 - 13: Hộp thoại tải một số thành phần trong Android Studio

Đợi một lúc và nhấn tiếp Finish. Quá trình download file bắt đầu.

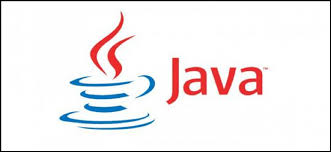


Hình 1 - 14: Hộp thoại tải hoàn thành

Khi hoàn tất quá trình download, nhấn Finish để kết thúc. Vậy là bạn đã hoàn thành quá trình cái đặt.

* + 1. **Ngôn ngữ lập trình Java.**

#### Giới thiệu, lịch sử phát triển Java.



Hình 1 - 15: Logo Java

Năm 1990, Sun MicroSystems thực hiện dự án Green nhằm phát triển phần mềm trong các thiết bị dân dụng. James Gosling, chuyên gia lập trình đã tạo ra một ngôn ngữ lập trình mới có tên là Oak. Ngôn ngữ này có cú pháp gần giống như C++ nhưng bỏ qua các tính năng nguy hiểm của C++ như truy cập trực tiếp tài nguyên hệ thống, con trỏ, định nghĩa chồng các tác tử…

Khi ngôn ngữ Oak trưởng thành, WWW cũng đang vào thời kỳ phát triển mạnh mẽ, Sun cho rằng đây là một ngôn ngữ thích hợp cho Internet. Năm 1995, Oak đổi tên thành Java và sau đó đến 1996  Java đã được xem như một chuẩn công nghiệp cho Internet.

#### Khả năng của ngôn ngữ Java

Là một ngôn ngữ bậc cao như C, C++, Perl, SmallTalk,.. cho nên có thể được dùng để tạo ra các ứng dụng để giải quyết các vấn đề về số, xử lý văn bản, tạo ra trò chơi, và nhiều thứ khác.Có các môi trường lập trình đồ họa như  Visual Java, Symantec Cafe, Jbuilder, Jcreator, ...Có khả năng truy cập dữ liệu từ xa thông qua cầu nối JDBC (Java DataBase Connectivity). Hỗ trợ các lớp hữu ích, tiện lợi trong lập trình các ứng dụng mạng (Socket) cũng như truy xuất Web. Hỗ trợ lập trình phân tán  (Remote Method Invocation ) cho phép một ứng dụng có thể được xử lý phân tán trên các máy tính khác nhau vaà luôn được bổ sung các tính năng cao cấp khác trong các phiên bản sau.

#### Những đặc điểm của ngôn ngữ Java

Ngôn ngữ hoàn toàn hướng đối tượng.Ngôn ngữ đa nền cho phép một chương trình có thể thực thi trên các hệ điều hành khác nhau (MS Windows, UNIX, Linux) mà không phải biên dịch lại chương trình. Phương châm của java là **"Viết một lần ,  Chạy trên nhiều nền"**  (Write Once, Run Anywhere). Ngôn ngữ đa luồng, cho phép trong một chương trình có thể có nhiều luồng điều khiển được thực thi song song nhau, rất hữu ích cho các xử lý song song. Ngôn ngữ phân tán, cho phép các đối tượng của một ứng dụng được phân bố và thực thi trên các máy tính khác nhau. Ngôn ngữ động, cho phép mã lệnh của một chương trình được tải từ một máy tính về máy của người yêu cầu thực thi chương trình. Ngôn ngữ an toàn, tất cả các thao tác truy xuất vào các thiết bị vào ra đều thực hiện trên máy ảo nhờ đó hạn chế các thao tác nguy hiểm cho máy tính thật.Ngôn ngữ đơn giản, dễ học, kiến trúc chương trình đơn giản, trong sáng.

#### Máy ảo Java (JVM - Java Virtual Machine)

Để đảm bảo tính đa nền, Java sử dụng cơ chế **Máy ảo của Java**. ByteCode đó là ngôn ngữ máy của Máy ảo Java tương tự như các lệnh nhị phân của các máy tính thực. Một chương trình sau khi được viết bằng ngôn ngữ Java (có phần mở rộng là .java) phải được biên dịch thành tập tin thực thi được trên máy ảo Java (có phần mở rộng là .class).  Tập tin thực thi này chứa các chỉ thị dưới dạng mã Bytecode mà máy ảo Java hiểu được phải làm gì.

Khi thực hiện một chương trình, máy ảo Java lần lượt thông dịch các chỉ thị dưới dạng Bytecode thành các chỉ thị dạng nhị phân của máy tính thực và thực thi thực sự chúng trên máy tính thực.

Máy ảo thực tế đó là một chương trình thông dịch. Vì thế các hệ điều hành khác nhau sẽ có các máy ảo khác nhau. Để thực thi một ứng dụng của Java trên một hệ điều hành cụ thể, cần phải cài đặt máy ảo tương ứng cho hệ điều hành đó.

#### Hai kiểu ứng dụng dưới ngôn ngữ java:

Khi bắt đầu thiết kế một ứng dụng dưới ngôn ngữ Java, bạn phải chọn kiểu cho nó là **Application** hay **Applet**.

**Applet:** Là một chương trình ứng dụng được nhúng vào các trang web. Mã của chương trình được tải về máy người dùng từ Web server khi người dùng truy xuất đến trang web chứa nó.

**Application:** Là một chương trình ứng dụng được thực thi trực tiếp trên các máy ảo của Java.

#### Bộ phát triển ứng dụng Java (JDK- Java Development Kit)

JDK là một bộ công cụ cho phép người lập trình phát triển và triển khai các ứng dụng bằng ngôn ngữ java được cung cấp miễn phí bởi công ty JavaSoft (hoặc Sun). Có các bộ JDK cho các hệ điều hành khác nhau. Các ấn bản của JDK không ngừng được phát hành.

* + - 1. Bộ công cụ này gồm các chương trình thực thi đáng chú ý sau:

**javac**: Chương trình biên dịch các chương trình nguồn viết bằng ngôn ngữ java ra các tập tin thực thi được trên máy ảo Java.

**java**: Đây là chương trình làm máy ảo của Java, thông dịch mã Bytecode của các chương trình kiểu application thành mã thực thi của máy thực.

**appletviewer**: Bộ thông dịch, thực thi các chương trình kiểu applet.

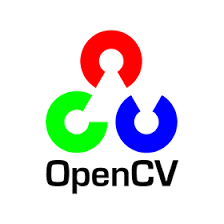
**javadoc**: Tạo tài liệu về chú thích chương trình nguồn một cách tự động.

**jdb**: Trình gở rối.

**rmic**: Tạo Stub cho ứng dụng kiểu RMI.

**rmiregistry**: Phục vụ danh bạ (Name Server) trong hệ thống RMI

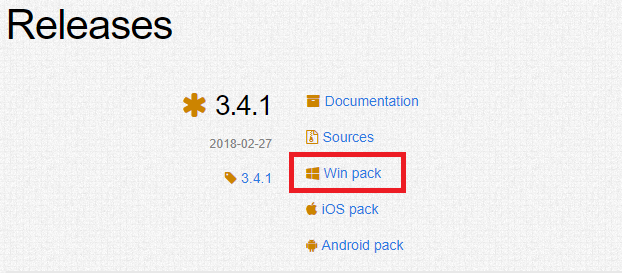
* + 1. **Framework OpenCV**
       1. Giới thiệu Framework OpenCV



Hình 1 - 16: Logo Framework OpenCV

**OpenCV** (Open Computer Vision) là một thư viện mã nguồn mở chuyên dùng để xử lý các vấn đề liên quan đến thị giác máy tính. Nhờ một hệ thống các giải thuật chuyên biệt, tối ưu cho việc xử lý thị giác máy tính, vì vậy tính ứng dụng của OpenCV là rất lớn, có thể kể đến như: Nhận dạng ảnh: nhận dạng khuôn mặt, các vật thể … Xử lý ảnh: khử nhiễu, điều chỉnh độ sáng … Nhận dạng cử chỉ và còn rất nhiều ứng dụng khác nữa.

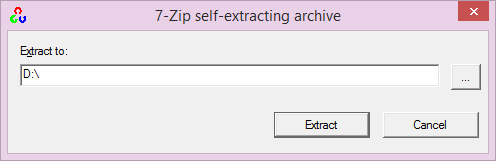
* + - 1. Cài đặt OpenCV
* Tải bộ cài đặt đầy đủ Framework OpenCV tại đường dẫn: <https://opencv.org/releases.html>



Hình 1 - 17: Tải Framework OpenCV

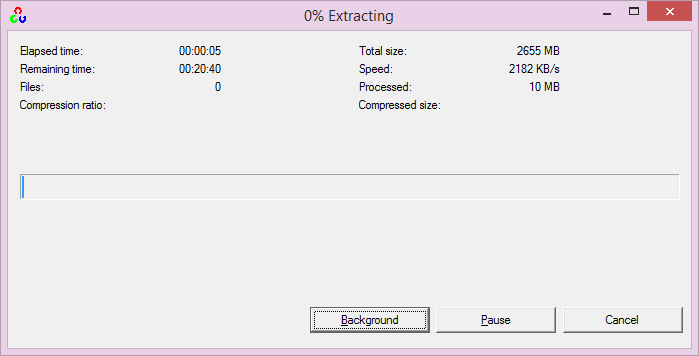
* Cài đặt OpenCV

Thư viện OpenCV phiên bản 3.0 sau khi tải về có dạng **opencv-3.0.0.exe**. Tiến hành chạy file cài đặt, ta chọn đường dẫn để bộ cài đặt giải nén thư viện OpenCV



Hình 1 - 18: Hộp thoại chọn đường dẫn lưu OpenCV

Quá trình giải nén diễn ra

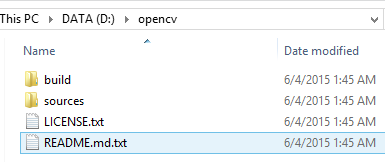


Hình 1 - 19: Hộp thoại giải nén

Sau khi giải nén, ta thu được 2 folder:

**Build**: chứa thư viện OpenCV đã được biên dịch thành thư viện cho nhiều ngôn ngữ khác nhau: C++, Java, Python.

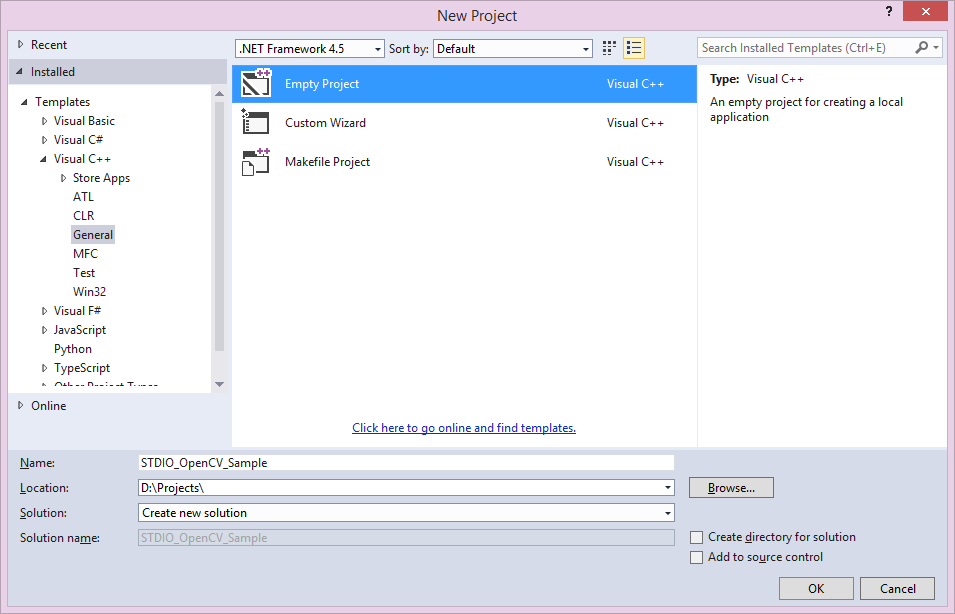
**Sources**: tài liệu hướng dẫn, ví dụ và mã nguồn của OpenCV.



Hình 1 - 20: Nơi lưu trữ 2 folder Build & Sources

* + - 1. Tích hợp Framework OpenCV vào project Visual Studio

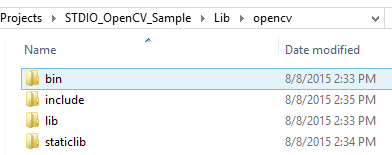
Khởi tạo một project trống và đặt tên là STDIO\_OpenCV\_Sample



Hình 1 - 21: Hộp thoại tạo project trên Visual Studio

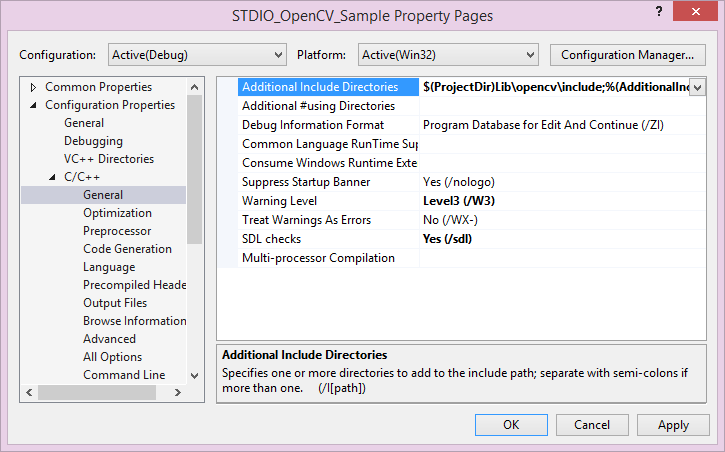
Tại thư mục chứa project, ta tạo folder **Lib/opencv** và copy toàn bộ nội dung ở **<extracted\_path>/build/x86/vc12** và **<extracted\_path>/build/include** vào. Trong đó, **<extracted\_path>** là đường dẫn cài đặt OpenCV ở trên.

Sau khi copy, ta có được cấu trúc folder như sau:



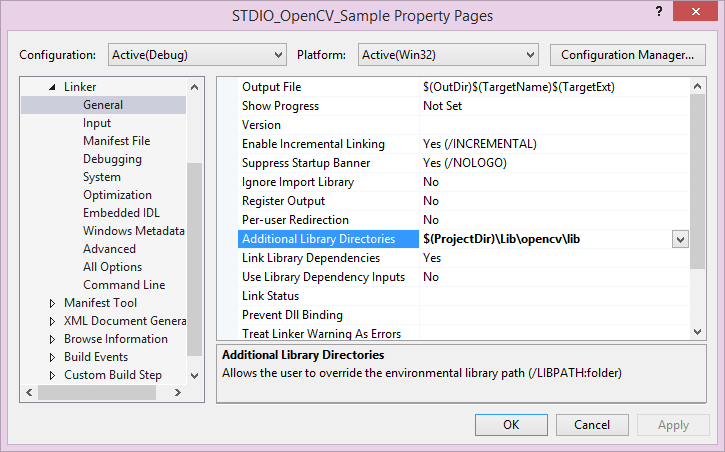
Hình 1 - 22: Cấu trúc cấu hình project

Tại giao diện của Visual Studio, chọn **Project Properties > Configuration Properties > C/C++ > General > Addition Include Directories**, ta thêm đường dẫn **$(ProjectDir)\Lib\opencv\include**



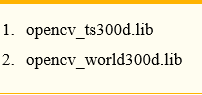
Hình 1 - 23: Hộp thoại cấu hình project C/C++

Tiếp theo, cũng tại trang Project Properties, chọn **Linker > General > Addition Library Directories**, thêm vào đường dẫn **$(ProjectDir)\Lib\opencv\lib**



Hình 1 - 24: Hộp thoại cấu hình project Linker

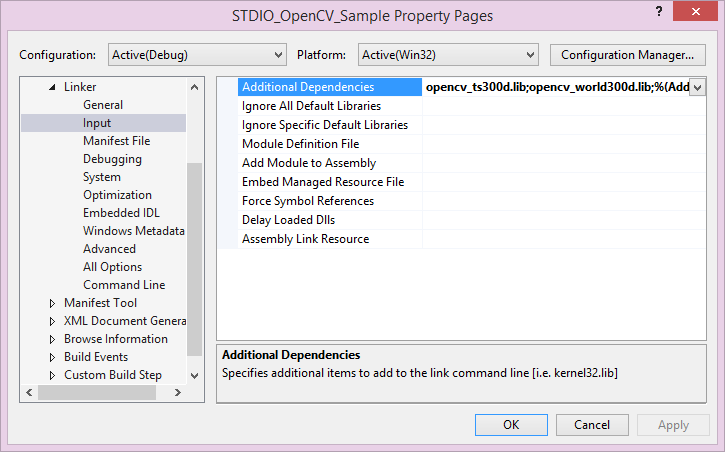
Và tại **Linker > Input**, ta thêm vào các thư viện sau:



Hình 1 - 25: Thêm thư viện

Cuối cùng, ta copy file **opencv\_world300d.dll** trong **Lib\opencv\bin** vào folder **Debug**của project.

Tới đây, ta đã kết thúc việc tích hợp thư viện OpenCV vào project.



Hình 1 - 26: Hộp thoại sau khi thêm thư viện

* + - 1. Ví dụ minh họa Framework OpenCV

Trong ví dụ minh họa này, chương trình của ta có khả năng: Load một ảnh bất kì, đọc vào một đối tượng Mat của OpenCV, hiển thị lên màn hình.

Tạo một file sameple.cpp có nội dung như sau:

1. #include <stdio.h>
2. #include "opencv2/core/core.hpp"
3. #include "opencv2/highgui/highgui.hpp"
5. using namespace cv;
7. int main()
8. {
9. // Read image
10. Mat image = imread("Lena.png", CV\_LOAD\_IMAGE\_COLOR);
12. // Check for valid
13. if (!image.data)
14. {
15. printf("Could not open or find the image\n");
16. return -1;
17. }
19. // Create and show image in window
20. imshow("STDIO OpenCV Sample", image);
22. // Wait input and exit
23. waitKey(0);
25. return 0;
26. }

Dòng 10: Đọc file ảnh lên với hàm imread cùng CV\_LOAD\_IMAGE\_COLOR (đọc ảnh màu). Hàm này trả về một ma trận các pixel màu. Ý nghĩa của các tùy chọn:

**CV\_LOAD\_IMAGE\_UNCHANGED**: load ảnh bình thường (bao gồm cả kênh alpha nếu có)

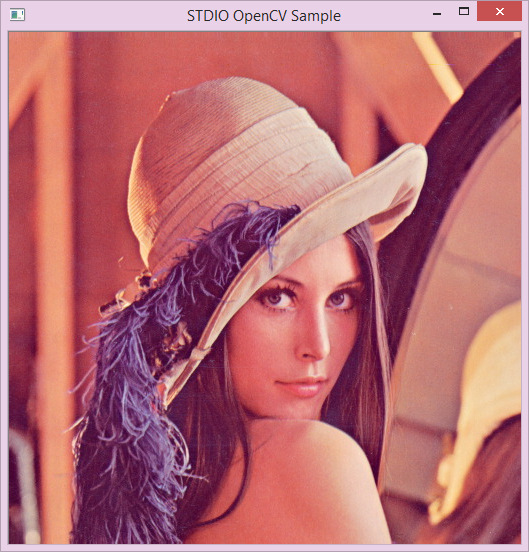
**CV\_LOAD\_IMAGE\_GRAYSCALE**: load ảnh và chuyển về trắng đen

**CV\_LOAD\_IMAGE\_COLOR**: load ảnh phần màu (RGB), bỏ qua kênh alpha

Dòng 13 – 17: kiểm tra việc đọc ảnh có thành công hay không.

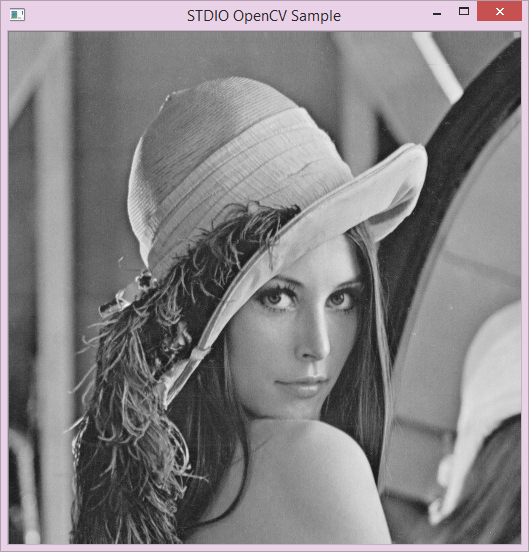
Dòng 20:  Hiển thị ảnh đã load sử dụng hàm imshow với tham số là tiêu đề cửa sổ và nội dung ảnh.

Kết quả của đoạn chương trình trên



Hình 1 - 27: Kết quả chạy chương trình 1

Thử thay đổi tùy chọn tại dòng 10 từ **CV\_LOAD\_IMAGE\_COLOR** thành **CV\_LOAD\_IMAGE\_GRAYSCALE** thì ta được kết quả như sau:



Hình 1 - 28: Kết quả chạy chương trình 2

* + 1. **Phát hiện và nhận diện khuôn mặt với Luxand FaceSDK.**
       1. Giới thiệu Luxand FaceSDK



Hình 1 - 29: Logo Luxand FaceSDK

Luxand FaceSDK là một thư viện hỗ trợ đa nền tảng, dễ dàng tích hợp vào ứng dụng, hỗ trợ về lĩnh vực nhận dạng, nhận diện khuôn mặt. Luxand FaceSDK cung cấp API giúp các nhà phát triễn ứng dụng phát hiện, theo dõi và các biểu hiện trên khuôn mặt để nhận biết giới tính, tuổi tác, nét mặt (đang cười, mắt nhắm lại hay đang mở ra) và nhận diện khuôn mặt trên ảnh tĩnh hoặc là trên video (nhận diện với thời gian thực – Live Face Recognition).

FaceSDK cung cấp API Tracker cho phép theo dõi và nhận dạng khuôn mặt trong video trực tiếp. API theo dõi đơn giản hóa hoạt động với các luồng video, cung cấp các chức năng để gắn thẻ các chủ đề có tên và nhận ra chúng

FaceSDK cung cấp tọa độ của 70 điểm dặc trưng trên khuôn mặt (bao gồm mắt, lông mày, miệng, mũi và đường nét mặt). Luxand FaceSDK sử dụng nhiều lỗi xử lý để tăng tốc độ nhận dạng, giúp hiêu suất của các ứng dụng được cải thiện rất nhiều.

Luxand FaceSDK là một thư viện hỗ trợ đa nên tảng, liên kết động cho các phiên bản 32 bit và 64 bit của Windown, Linux, hệ điều hành MacOS 64 bit, hệ điều hành di động IOS, Android. FaceSDK có rất nhiều ứng dụng mẫu cho C++ crosoft Visual C ++ 6.0 / 2005/2008/2010/2012/2013/2015/2017, Visual Basic .NET 2005/2008 / 2010+, Microsoft C # .NET 2005/2008 / 2010+, Borland Delphi 6.0 + / 7.0, Netbeans (Java), Xcode 4.2+ (iOS), Eclipse ADT (Android), Android Studio (Android), Visual Basic 6.0 và C ++ Builder 6.0.

* + - 1. Các nền tảng và cấu hình tối thiểu FaceSDK hỗ trợ
* Thư viện FaceSDK hỗ trợ các nền tảng như sau:

Windows XP SP3/2003 SP2/Vista/2008/2012, Windows 7, Windows 8, Windows 10.

Linux (RHEL 5+, CentOS 5+ and other).

Linux/ARMv7 (Raspberry Pi2+).

MacOS 10.7+ x86\_64.

IOS 5.0+, armv7/x86 (iPhone 3GS+, iPad 1+, simulator).

IOS 7.0+, arm64/x86\_64 (iPhone 5S+, iPad Air+, iPad mini retina+, simulator).

Android 4.0+ (platform version 14+), armv7 (armeabi-v7a)/x86.

* Hiêu suất được khuyến khích với bộ xử lý Intel

Yêu cầu hệ thống tối thiểu:

* 1.6 GHz processor
* 256 MB RAM

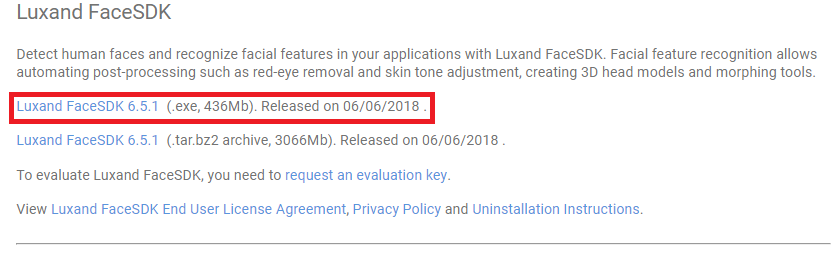
Yêu cầu hệ thống được đề xuất:

* Intel Core i7 or Xeon processor
* 2 GB RAM
* DirectShow-compatible webcam (on Windows)
* IP camera with MJPEG interface (like AXIS IP cameras)
  + - 1. FaceSDK với các thông số kỹ thuật

Thư viện FaceSDK với một vài thông số kỹ thuật như sau:

* Phát hiện khuôn mặt:
  + Phát hiện khuôn mặt nhanh, mạnh mẽ.
  + Phát hiện nhiều khuôn mặt trong một bức ảnh.
  + Hỗ trợ xoay đầu từ -30 độ 🡪 30 độ trong và ngoài mặt phẳng.
  + Xác định góc xoay.
  + Phát hiện khuôn mặt trong thời gian.
  + Thông tin trả về cho mỗi khuôn mặt được phát hiện: (x, y) tọa độ của mặt trung tâm, chiều rộng khuôn mặt và góc quay
  + Dễ dàng cấu hình các thông số phát hiện khuôn mặt
* Nhận dạng video trực tiếp với Tracker API
  + Chỉ định ID duy nhất cho mỗi khuôn mặt trong video.
  + Cho phép gắn thẻ bất kỳ khuôn mặt nào có trong video.
  + Học liên tục những đặc trưng của mỗi khuôn mặt.
  + Cung cấp tỉ lệ chấp nhận sai và tỷ lệ nhận dạng khuôn mặt.
  + Theo dõi khuôn mặt và đặc điểm của chúng.
  + Nhận biết giới tính nam nữ.
  + Nhận dạng tuổi.
  + Nhận dạng nét mặt.
* Phát hiện đặc trưng khuôn mặt.
  + Phát hiện 70 đặc trưng trên khuôn mặt (mắt, lông mày, miệng, mũi, đường viền khuôn mặt).
  + Cho phép xoay đầu: -30 độ 🡪 30 độ quay trong mặt phẳng, xoay -20 độ 🡪 20 độ bên ngoài mặt phẳng.
  + Thông tin trả về: Mảng tọa độ (x,y) gồm 70 đặc trưng trên khuôn mặt.
* Phát hiện vị trí trung tâm của mắt.
  + Thông tin trả về: hai tọa độ (x,y) là trung tâm của mắt trái và mắt phải.
* Nhận thức giới tính.
  + Công nhận giới tính khác nhau.
  + Thông tin trả về: Mức độ tin cậy của mỗi giới tính.
* Nhận dạng tuổi.
  + Công nhận độ tuổi
  + Thông tin trả về: Số tuổi của một người.
* Nhận dạng biểu hiện khuôn mặt.
  + Nhận ra khuôn mặt đang cười, hoặc mắt nhắm hay mở.
* Hỗ trợ đa lõi
  + Thư viện sử dụng tất cả các lõi bộ xử lý có sẵn khi thực hiện các chức năng dò tìm khuôn mặt hoặc nhận dạng, để tối đa hóa hiệu suất.
    - 1. Cài đặt Luxand FaceSDK.
* Tải bộ cài đặt Luxand FaceSDK tại đường dẫn:

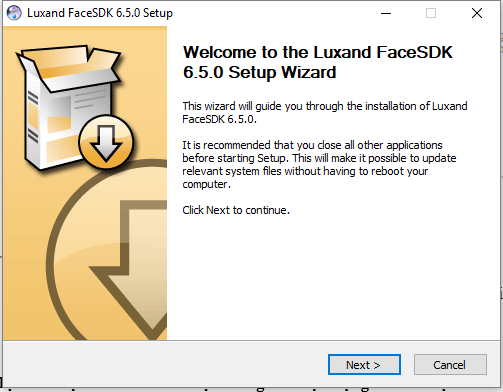
<https://www.luxand.com/download/>



Hình 1 - 30: Download Luxand FaceSDK

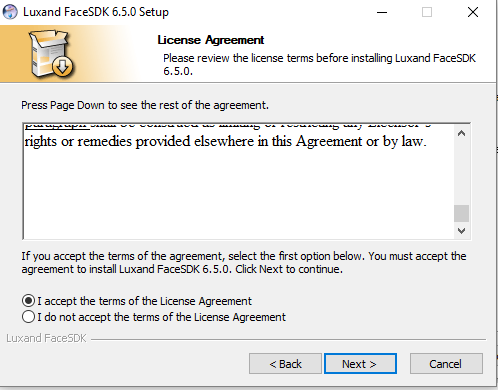
* Cài đặt Luxand FaceSDK.

Sau khi tải hoàn tất, chạy file Luxand\_FaceSDK\_Setup.exe. Hộp thoại Welcome to Luxand FaceSDK xuất hiện.



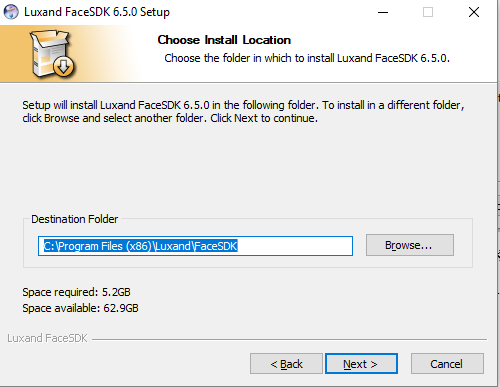
Hình 1 - 31: Hộp thoại Welcome to Luxand FaceSDK

Chọn next để tiếp tục cài đặt, sau đó xuất hiện hộp thoại thỏa thuận cấp phép sử dụng Luxand, chọn **I accept the term of the License Agreement**, chọn next chuyển sang bước cài đặt kế tiếp.



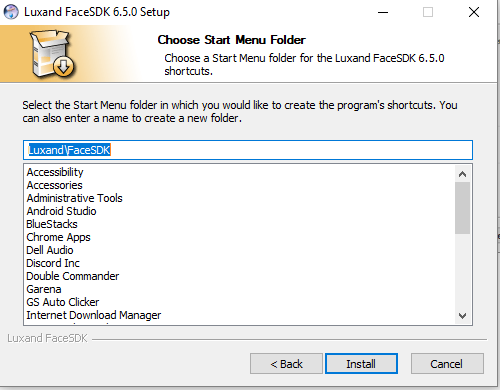
Hình 1 - 32: Hộp thoại thỏa thuận cấp phép sử dụng Luxand FaceSDK

Hộp thoại chọn nơi lưu FaceSDK, giữ nguyên đường dẫn mặc định, chọn next.



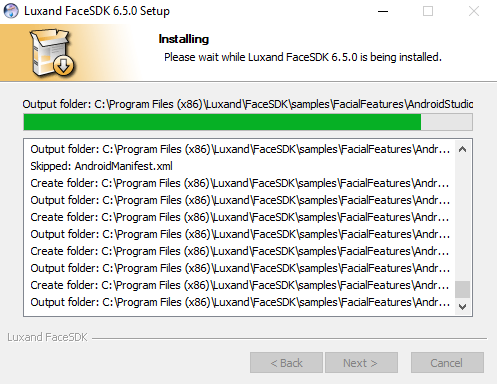
Hình 1 - 33: Hộp thoại chọn nơi lưu Luxand FaceSDK

Hộp thoại cài đặt, nhấn Install và chờ tiến trình cài đặt hoàn tất.



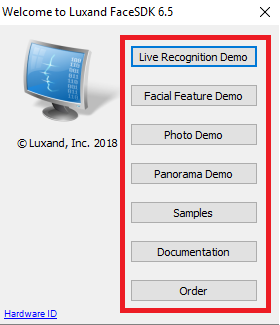
Hình 1 - 34: Hộp thoại cài đặt Luxand FaceSDK

Quá trình cài đặt đang diễn ra.



Hình 1 - 35: Hộp thoại diễn ra quá trình cài đặt

Giao diện sau khi cài đặt hoàn thành Luxand FaceSDK, và một số ứng dụng mẫu.



Hình 1 - 36: Giao diện Luxand FaceSDK.

#### Cấu trúc thư mục Luxand FaceSDK

Thư mục FaceSDK chứa các thư mục và tệp sau:

|  |  |
| --- | --- |
| bin\ | FaceSDK binary files |
| bin\android | FaceSDK Android binaries |
| bin\iOS | FaceSDK iOS binaries |
| bin\linux\_armv7 | FaceSDK Linux/ARMv7 binaries |
| bin\linux\_x86 | FaceSDK Linux 32-bit binaries |
| bin\linux\_x86\_64 | FaceSDK Linux 64-bit binaries |
| bin\osx\_x86\_64 | FaceSDK macOS 64-bit binaries |
| bin\vcredist | Visual C++ 2017 redistributables (used by FaceSDK.NET.dll) |
| bin\win32 | FaceSDK Widows 32-bit binaries and stub library files |
| bin\win64 | FaceSDK Windows 64-bit binaries and stub library files |
| demo\ | Demo applications (win32) |
| include\ | Header files |
| samples\ | Sample applications |

#### Một số ứng dụng mẫu FaceSDK

* Luxand FaceSDK với một số ứng dụng mẫu sau đây (có thể tìm thấy chúng ở FaceSDK samples\ directory).
  + Live Recognition: Ứng dụng này nhận một video từ camera, cho phép gắn thẻ bấy kỳ khuôn mặt nào bằng tên, và hiển thị tên đó (nhận dạng khuôn mặt).
  + Face Tracking: Ứng dụng này nhận một video từ camera và làm nổi bật tất cả các khuôn mặt được phát hiện với hình chữ nhật.
  + Lookalikes: Ứng dụng này cho phép người dùng tạo cơ sở dữ liệu chứa những khuôn mặt và tìm kiếm khuôn mặt phù hợp nhất (khuôn mặt tương tự nhất từ ​​cơ sở dữ liệu được hiển thị)
  + Live Facial Features: Ứng dụng này theo dõi cái tính năng của khuôn mặt trong thời gian thực bằng camera.
  + Age Gender Recognition: Sử dụng API tracker, ứng dụng này giúp nhận ra giới tính và độ tuổi của một người tìm kiếm từ một webcam.
  + Expression Recognition: Từ webcam ứng dụng này có thể nhận biết khuôn mặt đang có biển hiện mỉm cười, mắt nhắm hay mở.
  + Facial Features: Ứng dụng này mở một tấm ảnh và phát hiện khuôn mặt trong ảnh (chỉ một khuôn mặt được phát hiện tốt nhất) sau đó phát hiện ra những tính năng khuôn mặt và vẽ hình tròn lên chúng.

#### Các tình huống sử dụng

Mức độ sử dụng thư viện phụ thuộc vào chức năng có được từ Luxand FaceSDK. Nếu bạn làm việc với video, hãy cân nhắc sử dụng API Tracker, vì API này cung cấp những tính năng cao cấp giúp nhận dạng khuôn mặt và gắn thẻ chúng với tên, theo dõi khuôn mặt và những đặc điểm của nỏ, nhận ra giới tính, tuổi tác nét mặt…

* Kịch bản sử dụng FaceSDK điển hình
  + Kích hoạt FaceSDK bằng cách gọi hàm FSDK\_ActivateLibrary bằng khóa gửi đến mail từ Luxand.
  + Khởi tạo FaceSDK bằng cách gọi hàm FSDK\_Initialize.
  + Tải hình ảnh từ tệp, bộ đệm hoặc HBITMAP bằng cách sử dụng hàm (FSDK\_LoadImageFromFile, FSDK\_LoadImageFromBuffer,

FSDK\_LoadImageFromHBitmap functions).

* + Đặt thông số phát hiện khuôn mặt nếu cần (FSDK\_SetFaceDetectionParameters, FSDK\_SetFaceDetectionThreshold)
  + Sử dụng các hàm FaceSDK:
    - Phát hiện khuôn mặt (FSDK\_DetectFace) hoặc nhiều khuôn mặt

(FSDK\_DetectMultipleFaces) trong một hình ảnh

* + - Phát hiện các tính năng trên khuôn mặt nếu cần (FSDK\_DetectFacialFeatures, FSDK\_DetectFacialFeaturesInRegion)
    - Trích xuất khuôn mặt từ hình ảnh (FSDK\_GetFaceTemplate, FSDK\_GetFaceTemplateInRegion,

FSDK\_GetFaceTemplateUsingFeatures)

* + - Phù hợp với các mẫu khuôn mặt(FSDK\_MatchFaces) và có được mặt-mức độ tương tự với các khuôn mặt đó.
  + Hoàn thành sử dụng thư viện FaceSDK (hàm SDK\_Finalize).
* Kịch bản làm việc với camera
  + Khởi tạo máy ảnh (FSDK\_InitializeCapturing).
  + Nhận danh sách các máy ảnh trong hệ thống (FSDK\_GetCameraList)
  + Nhận danh sách các định dạng video được máy ảnh hỗ trợ (FSDK\_GetVideoFormatList).
  + Đặt định dạng video cho máy ảnh đã chọn (FSDK\_SetVideoFormat).
  + Mở máy ảnh web (FSDK\_OpenVideoCamera) hoặc camera IP

(FSDK\_OpenIPVideoCamera).

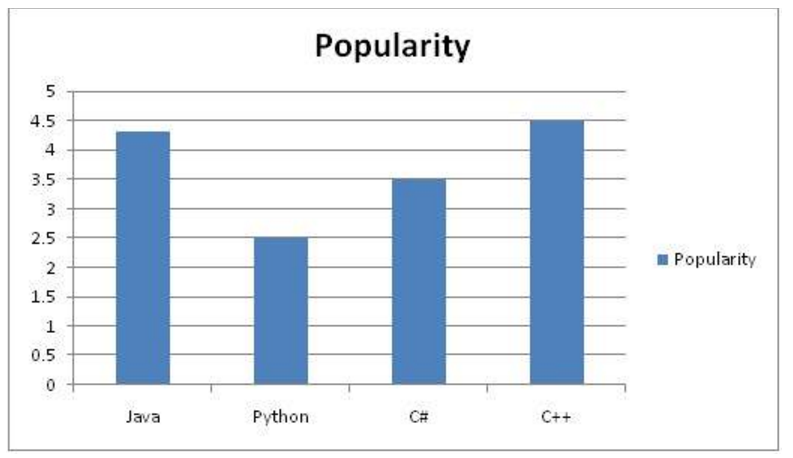
* + Lấy khung ảnh (FSDK\_GrabFrame) trong một vòng lặp, hiển thị chúng và phát hiện / nhận ra khuôn mặt.
  + Đóng máy ảnh (FSDK\_CloseVideoCamera).
  + Xóa danh sách các định dạng video (FSDK\_FreeVideoFormatList).
  + Xóa danh sách máy ảnh (FSDK\_FreeCameraList).
  + Hoàn tất (FSDK\_FinalizeCapturing).
  1. **Một số thuật toán rút trích đặc trưng và nhận dạng khuôn mặt.**
     1. **Giới thiệu một vài nhóm thuật toán Machine Learning**
* Classification (Phân loại): Một bài toán được gọi là classification nếu các label của dữ liệu đầu vào được chia thành một số hữu hạn nhóm. Ví dụ: Gmail xác định xem một email có phải là spam hay không, các hãng tín dụng xác định xem một khách hàng có khả năng thanh toán nợ hay không. hai ví dụ phía trên được chia vào loại này.
* Regression (Hồi quy): Nếu label không được chia thành các nhóm mà là một giá trị thực cụ thể. Ví dụ: một căn nhà rộng x m2, có y phòng ngủ và cách trung tâm thành phố z km sẽ có giá là bao nhiêu? Gần đây [Microsoft có một ứng dụng dự đoán giới tính và tuổi dựa trên khuôn mặt](https://how-old.net/). Phần dự đoán giới tính có thể coi là thuật toán **Classification**, phần dự đoán tuổi có thể coi là thuật toán **Regression**. Chú ý rằng phần dự đoán tuổi cũng có thể coi là ***Classification*** nếu ta coi tuổi là một số nguyên dương không lớn hơn 150, chúng ta sẽ có 150 class (lớp) khác nhau.
  + 1. **Đặc trưng ảnh “ Histogram ”.**
       1. Giới thiệu Histogram.

Histogram là một đồ thị. Histogram thể hiện tần số của mọi thứ. Thông thường Histogram có các thanh đại diện cho tần số xuất hiện của dữ liệu hay nói cách khác là Histogram hiển thị sự phân bố tần số của một tập dữ liệu. Một Histogram có hai trục: trục x và trục y.

* Trục x chứa các điểm có sự phân bố tần số.
* Trục y chứa tần số.

Các độ cao khác nhau của thanh cho thấy tần số khác nhau của sự xuất hiện của dữ liệu.

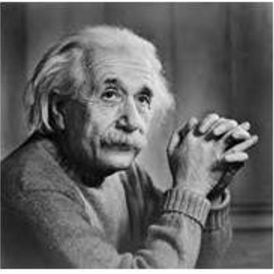
Thông thường, một biểu đồ trông như thế này.



Hình 1 - 37: Ví dụ đơn giản về histogram

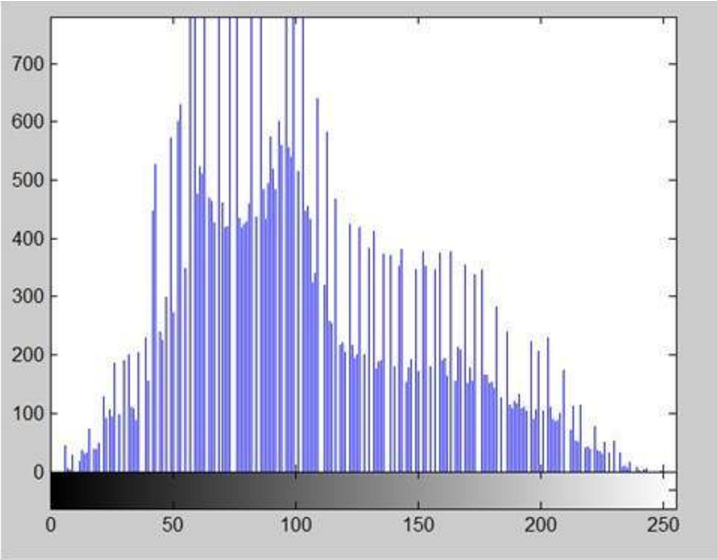
* + - 1. Histogram của ảnh.

Histogram của một hình ảnh cũng giống như histograms khác, nó cũng cho thấy sự phân bố tần số. Nhưng biểu đồ hình ảnh hiển thị tần số của các giá trị cường độ điểm ảnh. Trong một biểu đồ hình ảnh, trục x hiển thị cường độ mức xám và trục y hiển thị tần số của cường độ.



Hình 1 - 38: Ảnh minh họa histogram

Biểu đồ histogram của ảnh trên:



Hình 1 - 39: Histogram của ảnh

Trục x của biểu đồ thể hiện phạm vi của các giá trị pixel. Từ một hình ảnh 8 bpp của nó, có nghĩa là nó có 256 mức xám. Đó là lý do trục x bắt đầu từ 0 và kết thúc tại 255 với một khoảng cách là 50. Trong khi đó, trên trục y thể hiện giá trị cường độ.

Ứng dụng của Histogram: dùng để phân tích hình ảnh, cân bằng một hình ảnh, tạo ngưỡng cho ảnh. Chúng ta có thể dự đoán về một hình ảnh bằng cách chỉ nhìn vào biểu đồ của nó.

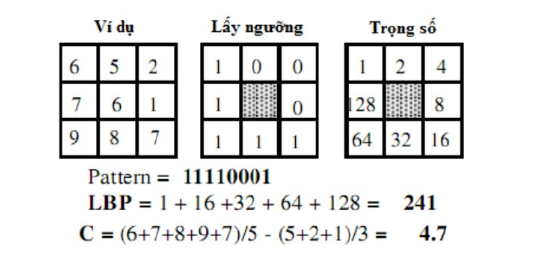
* + 1. **Thuật toán rút trích đặt trưng Local Binary Patterns (LBP).**
       1. Giới thiệu thuật toán rút trích đặt trưng LBP.

Có một số phương pháp để trích xuất các đặc trưng hữu ích nhất từ hình ảnh (được trích xuất trước) để thực hiện chức năng nhận diện khuôn mặt. Một trong số những phương pháp này là phương pháp rút trích đặt trưng LBP hay còn gọi là Local Binary Patterns. Thuật toán được giới thiệu vào năm 1996 bới Ojala. Với LBP, chúng ta có thể mô tả kết cấu và hình dạng của một tấm ảnh kỹ thuật số.

* + - 1. Phân tích toán học.

Phiên bản đầu tiên của LBP được dùng 8 điểm ảnh xung quanh và sử dụng giá trị của điểm ảnh trung tâm làm ngưỡng. Giá trị LBP được xác định bằng cách nhân giá trị ngưỡng với trọng số tương ứng với mỗi điểm ảnh sau đó cộng tổng lại.

Kể từ khi đưa ra, theo định nghĩa là bất biến với những thay đổi đợn điệu tronh ảnh trắng đen. LBP đã bổ sung thêm phương pháp tính độ tương phản trực quan (C) nó là ký hiệu cấp độ xám trung bình của các điểm ảnh lớn hơn hoặc bằng ngưỡng với các điểm ảnh thấp hơn ngưỡng. Phân phối hai chiều của LBP và độ tương phản cục bộ được lấy làm đặc trưng gọi hàm là LBP/C.



Hình 1 - 40: Ví dụ minh họa về cách tính LBP và độ tương phản trực giao (C)

Tiến trình hoạt động của thuật toán LBP:

Bước 1: Xác định bán kính làm việc

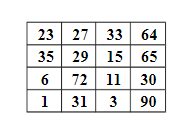
Bước 2: Tính giá trị LBP cho pixel ở trung tâm (xc,yc) khối ảnh dựa trên thông tin của các pixel lân cận.

Trong đó, (gp) là giá trị grayscale của các pixel lân cận, (gc) là giá trị grayscale của các trung tâm và (s) là hàm nhị phân được xác định như sau: s(z) = 1 nếu z ≥ 0.

Một mẫu nhị phân được gọi là đồng dạng (uniform) khi xét chuỗi bit xoay vòng thì có nhiều nhất là hai lần thay đổi (transitions) từ giá trị 0 sang 1 và ngược lại 1 sang 0. Ví dụ: 00000000 có 0 transitions, 01110000 có 2 transitions, 11001111 có 2 transitions nên đây là uniform LBL, 11001001 có 4 transitions, 01010011 có 6 transitions nên đây không phải là uniform LBP.

* + - 1. Thử nghiệm LBP.

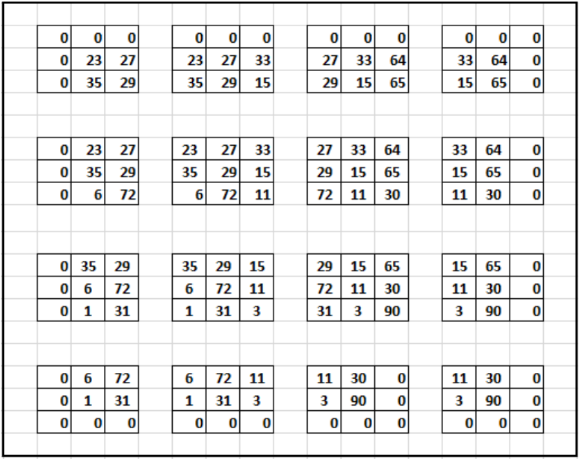
Với một ảnh có kích thước 4x4:



Hình 1 - 41: Ảnh có kích thước 4x4

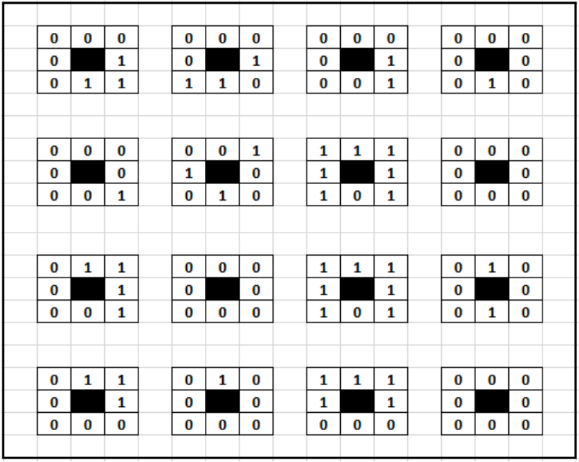
Chúng ta sẽ tính giá trị đặc trưng LBP như sau:

Bước 1: Chọn bán kính làm việc 3x3



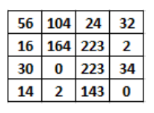
Hình 1 - 42: Tách hình mẫu theo bán kính 3x3

Tính giá trị nhị phân s(z) của tưng pixel trên mỗi khung ảnh kích thước 3x3



Hình 1 - 43: Kết quả sau khi tính giá trị nhị phân

Bước 2: Rút trích đặt trưng LBP



Hình 1 - 44: Kết quả rút trích đặc trưng LBP

* + - 1. Thảo luận.

Ưu điểm: Thuật toán rút trích đặc trưng LBP cài đặt đơn giản, thời gian tính toán giá trị đặc trưng nhanh vì nó làm việc với số nguyện.

Nhược điểm: Độ chính xác không cao.

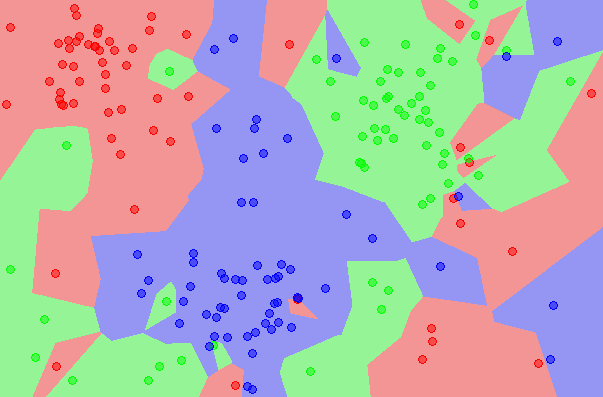
Ứng dụng: Được ứng dụng vào các bài toán nhận diện khuôn mặt.

* + 1. **Thuật toán K-Nearest Neighbors (KNN).**
       1. Giới thiệu thuật toán K-Nearest Neighbors (KNN).

K-nearest neighbor (KNN) là một trong những thuật toán supervised learning đơn giản. Khi huấn luyện, thuật toán này không học một điều gì từ dữ liệu huấn luyện mà nhớ lại một cách máy móc toàn bộ dữ liệu đó. Đây cũng là lý do thuật toán này được xếp vào loại lazy learning, mọi tính toán được thực hiện khi nó cần dự đoán đầu ra của dữ liệu mới. KNN có thể áp dụng được vào cả classification và regression. KNN còn được gọi là một thuật toán instance-based [AKA91] hay memory-based learning.

Với KNN, trong bài toán classification, nhãn của một điểm dữ liệu mới được suy ra trực tiếp từ K điểm dữ liệu gần nhất trong tập huấn luyện. Nhãn đó có thể được quyết định bằng bầu chọn theo đa số (major voting) trong số K điểm gần nhất, hoặc nó có thể được suy ra bằng cách đánh trọng số khác nhau cho mỗi trong các điểm gần nhất đó rồi suy ra kết quả. Chi tiết sẽ được nêu trong phần tiếp theo. Trong bài toán regresssion, đầu ra của một điểm dữ liệu sẽ bằng chính đầu ra của điểm dữ liệu đã biết gần nhất (trong trường hợp K = 1), hoặc là trung bình có trọng số của đầu ra của những điểm gần nhất, hoặc bằng một mối quan hệ dựa trên các điểm gần nhất đó và khoảng cách tới chúng.

Một cách ngắn gọn, KNN là thuật toán đi tìm đầu ra của một điểm dữ liệu mới bằng cách chỉ dựa trên thông tin của K điểm dữ liệu gần nhất trong tập huấn luyện.



Hình 1 - 45: Ví dụ về KNN

Hình 1 - 37 mô tả một bài toán phân lớp với ba class: đỏ, lam, lục. Các hình tròn nhỏ với màu khác nhau thể hiện dữ liệu huấn luyện của các class khác nhau. Các vùng màu nền khác nhau thể hiện lãnh thổ của mỗi class. Tại một điểm bất kỳ, class của nó được xác định dựa trên class của điểm gần nó nhất trong trong tập huấn luyện. Trong hình này, có một vài vùng nhỏ xem lẫn vào các vùng lớn hơn khác màu. Ví dụ có một điểm màu lục ở gần góc 11 giờ nằm giữa hai vùng lớn với nhiều dữ liệu màu đỏ và lam. Điểm này rất có thể là nhiễu. Việc này nhiều khả năng sẽ dẫn đến việc phân lớp sai cho một điểm dữ liệu kiểm thử rơi vào khu vực này.

KNN là một ví dụ rõ nhất của overfitting. Với mô hình này, mọi điểm trong tập huấn luyện đều được mô hình mô tả một cách chính xác, vì vậy, nó rất nhạy cảm với nhiễu.

Mặc dù có nhiều hạn chế, KNN vẫn là một giải pháp đầu tiên nên nghĩ tới khi giải quyết một bài toán machine learning. Khi làm các bài toán machine learning nói chung, không có mô hình đúng hay sai, chỉ có mô hình cho kết quả tốt hơn. Chúng ta luôn cần một mô hình đơn giản để giải quyết bài toán, sau đó dần dần tìm cách tăng chất lượng của mô hình.

* + - 1. **Phân tích toán học**

KNN là một thuật toán lazy learning, không có mất mát hàm và bài toán tối ưu nào phải thực hiện trong quá trình huấn luyện. Mọi tính toán được thực hiện ở bước kiểm thử. Vì KNN ra quyết định dựa trên các điểm gần nhất nên có hai vấn đề ta cần lưu tâm. Thứ nhất, khoảng cách được định nghĩa như thế nào. Thứ hai, cần phải tính toán khoảng cách như thế nào cho hiệu quả.

Với vấn đề thứ nhất, mỗi điểm dữ liệu được thể hiện bằng một vector đặc trưng, khoảng cách giữa hai điểm chính là khoảng cách giữa hai vector đó. Để đo khoảng cách trong không gian nhiều chiều thì đó chính là khoảng cách Euclid quen thuộc.

Ta cần lưu ý tới vấn đề thứ hai hơn, đặc biệt với các bài toán với tập huấn luyện lớn và vector đặc trưng có kích thước lớn (large-scale problem). Giả sử các vector đặc trưng huấn luyện là các cột của ma trận X ∈ Rd×N với d và N lớn, KNN sẽ phải tính toán khoảng cách từ một điểm dữ liệu mới z ∈ Rd đến toàn bộ N điểm dữ liệu đã cho và chọn ra K khoảng cách nhỏ nhất. Nếu không có một cách tính hiệu quả, khối lượng tính toán ở đây sẽ rất lớn. Đây cũng là cái giá phải trả cho việc lazy learning.

Trong bài toán nhận diện khuôn mặt, với một tập train cho trước, ta đưa vào một khuôn mặt để nhận diện (tạm gọi F), kết quả cho ta biết F được gắn với nhãn nào trong tập train.

Để giải quyết bài toán này, ta phải tính và tìm ra khoảng cách gần nhất giữa F và tập train, ta thêm một tham số K = 3 ( K được gọi là ngưỡng với giá trị do người lập trình chọn) với K = 3 ta sẽ tìm ra 3 khoảng cách gần nhất của F so với tập train, sau đó lựa chọn ra khoảng cách nào mà ta thấy ra chính xác nhất bằng cách đếm số lượng loại trong tập K xem loại nào nhiều hơn thì ta chọn. Ta thấy, việc thêm K vào sẽ giúp ta giảm bớt đi tỉ lệ nhận diện sai khi tập train bị nhiễu (xem ví dụ Hình 1 – 37 để hiểu rõ hơn).

* + - 1. Thử nghiệm KNN
* Bộ dữ liệu lớp CĐTH 16A

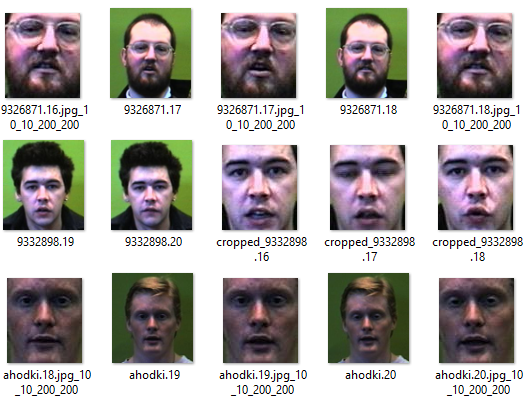
****

Hình 1 - 46: Bộ dữ liệu CĐTH 16A

|  |  |  |  |
| --- | --- | --- | --- |
| **KNN** | **Cách tính** | **Đặc trưng** | **Kết quả** |
| Tính khoảng cách giữa 2 vector | LBP | 55% |
| Histogram | 64% |
| Tìm điểm tương đồng lớn nhất với tất cả các ảnh trong tập train | LBP | 60% |
| Histogram | NA |

Bảng 1: Thử nghiệm KNN với bộ dữ liệu CĐTH 16A

* Bộ dữ liệu mẫu



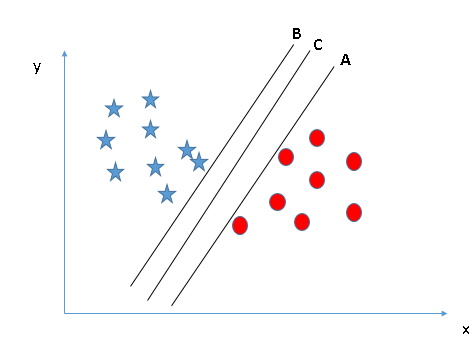
Hình 1 - 47: Bộ dữ liệu mẫu

|  |  |  |  |
| --- | --- | --- | --- |
| **KNN** | **Cách tính** | **Đặc trưng** | **Kết quả** |
| Tính khoảng cách giữa 2 vector | LBP | 70% |
| Histogram | NA |
| Tìm điểm tương đồng lớn nhất với tất cả các ảnh trong tập train | LBP | 90% |
| Histogram | NA |
| Tìm điểm tương đồng lớn nhất với tất cả các ảnh trong các tập hợp | LBP | 60% |
| Histogram | NA |

Bảng 2: Thủ nghiệm KNN với bộ dữ liệu mẫu

* + - 1. Thảo luận
* Ưu điểm KNN
  + Độ phức tạp tính toán của quá trình huấn luyện là bằng 0.
  + Việc dự đoán kết quả của dữ liệu mới rất đơn giản (sau khi đã xác định được các điểm lân cận).
  + Không cần giả sử về phân phối của các class.
* Nhược điểm KNN
  + KNN rất nhạy cảm với nhiễu khi K nhỏ.
  + Như đã nói, KNN là một thuật toán mà mọi tính toán đều nằm ở khâu kiểm thử. Trong đó việc tính khoảng cách tới từng điểm dữ liệu trong tập huấn luyện tốn rất nhiều thời gian, đặc biệt là với các cơ sở dữ liệu có số chiều lớn và có nhiều điểm dữ liệu. Với K càng lớn thì độ phức tạp cũng sẽ tăng lên. Ngoài ra, việc lưu toàn bộ dữ liệu trong bộ nhớ cũng ảnh hưởng tới hiệu năng của KNN.
    1. **Thuật toán Support Vector Machines (SVM).**
       1. Giới thiệu thuât toán SVM

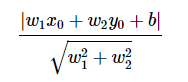
SVM là một thuật toán giám sát, nó có thể sử dụng cho cả việc phân loại hoặc hồi quy. Tuy nhiên nó được sử dụng chủ yếu cho việc phân loại. Trong thuật toán này, chúng ta vẽ đồ thị dữ liệu là các điểm trong n chiều ( ở đây n là số lượng các tính năng bạn có) với giá trị của mỗi tính năng sẽ là một phần liên kết. Sau đó chúng ta thực hiện tìm "đường bay" phân chia các lớp. Đường bay - nó chỉ hiểu đơn giản là 1 đường thằng có thể phân chia các lớp ra thành hai phần riêng biệt, hiểu một cách đơn giản là các đối tượng trên đồ thị tọa độ quan sát. Support Vector Machine là một biên giới để chia hai lớp một cách tốt nhất.



Hình 1 - 48: Support Vector Machine

* + - 1. Phân tích toán học
* Khoảng cách từ một điểm tới một siêu mặt phẳng

Trong không gian 2 chiều, ta biết rằng khoảng cách từ một điểm có tọa độ **(x0,y0)** tới đường thẳng có phương trình **w1x+w2y+b=0** được xác đinh bởi

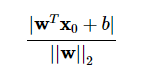


Trong không gian 3 chiều, ta biết rằng khoảng cách từ một điểm có tọa độ **(x0,y0,z0)** tới đường thẳng có phương trình **w1x + w2y + w3z + b = 0** được xác đinh bởi



Hơn nữa, nếu ta bỏ dấu trị tuyệt đối ở tử số, chúng ta có thể xác định được điểm đó nằm về phía nào của đường thẳng hay mặt phẳng đang xét. Những điểm làm cho biểu thức trong dấu giá trị tuyệt đối mang dấu dương nằm về cùng 1 phía (tôi tạm gọi đây là phía dương của đường thẳng), những điểm làm cho biểu thức trong dấu giá trị tuyệt đối mang dấu âm nằm về phía còn lại (tôi gọ là phía âm). Những điểm nằm trên đường thẳng/măt phẳng sẽ làm cho tử số có giá trị bằng 0, tức khoảng cách bằng 0.

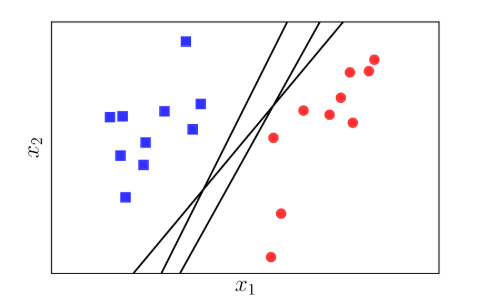
Việc này có thể được tổng quát lên không gian nhiều chiều: Khoảng cách từ một điểm (vector) có toạ độ **x0** tới siêu mặt phẳng (hyperplane) có phương trình wTx + b = 0 được xác định bởi:



Với ||w||2 = với d là số chiều của không gian.

* Nhắc lại bài toán phân chia hai classes

Chúng ta cùng quay lại với bài toán trong Perceptron Learning Algorithm (PLA). Giả sử rằng có hai class khác nhau được mô tả bởi các điểm trong không gian nhiều chiều, hai classes này linearly separable, tức tồn tại một siêu phẳng phân chia chính xác hai classes đó. Hãy tìm một siêu mặt phẳng phân chia hai classes đó, tức tất cả các điểm thuộc một class nằm về cùng một phía của siêu mặt phẳng đó và ngược phía với toàn bộ các điểm thuộc class còn lại. Chúng ta đã biết rằng, thuật toán PLA có thể làm được việc này nhưng nó có thể cho chúng ta vô số nghiệm như Hình 1 – 41dưới đây:



Hình 1 - 49: Các mặt phân cách hai classes lineary separable

Câu hỏi đặt ra là: trong vô số các mặt phân chia đó, đâu là mặt phân chia tốt nhất theo một tiêu chuẩn nào đó? Trong ba đường thẳng minh họa trong

Hình 1 - 41 phía trên, có hai đường thẳng khá lệch về phía class hình tròn đỏ. Điều này có thể khiến cho lớp màu đỏ không vui vì lãnh thổ xem ra bị lấn nhiều quá. Liệu có cách nào để tìm được đường phân chia mà cả hai classes đều cảm thấy công bằng và hạnh phúc nhất hay không?

Chúng ta cần tìm một tiêu chuẩn để đo sự hạnh phúc của mỗi class. Hãy xem Hình 1 - 42 dưới đây:



Hình 1 - 50: Margin của hai classes là bằng nhau và lớn nhất có thể

Nếu ta định nghĩa mức độ hạnh phúc của một class tỉ lệ thuận với khoảng cách gần nhất từ một điểm của class đó tới đường/mặt phân chia, thì ở Hình 1 - 42 trái, class tròn đỏ sẽ không được hạnh phúc cho lắm vì đường phân chia gần nó hơn class vuông xanh rất nhiều. Chúng ta cần một đường phân chia sao cho khoảng cách từ điểm gần nhất của mỗi class (các điểm được khoanh tròn) tới đường phân chia là như nhau, như thế thì mới công bằng. Khoảng cách như nhau này được gọi là margin (lề).

Đã có công bằng rồi, chúng ta cần văn minh nữa. Công bằng mà cả hai đều kém hạnh phúc như nhau thì chưa phải là văn mình cho lắm.

Chúng ta xét tiếp Hình 1 - 42 bên phải khi khoảng cách từ đường phân chia tới các điểm gần nhất của mỗi class là như nhau. Xét hai cách phân chia bởi đường nét liền màu đen và đường nét đứt màu lục, đường nào sẽ làm cho cả hai class hạnh phúc hơn? Rõ ràng đó phải là đường nét liền màu đen vì nó tạo ra một margin rộng hơn.

Việc margin rộng hơn sẽ mang lại hiệu ứng phân lớp tốt hơn vì sự phân chia giữa hai classes là rạch ròi hơn. Việc này, sau này các bạn sẽ thấy, là một điểm khá quan trọng giúp Support Vector Machine mang lại kết quả phân loại tốt hơn so với Perceptron Learning Algorithm.

Bài toán tối ưu trong Support Vector Machine (SVM) chính là bài toán đi tìm đường phân chia sao cho margin là lớn nhất.

* + - 1. Thử nghiệm SVM
* Bộ dữ liệu lớp CĐTH 16A

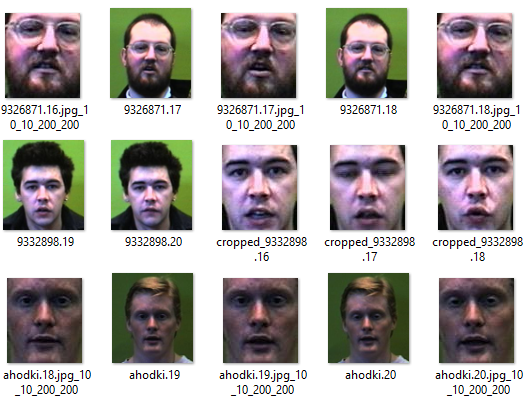
****

Hình 1 - 51: Bộ dữ liệu CĐTH 16A

|  |  |  |
| --- | --- | --- |
| **SVM** | **Đặc trưng** | **Kết quả** |
| LBP | 63.47% |
| Intensity | 79.1% |

Bảng 3: Thử nghiệm SVM với bộ dữ liệu CĐTH 16A

* Bộ dữ liệu mẫu



Hình 1 - 52: Bộ dữ liệu mẫu

|  |  |  |
| --- | --- | --- |
| **SVM** | **Đặc trưng** | **Kết quả** |
| LBP | 91% |
| Intensity | 99% |

Bảng 4: Thử nghiệm SVM với bộ dữ liệu mẫu

* + - 1. Thảo luận
* Ưu điểm SVM
  + Xử lý trên không gian nhiều chiều
  + Tiết kiệm bộ nhớ
  + Thích hợp với các bài toán phân loại như nhận dạng khuôn mặt
* Nhược điểm SVM
  + Thời gian train dữ liệu lâu.
  + Chưa thể hiện rõ tính xác suất.

# **CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ DỮ LIỆU**

* 1. **Đặc tả chức năng.**
     1. **Quản lý lớp học.**
        1. Thêm lớp học.

Người dùng có thể thêm lớp học và ứng dụng để việc quản lý điểm danh theo lớp học một cách đơn giản, dễ dàng hơn.

* + - 1. Xóa lớp học.

Người dùng có thể xóa lớp. Có thể xóa trong một số trường hợp như là lớp học mà họ không còn giảng dạy, hay một lý do khác.

* + - 1. Sửa tên lớp học

Có thể đặt lại tên lớp theo nhu cầu của người dùng. Có thể dùng trong một số trường hợp như là đặt sai tên, hoặc sửa lại tên lớp học cho dễ phân biệt với những lớp khác.

* + - 1. Cập nhật số lượng học viên

Chức năng cập nhật số lượng học viên được tự động, việc cập nhật này giúp cho việc thông kê, quản lý lớp học một cách hiệu quả và rõ ràng hơn.

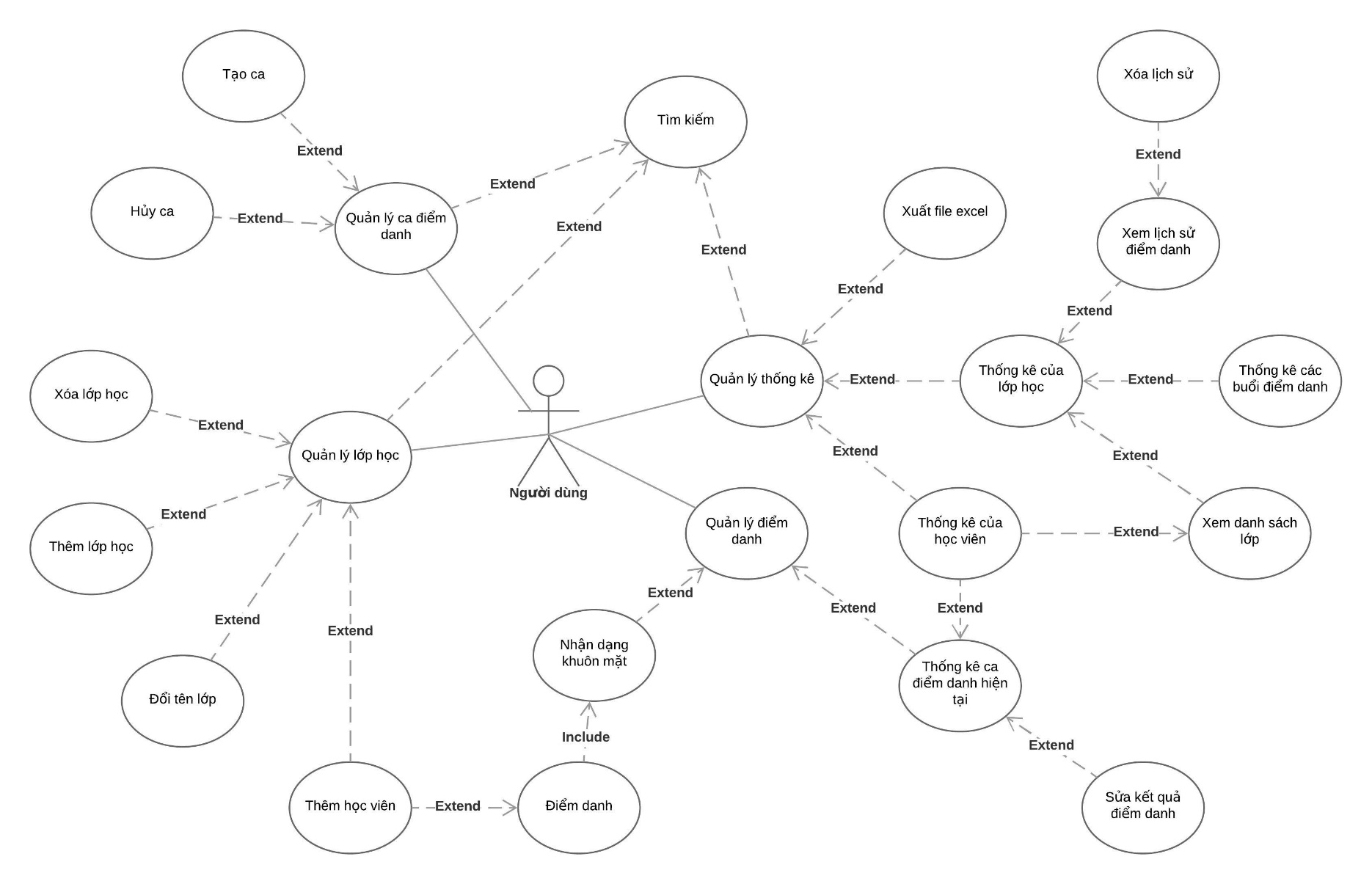
* + 1. **Điểm danh.**
       1. Quản lý ca điểm danh.
* Tạo ca điểm danh: Người dùng có thể tạo ca để điểm danh. Dựa trên ý tưởng điểm danh theo tiết học. Người dùng có thể tạo ca theo giờ ví dụ: Ca 1( gồm 2 tiết đầu buổi sáng): thì người dùng có thể tạo ca với thời gian bắt đầu là thời điểm người dùng tạo ca (ứng dụng sẽ lấy thời gian hiện tại), và thời gian kết thúc do người dùng nhập vào là 8h.
* Hủy ca điểm danh: Người dùng có thể hủy ca đang điểm danh, và tạo lại ca khác nếu có sai sót.
  + - 1. Điểm danh bằng cách nhận dạng khuôn mặt.

Điểm danh bằng cách nhận diện khuôn mặt. Dựa trên ý tưởng phong trào “ tự sướng ” (selfie) của giới trẻ hiện nay.

* + 1. **Thống kê.**
       1. Thống kê theo lớp học.
* Xem danh sách học viên theo từng lớp học.
* Thống kê cách buổi điểm danh theo lớp học.
* Xem lịch sử điểm danh theo từng lớp.
  + - 1. Thống kê theo học viên của từng lớp học.
* Thông kê tỉ lệ có mặt của một học viên theo ngày trong tuần.
* Thống kê tỉ lệ có mặt của một học viên theo tất cả lần điểm danh.
  + - 1. Thống kê ca điểm danh.
* Xem danh sách các ca đã điểm danh.
* Thống kê tỉ lệ có mặt của học viên theo một ca.
* Thống kê danh sách học viên có, vắng mặt trong một ca.
  + - 1. Xuất file Excel.

Xuất file Exel thông tin điểm danh của một lớp.

* + 1. **Ghi lịch sử**
* Ghi lại lịch sử điểm danh, hủy ca, đổi tên lớp.
* Xem lịch sử điểm danh theo từng lớp.
* Xóa lịch sử.
  1. **Use Case.**

****

* + 1. Thêm lớp học.

**Tác nhân**: Người dùng (giảng viên, giáo viên) .

**Mô tả chung**: Người dùng tạo lớp mới.

**Luồng sự kiện chính**:

**Hành động của tác nhân:**

B2 - Tác nhân chọn vào nút thêm.

B4 - Điền tên lớp mới.

B5 - Nhấn xác nhận.

**Phản ứng của hệ thống:**

B1 - Hiển thị giao diện chính.

B3 - Hiển thị khung nhập tên lớp.

B6 - Kiểm tra tên lớp thỏa điều kiện chưa. Nếu thỏa sang bước 7, ngược lại không cho thêm lớp.

B7 - Thêm lớp mới vào cơ sở dữ liệu.

**Các yêu cầu cụ thể**: Tên lớp không được để trống và không trùng với lớp đang có.

**Điều kiện trước**: Không có.

**Điều kiện sau**: Thông báo thêm thành công/ thất bại.

**Các điểm mở rộng**: Không có.

* + 1. Đổi tên lớp.

**Tác nhân**: Người dùng (giảng viên, giáo viên) .

**Mô tả chung**: Người dùng đổi tên lớp đã có.

**Luồng sự kiện chính**:

**Hành động của tác nhân:**

B1 - Tác nhân chọn nút đổi tên lớp.

B3 - Tác nhân nhập tên muốn đổi.

B4 - Nhấn xác nhận.

**Phản ứng của hệ thống:**

B2 - Hiển thị giao diện đổi tên mới.

B5 - Kiểm tra tên lớp thỏa điều kiện chưa. Nếu thỏa sang bước 6, ngược lại không cho thêm lớp.

B6 - Đổi tên lớp trong cơ sở dữ liệu.

B7 - Ghi lại lịch sử thay đổi.

**Các yêu cầu cụ thể**: Tên lớp không được để trống và không trùng với lớp đang có.

**Điều kiện trước**: Không có.

**Điều kiện sau**: Thông báo đổi tên thành công/ thất bại.

**Các điểm mở rộng**: Không có.

* + 1. Xóa lớp đang có.

**Tác nhân**: Người dùng (giảng viên, giáo viên) .

**Mô tả chung**: Người dùng xóa lớp đã có.

**Luồng sự kiện chính**:

**Hành động của tác nhân:**

B1 - Tác nhân nhấn giữ vào lớp muốn xóa.

B3 - Tác nhân chọn "Đã rõ" 🡪 sang bước 4. Tác nhân chọn "Hủy" 🡪 kết thúc.

**Phản ứng của hệ thống:**

B2 - Hiển thị cảnh báo xóa các dữ liệu liên quan.

B4 - Xóa tất cả dữ liệu liên quan đến lớp.

**Các yêu cầu cụ thể**: Không có.

**Điều kiện trước**: Không có.

**Điều kiện sau**: Thông báo xóa thành công/ thất bại.

**Các điểm mở rộng**: Tác nhân có thể tìm kiếm lớp theo tên.

* + 1. Tạo ca điểm danh.

**Tác nhân**: Người dùng (giảng viên, giáo viên) .

**Mô tả chung**: Người dùng tạo mới ca điểm danh trong ngày.

**Luồng sự kiện chính**:

**Hành động của tác nhân:**

B2 - Tác nhân chọn vào lớp muốn điểm danh. Nếu đang trong ca điểm danh thì sang bước 7, ngược lại làm tiếp bước 3.

B4 - Tác nhân chọn thời gian bằng giao diện, nhấn xác nhận.

**Phản ứng của hệ thống:**

B1 - Hiển thị giao diện lớp học (bao gồm thông tin có đang trong ca điểm danh hay không).

B3 - Hiển thị giao diện chọn thời gian kết thúc lần điểm danh mới và lấy thời gian hiện tại làm thời gian bắt đầu ca.

B5 - Kiểm tra nếu thời gian kết thúc không thỏa điều kiện thì hiển thị thông báo cho tác nhân, ngược lại sang bước 6.

B6 - Thêm ca điểm danh mới vào cơ sở dữ liệu, ghi lịch sử thay đổi và sang bước 7.

B7 - Chuyển sang giao diện điểm danh.

**Các yêu cầu cụ thể**: Thời gian kết thúc điểm danh phải lớn hơn thời gian hiện tại.

**Điều kiện trước**: Không có.

**Điều kiện sau**: Thông báo thêm ca thành công/ thất bại và thông tin tên lớp + ngày + thời gian bắt đầu và kết thúc ca điểm danh đó.

**Các điểm mở rộng**: Tác nhân có thể tìm kiếm lớp theo tên.

* + 1. Điểm danh.

**Tác nhân**: Người dùng (giảng viên, giáo viên) .

**Mô tả chung**: Người dùng điểm danh học viên theo lớp đã chọn.

**Luồng sự kiện chính**:

**Hành động của tác nhân:**

B2 - Tác nhân yêu cầu học viên đưa mặt vào camera để tiến hành điểm danh.

B4 - Yêu cầu học viên nhấp chọn vào khung nhận dạng. Nếu đã hiển thị tên 🡪 sang bước 8, ngược lại sang bước 5.

B6 - Tác nhân điền tên học viên mới vào khung và nhấn xác nhận.

**Phản ứng của hệ thống:**

B1 - Hiển thị giao diện điểm danh (gồm camera, tên lớp, ca điểm danh, nút chuyển sang trang thống kê).

B3 - Phát hiện khuôn mặt (vẽ khung xanh bao quanh mặt) và nhận dạng học viên (hiển thị tên học viên). Nếu phát hiện được khuôn mặt thì sang bước 4.

B5 - Hiển thị khung nhập tên để thêm học viên mới vào lớp.

B7 - Lưu ảnh của học viên và thêm học viên với tên được nhập vào cơ sở dữ liệu.

B8 - Điểm danh học viên trên vào ca hiện tại và ghi lịch sử thay đổi.

**Các yêu cầu cụ thể**: Không có.

**Điều kiện trước**: Không có.

**Điều kiện sau**: Thông báo điểm danh học viên thành công.

**Các điểm mở rộng**: Không có.

* + 1. Thống kê ca điểm danh hiện tại.

**Tác nhân**: Người dùng (giảng viên, giáo viên) .

**Mô tả chung**: Cho phép người dùng xem thông tin ca điểm danh hiện tại (gồm tỉ lệ học viên đi học, danh sách học viên có mặt / vắng / vắng có phép).

**Luồng sự kiện chính**:

**Hành động của tác nhân:**

B2 - Tác nhân chọn hiển thị danh sách theo có mặt hoặc vắng. Nếu chọn theo có mặt 🡪 sang bước 3, ngược lại sang bước 4.

B5 – Tác nhân có thể nhấn giữ tên học viên để thay đổi kết quả điểm danh.

**Phản ứng của hệ thống:**

B1 - Hiển thị giao diện thống kê.

B3 - Tải danh sách học viên có mặt lên giao diện.

B4 - Tải danh sách học viên vắng có phép hoặc không phép lên giao diện.

B7 - Tác nhân chọn kết quả trên giao diện (có mặt / vắng / vắng có phép) và nhấn xác nhận.

B6 - Hiển thị giao diện đổi kết quả.

B8 - Cập nhật kết quả điểm danh vào cơ sở dữ liệu và ghi lịch sử thay đổi.

**Các yêu cầu cụ thể**: Không có.

**Điều kiện trước**: Không có.

**Điều kiện sau**: Thông báo cập nhật kết quả điểm danh thành công/thất bại.

**Các điểm mở rộng**: Tác nhân có thể tìm kiếm theo tên học viên.

* + 1. Thống kê của lớp.

**Tác nhân**: Người dùng (giảng viên, giáo viên) .

**Mô tả chung**: Cho phép người dùng xem các dữ liệu thống kê của lớp qua các lần điểm danh.

**Luồng sự kiện chính**:

**Hành động của tác nhân:**

B2 - Tác nhân chọn vào nút thống kê.

**Phản ứng của hệ thống:**

B1 - Hiển thị giao diện danh sách lớp.

B3 - Chuyển sang giao diện thống kê của lớp học.

B4 - Tải dữ liệu lên 4 tab thống kê.

* + - 1. Thống kê các lần điểm danh.

**Tác nhân**: Người dùng (giảng viên, giáo viên) .

**Mô tả chung**: Cho phép người dùng xem thông tin các lần điểm danh trong quá khứ và có thể xóa 1 ca bất kỳ.

**Luồng sự kiện chính**:

**Hành động của tác nhân:**

B3 - Tác nhân có thể chọn ngày hoặc thời gian bắt đầu/kết thúc khác nhau.

B5 - Tác nhân có thể nhấn vào nút xóa ca điểm danh.

B7 - Chọn xác nhận.

**Phản ứng của hệ thống:**

B1 - Tải danh sách ngày + thời gian bắt đầu + thời gian kết thúc. Tải danh sách học viên có mặt / vắng / vắng không phép.

B2 - Hiển thị giao diện thống kê các lần điểm danh.

B4 - Tải lại danh sách học viên theo ca đang chọn.

B6 - Hiển thị hộp thoại xác nhận xóa ca.

B8 – Xóa ca điểm danh và các dữ liệu liên quan và ghi lịch sử thay đổi. Tải lại danh sách ngày + thời gian bắt đầu + thời gian kết thúc 🡪 Hiển thị lại giao diện.

**Các yêu cầu cụ thể**: Không có.

**Điều kiện trước**: Không có.

**Điều kiện sau**: Thông báo xóa ca điểm danh thành công/thất bại.

**Các điểm mở rộng**: Tác nhân có thể tìm kiếm theo tên học viên.

* + - 1. Xem lịch sử thay đổi.

**Tác nhân**: Người dùng (giảng viên, giáo viên) .

**Mô tả chung**: Cho phép người dùng xem lịch sử thay đổi trong quá khứ.

**Luồng sự kiện chính**:

**Hành động của tác nhân:**

B3 - Tác nhân có thể chọn nút xóa lịch sử 🡪 sang bước 4.

**Phản ứng của hệ thống:**

B1 - Tải danh sách lịch sử.

B2 - Hiển thị giao diện xem lịch sử thay đổi.

B4 – Xóa tất cả lịch sử thay đổi khỏi cơ sở dữ liệu.

B6 - Hiển thị lại giao diện xem lịch sử thay đổi.

**Các yêu cầu cụ thể**: Không có.

**Điều kiện trước**: Không có.

**Điều kiện sau**: Thông báo xóa lịch sử thành công/thất bại.

**Các điểm mở rộng**: Không có.

* 1. **Cơ sở dữ liệu.**
     1. **Danh mục các bảng.**
        1. Bảng LOP\_HOC (Lớp Học).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Rỗng** | **Ghi chú** |
| ID | Integer (Auto Increment) | Khóa chính | Không | Mã lớp, được tăng tự động |
| TenLop | Text | N/A | Không | Tên lớp |
| NgayTao | Text | N/A | Không | Ngày tạo lớp |
| SoLuong | Integer | N/A | Không | Số lượng học viên |

Bảng 5: Bảng LOP\_HOC

* + - 1. Bảng HOC\_VIEN (Học Viên).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Rỗng** | **Ghi chú** |
| ID | Integer (Auto Increment) | Khóa chính | Không | Mã học viên, được tăng tự động |
| HoTen | Text | N/A | Không | Tên lớp |
| IDLop | Interger | Khóa ngoại | Không | Mã lớp của học viên |

Bảng 6: Bảng HOC\_VIEN.

* + - 1. Bảng DIEM\_DANH (Điểm Danh).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Rỗng** | **Ghi chú** |
| ID | Integer (Auto Increment) | Khóa chính | Không | Mã điểm danh, được tăng tự động |
| IDLop | Integer | Khóa ngoại | Không | Mã lớp được điểm danh |
| NgayDiemDanh | Text | N/A | Không | Ngày điểm danh |
| ThoiGianBatDau | Text | N/A | Không | Thời gian bắt đầu |
| ThoiGianKetThuc | Text | N/A | Không | Thời gian kết thúc |

Bảng 7: Bảng DIEM\_DANH

* + - 1. Bảng CHI\_TIET\_DIEM\_DANH (Chi Tiết Điểm Danh).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Rỗng** | **Ghi chú** |
| IDDiemDanh | Integer | Khóa chính | Không | Mã điểm danh |
| IDHocVien | Integer | Khóa chính | Không | Mã học viên |
| KetQuaDiemDanh | Integer | N/A | Không | Kết quả điểm danh |

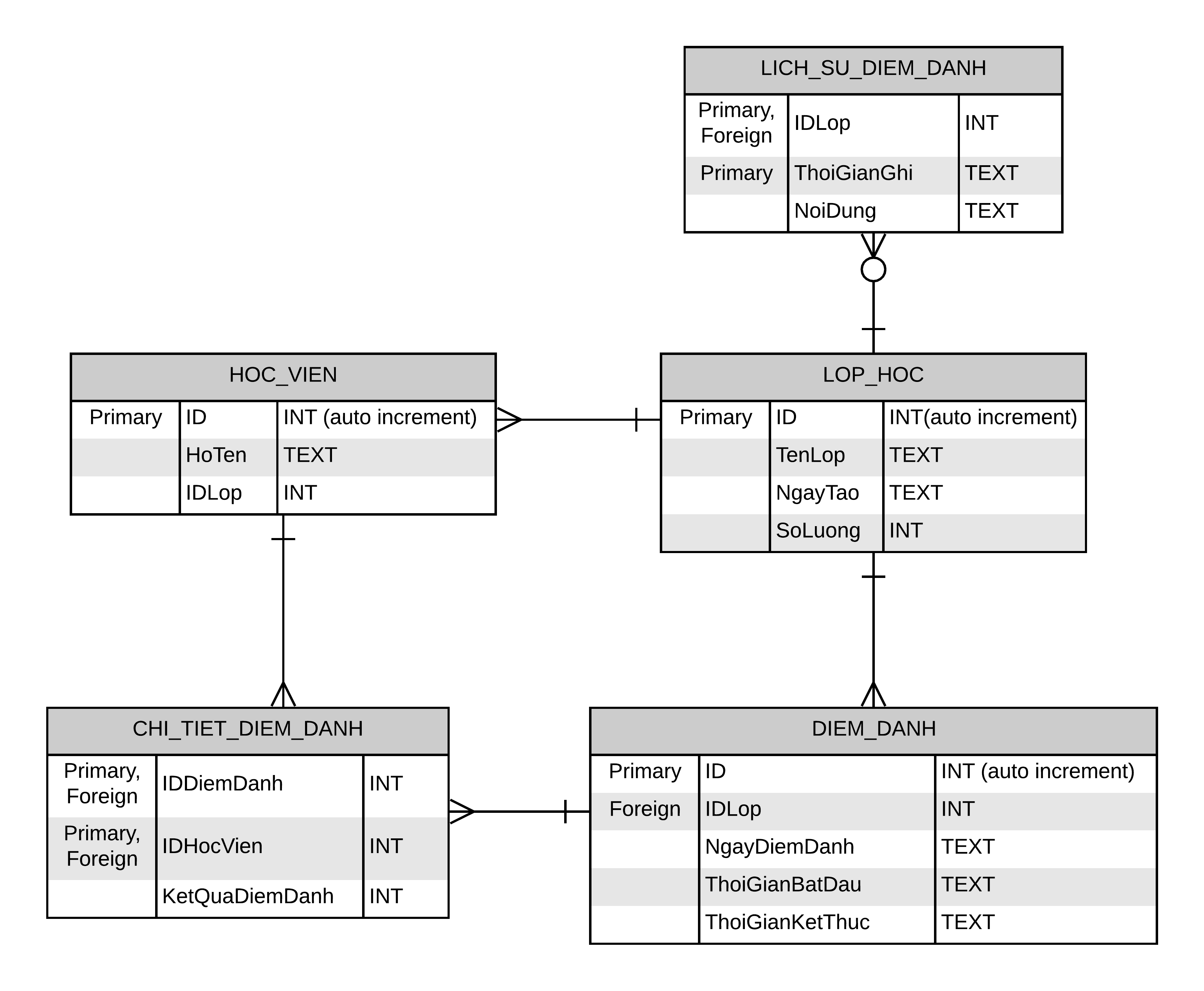
Bảng 8: Bảng CHI\_TIET\_DIEM\_DANH

* + - 1. Bảng LICH\_SU\_DIEM\_DANH (Lịch Sử Điểm Danh).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Rỗng** | **Ghi chú** |
| IDLop | Integer | Khóa chính, khóa ngoại | Không | Mã lớp |
| ThoiGianGhi | Text | Khóa chính | Không | Thời gian ghi |
| NoiDung | Text | N/A | Không | Nội dung lịch sử |

Bảng 9: Bảng LICH\_SU\_DIEM\_DANH

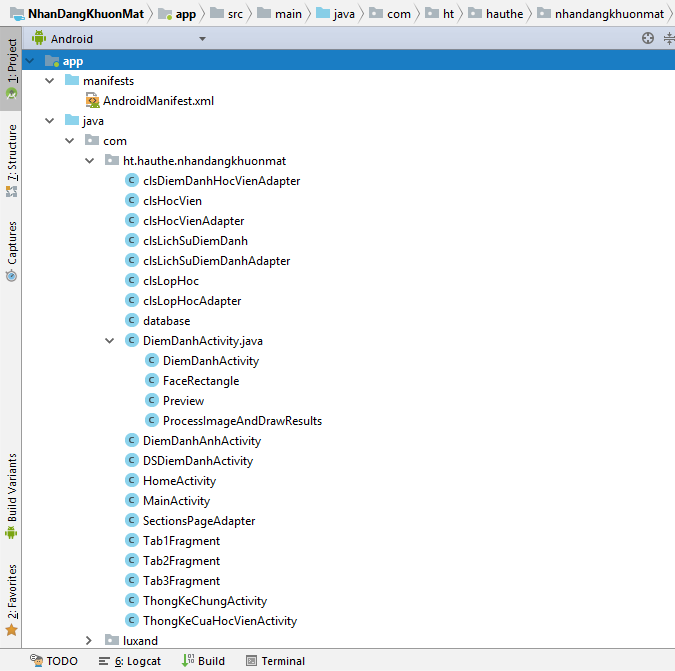
* + 1. **Mô hình cơ sở dữ liệu**



Hình 2 - 1: Mô hình cơ sở dữ liệu.

**CHƯƠNG 3: CÀI ĐẶT GIAO DIỆN VÀ CHƯƠNG TRÌNH**

2. 1. **Cấu trúc thư mục đồ án.**



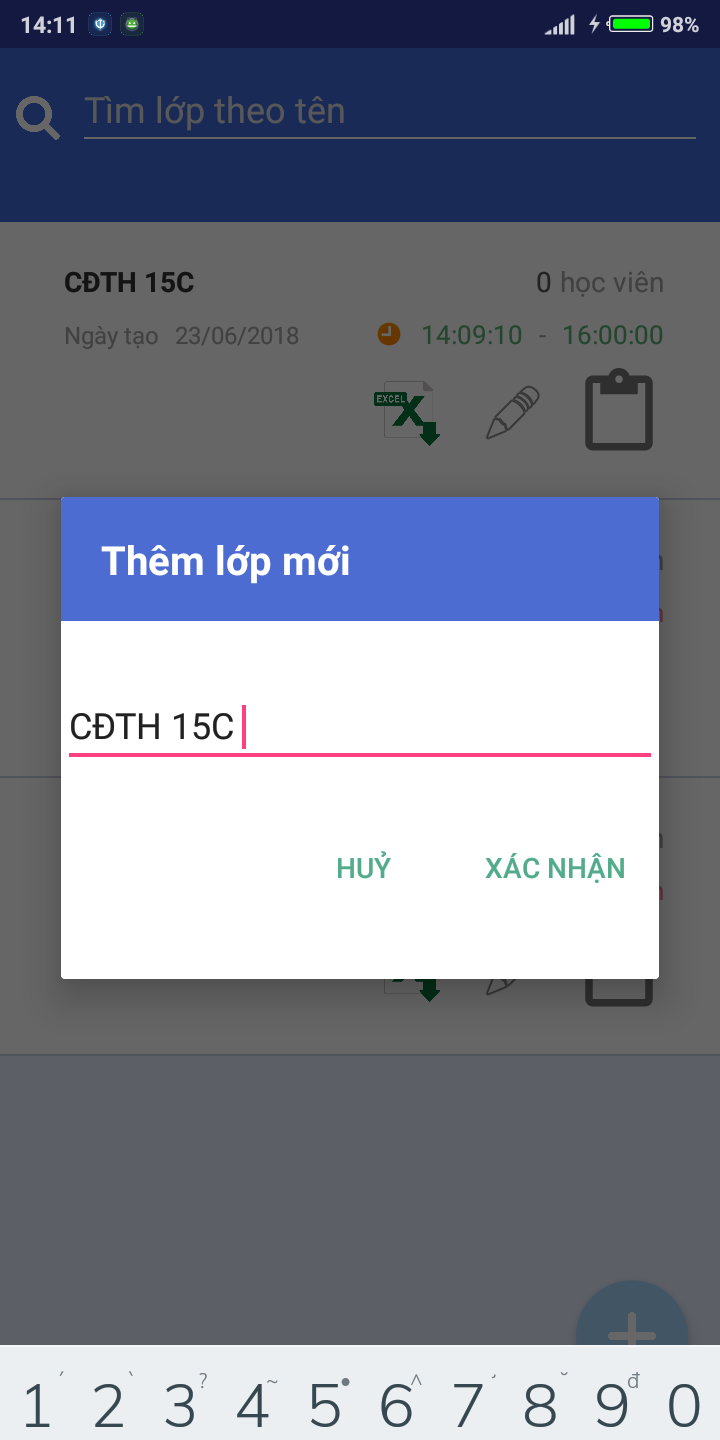
Hình 3 - 1: Cấu trúc thư mục đồ án

## **Giao diện ứng dụng.**

* + 1. **Quản lý lớp học.**
       1. Thêm lớp học.

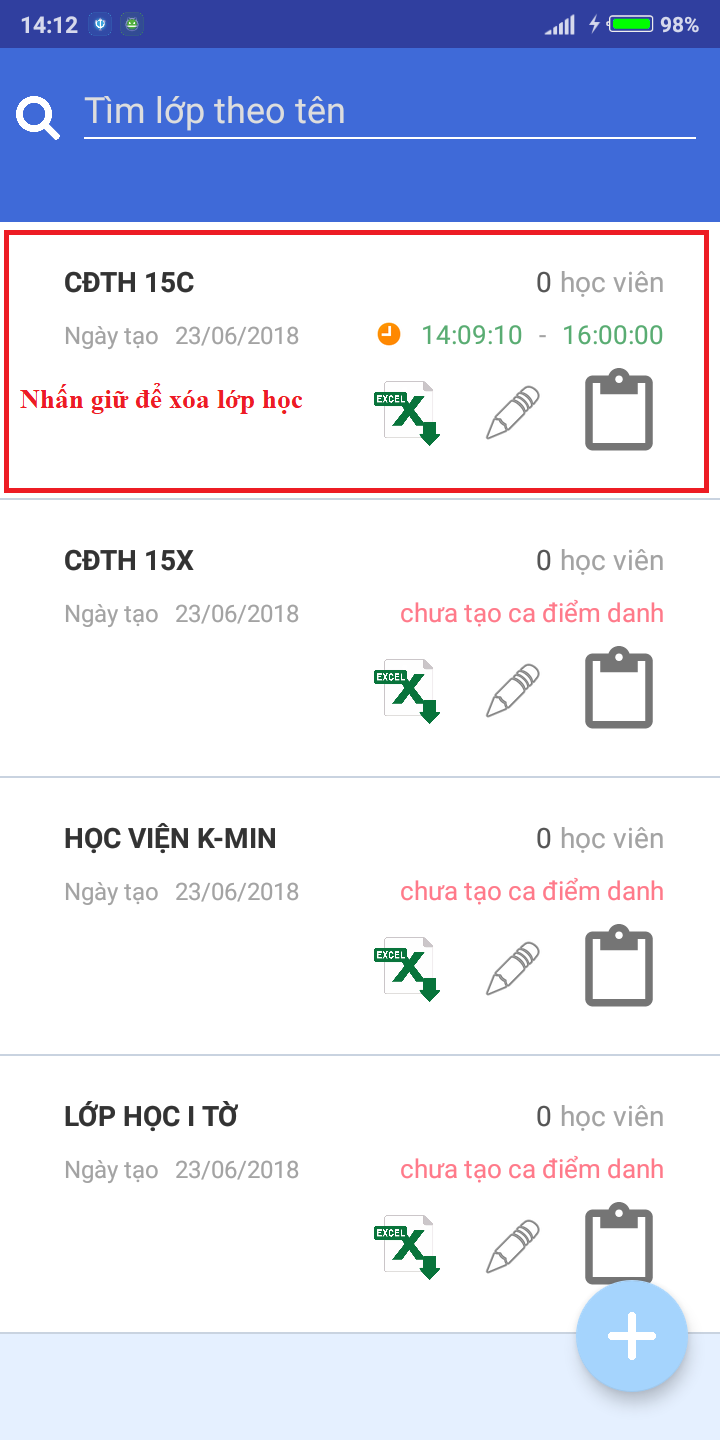


Hình 3 - 2: Nút tạo lớp học

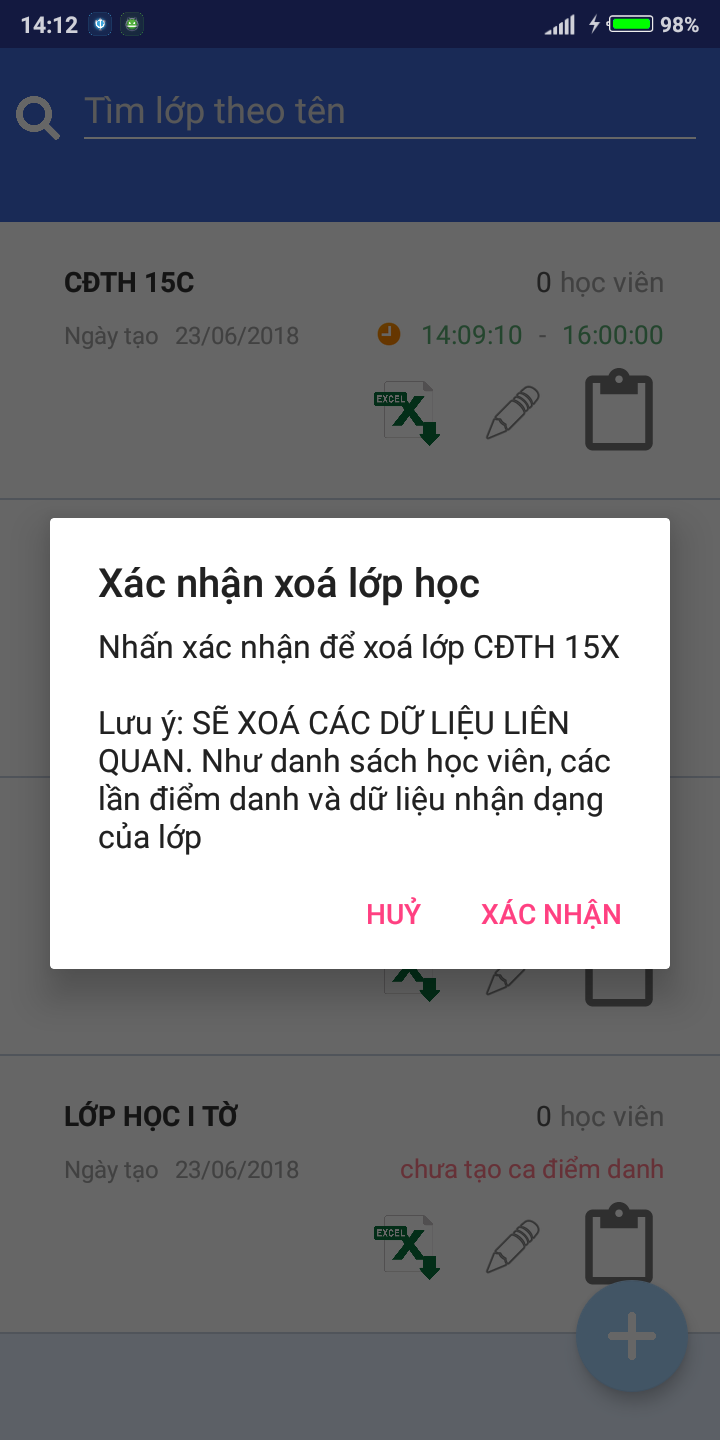


Hình 3 - 3: Giao diện thêm lớp học

* + - 1. Xóa lớp học.



Hình 3 - 4: Cách xóa lớp học

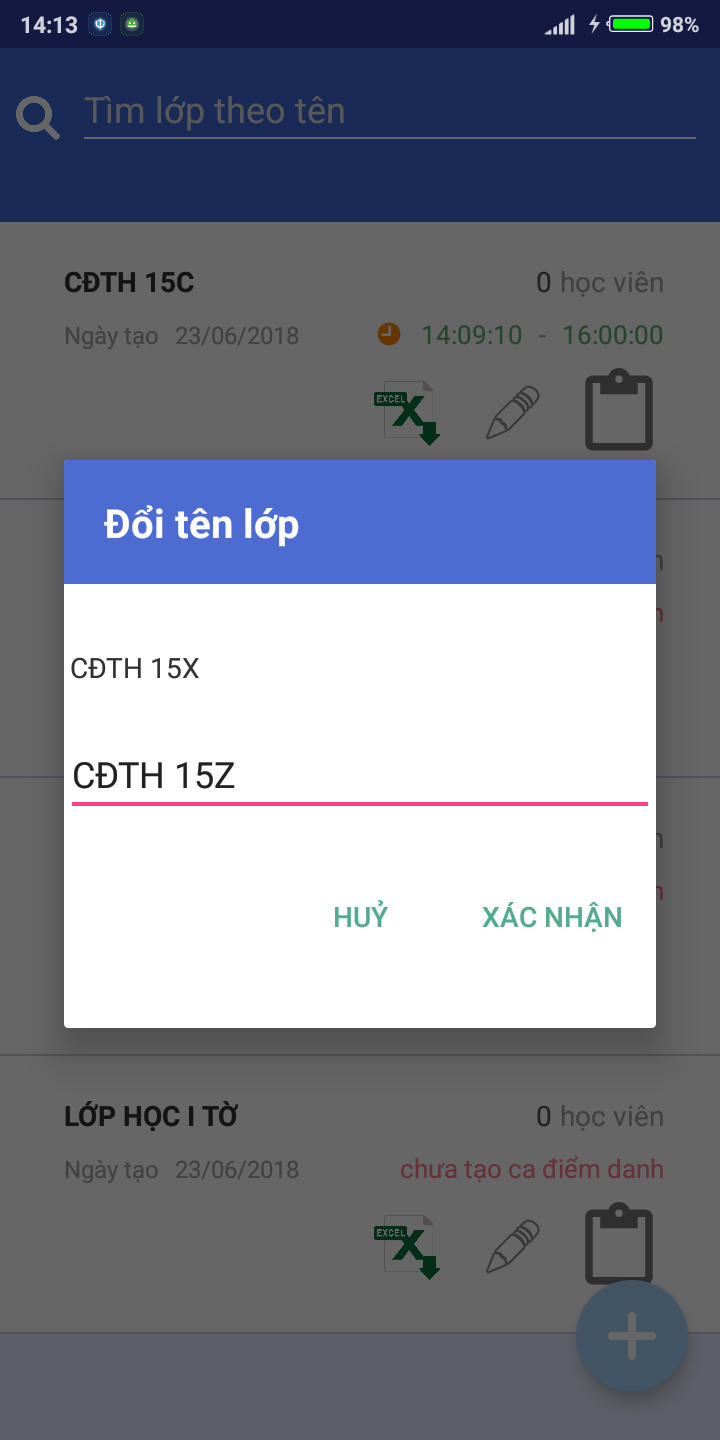


Hình 3 - 5: Giao diện xóa lớp học

* + - 1. Đổi tên lớp học.



Hình 3 - 6: Nút đổi tên lớp học



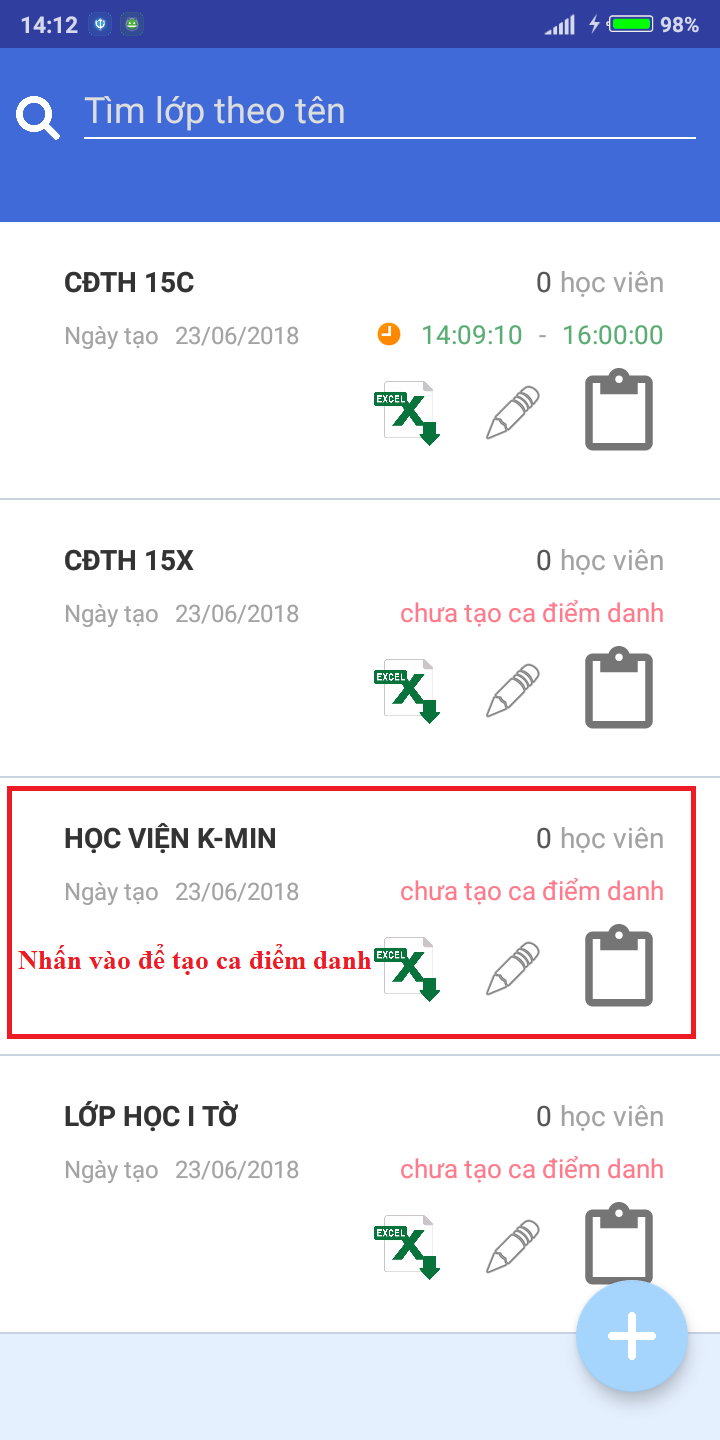
Hình 3 - 7: Giao diện đổi tên lớp học

* + - 1. Danh sách lớp học.

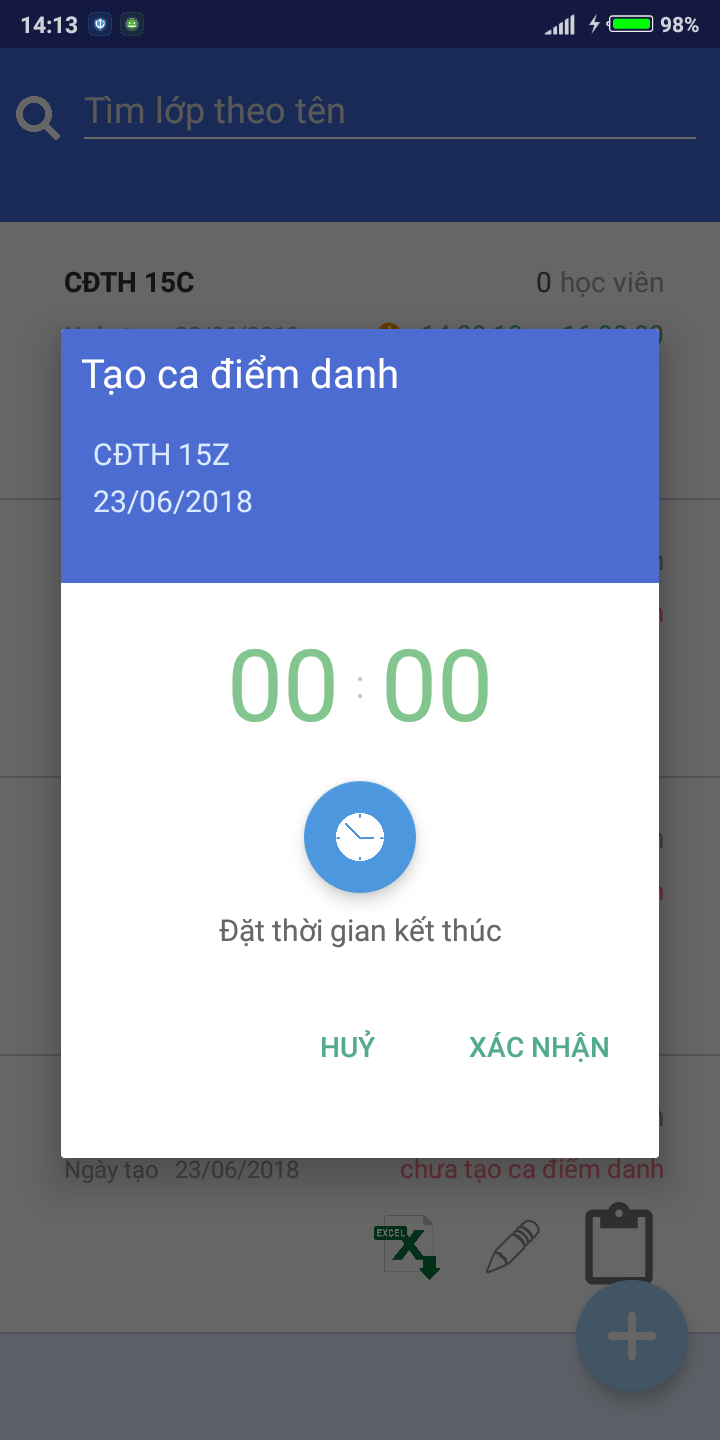


Hình 3 - 8: Giao diện danh sách lớp học

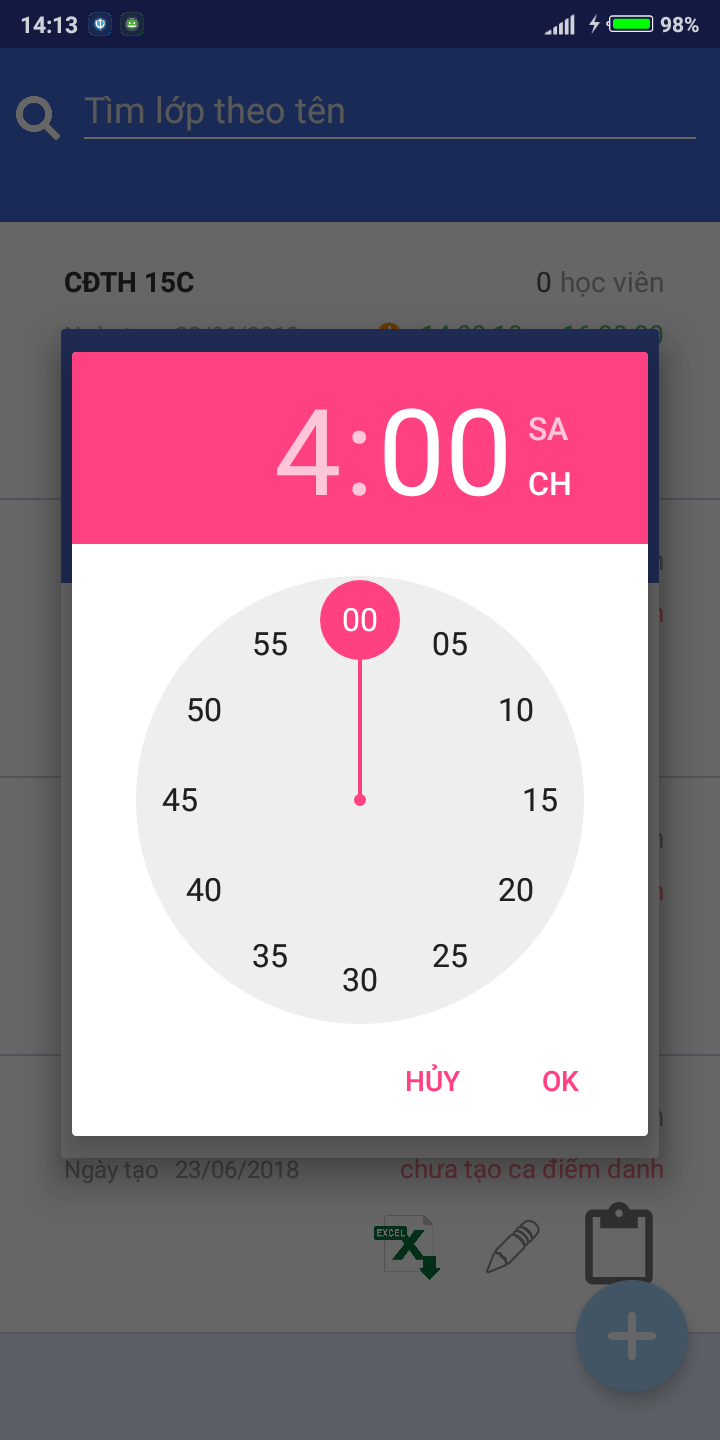
* + 1. **Điểm danh.**
       1. Tạo ca điểm danh.



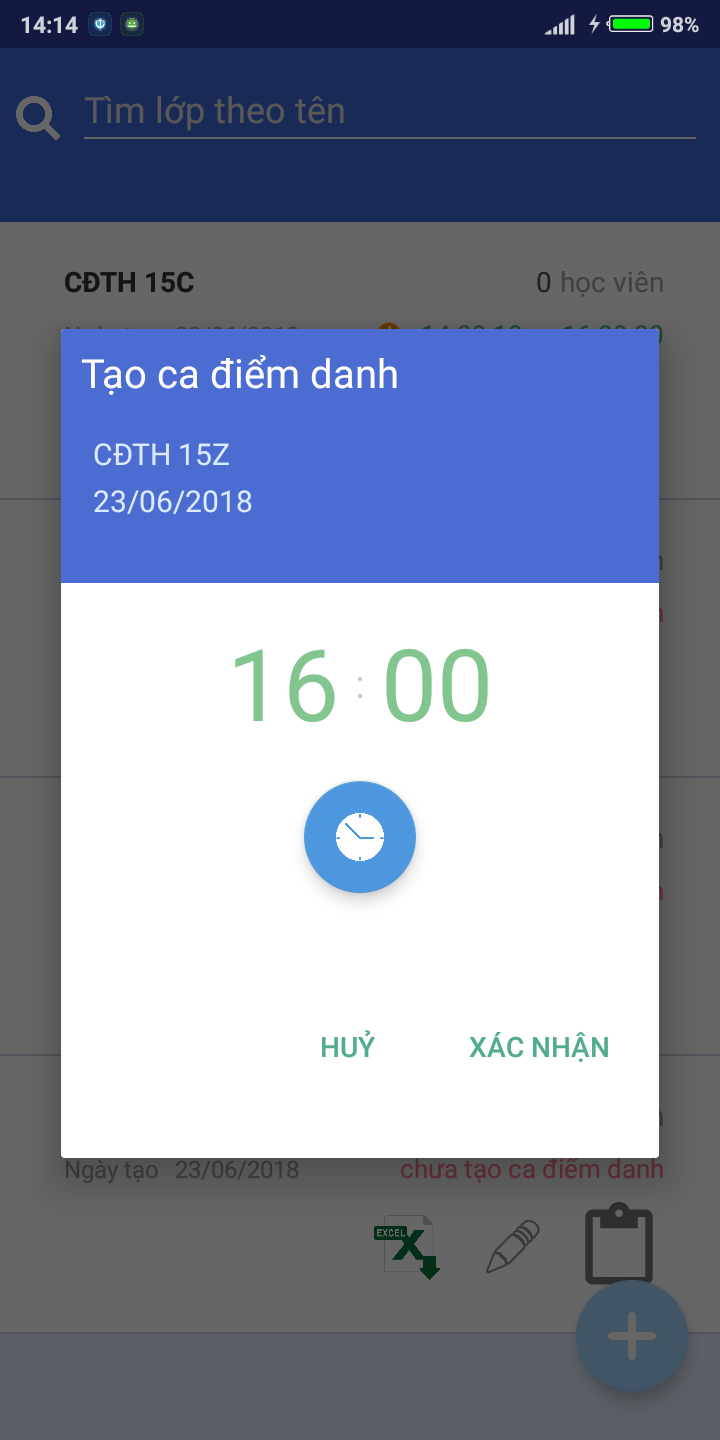
Hình 3 - 9: Bước 1 tạo ca điểm danh



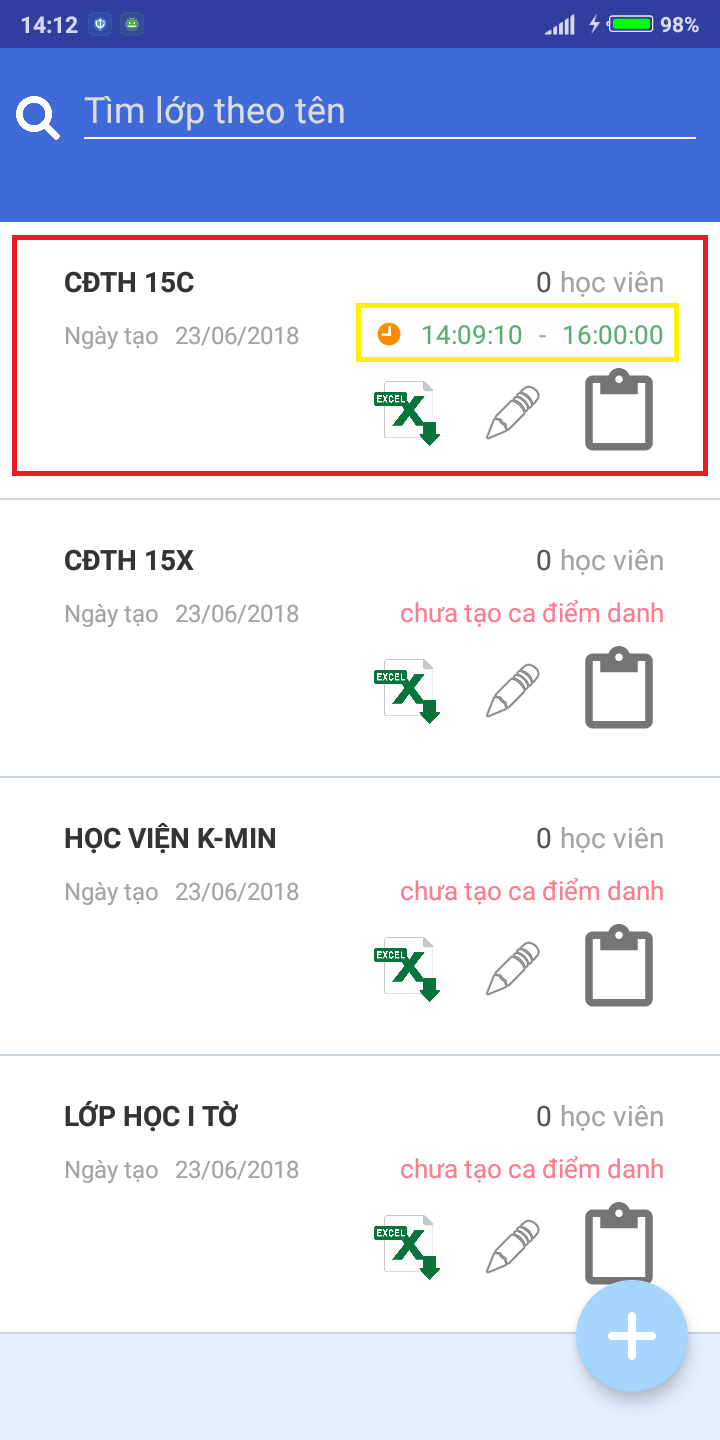
Hình 3 - 10: Bước 2 đặt giờ kết thúc



Hình 3 - 11: Bước 3 chọn giờ

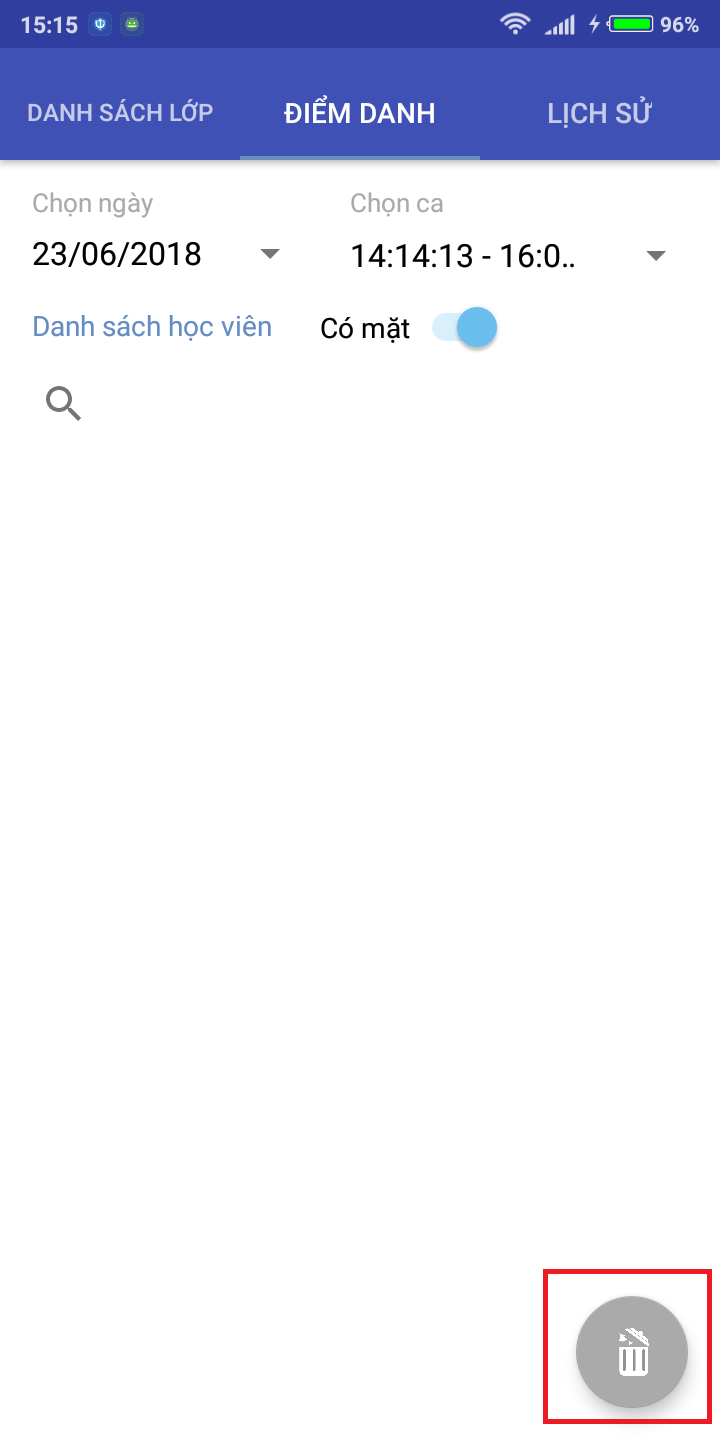


Hình 3 - 12: Bước 4 xác nhận.

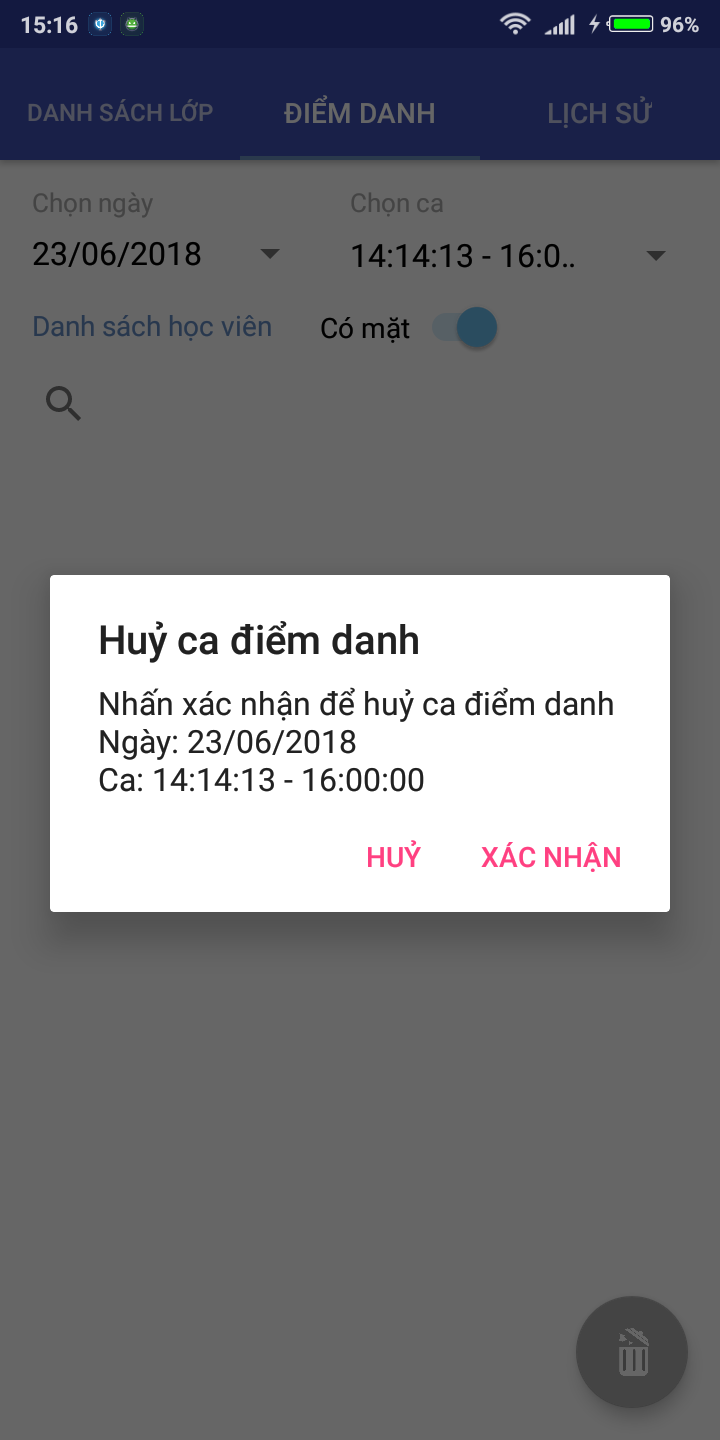


Hình 3 - 13: Giao diện tạo ca điểm danh hoàn tất.

* + - 1. Hủy ca điểm danh.

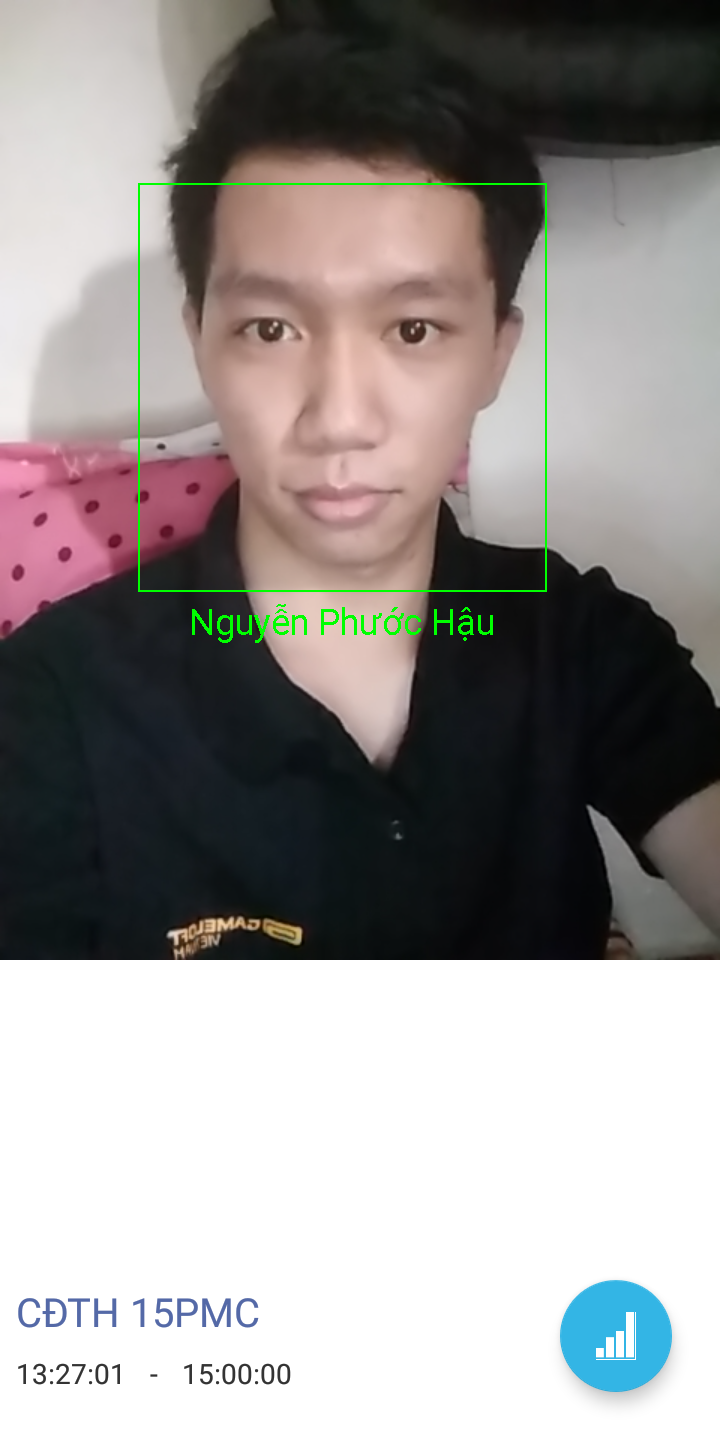


Hình 3 - 14: Nút hủy ca điểm danh



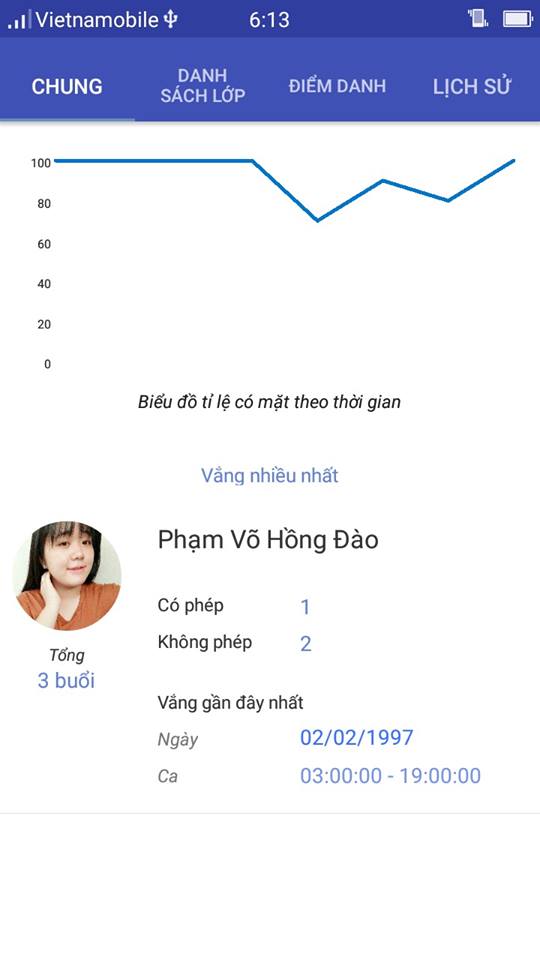
Hình 3 - 15: Giao diện hủy ca điểm danh.

* + - 1. Điểm danh bằng cách nhận diện khuôn mặt

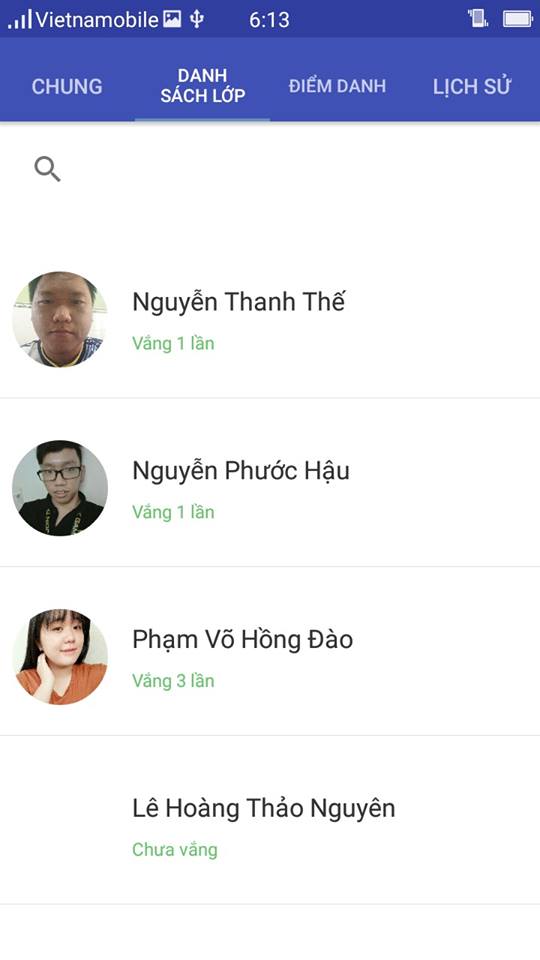


Hình 3 - 16: Giao diện nhận diện khuôn mặt.

* + 1. **Thống kê.**
       1. Thống kê theo lớp học.



Hình 3 - 17: Thống kê tỉ lệ có mặt theo thời gian điểm danh và ai vắng nhiều nhất

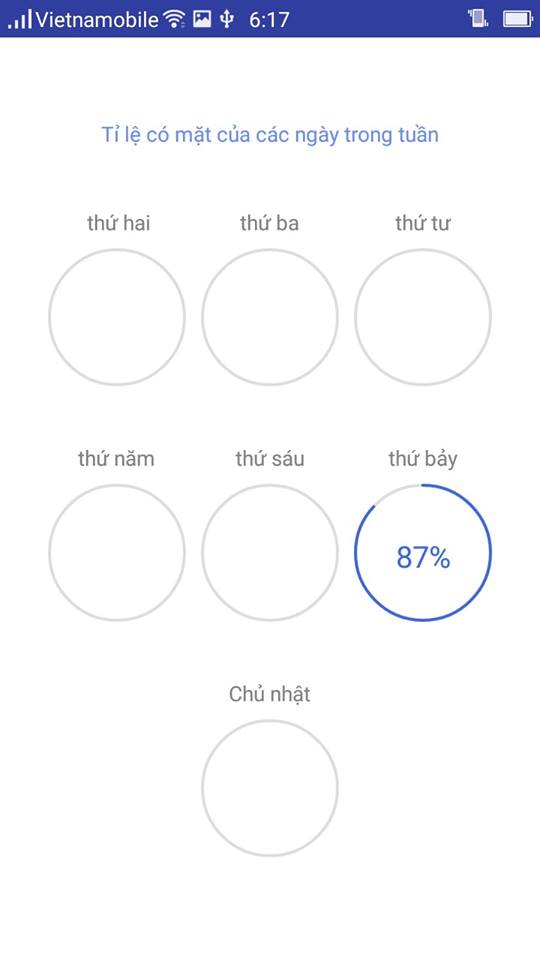


Hình 3 - 18: Danh sách học viên của lớp

* + - 1. Thống kê theo học viên của từng lớp học.

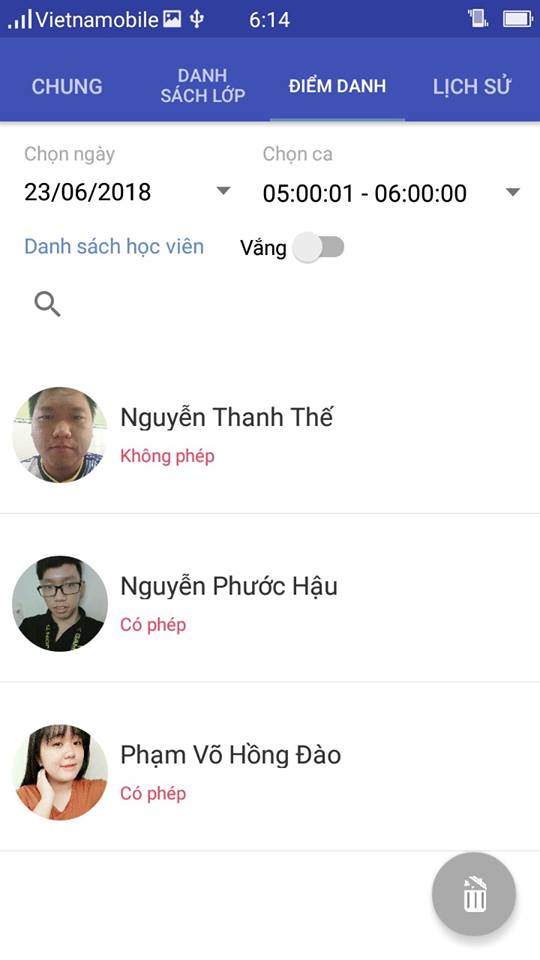


Hình 3 - 19: Thống kê tỉ lệ có mặt trong các lần điểm danh



Hình 3 - 20: Thống kê tỉ lệ có mặt vào các thứ trong tuần

* + - 1. Thống kê ca điểm danh.



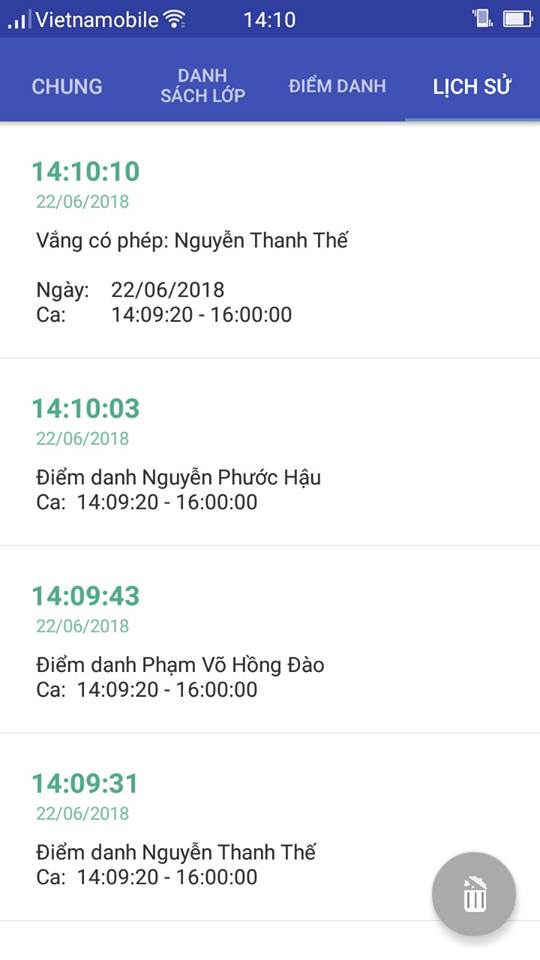
Hình 3 - 21: Xem lại các ca điểm danh trước đó

* + - 1. Xuất file Excel
* Giao diện xuất file Excel.



Hình 3 - 22: Nút xuất file Excel

* Định dạng file Excel sau khi xuất.
  + 1. **Ghi lịch sử.**

****

Hình 3 - 23: Giao diện lịch sử điểm danh

**CHƯƠNG 4: TỔNG KẾT VÀ HƯỚNG PHÁT TRIỄN**

1. **Kết quả đạt được.**

Ứng dụng điểm danh thông minh DiDan sau khi bắt đầu thực hiện đến nay về cơ bản ứng dụng đã hoàn thiện, đáp ứng rất tốt cho người dùng thông qua việc điểm danh, quản lý và thông kê cho nhóm học, lớp học, tổ chức hay doanh nghiệp vừa và nhỏ.

Ứng dụng được viết tập trung vào nhu cầu sử dụng của giáo viên và giảng viên. Vì vậy giáo viên hay giảng viên khi sử dụng ứng dụng sẽ rất tiện lợi cho việc quản lý ,điểm danh và thống kê học sinh, sinh viên lớp mình giảng dạy. Qua thời gian trải nghiệm có thể xem DiDan là một ứng dụng phục vụ rất tốt cho việc điểm danh với công nghệ nhận diện khuôn mặt. Giúp cho người dùng tiết kiệm thởi gian trong việc điểm danh, bên cạnh đó việc quản lý điểm danh theo ngày, tháng, năm cũng dễ dàng hơn bao giờ hết.

1. **Phần hạn chế của đề tài.**

Do thời gian, kiến thức, kinh nghiệm còn giới hạn nên đồ án của nhóm chúng em không tránh khỏi những thiếu sót.

Thời gian ứng dụng được trải nghiệm chỉ có một, hai tuần, vì vậy những lỗi gây crash ứng dụng và những lỗi về tốc độ xử lý của ứng dụng khi sử dụng với dữ liệu lớn vẫn chưa được phát hiện, nhưng không chắc chắn là không có.

Bên cạnh đó, số lượng thiết bị sử dụng để kiểm tra ứng dụng vẫn còn hạn chết (5 thiết bị android). Cho nên không thể phát hiện những lỗi về giao diện hay ứng dụng có đáp ứng tốt trên cái điện thoại android khác hay không.

Có thể nói nhận diện khuôn mặt là một lĩnh vực hoàn toàn mới đối với chúng em, vì vậy ứng dụng chỉ nhận diện được một người một lần, không thể nhận diện được nhiều người cùng lúc. Thông qua đó việc nhận diện còn phụ thuộc vào điều kiện ánh sáng, nếu camera được đặt ở vị trí ngược sáng thì có thể không nhận diện được, khả năng nhận diện còn sai với một số trường hợp như là: xoay đầu, khi gắn thẻ cho khuôn mặt với điều kiện ánh sáng không tốt.

1. **Hướng phát triễn.**

Đối với công nghệ dận diện khuôn mặt cần cải thiện một số chức năng sau:

* Nhận diện nhiều khuôn mặt cùng một thời điểm
* Khắc phục việc nhận diện sai khi xoay đầu hay gắn thẻ khuôn mặt với điều kiện ánh sáng không tốt
* Cải thiện tốc độ nhận diện nhanh hơn, mượt mà hơn.

Đối với ứng dụng điểm danh DiDan:

* Đảm bảo ứng dụng hoàn hảo khi sử dụng với dữ liệu lớn.
* Giao diện giúp cho người dùng sử dụng ứng dụng một cách dễ dàng hơn, trông đẹp mắt hơn.
* Phục vụ tốt cho tất cả loại điện thoại nền tảng Android.
* Đưa ứng dụng lên Google Play Store.
* Đưa ứng dụng sang các nền tảng khác như là: IOS, Windows, Windows Phone, MacOS…

# **TÀI LIỆU THAM KHẢO**

1. <https://www.stdio.vn/articles/opencv-cai-dat-va-vi-du-minh-hoa-su-dung-313>
2. <https://www.stdio.vn/articles/cai-dat-moi-truong-lap-trinh-android-studio-tren-windows-va-tao-project-dau-tien-227>
3. <https://123doc.org/document/3716239-nhan-dang-bien-so-xe-thuat-toan-trich-rut-dac-trung-lbp.htm>
4. <http://home.zcu.cz/~pkral/papers/icae_kral15.pdf>
5. <http://vimach.net/threads/gioi-thieu-ve-histogram-trong-xu-ly-anh.460/>
6. <https://www.luxand.com/facesdk/documentation/requirements.php>
7. <https://www.luxand.com/download/Luxand_FaceSDK_Documentation.pdf>
8. [https://machinelearningcoban.com/2016/12/27/categories/#classification-phan-loai](https://machinelearningcoban.com/2016/12/27/categories/%23classification-phan-loai)
9. <https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm>
10. <https://viblo.asia/p/mot-chut-ve-thuat-toan-svm-support-vector-machine-algorithm-OeVKBgGAZkW>
11. Sách tham khảo: Machine Learning Cơ Bản – Vũ Hữu Tiệp