BỘ CÔNG THƯƠNG

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM

KHOA CÔNG NGHỆ THÔNG TIN

---🙠🕮🙢---



**Đồ Án**

**Phần Mềm Quản Lý Điểm Danh Sinh Viên**

Giáo viên hướng dẫn: **Bùi Công Danh**

Sinh viên thực hiện:

Nguyễn Minh Hiến - 2001180169

Trần Đình Văn - 2001180274

TP. HỒ CHÍ MINH, tháng 12 năm 2020

MỤC LỤC

[MỤC LỤC 1](#_Toc59475612)

[CHƯƠNG I: GIỚI THIỆU ĐỒ ÁN 2](#_Toc59475613)

[1. Lý do lựa chọn đề tài 2](#_Toc59475614)

[2. Mục tiêu của đề tài 2](#_Toc59475615)

[3. Nhiệm vụ 2](#_Toc59475616)

[4. Sự thực tiễn của đề tài 2](#_Toc59475617)

[CHƯƠNG II: NHẬN DIỆN KHUÔN MẶT OPENCV 3](#_Toc59475618)

[1. Giới thiệu EmguCV 3](#_Toc59475619)

[1.1 EmguCV là gì? 3](#_Toc59475620)

[1.2 Cài đặt EmguCV 3](#_Toc59475621)

[2. Giới thiệu OpenCV 3](#_Toc59475622)

[2.1 *OpenCV* là gì? 3](#_Toc59475623)

[2.2 Cấu trúc tổng quan 4](#_Toc59475624)

[2.3 Hướng dẫn sử dụng các thư viện OpenCV. 4](#_Toc59475625)

[3. Phát hiện khuôn mặt bằng phương pháp Haar-Cascade 4](#_Toc59475626)

[CHƯƠNG III: THIẾT KẾ CSDL VÀ THỰC HIỆN 10](#_Toc59475627)

[1. Sơ đồ quan hệ 10](#_Toc59475628)

[2. Các chức năng của phần mềm 10](#_Toc59475629)

[2.1 Form đăng nhập 10](#_Toc59475630)

[2.2 Form danh mục 11](#_Toc59475631)

[2.3 Chi tiết các chức năng của phần mềm 12](#_Toc59475632)

[CHƯƠNG IV: KẾT LUẬN VÀ ĐỊNH HƯỚNG PHÁT TRIỂN 16](#_Toc59475633)

[1. Kết luận 16](#_Toc59475634)

[2. Định hướng phát triển 16](#_Toc59475635)

[TÀI LIỆU THAM KHẢO 17](#_Toc59475636)

CHƯƠNG I: GIỚI THIỆU ĐỒ ÁN

1. Lý do lựa chọn đề tài

Việc điểm danh sinh viên bằng phương pháp thủ công trong mỗi buổi học là một quá trình mất nhiều thồi gian, đặc biệt là khi số lượng sinh viên trong một lớp là tương đối nhiều: từ 80 sinh viên đến 500 sinh viên. Với yêu cầu thì trong mỗi buổi học, giáo viên sẽ thực hiện điểm danh trong mỗi buổi học để đảm bảo tính điểm quá trình cho sinh viên. Nói cách khác, trong mỗi buổi học giáo viên sẽ tốn nhiều thừi gian để thực hiện điểm danh cho một lần duy nhất trong mỗi buổi học. Theo khảo sát thì em thấy quá trình điểm danh sinh viên còn gặp nhiều khó khăn như: tốn quá nhiều thời gian của buổi học, tình trạng điểm danh hộ,..

Xuất phát từ những khó khăn trong việc điểm danh sinh viên nhóm chúng em đã nghiên cứu đề tài “Xây dựng phần mềm quản lý điểm danh sinh viên” đưa ra giải pháp ứng dụng công nghệ thông tin trong việc quản lý điểm danh sinh viên. Giúp cho công việc quản lý của giáo viên nhanh chóng và chính xác. Đề tài giúp cho việc điểm danh sinh viên trở nên khoa học, chuyên nghiệp và chính xác hơn.

1. Mục tiêu của đề tài

Tìm hiểu những ưu điểm nhược điểm quá trình điểm danh sinh viên theo quy định truyền thống.

Nghiên cứu điểm danh sinh viên bằng phương pháp nhận diện khuôn mặt bằng camera.

Nghiên cứu đề tài để xây dựng kinh nghiệm hỗ trợ giảng dạy.

1. Nhiệm vụ

* Tìm hiểu vệ hệ thống nhận dạng khuôn mặt
* Tìm hiểu những ưu điểm nhược điểm quá trình điểm danh sinh viên quy định truyền thống
* Xây dựng quy trình điểm danh sinh viên
* Trình bày đầy đủ những nội dung đã thực hiện
* Xây dựng chức năng quản lý thống kê báo cáo danh sách sinh viên đi học.
* Hiển thị kết quả tìm kiếm cho sinh viên kiểm tra thông tin đi học của mình.

1. Sự thực tiễn của đề tài

Ý tưởng xuất phát từ nhu cầu thực tế nên đề tài mang tính ứng dụng cao.

CHƯƠNG II: NHẬN DIỆN KHUÔN MẶT OPENCV

1. Giới thiệu EmguCV
   1. EmguCV là gì?

Là một cross platform .NET, một thư viện xử lý ảnh mạnh mẽ dành riêng cho ngôn ngữ .NET, cho phép gọi được chức năng của OpenCV là từ .NET.

Tương thích ngôn ngữ như: C#, VB, VC++, Iron Python…

Wrapper có thể được biên dịch bởi Visual Studio, Xamarin Studio và Unity.

Có thể chạy trên Windows, Linux, Mac OS X, IOS, Android và Windows phone.

* 1. Cài đặt EmguCV

Download Emgucv tại: <https://sourceforge.net/projects/emgucv/files/emgucv/>.

Hoặc vào trên thanh công cụ visual studio chọn Tools – NuGetPackageManage – Manage Package NuGet solution…

1. Giới thiệu OpenCV

2.1 OpenCV là gì?

OpenCv (Open Source Computer Vision) là một thư viện mã nguồn mở về thị giác máy với hơn 500 hàm và hơn 2500 các thuật toán đã được tối ưu về XLA, và các vấn đề liên quan tới

thị giác máy. OpenCv được thiết kế một cách tối ưu, sử dụng tối đa mạnh của các dòng chip đa

lõi… để thực hiện các phép tính toán trong thời gian thực, nghĩa là tốc độ đáp ứng của nó thể đủ nhanh cho các ứng dụng thông thường.

OpenCv là thư viện được thiết kế để chạy trên nhiều nền tảng khác nhau (cross-platform),

nghĩa là nó có thể chạy trên hệ điều hành Window, Linux, Mac, iOS… Việc sử dụng thư viện OpenCv tuân theo các quy định về *sử dụng phần mềm mã nguồn mở BSD* do đó bạn có thể sử dụng thư viện này một cách miễn phí cho các mục đích phi thương mại lẫn thương mại.

Dự án về OpenCv được khởi động từ những năm 1999, đến năm 2000 nó được giới thiệu trong một *hội nghị của IEEE về các vấn đề trong thị giác máy và nhận dạng*, tuy nhiên bản OpenCV 1.0 mãi tới tận năm 2006 mới chính thức được công bố và năm 2008 bản 1.1 (pre-release) mới được ra đời. Tháng 10 năm 2009, bản OpenCV thế hệ thứ hai ra đời (thường gọi là phiên bản 2.x), phiên bản này có giao diện của C++ (khác với phiên bản trước có giao diện của C) và có nhiều điểm khác biệt so với phiên bản thứ nhất.

Thư viện OpenCV ban đầu được sự hỗ trợ từ Intel, sau đó được hỗ trợ bởi Willow Garage, một phòng thí nghiệm chuyên nghiên cứu về công nghệ robot. Cho đến nay, OpenCV vẫn là thư viện mở, được phát triển bởi nguồn quỹ không lợi nhuận (none-profit foundation) và được sự hưởng ứng rất lớn của cộng đồng.

2.2 Cấu trúc tổng quan

Cấu trúc tổng quan của OpenCv gồm 5 phần chính. Hình vẽ dưới đây mô tả 4 trong 5 thành phần đó.

2.3 Hướng dẫn sử dụng các thư viện OpenCV.

Để sử dụng các thư viện của OpenCV, có thể tải về một bản cài đặt OpenCV bằng cáchtruy cập địa chỉ:[http://sourceforge.net/projects/opencvlibrary/ đ](http://sourceforge.net/projects/opencvlibrary/%20đ)ể tải về phiên bản mới nhất. Sau khi tải về, tiến hành cài đặt bình thường, với thư mục mặc định là *C:\Emgu.* Sau đó, cần tiến hành tùy chỉnh để có thể làm việc với OpenCV qua hai IDE thông dụng là Microsoft Visual Studio hoặc Eclipse CDT. Tuy nhiên, việc tùy chỉnh này có thể gây khó khăn cho một số người mới bắt đầu, vì vậy, ở đây trình bày một cách khác để có thể sử dụng các thư viện của OpenCV. Đó chính là sử dụng gói phần mềm EmguCV.

*EmguCV là một gói bao bên ngoài (wrapper) thư viện xử lý ảnh OpenCV của Intel, cho phép lập trình viên có thể gọi các hàm của OpenCV từ trong các ngôn ngữ .NET C#, VB,IronPython… Gói này có thể được biên dịch trong Mono và chạy trên Linux / Mac OS X.*

Việc sử dụng và lập trình với EmguCV cũng rất đơn giản, chỉ cần thêm các file DLL trong thư mục cài đặt của EmguCV vào chương trình C# thì có thể gọi tất cả các thuộc tính của EmguCV.

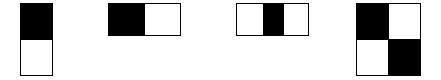
1. Phát hiện khuôn mặt bằng phương pháp Haar-Cascade

Ở bước này, chúng ta sẽ cần một mô hình có khả năng nhận biết ở đâu trong một bức ảnh có sự xuất hiện của mặt người, mô hình sẽ trả về cho chúng ta các toạ độ (x1, y1, x2, y2) hoặc (x, y, w, h) để hợp thành các ô vuông là các boundingbox, các boundingbox này sẽ chứa mặt người bên trong từ đó giúp chúng ta thực hiện cắt khuôn mặt, trích xuất ra đặc trưng là mặt của bức ảnh. Ngoài ra, mô hình cũng sẽ trả về cho chúng ta các tham số như: confidence (độ tự tin vị trí này có mặt người), và các toạ độ trên khuôn mặt như vị trí của mắt, mũi, miệng, ...

Phát hiện khuôn mặt sử dụng Haar-cascade là một phương pháp phát hiện khuôn mặt được đề xuất bởi Paul Viola và Michael Jones trong bài báo của họ, “Rapid Object Detection using Bossted Cascade of Simple Features” năm 2001.

**Đặc trưng haar-like**

Mỗi đặc trưng là một giá trị duy nhất thu được bằng cách trừ tổng số pixel trong vùng trắng cho tổng số pixel trong vùng đen.



Đặc trưng haar-like

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

* Đặc trưng cạnh (edge feature):

https://images.viblo.asia/47709099-37f8-48f2-be1b-4a5dc6a41c5b.png

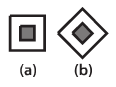
Đặc trưng cạnh

* Đặc trưng đường (line feature):



Đặc trưng đường

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



Đặc trưng xung quanh tâm

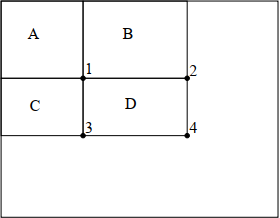
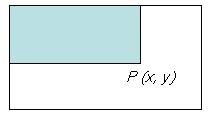
Ví dụ như sau:



Minh họa cách hoạt động của bộ lọc

Trong hình trên, cửa sổ trượt được đặt ngay ngắn vừa gọn để nhìn được toàn bộ ảnh. Các bạn có thể nhận ra rằng bộ lọc đầu trong đó đang tìm một "cạnh" phân cách giữa mắt/lông mày với mũi, vì ở đoạn đọc có chênh lệch về màu đáng kể; và ở bộ lọc sau, mô hình đang tìm đường sống mũi, vì ở đó sẽ có màu sáng hơn so với 2 bên (vì nó cao hơn dễ bắt sáng). Và như đã nói trên, bộ lọc Haar chỉ nhìn cụ thể vào một vùng trong cửa sổ để tìm thôi: trong khuôn mặt thì mũi lúc nào cũng ở chính giữa chứ không ở các góc, nên không cần nhìn các góc để làm gì cả.

Để tăng tốc khi tính toán các bộ lọc trên, chúng ta thường sử dụng Integral Image, là một mảng 2 chiều với kích thước bằng với kích thước của ảnh cần tính đặc trưng Haar-Like, với mỗi phần tử của mảng này được tính bằng cách tính tổng của điểm ảnh phía trên (dòng-1) và bên trái (cột-1) của nó.



Minh họa công thức Intergral Image

Công thức tính Intergral Image:

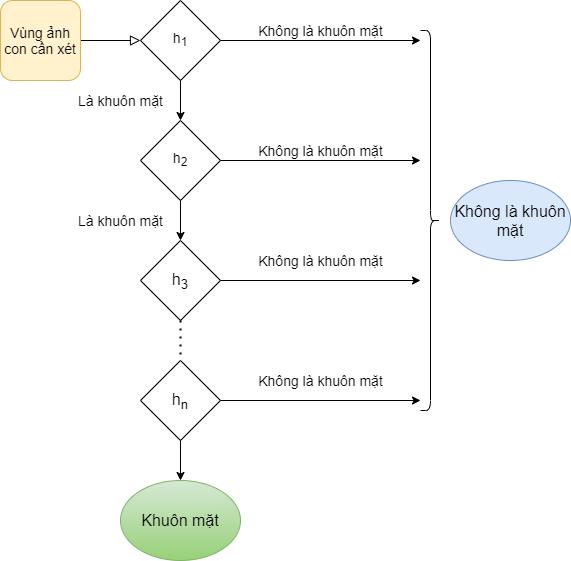
Việc tính tổng mức xám của một vùng bất kỳ được tính theo cách sau:

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

***AdaBoost***

AdaBoost là một bộ phân loại mạnh phi tuyến phức dựa trên hướng tiếp cận boosting được Freund và Schapire đưa ra vào năm 1996 [5]. Adaboost cũng hoạt động trên nguyên tắc kết hợp tuyến tính các weak classifiers [6] để hình thành một trong các classifiers.

AdaBoost được dùng để kết hợp các bộ phân loại yếu sử dụng các đặc trưng Haar-Like theo mô hình phân tầng (cascade) như sau:



Hình 3.13 Mô hình phân tầng (cascade)

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

, nếu và ngược lại

Với:

X: cửa sổ con cần xét.

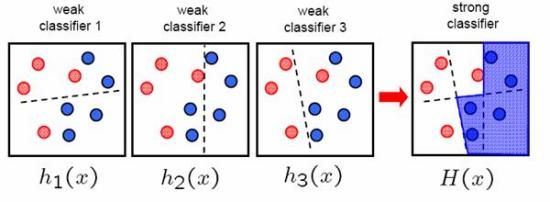
: Ngưỡng.

: Giá trị của đặc trưng Haar-Like.

: Hệ số quyết định chiều của phương trình.

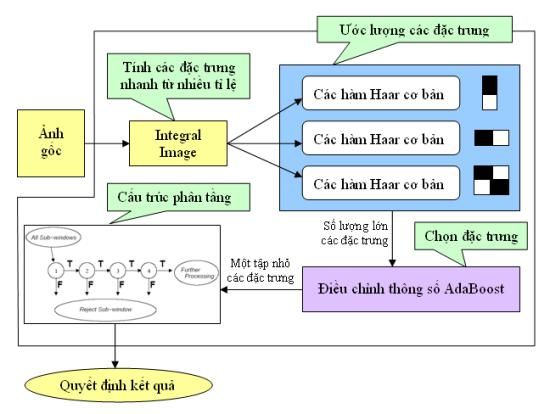
AdaBoost sẽ kết hợp các bộ phân loại yếu thành bộ phân loại mạnh như sau:

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



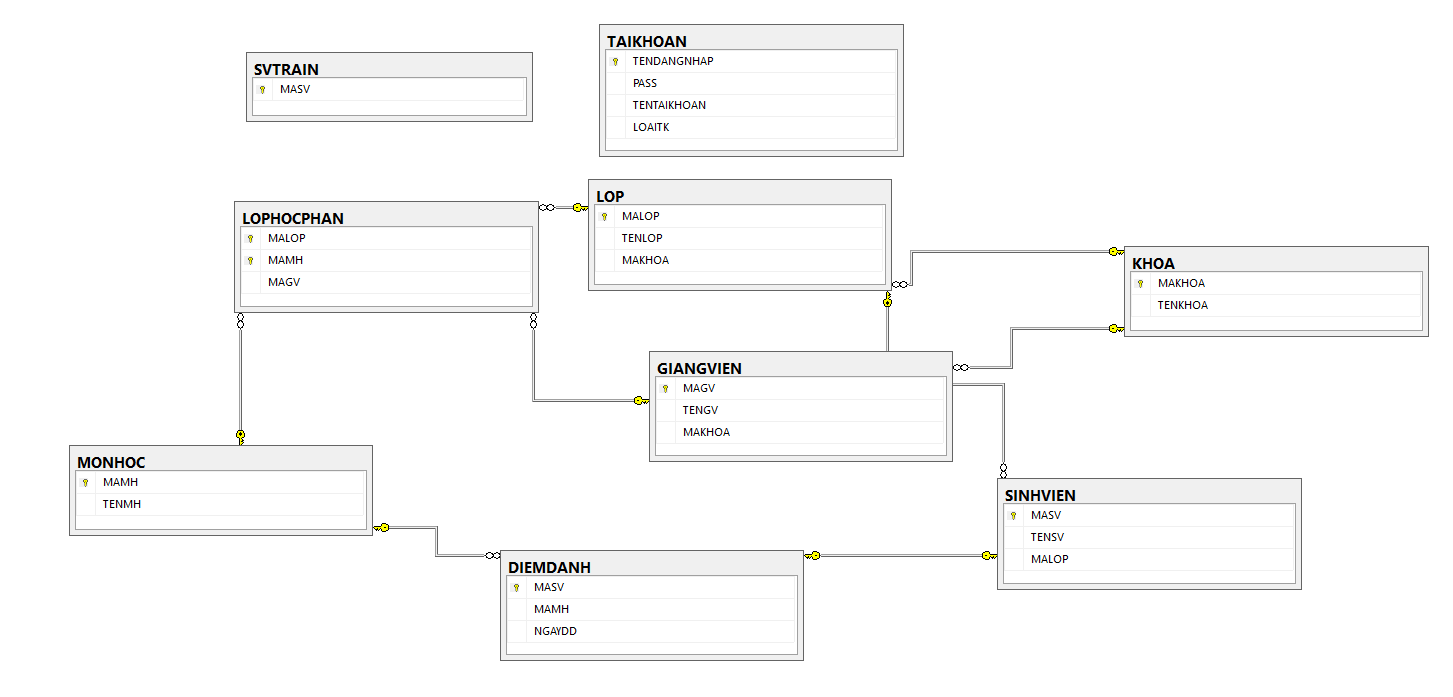
Adaboost kết hợp các classifier

Sơ đồ nhận dạng khuôn mặt:



Sơ đồ nhận dạng khuôn mặt

CHƯƠNG III: THIẾT KẾ CSDL VÀ THỰC HIỆN

1.  Sơ đồ quan hệ

Sơ đồ diagram database

1. Các chức năng của phần mềm
   1. Form đăng nhập

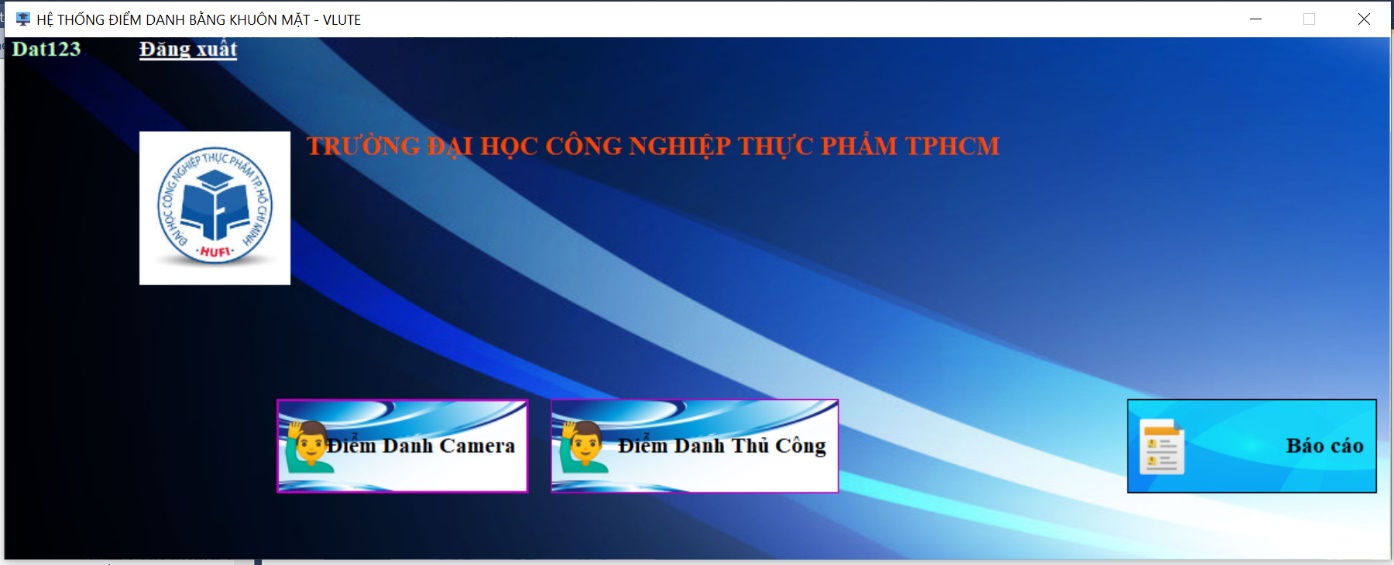
Trước khi vào bên trong của phần mềm thì bạn phải cần đăng nhập. Phần tài khoản đăng nhập. Chúng em chia ra làm 2 loại tài khoản là: tài khoản dành cho giảng viên và tài khoản dành cho admin. Ở form đăng nhập thì thầy cô có thể xem lại lịch dạy của mình bằng cách nhấn vào *Thời khóa biểu*. Và khi phần mềm gặp sự cố thì giảng viên có thể báo cho admin bằng facebook hoặc gmail.

****

2.2 Form danh mục

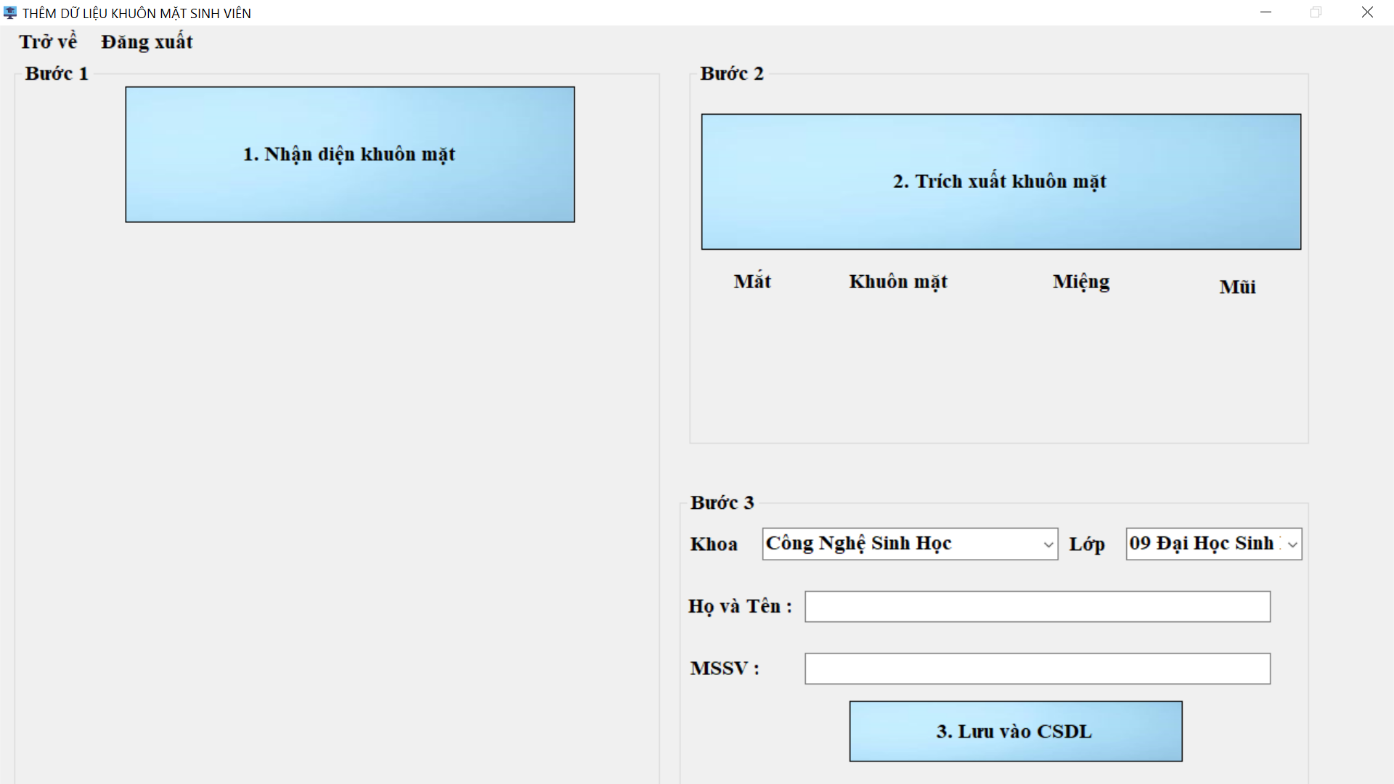
Form danh mục dành cho admin:

Admin quản lý tất cả tài khoản của giảng viên và chỉ admin mới có quyền thêm sinh viên và quản lý thông tin của sinh viên.

Form danh mục dành cho giảng viên:

Giảng viên chỉ có 3 chức năng là điểm danh bằng camera, điểm danh thủ công và phần báo cáo sau mỗi buổi học.

2.3 Chi tiết các chức năng của phần mềm

**2.3.1. Chức năng thêm sinh viên**

Phần thêm sinh viên gồm 3 bước:

* Đầu tiên chúng ta nhấn vào *nhân diện khuôn mặt*
* Nhấn trích xuất khuôn mặt để lấy ra các phần trên khuôn mặt. Đây là bước chuẩn bị dữ liệu để train khuôn mặt.
* Điền đầy đủ thông tin của sinh viên. Sau đó lưu xuống CSDL.

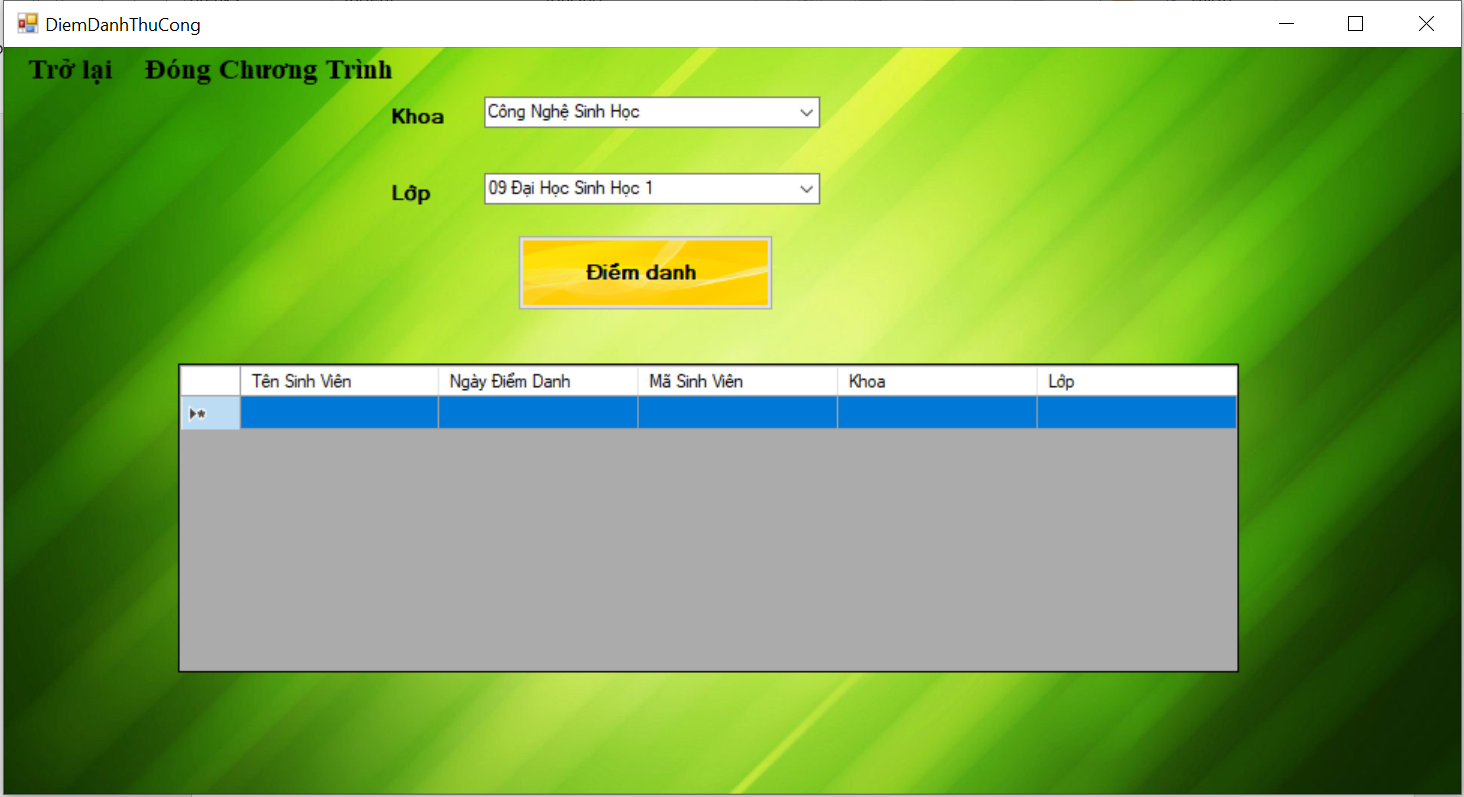
**2.3.2. Chức năng điểm danh bằng camera**

Khi nhấn vào chức năng điểm danh bằng camera thì phần mềm sẽ chuyển qua form mới:



Sau khi vào giảng viên nhấn chọn *Bắt đầu điểm danh* thì phần mềm sẽ bắt đầu điểm danh tự động. Mỗi khi phát hiện khuôn mặt sinh viên trùng trong tập train thì phần mềm sẽ tự động điểm danh và hiện lên datagrid view. Gần cuối buổi học giảng viên có thể điểm danh lại xem có đủ giống ban đầu không bằng cách nhấn chọn *Điểm danh lại.*

**2.3.3 Chức năng điểm danh thủ công**

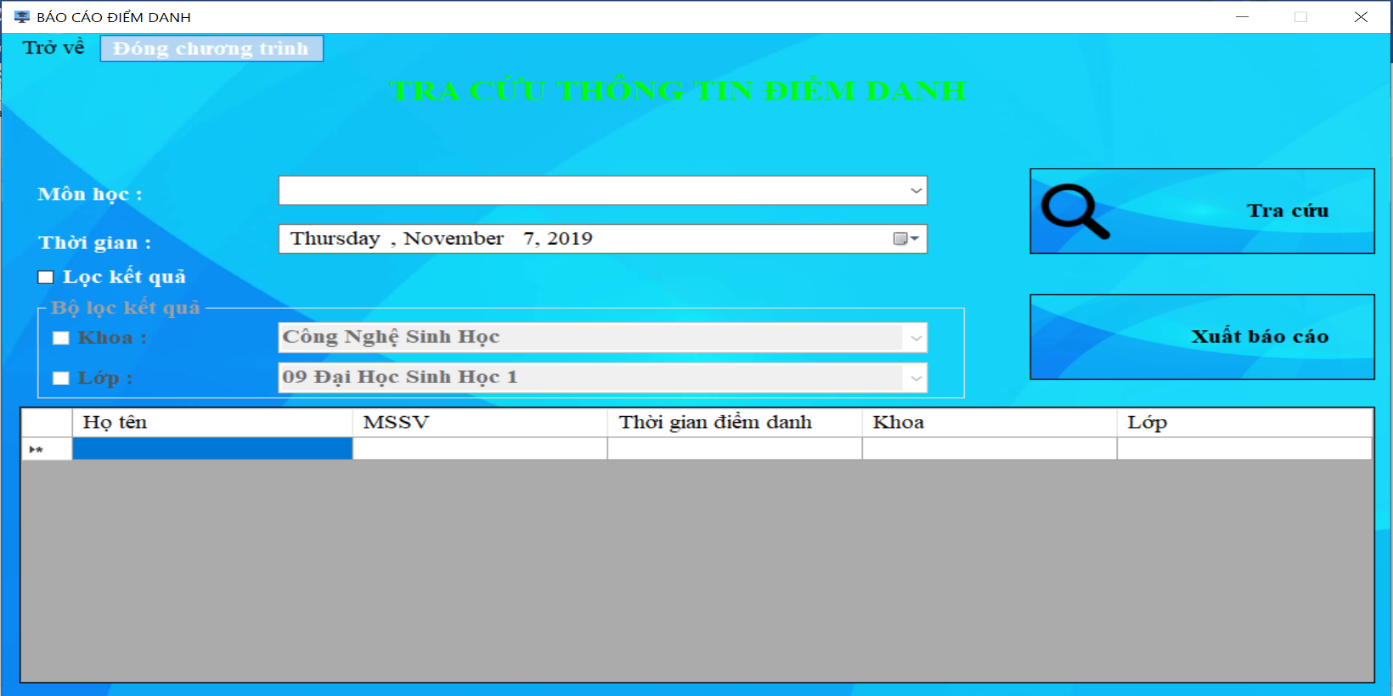
 Đây là chức năng, giúp cho một số bạn sinh viên, khi mà phần mềm không thể nhận diện được, thì giảng viên có thể điểm danh bằng thủ công.

**2.3.4 Chức năng quản lý sinh viên**

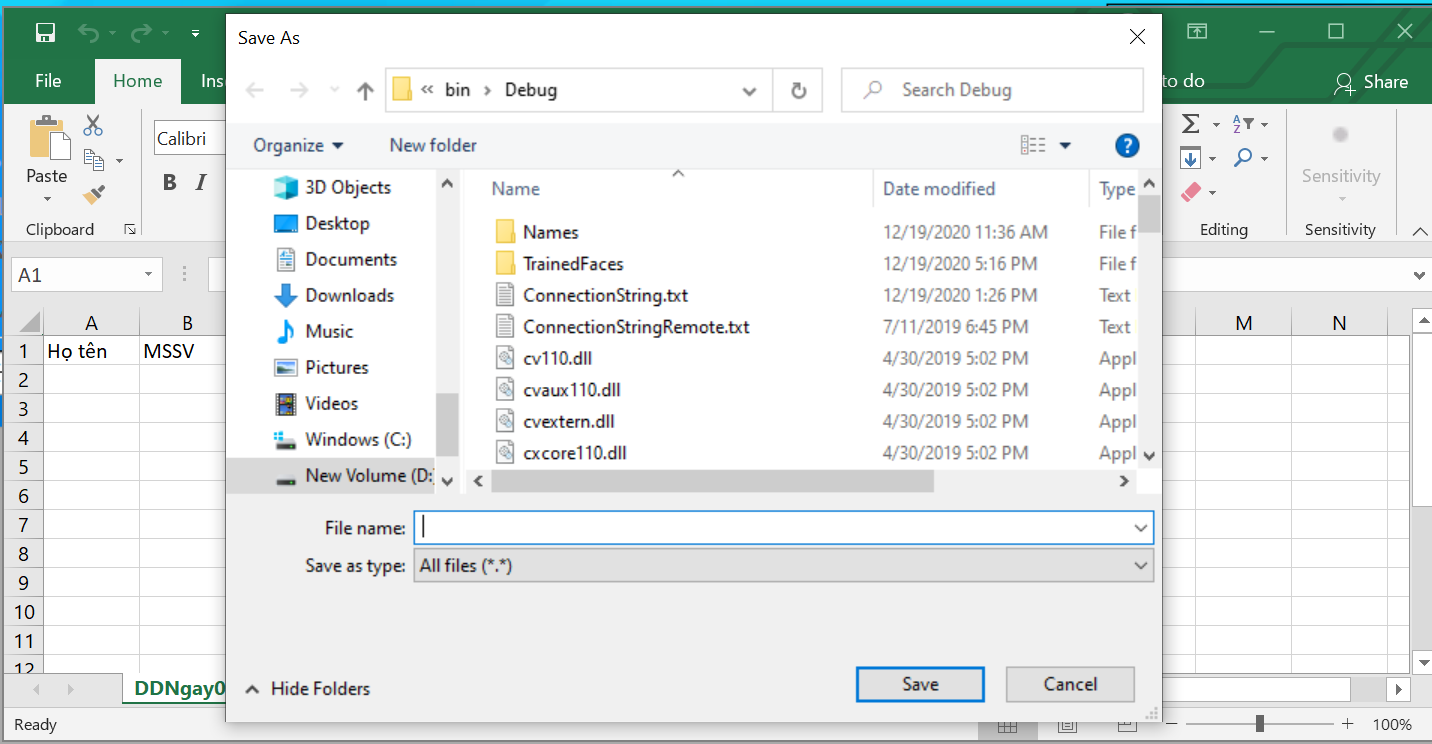


Admin có thể chỉnh sửa và xóa thông tin của sinh viên.

**2.3.5. Chức năng**



Đây là chức năng giúp cho giảng viên xuất ra file excel các sinh viên đã được điểm danh.

*Giao diện khi nhấn chọn xuất báo cáo.*

CHƯƠNG IV: KẾT LUẬN VÀ ĐỊNH HƯỚNG PHÁT TRIỂN

1. Kết luận

Chúng em đã tìm hiểm về Emgu/OpenCV, làm được các tác vụ cơ bản của phần mềm.

Chúng em chưa làm được phần môn học, đó là khi điểm danh bằng khuôn mặt thì chưa xác định được sinh viên đó thuộc vào môn nào, nhận diện có độ chính xác chưa cao.

1. Định hướng phát triển

Khắc phục các phần chưa làm được, cải thiện thuật toán để phần mềm nhận diện tốt hơn, thêm một số chức năng để tiện cho giảng viên trong quá trình điểm danh để không mất qua nhiều thời gian.

TÀI LIỆU THAM KHẢO

1. [https://4fire.wordpress.com/2012/07/31/nhan-dang-mat-nguoi-face-recognition-bangpca-eigen-face-matlab-code-on-orl-database/](https://4fire.wordpress.com/2012/07/31/nhan-dang-mat-nguoi-face-recognition-bang-pca-eigen-face-matlab-code-on-orl-database/)
2. Nguyễn Văn Long. “*Ứng dụng xử lý ảnh trong thực tế với thư viện OpenCV C/C++*”
3. https://www.stdio.vn/computer-vision/emgucv-opencv-cho-dot-net-gioi-thieu-va-cach-tich-hop-vao-dot-net-cAicg
4. Lê Mạnh Tuấn. Khóa luận tốt nghiệp ĐH hệ chính quy: “*Phát hiện mặt người trong ảnh và ứng dụng*”. ĐH Công Nghệ. Đại Học Quốc Gia Hà Nội.